

Streitberger, Werner

**Working Paper**

## Einsatz von Risikomanagement bei der Steuerung von Grid-Systemen: Eine Analyse von Versicherungen anhand einer simulierten Grid-Ökonomie

Bayreuther Arbeitspapiere zur Wirtschaftsinformatik, No. 35

**Provided in Cooperation with:**

University of Bayreuth, Chair of Information Systems Management

*Suggested Citation:* Streitberger, Werner (2009) : Einsatz von Risikomanagement bei der Steuerung von Grid-Systemen: Eine Analyse von Versicherungen anhand einer simulierten Grid-Ökonomie, Bayreuther Arbeitspapiere zur Wirtschaftsinformatik, No. 35, Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik, Bayreuth, <https://nbn-resolving.de/urn:nbn:de:bvb:703-opus-9816>

This Version is available at:

<https://hdl.handle.net/10419/191451>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/3.0/deed.de>



## Bayreuther Arbeitspapiere zur Wirtschaftsinformatik

Werner Streitberger

---

Einsatz von Risikomanagement bei der Steuerung von  
Grid-Systemen – eine Analyse von Versicherungen  
anhand einer simulierten Grid-Ökonomie

Bayreuth Reports on Information Systems Management



Die Arbeitspapiere des Lehrstuhls für Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i. d. R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

Alle Rechte vorbehalten. Insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

**Authors:**

Werner Streitberger (University of Bayreuth)

The Bayreuth Reports on Information Systems Management comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

All rights reserved. No part of this report may be reproduced by any means, or translated.

**Information Systems Management  
Working Paper Series**

**Edited by:**

Prof. Dr. Torsten Eymann

**Managing Assistant and Contact:**

Universität Bayreuth  
Lehrstuhl für Wirtschaftsinformatik (BWL VII)  
Prof. Dr. Torsten Eymann  
Universitätsstrasse 30  
95447 Bayreuth  
Germany

Email: [wi@uni-bayreuth.de](mailto:wi@uni-bayreuth.de)

**ISSN** 1864-9300

EINSATZ VON RISIKOMANAGEMENT BEI DER STEUERUNG VON  
GRID-SYSTEMEN – EINE ANALYSE VON VERSICHERUNGEN  
ANHAND EINER SIMULIERTEN GRID-ÖKONOMIE

D i s s e r t a t i o n  
zur Erlangung des Grades eines Doktors der Wirtschaftswissenschaft  
der Rechts- und Wirtschaftswissenschaftlichen Fakultät  
der Universität Bayreuth

vorgelegt  
von  
Dipl.-Inf. Univ. Werner Streitberger  
aus Donauwörth

Dekan: Prof. Dr. Jochen Sigloch  
Erstberichterstatter: Prof. Dr. Torsten Eymann  
Zweitberichterstatter: Prof. Dr. Klaus Schäfer  
Tag der mündlichen Prüfung: 4. Mai 2009

## **Erklärung**

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- oder Beratungsdiensten (Promotionsberaterinnen oder Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Ferner erkläre ich hiermit, dass ich mich bislang weder an einer in- noch an einer ausländischen wissenschaftlichen Hochschule um die Promotion beworben habe noch mich gleichzeitig bewerbe. Gegen mich ist kein Ermittlungs- oder Strafverfahren wegen eines Verbrechens oder vorsätzlichen Vorgehens anhängig und ich bin weder wegen einer solchen Tat rechtskräftig verurteilt noch verbüße ich eine Freiheitsstrafe.

Bayreuth, den 12. Juni 2009

Werner Streitberger



# Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter des Lehrstuhls für Wirtschaftsinformatik an der Universität Bayreuth. An dieser Stelle möchte ich allen herzlich danken, die während dieser Zeit mit ihren Anregungen, ihrer Unterstützung und ihren Ermutigungen zum Gelingen meiner Dissertation beigetragen haben.

Mein besonderer Dank gilt meinem sehr geschätzten Doktorvater Professor Dr. Torsten Eymann, Inhaber des Lehrstuhls für Wirtschaftsinformatik an der Universität Bayreuth, der durch eine Vielzahl von Informationen, Hinweisen und fachlichen Anmerkungen diese Arbeit entscheidend beeinflusst und ihren Erfolg mitgeprägt hat. Seine Unterstützung bei meiner wissenschaftlichen Tätigkeit, seine ständige Bereitschaft zu fachlichen Diskussionen mit vielen wertvollen Ratschlägen und konstruktiven Anregungen sowie sein unentwegter Eifer auf dem Gebiet der Motivation bildeten ein herausragendes Fundament für meine fachliche und persönliche Weiterentwicklung.

Des Weiteren möchte ich mich bei meinen Kolleginnen und Kollegen des Lehrstuhls für Wirtschaftsinformatik an der Universität Bayreuth bedanken, die einen großen Teil zum Gelingen meiner Arbeit beigetragen haben. Mein ganz besonderer Dank gilt Dr. Michael Reinicke für die intensive Zusammenarbeit in dem europäischen Forschungsprojekt CATNETS und für die Unterstützung zu Beginn meiner Arbeit. Mit ihm verbindet mich noch immer eine enge Freundschaft. Ich möchte mich auch bei meinen Kollegen Stefan König, Falk Zwicker und Raimund Matros bedanken, die stets ein offenes Ohr für mich hatten und mir mit ihrem Expertenwissen sehr weitergeholfen haben.

Mein Dank gilt auch dem Systemadministrator des Lehrstuhls Axel Pürckhauer, der mich immer mit seinem exzellenten Fachwissen bei der Lösung aller IT-Probleme während der Durchführung meiner Simulationen unterstützt hat. Ferner möchte ich den Sekretärinnen des Lehrstuhls für Wirtschaftsinformatik Frau Mandel und Frau Tauscher danken, die zu einem sehr guten Lehrstuhlklima beigetragen haben.



Die Grundlagen dieser Arbeit entstanden in dem europäischen Forschungsprojekt CATNETS. An dieser Stelle möchte ich mich bei dem gesamten Projektteam bedanken, ganz besonders bei Isaac Chao von der Universitat Politècnica de Catalunya für die immer sehr leidenschaftlich geführten, fachlichen Diskussionen im Projekt und die unvergessliche Zeit in Barcelona.

Außerdem danke ich an dieser Stelle all meinen Freunden für ihre Geduld, ihre Unterstützung und für unsere offenen Gespräche, die es mir ermöglicht haben, diese Arbeit fertig zu stellen.

Für die materielle Förderung meines Dissertationsprojekts bedanke ich mich bei dem Land Bayern, der Europäischen Union im Rahmen der Forschungsprojekte CATNETS und eRep und der Universität Bayreuth.

Mein besonderer Dank gilt meinen Eltern, die mich während meiner gesamten Ausbildung gefördert haben und mir Mut und Kraft gegeben haben. Zu guter Letzt möchte ich mich bei meiner lieben Schwester Miriam bedanken, die immer für mich da war und zum sehr guten Erfolg meiner Arbeit tatkräftig beigetragen hat.

München, Mai 2009  
Werner Streitberger

# Zusammenfassung

In Unternehmen hat sich der Einsatz von rechenintensiver Informationstechnologie (IT) bereits für den Geschäftsbetrieb als unverzichtbar erwiesen, um Geschäftsprozesse besser auszurichten und neue Geschäftslösungen mit größerer Flexibilität und Geschwindigkeit bereitzustellen. Dieser Situation gegenüber stehen die Kosten für die Anschaffung, den Betrieb und die Wartung der IT. Diese Kosten rechtfertigen jedoch nur selten die vollständige Abdeckung des potenziellen, maximal erwarteten Bedarfs von Software und Ressourcen wie Speicher- und Rechenleistung. So müssen Unternehmen neben Effizienz- und Geschwindigkeitsverbesserungen auch Kosteneinsparungen für ihre Infrastruktur realisieren, um wettbewerbsfähig zu bleiben.

Grid-Computing kann hierzu der nächste Schritt sein, IT-Dienste zu verbessern und bestehende Kapazitäten besser auszulasten. Das hinter dem Begriff Grid-Computing stehende Konzept beschreibt verschiedene Lösungsansätze zur Umsetzung eines dynamischen Bezugs von IT-Ressourcen und Diensten innerhalb eines Unternehmens und über Unternehmensgrenzen hinweg. Im Grid-Computing-Paradigma werden Informationen auf Rechnern im Internet gespeichert, diese werden dann den Benutzern auf Anforderung durch Dienstleister zur Verfügung gestellt. Jedoch gehen mit dem Einsatz von Grid-Computing-Systemen technische Risiken einher, deren Ursachen meist auf fehlerhafte Kommunikation und/oder auf den Ausfall von Ressourcen eines Standorts zurückzuführen sind. Dies schränkt die Zuverlässigkeit und Verfügbarkeit der Systeme ein und erfordert den Einsatz von Verfahren zur Behandlung dieser Risiken, da die Benutzer zuverlässige Ressourcen von einem Grid-Computing-System erwarten.

In den Wirtschaftswissenschaften sind verschiedene Verfahren bekannt, mit denen Risiken bewältigt werden können. Diese Aufgabe übernehmen häufig Versicherungen als Instrument, um ein bestimmtes Risiko zu transferieren. Versicherungen folgen dem Grundprinzip der kollektiven Risikoübernahme: Viele zahlen einen Versicherungsbetrag in den Geldtopf der Versicherung ein, um beim Eintreten des Versicherungsfalls aus diesem Geldtopf einen Schadensausgleich zu erhalten. Für den Konsumenten von Grid-Diensten bietet die Versicherung den Vorteil, dass sie

neben der monetären Kompensation eines Schadens als Kompensationsleistung auch Ersatzressourcen vermitteln kann. In letzterem Fall werden Ressourcen zur Kompensation vorgehalten und im Schadensfall bereitgestellt, die die Ausführung des Dienstes übernehmen und so zu einer erhöhten Verfügbarkeit und Zuverlässigkeit der Grid-Systeme beitragen.

Die Arbeit identifiziert technische und ökonomische Risiken in Grid-Computing-Systemen und untersucht sie hinsichtlich ihrer Versicherbarkeit. Anforderungen an die Versicherung für Grid-Systeme werden abgeleitet und ein Prämienberechnungsmodell aus der Kraftfahrzeugversicherung ausgewählt und an Grid-Systeme angepasst. Die Umsetzung der Grid-Versicherung in ein rechnergestütztes Simulationsmodell erfolgt mithilfe von Multi-Agenten-Technologie mit der ein elektronischer Marktplatz zum Handel von Grid-Ressourcen simuliert wird. Die Versicherung wird mit gridspezifischen Performanzkennzahlen und Metriken der Versicherungsökonomie evaluiert. Die in dieser Arbeit durchgeführten Simulationen zeigen, dass die Einführung einer Versicherung einen nachweislichen Einfluss auf die analysierten technischen und ökonomischen Kennzahlen haben und eine Verbesserung der Verfügbarkeit und Zuverlässigkeit der untersuchten Utility-Grid-Systeme erwarten lassen. Ressourcenkonsumenten haben bei der Festlegung ihrer Strategie zur Behandlung der technischen Risiken einen flexiblen Ansatz zur Verfügung, den sie in ihrer Risikomanagementstrategie berücksichtigen können. Der Risikotransfer auf eine Grid-Versicherung bietet den Ressourcenkonsumenten die Möglichkeit, ihre Risikokosten im Vergleich zu einer Redundanzstrategie zu senken. Aus Sicht der Versicherungswirtschaft besteht die Möglichkeit neue Versicherungstarife im Bereich von Grid-Systemen anzubieten und damit ihr Angebot an Versicherungstarifen zu erweitern.

# Abstract

Computational-intensive information technology (IT) is one of the key success factors for businesses, which allows them to align their business processes with greater flexibility and dynamics to current needs. But, costs for initial purchase of hardware and software, operational and maintenance expenses could lower the expected profit because of fluctuating demand and peak load capacity planning. These costs could be justified only in a few business cases where the complete coverage of the potential maximum supply of software applications and computing resources like storage and bandwidth is required. In most cases, businesses are forced to realize benefits from efficiency and speed of their information technology as well as reducing their process and infrastructure costs to stay competitive.

The Grid Computing paradigm has potential to be the next big step improving the delivery of IT services and utilizing existing capacities more efficiently. The concept behind the Grid Computing paradigm encompasses different solutions for dynamic delivery of IT resources and services within companies and beyond the borders of a single company. Grid Computing systems store information on Internet-accessible resources, which are requested by users and provided by Grid providers. But, Grid systems come along with a number of technical and operational risks, which have their root cause in communication failures and/or failures of services and resources of a Grid site. This reduces the reliability and availability of Grid systems and requires the introduction of new risk management strategies to be taken into account by Grid users.

In economics, there are several risk management strategies known, which could be applied to address these Grid-specific risks. The risk transfer using insurances is a well known risk management strategy which covers a certain risk by paying a premium. Insurances follow the principle of collective risk coverage: Many individuals pay a premium into a pool from which a claimant receives the compensation of his claim. Consumers of Grid services could profit from the existence of an insurance service twofold: a Grid service insurance could offer beside a monetary compensation also a compensation with spare resources, which continue the service execution

and prevent the failure of the service. Depending on the selected compensation model, an insurance risk transfer strategy could result in higher availability and reliability of a Grid system.

This thesis identifies technical and economical risks of Grid systems and analyzes them regarding to their insurability. Requirements of insurances for Grid systems are deducted from the analysis. A premium computation model derived from the car insurance domain is selected and adapted to Grid systems. The implementation of the Grid insurance service in a computer-based simulation model uses multi-agent technology to realize an electronic market place for trading Grid services, resources and insurance policies. The model is simulated and evaluated with Grid specific performance metrics as well as metrics from the insurance economy domain. The simulation runs show that the introduction of an insurance on a Grid market has strong influence on the technical and economical metrics and results in better availability and reliability of the simulated Grid system. Resource consumers could profit from a Grid service insurance, which extends their risk management strategy portfolio beside risk acceptance and redundancy with a third option. The risk transfer strategy with insurances enables a Grid consumer to lower his risk costs compared to the existing risk acceptance and redundancy strategies. As the simulations results of this thesis show, insurances services for Grid systems have the potential for insurance companies to create new insurance policy offerings and to extend their traditional insurance business to Grid systems.

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>V</b>
<b>Zusammenfassung</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Inhaltsverzeichnis</b>	<b>XI</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problemstellung und Relevanz der Arbeit . . . . .	6
1.3 Die Forschungsmethode Simulation . . . . .	10
1.3.1 Abgrenzung von analytischer Forschungsmethode und realem Experiment	11
1.3.2 Vorteile und Nachteile von Simulationen . . . . .	13
1.4 Aufbau der Arbeit . . . . .	14
1.5 Glossar des Risikobegriffs . . . . .	15
<b>2 Grid-Technologien und ihre Risiken</b>	<b>19</b>
2.1 Aufbau von Grid-Systemen . . . . .	21
2.1.1 Allgemeine Entwurfsprinzipien . . . . .	21
2.1.2 Schichtenmodell der Grid-Architektur . . . . .	23
2.1.3 Klassifizierung der technischen Risiken . . . . .	25
2.2 Die Open-Grid-Services-Architektur . . . . .	29
2.2.1 Der Begriff des Grid-Services . . . . .	29
2.2.2 Grundlegende Grid-Dienste . . . . .	31
2.2.3 Serviceorientierte Architekturen und Web-Services . . . . .	32
2.3 Service-Level-Agreements (SLAs) . . . . .	34
2.3.1 Aufbau von Service-Level-Agreements . . . . .	35

2.3.2	Einsatz von Service-Level-Agreements in Grid-Transaktionen . . . . .	37
2.4	Anwendungsfelder von Grid-Systemen . . . . .	39
2.4.1	Computational-Grid . . . . .	39
2.4.2	Data-Grid . . . . .	43
2.4.3	Utility-Grid . . . . .	45
2.5	Organisationsformen von Grid-Systemen . . . . .	47
2.5.1	Enterprise-Grid . . . . .	47
2.5.2	Partner-Grid . . . . .	48
2.5.3	Open-Grid . . . . .	50
2.6	Studien über technische Risiken und Risikomodelle . . . . .	52
2.6.1	Studien zur Fehleranalyse in Computational-Grids . . . . .	52
2.6.2	Das Modell von Dai et al. zur Modellierung von Grid-Risiken . . . . .	55
2.7	Zusammenfassung . . . . .	59
<b>3</b>	<b>Die CATNETS Grid-Ökonomie</b>	<b>61</b>
3.1	Das Utility-Grid-Modell . . . . .	62
3.1.1	Service- und Ressourcenmarkt . . . . .	62
3.1.2	Netzwerkmodell . . . . .	67
3.2	Implementierung der CATNETS Grid-Ökonomie . . . . .	69
3.2.1	Flooding-basierte Dienstsuche . . . . .	70
3.2.2	Verhandlung der Grid-Services . . . . .	72
3.2.2.1	Allgemeine Prinzipien . . . . .	72
3.2.2.2	Variablen der Strategie . . . . .	73
3.2.2.3	Der Lernalgorithmus . . . . .	75
3.2.3	Ausführung eines Service auf Grid-Ressourcen . . . . .	77
3.3	Zusammenfassung . . . . .	78
<b>4</b>	<b>Risikomanagement im Grid-Computing</b>	<b>81</b>
4.1	Der Grid-Risikomanagementzyklus . . . . .	82
4.2	Verfahren zur Risikobehandlung in Grid-Ökonomien . . . . .	85
4.2.1	Risikoakzeptanz . . . . .	86
4.2.2	Risikovermeidung . . . . .	88
4.2.3	Risikoreduktion . . . . .	88
4.2.4	Risikotransfer . . . . .	89
4.3	Zusammenfassung . . . . .	94

<b>5</b>	<b>Realisierung einer Versicherung in einer Grid-Ökonomie</b>	<b>95</b>
5.1	Funktionen und Aufgaben von Versicherungen . . . . .	96
5.2	Bestandteile einer Versicherung . . . . .	97
5.3	Versicherbarkeit von Grid-Risiken . . . . .	99
5.4	Realisierung des Versicherungsmodells . . . . .	104
5.4.1	Schadensstatistik . . . . .	104
5.4.2	Versicherungsperiode . . . . .	104
5.4.3	Prämienberechnung . . . . .	107
5.4.3.1	Prämiendifferenzierung . . . . .	107
5.4.3.2	Prognose auf Basis von Tarifmerkmalen . . . . .	109
5.4.3.3	Versicherungstechnisches Risiko . . . . .	112
5.4.4	Erstattung im Schadensfall . . . . .	115
5.5	Einbettung des Grid-Versicherungsmodells in die CATNETS Grid-Ökonomie . .	116
5.5.1	Versicherungswettbewerb vs. Versicherungsmonopol . . . . .	116
5.5.2	Modellierung der Versicherungsnachfrage . . . . .	117
5.6	Zusammenfassung . . . . .	117
<b>6</b>	<b>Simulation einer Grid-Versicherung in einer Grid-Ökonomie</b>	<b>119</b>
6.1	Grundlagen von Simulationen . . . . .	120
6.1.1	Grundlegende Konzepte . . . . .	120
6.1.2	Prozess zur Durchführung von Simulationen . . . . .	122
6.2	Der CATNETS Grid-Simulator . . . . .	127
6.2.1	Aufbau der Simulationsumgebung . . . . .	128
6.2.2	Konfiguration des Simulators . . . . .	132
6.3	Modellimplementierung . . . . .	134
6.3.1	Implementierung des Grid-Versicherungsmodells . . . . .	134
6.3.2	Sequenzdiagramm eines beispielhaften Ablaufprozesses . . . . .	139
6.4	Simulationsentwurf . . . . .	141
6.4.1	Dynamik der Grid-Versicherung . . . . .	141
6.4.1.1	Versicherungsperiode . . . . .	141
6.4.1.2	Sicherheitszuschlag . . . . .	142
6.4.1.3	Versicherungsnachfrage . . . . .	142
6.4.2	Einfluss der Grid-Versicherung auf das Ressourcenmanagement . . . . .	143
6.5	Definition der Metrikpyramide zur Evaluierung der Systemperformanz . . . . .	144
6.5.1	Allgemeine Methode . . . . .	144



6.5.2	Die Metrikenpyramide . . . . .	146
6.5.2.1	Aufbau . . . . .	146
6.5.2.2	Technische Schicht der Messdaten . . . . .	148
6.5.2.3	Indikatoren - Erste Aggregation . . . . .	152
6.5.2.4	Einfache Indizes - Zweite Aggregation . . . . .	156
6.5.2.5	Der Social-Loss-Index - Dritte Aggregation . . . . .	157
6.6	Zusammenfassung . . . . .	159
<b>7</b>	<b>Evaluation der Grid-Versicherung</b>	<b>161</b>
7.1	Erzeugung eines realistischen Szenarios . . . . .	162
7.1.1	Netzwerkinfrastruktur und Ressourcenverteilung . . . . .	162
7.1.2	Szenario des Ressourcenmarkts . . . . .	164
7.1.3	Szenario des Servicemarkts . . . . .	168
7.1.4	Zuverlässigkeit der Basic-Service-Ausführung . . . . .	169
7.1.5	Versicherungstarif für die Basic-Service-Ausführung . . . . .	171
7.2	Simulationsergebnisse Teil 1: Dynamik der Versicherung . . . . .	173
7.2.1	Versicherungsperiode und Entwicklung der Tariffaktoren . . . . .	173
7.2.2	Sicherheitszuschlag . . . . .	181
7.2.3	Versicherungsnachfrage . . . . .	186
7.2.3.1	Konfiguration des Lernalgorithmus . . . . .	186
7.2.3.2	Analyse der Versicherungsnachfrage . . . . .	188
7.2.3.3	Analyse der Basic-Services . . . . .	197
7.3	Simulationsergebnisse Teil 2: Analyse der Gesamtperformanz . . . . .	204
7.3.1	Aufbau und Konfiguration der Szenarien . . . . .	204
7.3.2	Analyse der Indikatoren . . . . .	206
7.3.3	Analyse der Indizes . . . . .	212
7.4	Zusammenfassung . . . . .	215
<b>8</b>	<b>Zusammenfassung und Fazit</b>	<b>219</b>
8.1	Zusammenfassung der Arbeit . . . . .	220
8.2	Implikationen für die Forschung . . . . .	222
8.3	Implikationen für die Praxis . . . . .	225

<b>Anhang</b>	<b>227</b>
<b>A Implementierung der Grid-Versicherung</b>	<b>229</b>
A.1 Die Klasse <i>InsuranceAgent</i> . . . . .	229
A.2 Die Klasse <i>InsuranceModelResource</i> . . . . .	235
<b>B Konfiguration des Simulationsszenarios</b>	<b>253</b>
B.1 Simulatorkonfiguration . . . . .	253
B.2 Konfiguration der Versicherung . . . . .	257
B.3 Marktkonfiguration . . . . .	258
B.4 Konfiguration der Ressourcenbündel . . . . .	265
B.5 Konfiguration der Basic-Services . . . . .	265
B.6 Konfiguration der Complex-Services . . . . .	265
B.7 Konfiguration des Lernalgorithmus . . . . .	272
<b>C Evaluationsergebnisse</b>	<b>275</b>
C.1 Versicherungsperiode . . . . .	275
C.2 Metrikpyramide . . . . .	275
<b>Abbildungsverzeichnis</b>	<b>285</b>
<b>Tabellenverzeichnis</b>	<b>291</b>
<b>Abkürzungsverzeichnis</b>	<b>293</b>
<b>Literaturverzeichnis</b>	<b>295</b>
<b>Lebenslauf</b>	<b>327</b>



# Kapitel 1

## Einführung

*„The broad and rich foundation of the internet will unleash a 'service wave' of applications and experiences available instantly. Services designed to scale to tens or hundreds of millions [of users] will dramatically change the nature and cost of solutions deliverable to enterprises or small business. This new wave will be very disruptive.“ (Bill Gates) [Carr08, S. 63]*

### 1.1 Motivation

Unternehmen müssen sich in einem Umfeld vielfältiger und neuer Anforderungen behaupten. Sie müssen sich ständig wandelnden Herausforderungen stellen, die in einer wachsenden Dynamik des Marktes und einer zunehmenden Wettbewerbssituation begründet sind [Weil92] [Nath05]. Dies hat zur Folge, dass eine stetige Anpassung und Überprüfung des eingesetzten Wissens, der Technologie und insbesondere des eigenen Ressourceneinsatzes notwendig sind [PiRW03, S. 230f]. In Unternehmen hat sich der Einsatz von rechenintensiver Informationstechnologie (IT) bereits für den Geschäftsbetrieb als unverzichtbar erwiesen, um Geschäftsprozesse besser auszurichten und neue Geschäftslösungen mit größerer Flexibilität und Geschwindigkeit bereitzustellen [ZhTZ07] [Weil92]. Dieser Situation gegenüber stehen die Kosten für die Anschaffung, den Betrieb und die Wartung der IT. Diese Kosten rechtfertigen jedoch nur selten die vollständige Abdeckung des potenziellen, maximal erwarteten Bedarfs von Software und Ressourcen wie Speicher- und Rechenleistung [RoWe04]. So müssen Unternehmen neben Effizienz- und Geschwindigkeitsverbesserungen auch Kosteneinsparungen für ihre Infrastruktur realisieren, um wettbewerbsfähig zu bleiben [Orlo06].

*Grid-Computing* kann hierzu der nächste Schritt sein, IT-Dienste zu verbessern und bestehen-

de Kapazitäten besser auszulasten [Wlad04] [Haye08] [BMQL<sup>+</sup>07]. Das hinter dem Schlagwort stehende Konzept beschreibt verschiedene Lösungsansätze zur Umsetzung eines dynamischen Bezugs von IT-Ressourcen und Diensten innerhalb eines Unternehmens und über Unternehmensgrenzen hinweg [Stro05] [Weis07]. Im Grid-Computing-Paradigma werden Informationen auf Rechnern im Internet gespeichert, diese werden dann den Benutzern auf Anforderung zur Verfügung gestellt [Hewi08].

Jedoch gehen mit dem Einsatz von Grid-Computing-Systemen technische Risiken einher, deren Ursachen meist auf fehlerhafte Kommunikation [Birm93] und/oder auf den Ausfall von Ressourcen [BaBL02] eines Standorts zurückzuführen sind. Dies schränkt die Zuverlässigkeit und Verfügbarkeit der Systeme ein [RoWe03] und erfordert den Einsatz von Verfahren zur Behandlung dieser Risiken [BaBL02], da die Benutzer zuverlässige Ressourcen von einem Grid-Computing-System erwarten [ThLi04] [JaLG08].

Diese Risiken können einen signifikanten Einfluss auf die Geschäftsmodelle haben [JaLG08], die im Umfeld des Grid-Computing-Paradigmas anzutreffen sind und in Abbildung 1.1 gezeigt werden. Aktuelle Geschäftsmodelle sind:

- **Infrastructure-as-a-Service (IaaS):** Das Geschäftsmodell *Infrastructure-as-a-Service* umfasst die Lieferung einer virtualisierten Computerinfrastruktur als Service [NWGO<sup>+</sup>08] [MiLe08], auf der Softwareanwendungen eines Benutzers ausgeführt werden können [Vari08]. Die virtualisierte Infrastruktur umfasst sowohl Rechenleistung als auch Speicherkapazität und Netzwerkbandbreite, die nach Bedarf abgerechnet werden [MiLe08].
- **Software-as-a-Service (SaaS):** Ein weiteres Geschäftsmodell stellt *Software-as-a-Service* dar, das die Bereitstellung von Softwareanwendungen über eine Internetinfrastruktur beschreibt [Lawt08]. Die Bereitstellung der Software erfolgt üblicherweise über einen Application-Service-Provider (ASP), der die Installation, die Wartung und den Betrieb der Softwareanwendung übernimmt [TuBB03] [LaZV08].
- **Utility-Computing:** *Utility-Computing*, häufig auch *On-Demand-Computing* genannt, beschreibt ein Geschäftsmodell, in dem Unternehmen nur einen Teil ihrer IT-Dienste selbst bereitstellen und den variierenden, zusätzlichen Bedarf von einem Fremdanbieter beziehen [Stie07] [FoKe03]. Die IT-Dienste und IT-Ressourcen liegen als *Utility* vor, d. h., sie sind vollständig standardisiert und werden nach Bedarf angefordert und abgerechnet [BuYV08]. IT-Leistungen werden nicht mehr als Produkte, sondern als *Service* genutzt [Cusu08].

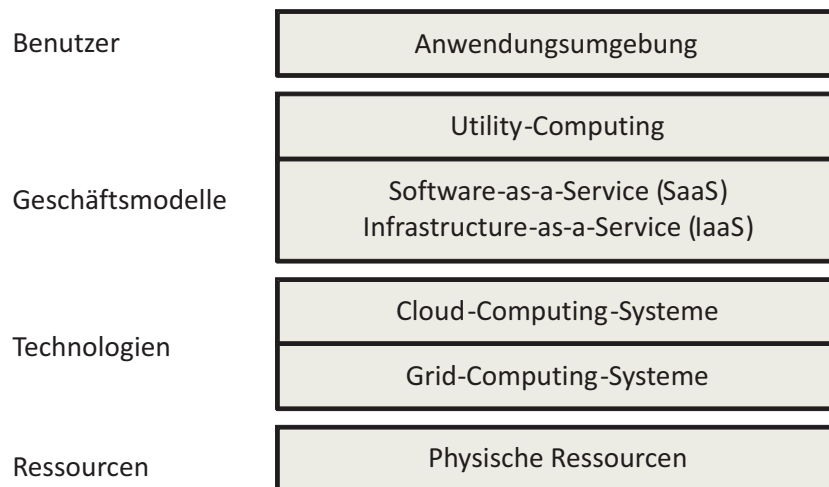


Abbildung 1.1: Technologien und Geschäftsmodelle zum Bezug von IT-Diensten und IT-Ressourcen

Verwandt zu den Grid-Computing-Systemen sind die *Cloud-Computing*-Systeme, die vor allem von kommerziellen Anbietern eingesetzt werden. Diese integrieren die Konzepte des *Utility-Computing* und des *Grid-Computing*, um Infrastrukturrressourcen und Softwareanwendungen auf einer höheren Abstraktionsebene dem Benutzer zur Verfügung zu stellen [Bégi08] [JhMF08].

Unternehmen, die in Betracht ziehen, Utility-Computing einzusetzen, wurden in einer Studie der META-Group [Balt04] nach ihren Zielen befragt. Sie nannten überwiegend folgende Erwartungen: Verbesserung der Ressourcenauslastung und der Geschäftsprozesse, verbesserte IT-Flexibilität und die Erstellung konsistenter Geschäftsprozesse für das gesamte Unternehmen. Diese Erwartungen können zu einer Verringerung der Kosten führen, die bei fast allen befragten Unternehmen als wichtigstes Kriterium im Vordergrund steht [Balt04]. Als weitere, potenzielle Vorteile werden für den Anwender von Utility-Computing-Services die folgenden Tatsachen genannt [Stie07]:

- Geringere Investitionsrisiken: Der Anbieter trägt die Kosten für die Anschaffung von Software oder IT-Infrastrukturkomponenten und übernimmt damit auch das Investitionsrisiko, während Kunden nur für die aktuelle Nutzung oder den tatsächlichen Verbrauch zahlen [JaLG08].
- Bessere Performance und Sicherheit: Spezialisierten Providern, für die der Betrieb der Informationstechnologie das Kerngeschäft ist, stehen in der Regel mehr Ressourcen für die Sicherstellung von Performance und Sicherheit zur Verfügung [Stie07].

Für die Anbieter von Utility-Computing-Diensten liegt der Reiz in der hohen Skalierbarkeit, die

es ermöglicht, Skaleneffekte durch Standardisierung zu erzielen [Garf07] [Carr05]. Zwar entstehen höhere Fixkosten beim Aufbau der Infrastruktur, die variablen Kosten durch Wartung und Support fallen jedoch wesentlich geringer aus. Aus technischer Sicht werden Utility-Computing-Ansätze durch Virtualisierungstechnologien und -konzepte unterstützt, die es möglich machen, Ressourcen in einem gemeinsamen Ressourcenpool zu nutzen. Mithilfe von Virtualisierungstechnologien werden physikalische IT-Ressourcen dabei in logische Einheiten aufgeteilt, die unterschiedlichen Benutzern zur Verfügung stehen und somit eine gleichzeitige Nutzung der Ressourcen durch verschiedene Benutzer ermöglichen. Einem IT-Dienst bzw. einer Anwendung wird kein bestimmter Server oder Speicherplatz mehr exklusiv zugeteilt, sondern werden nur bei Bedarf zusätzliche, freie Kapazitäten aus einem Ressourcenpool bezogen. Das heißt auch, dass mehrere Kunden logisch getrennt über eine gemeinsam genutzte Infrastruktur bedient werden können. Ein Anbieter von Utility-Computing-Services kann somit seine Infrastruktur besser auslasten [JaLG08].

Die IT-Dienste werden über kurzfristige, elektronische Verträge, sogenannte Service-Level-Agreements (SLAs) [KeLu03], angemietet und in die eigenen, betriebsinternen Prozesse integriert. Für Unternehmen bleiben die Kosten transparent, da die Abrechnung nach einem *pay-per-use* Modell erfolgt [AllM07]. *Pay-per-use* stellt ein Lizenzmodell für die Nutzung von IT-Diensten dar, bei dem nur der tatsächliche Verbrauch in Rechnung gestellt wird [DiSe68] [BoSc03]. Die Nutzung des IT-Dienstes wird hinsichtlich der im Service-Level-Agreement (SLA) festgelegten Kriterien gemessen und nach Dienstleistungserbringung abgerechnet.

Das *Grid-Computing*-Paradigma unterstützt dabei das *Utility-Computing* Geschäftsmodell. Es verfolgt die Idee, ungenutzte Rechenkapazitäten über ein privates oder öffentliches Netzwerk zusammenzuschließen und diese Kapazitäten flexibel nutzbar zu machen [Stro05]. Die Sichtweise geht hierbei von einer reinen Anbietersichtweise zu nachfragerorientierten Lösungen, die eine bessere Ausrichtung der Geschäftsprozesse und der IT insgesamt erlauben. Grid-Computing bietet die benötigte Flexibilität für die Bereitstellung von Rechenleistung und Softwareanwendungen zur Unterstützung von Geschäftsprozessen, indem geringer ausgelastete Ressourcen verfügbar gemacht werden und neue IT-basierte Geschäftslösungen unterstützt werden [BaBL02].

Drei Hauptvorteile lassen sich für den Einsatz von Grid-Computing in Unternehmen nennen [GiNu05]:

- Eine skalierbare und flexible IT-Infrastruktur: Grid-Computing gibt den Unternehmen die Möglichkeit, dynamische Ressourcen *on demand* zu bestehenden Ressourcen hinzuzufügen und wieder freizugeben. Mögliche Projektziele hängen damit nicht mehr davon ab,

ob genügend Rechenleistung oder Speicherkapazität vorhanden ist [GiNu05]. Das Überwachen von Metriken eines Grid-Systems erlaubt die Aggregation dieser technischen Metriken in ökonomische Metriken, die zusammen mit automatisierten Anpassungsmechanismen die Qualität sichern, die vom Geschäftsprozess gefordert wird. Service-Level-Agreements werden eingesetzt, um die Leistung zu messen und zu optimieren.

- **Reduzierte Betriebskosten:** Grid-Computing bietet Verfahren aus dem Autonomic-Computing an wie beispielsweise Self-Healing [BaBL02]. Dies führt zu hoher Verfügbarkeit und zur Fähigkeit der selbständigen Optimierung des Systems. IT-Systemadministratoren werden dadurch von einfachen Aufgaben entlastet und können sich auf komplexere Aufgaben konzentrieren [GiNu05].
- **Effizienter Einsatz bestehender Hardware und Ressourcen:** Durch die verteilte Natur des Grid-Computing kann es unternehmensweit dazu verwendet werden, große Mengen ungenutzter IT-Infrastrukturkapazitäten zu nutzen und die Anschaffung von neuer Hardware auf ein Minimum zu verringern [FoKe03].

Das technische Konzept des Grid-Computing ermöglicht es, hochwertige technische Ressourcen in einen auf einem Markt handelbaren Zustand zu überführen und schafft damit die Basis für eine bedarfsgerechte Inanspruchnahme von Rechnerkapazität [KeCh02] [KeCh04]. Durch die Virtualisierung und die dadurch ermöglichte vollkommene Teilbarkeit der Ressourcen ist es nicht mehr notwendig, umfangreiche Hardwareressourcen selbst zu unterhalten bzw. fest anzumieten, stattdessen kann im Idealfall vollkommen frei auf die benötigte Leistung zugegriffen werden [Carr05]. Von diesen hypothetischen Überlegungen abgesehen, besteht ein besonders einfach zu realisierender Vorteil aus Sicht des Nachfragers darin, dass die benötigten Systemressourcen nicht mehr nach dem Spitzenbedarf bemessen werden müssen. Stattdessen kann die Grid-Dienstleistung in Anspruch genommen werden, um kurzzeitige Leistungsbedarfsspitzen abzufangen, die eigentliche IT-Infrastruktur des Unternehmens kann reduziert werden, da sie nur den gewöhnlichen Bedarf decken muss. Diese Option erscheint bereits jetzt für viele Unternehmen als eine praktikable Möglichkeit, den IT-Bereich zu rationalisieren, ohne dauerhaft auf eventuell benötigte Leistungsreserven verzichten zu müssen [McKe05].

Der Fokus bei der Entwicklung zukünftiger *Grid-Computing*-Infrastrukturen liegt auf der effizienten Ausnutzung von verfügbaren Diensten und Ressourcen über öffentliche Netzwerke. Aus diesem Grund ist es eine Schlüsselanforderung an Grid-Infrastrukturen, skalierbare und sich der Umgebung anpassende Ressourcenmanagementsysteme zu unterstützen. Jedoch sind aktuelle Grid-Systeme lose verbundene Systeme, die von Natur aus sehr dynamisch sind, was neue



Herausforderungen an Ressourcenmanagementsysteme stellt [MoBa03]. Durch das dynamische Verhalten der Umgebung und die Delegation der Bereitstellung von Diensten und Ressourcen an einen Anbieter entsteht das Risiko, die vertraglich verhandelten und vereinbarten Garantien unvollständig zu bekommen. Wenn die zugesicherten Dienste nicht geliefert werden, d. h., es kommt zu keiner Dienstauführung auf den Grid-Ressourcen, entsteht der größte Schaden. So sind die Garantien in den Verträgen erst während oder nach Dienstonutzung überprüfbar. Dabei stellt sich in der Dienstaufwahl die Frage nach den Kriterien, nach denen der Konsument einen Anbieter selektiert, welches Risiko der Konsument bereit ist einzugehen und wie hoch sich die Kosten bei einer erneuten Allokation gestalten. Dabei soll mögliches Fehlverhalten (z. B. das Unterschreiten der vereinbarten Qualität) der Marktakteure in zukünftigen Transaktionen mit einbezogen werden.

Integriertes Risikomanagement ist notwendig, um auftretende technische wie auch ökonomische Risiken zu behandeln. Durch Einführung einer serviceorientierten Architektur und technischen Standards werden teilweise Risiken auf technischer Ebene angegangen und Lösungsansätze dargestellt, die einen Fremdbezug der IT-Dienste ermöglichen [Papa03]. Bisherige Untersuchungen fokussieren auf Sicherheitsrisiken, die durch den Fremdbezug von IT-Diensten entstehen können [Erne06]. Die Auswirkungen der technischen Risiken selbst – z. B. durch Benutzung eines öffentlichen Netzwerks – und deren Einfluss auf ökonomischer Ebene wird bisher kaum betrachtet. Für eine breite Umsetzung des Utility-Computing-Geschäftsmodells ist es jedoch unabdingbar, diese Risiken zu quantifizieren, um eine gleich bleibende Dienstqualität zu garantieren und mögliche Konsequenzen bei Nichteinhalten der vereinbarten Service-Level-Agreements zu behandeln.

## 1.2 Problemstellung und Relevanz der Arbeit

In den Wirtschaftswissenschaften sind verschiedene Verfahren bekannt, mit denen Risiken bewältigt werden können. Diese Aufgabe übernehmen häufig Versicherungen als Instrument, um ein bestimmtes Risiko zu transferieren. Die Versicherung ist eine vertrauenswürdige Institution, die das Risiko einer Vertragsverletzung begrenzt. Versicherungen folgen dem Grundprinzip der kollektiven Risikoübernahme [ZwEi07]: Viele zahlen einen Versicherungsbetrag in den Geldtopf der Versicherung ein, um beim Eintreten des Versicherungsfalles aus diesem Geldtopf einen Schadensausgleich zu erhalten.

Da der Versicherungsfall nur bei wenigen Versicherten eintreten wird, reicht der Geldtopf bei

bezahlbarem Beitrag aus. Wenn Versicherungen viele Versicherungsnehmer haben, dann können sie die Risiken breit verteilen und damit das Risiko des Einzelnen deutlich verringern. Sind die Risiken unabhängig voneinander und die Anzahl der Versicherungsnehmer ist groß, so können die individuellen Risiken effizient durch ein Versicherungsunternehmen beseitigt werden [MiRo92].

Für den Konsumenten von Utility-Computing-Diensten bietet das integrierte Risikomanagement eine Reihe von Vorteilen. So ist der Inhalt des Versicherungsvertrags ein Schutzversprechen des Versicherers, nach einem konkreten Schadensfall eine vertraglich vereinbarte Leistung zu erbringen. In bestehenden Ökonomien wird diese Leistung üblicherweise durch monetäre Kompensation erbracht [ZwEi07]. Diese Kompensationszahlung bietet dem Konsumenten die Möglichkeit, eine erneute Allokation der Ressourcen von einem anderen Anbieter durchzuführen, um seine Ziele zu erreichen. Neu ist die Möglichkeit in Grid-Ökonomien, dass die Versicherung als Kompensationsleistung Ersatzressourcen vermittelt. In diesem Fall werden Ressourcen zur Kompensation vorgehalten und im Schadensfall bereitgestellt, die die Ausführung des Dienstes übernehmen. Der Konsument kann in Grid-Ökonomien zwischen verschiedenen Modellen zur Kompensation seines Schadens wählen.

Aus Anbietersicht können Versicherungen von Grid-Transaktionen bestehende Risikomanagementpläne ergänzen. Nach einer Studie des AFCOM Data Center Instituts [Dunn06] werden bis 2010 fast 70% der Rechenzentren Grid-Computing-Technologien einsetzen, jedoch haben nur wenige Rechenzentren überhaupt Risikomanagementpläne, wobei diese sich hauptsächlich auf physikalische Hardwareschäden beziehen und keine Betrachtung der einzelnen Transaktionen durchführen. So kann das Konzept der Versicherungen Schäden durch Qualitätsabweichungen während der Dienstleistungserbringung übernehmen und diese durch monetäre oder nicht-monetäre Kompensation ersetzen. Dadurch wird eine Bewertung der Risiken durch den Konsumenten oder Anbieter ermöglicht.

Die zentralen Problemstellungen dieser Arbeit sind der Entwurf, die Implementierung und die Evaluierung eines Versicherungsmodells für Grid-Transaktionen. Der Begriff einer Grid-Transaktion wird im Rahmen dieser Arbeit analog zu einer Transaktion auf einem digitalen Marktplatz gesehen, der durch eine Grid-Ökonomie realisiert wird.

Eine Grid-Ökonomie beschreibt Verfahren und Methoden zum voll automatisierten Handel von Grid-Services und Grid-Ressourcen im Sinne des Utility-Computing-Paradigmas. Die traditionell in der Ökonomie behandelten Fragen der Produktion, der Allokation und des Konsums knapper Ressourcen werden hierbei auf Grid-Systeme übertragen [NeHO06] [ChKB05]. Die Ent-

scheidungen über Grid-Dienste und Grid-Ressourcen werden auf ökonomischer Basis getroffen [KeCh04]. Der Preis wird zum entscheidenden Faktor bei der Nutzung von Grid-Ressourcen [ChKe03]. Dies ist eine wichtige Voraussetzung für die Einführung einer Versicherung, da eine Grid-Ökonomie die Berechnung einer Versicherungsprämie erlaubt und damit den Grid-Risiken ein Wert zugeordnet werden kann. Die Grid-Dienste sind dabei meist standardisierte Leistungsversprechen, bei denen die Erfahrungseigenschaften dominieren und die Qualität des Grid-Services zum Zeitpunkt des Vertragsabschlusses noch nicht feststeht [BHHK08].

Die in der Grid-Ökonomie gehandelten IT-Dienste haben dabei die Eigenschaft, dass ihre Bereitstellung und Ausführung meist sofort und online<sup>1</sup> erfolgt. Findet keine sofortige Bereitstellung der Grid-Dienste statt, kann es durch die Dynamik eines Grid-Systems zu starken Schwankungen des Risikos bis zur Bereitstellung kommen, was eine Bewertung des Risikos erschwert. Aus diesem Grund müssen Risikomanagementverfahren für Grid-Systeme allgemein und die Versicherung im konkreten Fall als Dienste entworfen werden, die zum einen online und so weit wie möglich in Echtzeit ablaufen. Ein solches Modell ist neu und in keiner dem Autor bisher bekannten Arbeit betrachtet worden. Als Forschungsmethode kommt die Methode einer Simulationsstudie zum Einsatz, da bisher noch keine im Produktivbetrieb befindliche Grid-Ökonomie existiert.

Im Detail werden im Rahmen dieser Arbeit folgende Forschungsfragen untersucht:

- Forschungsfrage 1: Identifikation der technischen und ökonomischen Risiken im Grid-Computing

*Was sind die technischen und ökonomischen Risiken der Grid-Benutzer, die bei der Verwendung von Grid-Ressourcen auftreten können?*

Um diese Frage zu beantworten, wird eine Literaturrecherche nach technischen und ökonomischen Risikoquellen durchgeführt. Es werden verschiedene Grid-Anwendungsfelder und Organisationsformen betrachtet und die identifizierten Risiken hinsichtlich ihrer Versicherbarkeit überprüft. Weiterhin werden in dieser Forschungsfrage Modelle zur Quantifizierung der Risiken vorgestellt, die die Grundlage für das Versicherungsmodell bilden. Diese umfassen technische Gegebenheiten eines Grid-Systems, die Auswirkungen auf das Risiko eines Einsatzes von Grid-Systemen haben und damit entscheidend die zu erwartete Dienstleistungsqualität beeinflussen. Am Ende dieser Analysephase stehen die Anforderungen an das Versicherungsmodell fest.

---

<sup>1</sup>Der Begriff *online* bedeutet in diesem Zusammenhang, dass nicht sämtliche Daten eines Problems vorab bekannt sind. Entscheidungen müssen auf unvollständigem Wissen getroffen werden.

- Forschungsfrage 2: Entwurf eines Versicherungsmodells für Grid-Transaktionen  
*Welche Versicherungsmodelle eignen sich für die in Forschungsfrage 1 identifizierten und quantifizierten Risiken?*

Hinsichtlich der Anforderungen an die Versicherung werden verschiedene Versicherungsmodelle und Szenarien zum Einsatz von Versicherungen in Grid-Systemen analysiert. Ein auf gridspezifischen Risikomerkmale basierendes Prämienberechnungsmodell stellt das grundlegende Modell der Grid-Versicherung dar. Hier wird auf ein statistisches Verfahren zur Prämienanpassung zurückgegriffen wie es in der deutschen Kraftfahrzeugversicherung eingesetzt wird [Meye05]. Anhand eines Szenarios wird exemplarisch die Berechnung und Anpassung einer Versicherungsprämie vorgestellt.

- Forschungsfrage 3: Implementierung des Versicherungsmodells in ein Softwaresystem  
*Wie lassen sich Versicherungsnachfrage und Versicherungsangebot voll automatisiert für technische Risiken realisieren?*

Die Umsetzung in ein rechnergestütztes Simulationsmodell für die Versicherung technischer Risiken ist das Ziel dieser Forschungsfrage. Das in der vorangegangenen Modellierungsphase entworfene Modell wird in einem Grid-Simulator umgesetzt, der eine Grid-Ökonomie simulieren kann. Algorithmen und Datenstrukturen werden in diesem Schritt entworfen und implementiert und es damit ermöglichen, die Versicherungsnachfrage und das Versicherungsangebot voll automatisiert ablaufen zu lassen. Die Implementierung der Versicherungsnachfrage orientiert sich dabei an der Risikoneigung menschlicher Nachfrager. Das Versicherungsangebot leitet sich aus den technischen Risiken ab, die durch den Einsatz von Grid-Systemen gegeben sind.

Neben dem Versicherungsangebot wird auch ein Modell der Versicherungsnachfrage entworfen. Dieses Modell basiert auf der Erwartungsnutzentheorie, die in der Entscheidungstheorie weit verbreitet ist und für Entscheidungen unter Unsicherheit verwendet wird. Abhängig von seiner individuellen Risikoneigung evaluiert ein potenzieller Versicherungsnehmer, ob er eine Versicherung abschließt oder nicht. Als Implementierungstechnologie kommen Multi-Agenten-Systeme zum Einsatz.

- Forschungsfrage 4: Evaluation der Versicherungsimplementierung  
*Welchen Einfluss hat die Einführung einer Versicherung auf die Performanz des Ressourcenmanagementsystems?*

Die Evaluation der Versicherung ist zweigeteilt. Zum einen wird der Einfluss der Ver-

sicherung auf das ausgewählte ökonomische Ressourcenallokationsverfahren untersucht und zum anderen werden Versicherungsangebot- und nachfrage analysiert. Kommen im ersten Fall gridspezifische Performanzkennzahlen zum Einsatz, so werden im zweiten Fall Metriken der Versicherungsökonomie für die Analyse herangezogen.

Das implementierte Modell wird mithilfe von verschiedenen Simulationsszenarien evaluiert, die charakteristische Eigenschaften real eingesetzter Grid-Systeme aufweisen. Als Analyseverfahren der gemessenen Metriken kommen sowohl Zeitserien als auch verschiedene Aggregationsebenen einer Metrikpyramide zum Einsatz. Dies deckt ein weites Feld der Analysemöglichkeiten von Simulationsstudien ab.

Im folgenden Abschnitt wird die Forschungsmethode *Simulation* eingeführt und hinsichtlich ihrer Vor- und Nachteile diskutiert.

### 1.3 Die Forschungsmethode Simulation

Die Forschungsmethode Simulation ist eine spezielle Art des Modellierens, die eine Vereinfachung eines Systems nachbildet [GiTr05, S. 2]. Ein System ist definiert als eine Ansammlung von Entitäten, die handeln und miteinander interagieren können, um ein bestimmtes Ziel zu erreichen [ScTa70]. Dabei verwenden Simulationen für die Nachbildung einen Computer, um zum einen das Modell *numerisch* zu evaluieren und zum anderen Daten zu sammeln, um die erwünschte Charakteristika des Modells *abzuschätzen* [Law07, S. 1]. Die Zielsetzung einer Simulation ist die Nachbildung eines Systems über ein bestimmtes Zeitintervall [Bank98]. Der Zustand eines Systems wird durch eine Ansammlung von Variablen während eines Simulationslaufs repräsentiert. Diese Variablen werden benötigt, um ein System zu einem bestimmten Zeitpunkt zu beschreiben [Law07, S. 3].

Im Kontext von Grid-Systemen werden Simulationen eingesetzt, um neuartige Grid-Systemkomponenten wie das integrierte Risikomanagement mit einer Grid-Versicherung zu simulieren und eine Sensitivitätsanalyse über die Verhaltensweise des simulierten Modells durchzuführen. Die Ergebnisse von Simulationen helfen dabei, zukünftige Grid-Systeme zu planen und umzusetzen. Die folgenden Abschnitte geben einen Überblick über die grundlegenden Konzepte von Simulation als Forschungsmethode und Techniken zur Durchführung einer Simulationsstudie.

Sowohl in den Sozialwissenschaften [Axe197] [Greg06] wie auch in der Informatik bzw. Wirtschaftsinformatik [MaSm95] [WiHe06] [GIRV04] ist die Anwendung von Simulationen eine

anerkannte Forschungsmethode. Simulationen können dabei zum Zweck der Vorhersage, des Beweises oder der Suche nach neuen Theorien eingesetzt werden [Greg06] [Axel97] [GiTr05, S. 4ff].

Die Simulation nimmt ein Modell, das zusammengesetzt ist aus der Struktur des Modells und den Regeln, die das Modell steuern, als Eingabe und produziert eine Ausgabe durch Überwachung des Systemverhaltens. Durch den Vergleich verschiedener Ergebnisse, die mit unterschiedlichen Strukturen und Steuerungsregeln erzeugt wurden, kann darauf geschlossen werden, was in einer realen Situation passieren würde, wenn ein solches Ereignis auftritt [Dool02].

### 1.3.1 Abgrenzung von analytischer Forschungsmethode und realem Experiment

Neben der in der Einleitung bereits erwähnten Möglichkeit einer rein analytischen Betrachtung ist die Anwendung eines Experiments an einem realen Grid-System eine weitere weit verbreitete Forschungsmethode für Grid-Systeme. Diese beiden werden im Folgenden von der Forschungsmethode Simulation abgegrenzt und hinsichtlich ihrer Stärken und Schwächen zur Evaluierung des Grid-Versicherungsmodells untersucht. Üblicherweise ist eine analytische Betrachtung der Simulation vorgeschaltet, während die Experimente an realen Grid-Systemen auf Simulationen folgen [BCCD<sup>+</sup>06].

**Analytische Forschungsmethode** Ein rein analytischer oder mathematischer Ansatz erlaubt den Beweis eines Theorems und ist dadurch nützlich, um grundlegende Prinzipien von Grid-Systemen zu verstehen. Dabei stellt es sich häufig heraus, dass viele Probleme wie die Zuweisung von Grid-Ressourcen oder das Routing von Datenpaketen zwischen Grid-Standorten NP-vollständig sind und aus diesem Grund nur für kleine Grid-Systeme überhaupt lösbar [Schn07].

Dies hat zur Konsequenz, dass die Anwendung eines analytischen Ansatzes ein sehr abstraktes Modell erfordert, das weit von den tatsächlich gegebenen Eigenschaften eines Grid-Systems entfernt ist und dadurch nur eine beschränkte Relevanz für den praktischen Einsatz in Grid-Systemen hat. Jedoch bilden Ergebnisse analytischer Betrachtungen häufig die Grundlage von Simulationen. Beispielsweise ist das Marginalsummenverfahren als Schätzverfahren der Grid-Risiken durch eine rein analytische Betrachtung untersucht und bewiesen worden und wird in der Simulation der Grid-Versicherung eingesetzt. Allerdings lässt eine gleichzeitige Betrachtung der verschiedenen Ebenen des Grid-Modells eine mathematische Berechnung als sehr schwierig

erscheinen, sodass hier die Anwendung von Simulationen vorteilhafter erscheint.

**Experiment an einem realen Grid-System** Experimente an realen Grid-Systemen sind die in der Grid-Forschung am häufigsten anzutreffende Forschungsmethode. Durch ihren Charakter – ein Experiment an einem realen Grid-System durchgeführt zu haben – sind sie glaubwürdig und zeigen, dass der Ansatz in der Praxis eingesetzt werden kann. Jedoch weisen sie eine Reihe von Nachteilen auf:

- Hohe Zeitdauer zur Durchführung der Experimente notwendig: Die Ausführung der Anwendungen erfolgt in Echtzeit, sodass ein Experiment Tage, Wochen oder Monate dauern kann.
- Arbeitsintensität: Sowohl die Anwendung als auch die Experimentumgebung müssen in ein reales System integriert werden. Dies erfordert einen erhöhten Implementierungs- und Testaufwand der Algorithmen und Experimente auf der Zielplattform, um die Korrektheit zu verifizieren.
- Testumgebung: Als Testumgebung kann entweder eine Laborumgebung, die das Verhalten des Systems emuliert oder eine Testumgebung im Produktivbetrieb ausgewählt werden. Die Laborumgebung weist Charakteristika einer kontrollierten, stabilen Umgebung mit einem wohl definierten Verhalten auf. Sie hat allerdings den Nachteil, nur wenig mit realen Grid-Systemen, die sich im Produktivbetrieb befinden, gemeinsam zu haben.

Die Auswahl eines realen, im Produktivbetrieb befindlichen Grid-Systems, bringt ebenfalls eine Reihe von Nachteilen mit sich, da sich das Experiment die Ressourcen des Systems mit anderen Benutzern teilen muss. Jeder Experimentlauf findet eine andere Umgebung vor, sodass die Ergebnisse der einzelnen Experimente, selbst mit gleichen Parametern nicht vergleichbar sind. Weiterhin können Fehler auftreten, die nicht oder nur wenig mit dem Experiment selbst zu tun haben, sondern durch andere Prozesse und Benutzer entstehen können. Nicht kontrollierbare Einflüsse können den zu untersuchenden Effekt überlagern und somit die Resultate verfälschen. Zusätzlich entsteht eine hohe Abhängigkeit zu einem Grid-System, was verhindert, belastbare statistische Aussagen über das untersuchte Verfahren zu erhalten und nur eine bedingte Überprüfung der Aussagen durch andere Forscher gestattet.

So haben prototypische Experimente mit dem vorliegenden Modell ergeben, dass Zusammenhänge aufgrund der sehr vielen Einflussfaktoren eines Grid-Systems nicht mehr be-



stimmt werden können. Auch die Installation, das Aufsetzen einer Testumgebung und die wiederholten Messungen sind sehr zeitaufwendig.

Im konkreten Fall der Evaluierung der Grid-Versicherung gibt es kein Produktivsystem einer Grid-Ökonomie, sodass die Forschungsmethode eines Experiments an einem realen Grid-System nicht angewandt werden kann. Im Gegensatz dazu bieten Simulationen die Möglichkeit, Erkenntnisse aus analytischen Betrachtungen zu verwenden und diese in einem abstrahierten Grid-System unter verschiedenen Szenarien zu evaluieren. Damit können Zeit und Kosten gespart werden, die bei Experimenten an realen Grid-Systemen berücksichtigt werden müssen. Die Ergebnisse aus Simulationen können schließlich beim Prototypenbau verwendet werden.

### 1.3.2 Vorteile und Nachteile von Simulationen

Obwohl in den vorangegangenen Abschnitten Simulationen als eine passende Forschungsmethode für die Evaluation der Grid-Versicherung in einer Grid-Ökonomie identifiziert wurden, hat die Forschungsmethode auch eine Reihe von Nachteilen. Die folgende Diskussion gibt einen kurzen Überblick über Vor- und Nachteile von Simulationen im zu untersuchenden Grid-Modell:

- Vorteile: Ein wichtiger Vorteil des Simulationsansatzes ist die Möglichkeit, Simulationen in einer kontrollierten Umgebung durchführen zu können und dadurch sicherzustellen, dass die Experimente immer wieder mit der gleichen Systemkonfiguration ausgeführt werden können. Dadurch ist eine Nachvollziehbarkeit der Ergebnisse durch andere Forscher gegeben, die in die Lage versetzt werden, ebenfalls diese Experimente durchzuführen. Einzelne Effekte durch veränderte Konfiguration der Grid-Versicherung oder des Grid-Systems lassen sich schnell nachprüfen und führen zu einer erhöhten Geschwindigkeit bei der Analyse verschiedener Szenarien. So können im gleichen Zeitabschnitt mehr Szenarien und Systemkonfigurationen analysiert werden als dies beispielsweise bei Experimenten an realen Grid-Systemen der Fall wäre. Neue, innovative Technologien wie die Grid-Versicherung können so in kurzer Zeitdauer analysiert werden.
- Nachteile: Die Modellierungs- und Implementierungsschritte können bei komplexen Simulationen fehlerbehaftet sein, sodass die Verifikation des Modells falsch sein kann und damit zu falschen Schlussfolgerungen führen kann. Aus diesem Grund ist es wichtig, das Modell auf einem validierten Modell aufzusetzen, sodass dies möglichst verhindert wird. Eine Überprüfung des Verhaltens der Grid-Versicherung muss ein ähnliches Verhalten wie real existierende Versicherungen zeigen, sodass die Aussagen der Simulationen mehr Ge-



wicht bekommen.

Zusammengefasst sind Simulationen nützlich für die Evaluierung der Grid-Versicherung in einer Grid-Ökonomie. Obwohl sie einige Nachteile hat, ist die Anwendung von Simulationen aktuell die einzige praktikable Möglichkeit, die Grid-Versicherung zu evaluieren. Die Ergebnisse der Simulation können Hinweise darauf geben, wie ein System einzustellen ist, damit die Grid-Versicherung auch in einem realen Grid-System ähnliche Wirkungsweisen zeigt.

## 1.4 Aufbau der Arbeit

Der Aufbau der Arbeit orientiert sich an den Forschungsfragen, die diese Arbeit betrachtet. Abbildung 1.2 zeigt den Gang der Untersuchung. Kapitel 2 gibt eine Einführung in den Aufbau von Grid-Systemen, deren Anwendungsgebiete und Organisationsformen sowie ihre Risiken, die die Verfügbarkeit und die Zuverlässigkeit von Grid-Systemen einschränken. Messergebnisse verschiedener empirischer Studien werden mit dem Ziel vorgestellt, diese Risiken quantitativ beschreiben zu können.

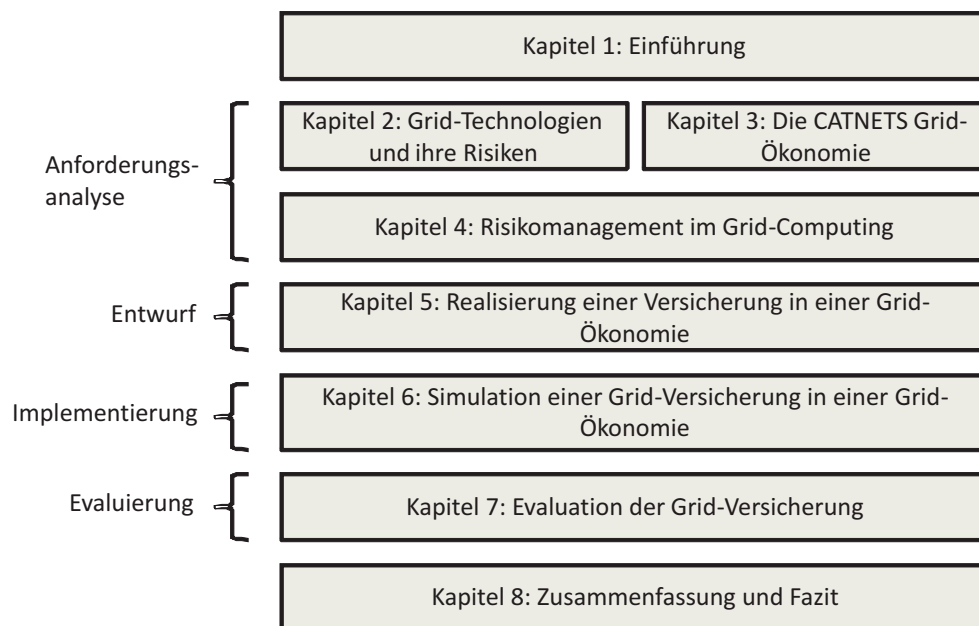


Abbildung 1.2: Gang der Untersuchung

Das anschließende Kapitel 3 gibt einen Überblick über aktuelle Grid-Ökonomien und ihre Risiken. Ausführlich werden in diesem Kapitel das Modell und die Umsetzung der CATNETS Grid-Ökonomie vorgestellt, die die Grundlage für die Simulationsstudie bildet.

Bestehende Risikomanagementarbeiten in Grid-Ökonomien auf Grundlage von Service-Level-Agreements stellt Kapitel 4 vor. Eine Abgrenzung bestehender Arbeiten von dieser Arbeit steht im Fokus dieses Kapitels.

Das Versicherungsmodell für Grid-Ökonomien beinhaltet Kapitel 5. Es baut dabei auf den Erkenntnissen der vorangegangenen Kapitel auf und verwendet diese bei der Umsetzung des Versicherungsmodells. Die Implementierung der Versicherung in die CATNETS Grid-Ökonomie zeigt Kapitel 6. Ausführlich wird der allgemeine Simulationsablauf, dessen Kalibrierung und das Vorgehen zur Verifikation der Implementierung durch Testläufe beschrieben.

Die Evaluierung der Versicherung unter Verwendung verschiedener technischer und ökonomischer Metriken präsentiert Kapitel 7. Anhand ausgewählter Grid-Szenarien wird eine Analyse mithilfe von Zeitreihen und einer Metrikpyramide durchgeführt. Die Ergebnisse werden mit Tabellen und Grafiken aufbereitet.

Kapitel 8 fasst die Ergebnisse zusammen und gibt einen Überblick über Implikationen für die weitere Forschung und Anwendungsmöglichkeiten in der Praxis.

## 1.5 Glossar des Risikobegriffs

Dieses Glossar definiert den Begriff des Risikos aus Sicht dieser Arbeit, da der Begriff in der wissenschaftlichen Literatur nicht immer deckungsgleich verwendet und meist speziell für eine Anwendungsdomäne definiert wird [Mein97, S. 22ff.]. Ein weiteres Ziel des Glossar ist es, den Begriff des *technischen Risikos* in den Kontext der Unternehmensrisiken einzuordnen, da dieser Begriff im Fokus der vorliegenden Arbeit steht. Abbildung 1.3 gibt eine Übersicht über die Risiken, die in diesem Glossar besprochen werden.

- Risiko: Die vorliegende Arbeit folgt der Risikodefinition von Knight [Knig21], der unter Risiko messbare Eintrittswahrscheinlichkeiten von Ereignissen versteht. Da es sich bei IT-Risiken um ausschließlich Verlustereignisse handelt, wird in dieser Arbeit nur das reine Risiko, also das Risiko mit negativen Folgen, betrachtet [Prok08, S. 8]. Risiken mit negativen Folgen stehen ebenfalls im Fokus von Versicherungen [ZwEi07, S. 2], die in dieser Arbeit betrachtet werden.
- Finanzwirtschaftliche Risiken: Finanzwirtschaftliche Risiken beziehen sich auf die Finanzströme eines Unternehmens und umfassen sämtliche Risiken des Finanzbereichs eines Unternehmens [Wied03].

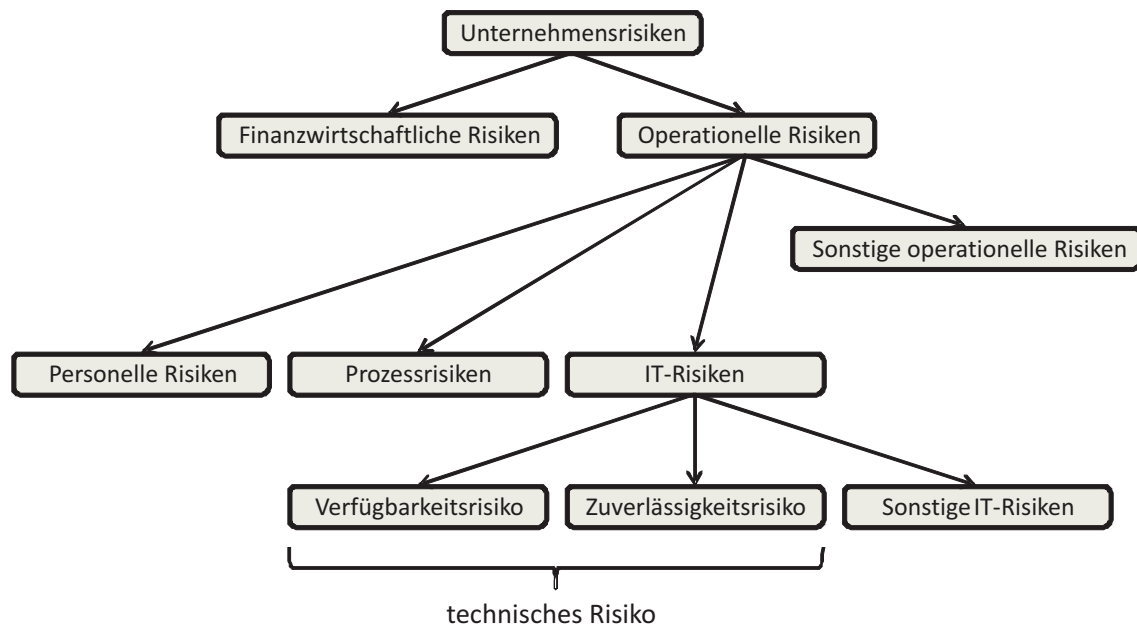


Abbildung 1.3: Einordnung des technischen Risikos in den Kontext der Unternehmensrisiken [Seib06] [Wolk07] [Prok08]

- **Operationelle Risiken:** Unter dem Begriff operationelles Risiko werden sämtliche Risiken zusammengefasst, die aus dem Geschäftsablauf entstehen und einen Verlust herbeiführen können [Seib06, S. 9] [Prok08, S. 10]. Das operationelle Risiko umfasst Personenrisiken, Prozessrisiken und IT-Risiken, die sowohl unternehmensintern als auch unternehmensextern – im Fall von Fremdbezug der IT-Dienste – auftreten können [Seib06, S. 16ff.] [Wolk07, S. 206ff.] [Schi01, S. 336ff.].
- **Personenrisiken:** Personenrisiken sind Risiken, die durch Verluste aufgrund menschlichen Versagens entstanden sind [Wolk07, S. 206].
- **Prozessrisiken:** Zu den Prozessrisiken gehören Verluste, die durch Störungen des Geschäftsablaufs entstehen und nicht auf Personenrisiken und IT-Risiken beruhen [Wolk07, S. 207].
- **IT-Risiko:** Unter einem IT-Risiko versteht man die Gefahr, anforderungsgerechte IT-Leistungen nicht erbringen zu können [Seib06, S. 11]. Dies schließt Verfügbarkeits- und Zuverlässigkeitsrisiken der IT-Systeme und Softwareanwendungen mit ein [Wolk07, S. 207].
- **Verfügbarkeitsrisiken:** Das Verfügbarkeitsrisiko bezieht sich auf mögliche Schäden durch Nichtverfügbarkeit von IT-Systemen – dazu zählen physische Ressourcen wie Netzwerke und Hardware – und Softwareanwendungen [BaBL02] [RoWe03] [Wolk07, S. 207].

- 
- **Zuverlässigkeitsrisiken:** Das Zuverlässigkeitsrisiko bezeichnet Verlustereignisse, die während der Nutzung der IT-Systeme und der Softwareanwendungen auftreten [YaAm01] [RoWe03].
  - **Technisches Risiko:** Das technische Risiko, das in dieser Arbeit im Mittelpunkt steht, bezeichnet mögliche Schäden sowohl durch die Nichtverfügbarkeit als auch durch die mangelhafte Zuverlässigkeit der betrachteten Grid- und Cloud-Computing-Systeme.



## Kapitel 2

# Grid-Technologien und ihre Risiken

*„In a Grid, resource failure is the rule rather than the exception.“ [BaBL02]*

In den letzten Jahren wurden Grid-Systeme durch eine wachsende Anzahl neuer Technologien, die geografisch verteiltes Rechnen ermöglichen, weiterentwickelt [FoKT01] [JoEF04]. Zum einen wurden Netzwerkkomponenten und Rechnerhardware virtualisiert [BDFH<sup>+</sup>03] [APST05] und zum anderen wurden verschiedene Middlewareprodukte und Programmierbibliotheken entwickelt, die es ermöglichen, dass Softwareanwendungen verteilte Rechenkapazitäten nutzen können [FoTu05] [Weis07]. Dies wurde durch den Entwurf von flexiblen Architekturen ermöglicht, die auf technischer Ebene heterogene Ressourcen verbinden und auf sozialer Ebene eine organisatorische Gliederung zulassen [FoTu05]. Mit der Größe der Grid-Systeme und ihrer verteilten Struktur gehen technische Risiken einher, die die Zuverlässigkeit des Grid-Systems beeinträchtigen und die damit verbundene Qualität der angebotenen Dienste [Dabr08]. Mit der Verwendung von Grid-Computing als Outsourcing-Plattform für dynamische Geschäftsprozesse können, ausgehend von den technischen Risiken, ökonomische Risiken entstehen [KeCh02] [RoWe04]. Ziel dieses Kapitels ist die Einführung der aktuellen Grid-Architektur und die Identifikation der technischen Risikoquellen von Grid-Systemen.

Anhand eines Application-Service-Provider-Szenarios wird ein beispielhafter Ablauf der Bereitstellung einer Anwendung unter Verwendung eines Grid-Systems gezeigt. Ein Application-Service-Provider bietet Anwendungen wie Enterprise-Resource-Planning (ERP)-Systeme oder Textverarbeitungsanwendungen über ein öffentliches Netzwerk an [OISS07]. Der Anbieter kümmert sich dabei um den Betrieb des Systems; der Konsument mietet die Software bei einem Serviceanbieter, anstatt wie bisher die Software zu kaufen und selbst die Wartung und den Betrieb durchzuführen.

Abbildung 2.1 zeigt den konzeptionellen Aufbau eines ASP-Szenarios. Ein Endbenutzer fragt eine bestimmte Anwendung, die er nutzen möchte, bei einem Serviceanbieter nach. Der Anbieter teilt ihm die Dienstgüte mit und beide einigen sich auf eine Dienstgütevereinbarung, ein Service-Level-Agreement. Der Kunde bezahlt anschließend den Dienst. Für die Bereitstellung der Anwendung benötigt der Application-Service-Provider zwei verschiedene Ressourcen: zum einen die Lizenz für die Software und das Softwarepaket zur Installation, die er vom Hersteller der Software bezieht und zum anderen ein Ressourcenbündel von einem Ressourcenanbieter zur Ausführung der Anwendung. Für dieses Ressourcenbündel vereinbart der Serviceanbieter ein Service-Level-Agreement zur Festlegung der Dienstgüte der Ressourcen. Sind sowohl Lizenzen als auch das Ressourcenbündel bereitgestellt und die Anwendung konfiguriert, wird der Dienst dem Konsumenten zur Verfügung gestellt.

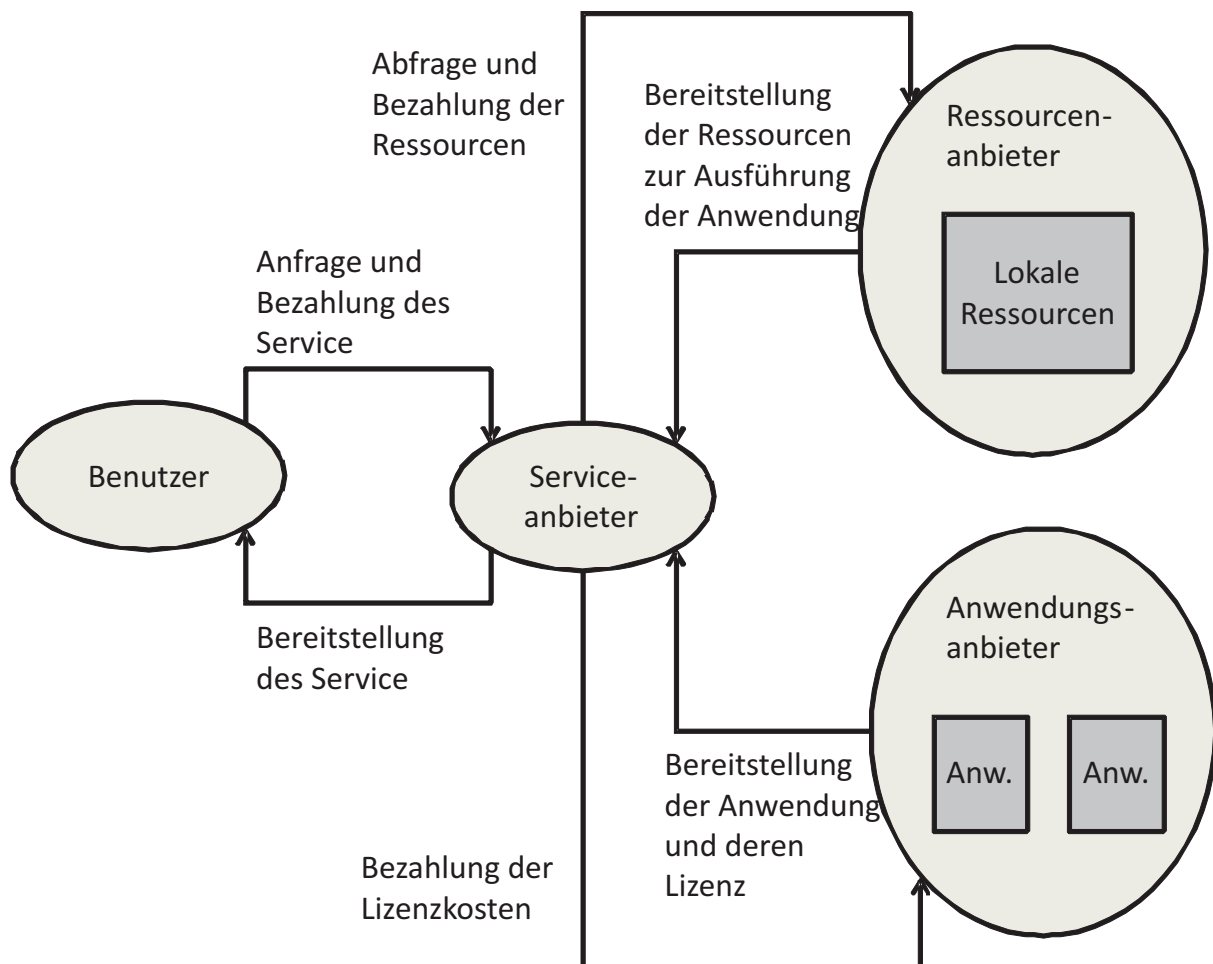


Abbildung 2.1: Konzeptionelles Modell eines Application-Service-Provider-Geschäftsmodells unter Verwendung eines Grid-Systems [OISS07]

In den folgenden Abschnitten werden detailliert die Technologien, die bei Umsetzung des oben beispielhaft dargestellten Szenarios zum Einsatz kommen, und die technischen Risiken, die durch den Fremdbezug einer Anwendung auftreten können, beschrieben. Den Anfang macht Kapitel 2.1, das den Aufbau aktueller Grid-Infrastrukturen vorstellt. Ausgehend von den allgemeinen Entwurfsprinzipien von Grid-Systemen, werden das Schichtenmodell der Grid-Architektur vorgestellt und Technologien zur Umsetzung dieser Architektur eingeführt.

Anhand der Open-Grid-Services-Architektur werden in Abschnitt 2.2 wichtige Begriffe der serviceorientierten Architektur eingeführt. Service-Level-Agreements als vertragliche Grundlagen für den kommerziellen Bezug von Grid-Diensten werden in Kapitel 2.3 vorgestellt. In den beiden folgenden Abschnitten, Abschnitt 2.4 und Abschnitt 2.5, werden typische Anwendungsfelder und Organisationsformen von Grid-Systemen beschrieben. Die Einsatzgebiete reichen von der Bereitstellung von Rechenleistungen bis hin zur Ausführung von Geschäftsprozessen auf Grid-Systemen. Die Größe der Grid-Systeme bzw. ihre Organisationsform kann sowohl einige wenige Rechner innerhalb eines Enterprise-Grids als auch mehrere hunderttausend Rechner eines Open-Grids umfassen. Aus den Anwendungsfeldern und Organisationsformen werden Risiken in Sektion 2.6 abgeleitet und klassifiziert, die signifikanten Einfluss auf die Verfügbarkeit und Leistungsfähigkeit eines Grid-Systems haben. Weiterhin werden aktuelle Modelle zur Beschreibung der technischen Risiken vorgestellt, die als Grundlage für die folgenden Kapitel dienen. Kapitel 2.7 fasst die Ergebnisse der Analyse der technischen Risiken von Grid-Systemen zusammen.

## 2.1 Aufbau von Grid-Systemen

### 2.1.1 Allgemeine Entwurfsprinzipien

Grid-Systeme bieten grundlegende Konzepte, um effizient verteilte Systeme und Anwendungen zu entwickeln [FoKe03]. Im Gegensatz zum klassischen Client-Server-Modell, in dem die Rollen *Client* und *Server* streng getrennt sind, ist es in Grid-Systemen möglich, jedem Teilnehmer eine Client- oder Server-Rolle zuzuordnen, abhängig davon, welche Funktion der Teilnehmer aktuell auf dem Netzwerk ausübt [TaTr03]. In dem am Beginn des Kapitels beschriebenen ASP-Szenario besitzt beispielsweise der Serviceanbieter eine Server-Rolle gegenüber dem Benutzer und eine Client-Rolle gegenüber dem Anwendungsanbieter und dem Ressourcenanbieter.

Des Weiteren besitzt das klassische Client-Server-Modell eine zentrale Architektur, die bei einer großen Anzahl an Clients zum Flaschenhals werden kann [TaTr03]. Auch dies wird bei dem Ent-



wurf von Grid-Systemen berücksichtigt, wobei in Grid-Systemen entweder eine hierarchische [KrBM02] oder eine dezentrale Architektur [ERSR<sup>+</sup>05] zum Einsatz kommt. Zusammenfassend bietet ein Grid-System einem Grid-Benutzer folgende 4 wichtige Merkmale an [FoKe99] [BaBL02]:

- *Mehrere administrative Domänen und lokale Autonomie:* Grid-Ressourcen sind geografisch über mehrere administrative Domänen verteilt und befinden sich im Besitz von verschiedenen Organisationen. Das Ressourcenmanagement des Grid-Systems bietet eine sichere Möglichkeit, geografisch verteilte Ressourcen zu überwachen und zu verwalten. Zusätzlich wird die Autonomie der Ressourcenbesitzer hinsichtlich ihres lokalen Ressourcenmanagements und ihrer Benutzungsrichtlinien berücksichtigt. Dies bedeutet, dass die Zuweisung der lokalen Ressourcen sich außerhalb des Einflussbereichs des Grid-Systems befindet. Das globale Ressourcenmanagement des Grids abstrahiert davon, indem es logische Grid-Ressourcen zuweist, die autonom vom lokalen Ressourcenmanagement verwaltet werden.

Im obigen ASP-Beispiel bleibt es dem Ressourcenanbieter selbst überlassen, welche Ressourcen er zur Ausführung der Anwendung zuweist, solange die vereinbarte Dienstgüte eingehalten wird. Er weist dem Serviceanbieter eine logische Ressource zu, die er intern mit seinem lokalen Ressourcenmanagement verwaltet.

- *Heterogenität:* Ein Grid-System bezieht eine Vielzahl an Ressourcen mit ein, die von Natur aus heterogen sind und eine breite Anzahl an Technologien umfassen. Ein Grid-System abstrahiert von den gegebenen physikalischen Ressourcen und Technologien mithilfe des Konzepts der Virtualisierung. Physikalische Ressourcen werden mithilfe der Methoden der Virtualisierung in einzelne Teilressourcen aufgeteilt, die als eigenständige Ressourcen behandelt werden können.
- *Skalierbarkeit:* Ein Grid kann von einigen wenigen Ressourcen bis hin zu mehreren Millionen wachsen. Dies wirft das Problem eines möglichen Leistungsverlusts bei zunehmender Größe auf. In der Konsequenz müssen die Anwendungen, die eine solche hohe Zahl an geografisch verteilten Ressourcen benötigen, tolerant gegenüber Fehlern sein. Gleichzeitig müssen das Ressourcenmanagement und weitere Verwaltungskomponenten im gleichen Maß skalieren, um den Leistungsverlust so gering wie möglich zu halten.
- *Anpassungsfähigkeit:* In Grid-Systemen ist der Ausfall einer Ressource eher die Regel als die Ausnahme [BaBL02]. Bei einer großen Zahl an Ressourcen sind auch deren Ausfallraten als hoch anzunehmen. Ressourcenmanagementsysteme und Anwendungen müssen

dynamisch ihr Verhalten anpassen und die verfügbaren Ressourcen und Dienste effizient und effektiv nutzen.

Diese vier Hauptaspekte eines Grid-Systems wurden durch eine Umfrage im Jahr 2006 unter 40 Grid-Forschern bestätigt [Stoc07], sodass sie immer noch als aktuell angenommen werden können. Die Forscher wurden dabei gebeten, ihre Definition eines Grid-Systems in Form eines kurzen Textes zu verfassen. Diese Definitionen wurden analysiert und in verschiedene Kategorien eingeteilt. Neben den vier Hauptaspekten eines Grid-Systems wurden auch verschiedene Anwendungs- und Organisationsformen genannt, über die sich Grid-Systeme definieren lassen und die in den Abschnitten 2.4 und 2.5 eingeführt und hinsichtlich ihres technischen Risikos untersucht werden.

Um die vier Hauptaspekte eines Grid-Systems umzusetzen, wird in den folgenden Abschnitten das Grid-Architekturmodell eingeführt, das den Betrieb und die Verwaltung von gemeinsam genutzten Ressourcen zwischen verschiedenen Organisationen ermöglicht. Diese Architektur identifiziert die grundlegenden Komponenten eines Grid-Systems. Die Grid-Architektur legt den Zweck und die Funktionen dieser Komponenten fest und zeigt auf abstrakter Ebene, wie diese Komponenten untereinander agieren [FoKT01].

### 2.1.2 Schichtenmodell der Grid-Architektur

Um die Komplexität eines Grid-Systems zu bewältigen, wird die Grid-Architektur in Schichten gegliedert. Die einzelnen Schichten stellen Abstraktionsebenen dar, mit denen eine einheitliche Modelldefinition möglich ist. Die Gliederung in Schichten bietet die Möglichkeit, Funktionen der Grid-Systemkomponenten zusammenzufassen und über eine einheitliche Schnittstelle anderen, darüberliegenden Schichten als Dienst anzubieten [PeDa00]. Einzelne Schichten lassen sich dabei unabhängig von anderen Schichten realisieren. Analog zu der Internet-Protokollarchitektur [Bake95], die die Netzwerkarchitektur des Internets beschreibt, lässt sich auch ein Grid-System in mehrere Schichten untergliedern.

Abbildung 2.2 zeigt die Schichten der Grid-Architektur in Vergleich zur Internet-Protokollarchitektur. Grid-Systeme lassen sich in 5 Schichten untergliedern [FoKT01] [JoEF04] [ABKN<sup>+</sup>06]:

- **Ressourcengeflecht:** Die unterste Schicht der Grid-Architektur bildet ein Geflecht aus verschiedenen lokalen, vernetzte Ressourcen, deren Zugriff über Protokolle des Grid-Systems erfolgen soll. Unter der Bezeichnung *lokale Ressource* ist die Zugehörigkeit zu einer Organisation gemeint, die diese Ressourcen besitzt und betreibt. Ressourcen umfassen Re-

chenressourcen, Netzwerke, Datenspeicher, Datenquellen, und Sensoren, die miteinander über ein Netzwerk verbunden sind [BaBL02]. Dabei kann eine Ressource auch eine logische Einheit sein, deren Eigenschaften allerdings außerhalb des Grid-Systems anzusiedeln ist.

Die Ressourcengeflecht-Schicht abstrahiert von der physikalischen Ressource und stellt den darüberliegenden Schichten Schnittstellen bereit, die unabhängig von den Eigenschaften der Ressource sind. Dies entspricht der Netzzugangsschicht der Internet-Protokollarchitektur. Auch hier wird von der konkreten physikalischen Netzwerkinfrastruktur abstrahiert.

Komponenten auf der Ressourcengeflecht-Schicht führen ressourcenspezifische Operationen in Abhängigkeit davon aus, um welchen Ressourcentyp es sich handelt. Für Rechenressourcen werden beispielsweise grundlegende Mechanismen zum Starten von Programmen und die Überwachung der ausführenden Prozesse gefordert. Sind die Ressourcen Datenspeicher, so werden Verfahren zum Speichern und Lesen von Dateien benötigt. Zusätzlich werden grundlegende Verfahren zur Ressourcenverwaltung für den jeweiligen Ressourcentyp bereitgestellt. Häufig übernehmen diese Aufgaben die Betriebssysteme der Ressourcen.

- **Verbindungsschicht:** Aufgabe der Verbindungsschicht ist die Bereitstellung von Protokollen zur Kommunikation und zur Authentifizierung, die für gridspezifische Netzwerktransaktionen benötigt werden. Kommunikationsprotokolle ermöglichen den Datenaustausch zwischen den verschiedenen Ressourcen des Ressourcengeflechts. Grid-Systeme abstrahieren hier wiederum von den konkreten Kommunikationsprotokollen, verwenden jedoch häufig die Protokolle der Internet-Protokollarchitektur. Authentifizierungsprotokolle bauen auf den Kommunikationsdiensten auf, um kryptografisch sichere Mechanismen für die Verifikation der Benutzer- und Ressourcenidentität bereitzustellen. Die Verbindungsschicht der Grid-Architektur entspricht den Schichten zur Vermittlung und zum Verbindungsaufbau der Internet-Protokollarchitektur, die durch ihre Protokolle die Kommunikation zwischen Rechnern im Internet regeln.
- **Ressourcenschicht:** Während die Anwendungsschicht in der Internet-Protokollarchitektur keine weitere Differenzierung dieser Schicht vornimmt, lassen sich in Grid-Systemen drei Teilschichten unterscheiden. Die Ressourcenschicht verwendet die Kommunikations- und Sicherheitsprotokolle der Verbindungsschicht, um zum einen Informationen über die Konfiguration oder den Zustand und zum anderen die Verwaltungsinformationen über die er-

brachten Dienste einer Ressource abzufragen. Die Ressourcenschicht ruft die Funktionen des Ressourcengeflechts auf, um auf die lokalen Ressourcen zuzugreifen. Auf dieser Schicht werden beispielsweise Entscheidungen der Kollektivschicht zur Reservierung oder Freigabe der Ressource umgesetzt. Dabei wird der Zustand des gesamten Grid-Systems nicht weiter betrachtet.

- **Kollektivschicht:** Die Aufgabe der Kollektivschicht ist die Koordination aller Ressourcen im Grid-System. Während die Ressourcenschicht eine einzelne Ressource verwaltet, ist die Kollektivschicht verantwortlich für das globale Ressourcenmanagement und die Interaktion mit dem Ressourcenkollektiv. Diese Schicht realisiert Protokolle zur gemeinsamen Nutzung von Ressourcen, wobei sie die Protokolle der Ressourcen- und Verbindungsschicht verwendet. Die Protokolle sind dabei nicht an eine spezifische Ressource gebunden, sondern universell für alle Ressourcen im Grid-System einsetzbar. Die in der Kollektivschicht bereitgestellten Dienste umfassen Dienste des Ressourcenmanagements von Grid-Systemen. Im Detail werden hierfür Dienste zur Suche, Auswahl, Verhandlung und Überwachung von Ressourcen benötigt.
- **Anwendungsschicht:** Die oberste Schicht der Grid-Architektur, die Anwendungsschicht, ermöglicht die Verwendung der Ressourcen einer Grid-Umgebung durch verschieden Protokolle der darunterliegenden Schichten. Anwendungen können die Dienste der Kollektivschicht, der Ressourcenschicht und der Verbindungsschicht verwenden, um ihre Funktionalität bereitzustellen. Abhängig von der jeweiligen Anwendung werden Dienste zur Ressourcenallokation, zur Kollaboration oder zum Datenaustausch verwendet.

Im Vergleich mit der Internet-Protokollarchitektur weist die Grid-Protokollarchitektur ähnliche Schichten auf den unteren Ebenen auf. Deutlich detaillierter gegliedert sind in der Grid-Protokollarchitektur die verschiedenen Schichten zur Verwaltung von Ressourcen, um eine möglichst große Unabhängigkeit von der jeweiligen Technologie zu erreichen. Von jeder dieser Schichten gehen technische Risiken aus, die im folgenden Abschnitt klassifiziert werden.

### 2.1.3 Klassifizierung der technischen Risiken

Um eine systematische Gruppierung der Fehlerquellen in Grid-Systemen zu ermöglichen, bietet sich eine Ontologie an, die Zusammenhänge zwischen verschiedenen Fehlerquellen aufzeigt. Eine solche Ontologie zeigt Abbildung 2.3. Diese Ontologie bezieht nicht nur Fehlerquellen der Grid-Ressourcen mit ein, sondern auch Fehlerquellen der Grid-Services, die IT-Dienste auf Grid-

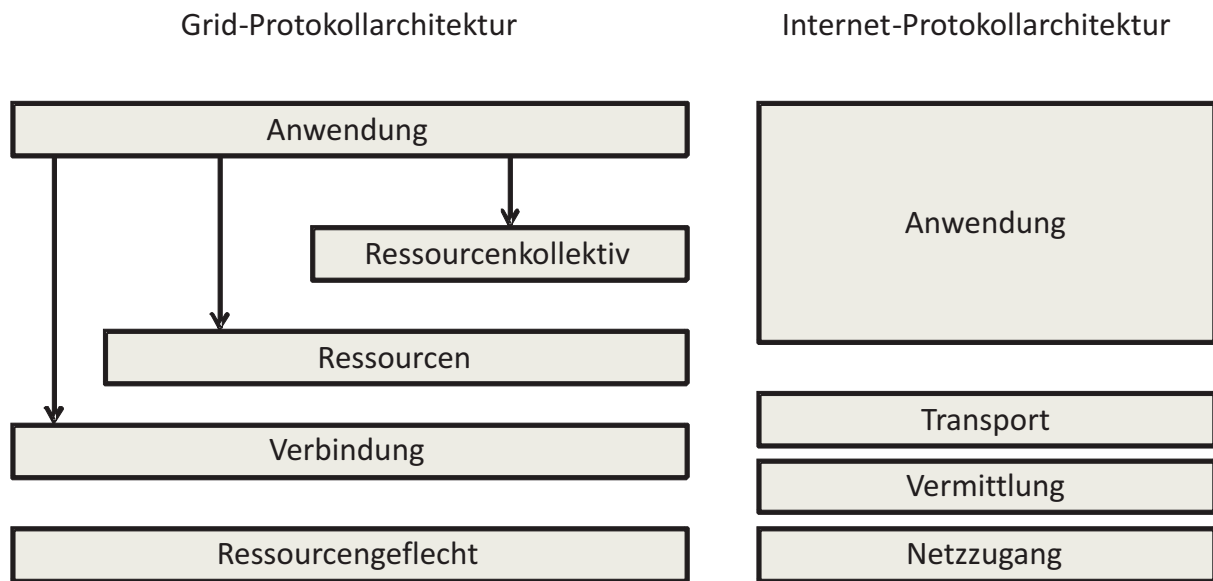


Abbildung 2.2: Schichtenmodelle der Grid- und Internet-Protokollarchitektur [JoEF04]

Ressourcen anbieten. Im einführenden ASP-Beispiel zu Beginn des Kapitels würden sich die Fehler der Grid-Ressourcen auf den Ressourcenanbieter beziehen, während die Grid-Services-Fehler vom Serviceanbieter ausgehen.

Bevor auf die Details der Ontologie näher eingegangen wird, wird die Ausführung eines Jobs auf einem Grid-System in zeitlicher Reihenfolge näher betrachtet, wobei nur die Schritte vorgestellt werden, die zur Durchführung der Simulation wichtig sind.

- **Anfrage einer Ressource:** Der Konsument schickt eine Anfrage nach einem Grid-Dienst oder nach Grid-Ressourcen an potenzielle Anbieter.
- **Datenübermittlung der Eingangsdaten:** Der Konsument übermittelt die Daten zur Berechnung. Dieser Schritt kann vernachlässigt werden, wenn die Datenmenge sehr klein ist.
- **Allokation der Ressourcen:** Es findet eine Allokation der Ressourcen auf Seiten des Anbieters statt.
- **Ausführung des Rechenjobs:** Der angefragte Dienst wird auf den zugewiesenen Ressourcen ausgeführt.
- **Freigabe der Ressourcen:** Die Ressourcen werden nach Ausführung des Dienstes wieder freigegeben.
- **Datenübermittlung der Ergebnisse:** Die Daten der Ergebnisse werden an den Konsumenten

übermittelt. Dieser Schritt ist ebenfalls vernachlässigbar, wenn die Datenmenge sehr klein ist.

- Antwort der Ressource: Die Ressource berichtet den Status der Dienstauführung.

Mithilfe einer für Grid-Systeme entwickelten Fehlerontologie von Hofer und Fahringer [HoFa08] werden die Fehlerarten, die für diese Arbeit wichtig sind, eingegrenzt. Dabei werden zwei Sichtweisen unterschieden, die der Grid-Ressourcen und die der Grid-Services. Nur die für diese Arbeit wichtigen Fehlerquellen wurden dabei ausgewählt und in Abbildung 2.3 dargestellt. Der Fokus liegt auf Fehlerquellen, die während der Ausführung auftreten und durch ihre versteckte Charakteristika Auswirkungen auf die Dienstleistung haben. Dabei wurden Fehler durch den Benutzer und Fehler, die auf Programmier- und Konfigurationsfehler zurückzuführen sind, ausgeblendet. Die wichtigen Fehlerquellen im Rahmen dieser Arbeit werden im Folgenden genannt:

- Policy: Policies umfassen alle Regeln und Vorgaben des lokalen Ressourcenmanagements eines Anbieters. Sie haben, wie z.B. in der Studie von [IJSE07] belegt, Einfluss auf die Verfügbarkeit der einzelnen Standorte eines Grid-Systems.
- Job: Die Ausführungszeiten eines Jobs auf Grid-Ressourcen können zum Teil deutliche Schwankungen aufweisen [IJSE07]. Dies kann dazu führen, dass die vereinbarte Dienstgüte nicht eingehalten wird.
- Netzwerk: Die fehlerhafte Konfiguration von Netzwerkkomponenten bzw. deren Ausfall oder Überlastung machen eine Dienstauführung auf Grid-Ressourcen nicht mehr möglich. Die noch verfügbaren Ressourcen müssen in diesem Fall eine zusätzliche Last bewältigen, was zu mehr Ausfällen durch die erhöhte Anfragefrequenz führt. Dies wurde durch Berechnung der Autokorrelationsfunktion zwischen Anfragefrequenz und Ausfall in der Studie von Li et al. [LGWT06] ermittelt.
- Datenübertragung: Noch stärker als die eigentliche Ausführung des Jobs auf den Grid-Ressourcen ist die Datenübertragung von Qualitätsschwankungen unterworfen [KHOS<sup>+</sup>06]. Anhängig von der jeweiligen Anwendung bzw. der zu übertragenden Datenmengen muss die Gewichtung dieser Fehlerquelle erfolgen. Bei einer datenintensiven Anwendung ist die Zuverlässigkeit der Datenübertragung über das Netzwerk von entscheidender Bedeutung.
- Servicequalität: Aus Sicht der Grid-Services kommt es zu Schwankungen der vereinbarten Servicequalität bei der Ausführung auf Grid-Ressourcen. Liegt die Ausführung eines Dienstes außerhalb der festgelegten Toleranzgrenzen, wird die Dienstauführung als fehlgeschlagen gezeichnet. Andernfalls ist die Dienstauführung als erfolgreich anzusehen.

- **Kommunikation:** Durch Netzwerkfehler wird die Kommunikation zwischen dem Benutzer und dem Grid-Dienst beeinträchtigt. Fehler in der Kommunikationsinfrastruktur führen zu einem Abbruch des Grid-Jobs und damit zu einer fehlerhaften Dienstausführung. Je datenintensiver der Grid-Dienst ist, desto stärker müssen Kommunikationsfehler mit einbezogen werden.

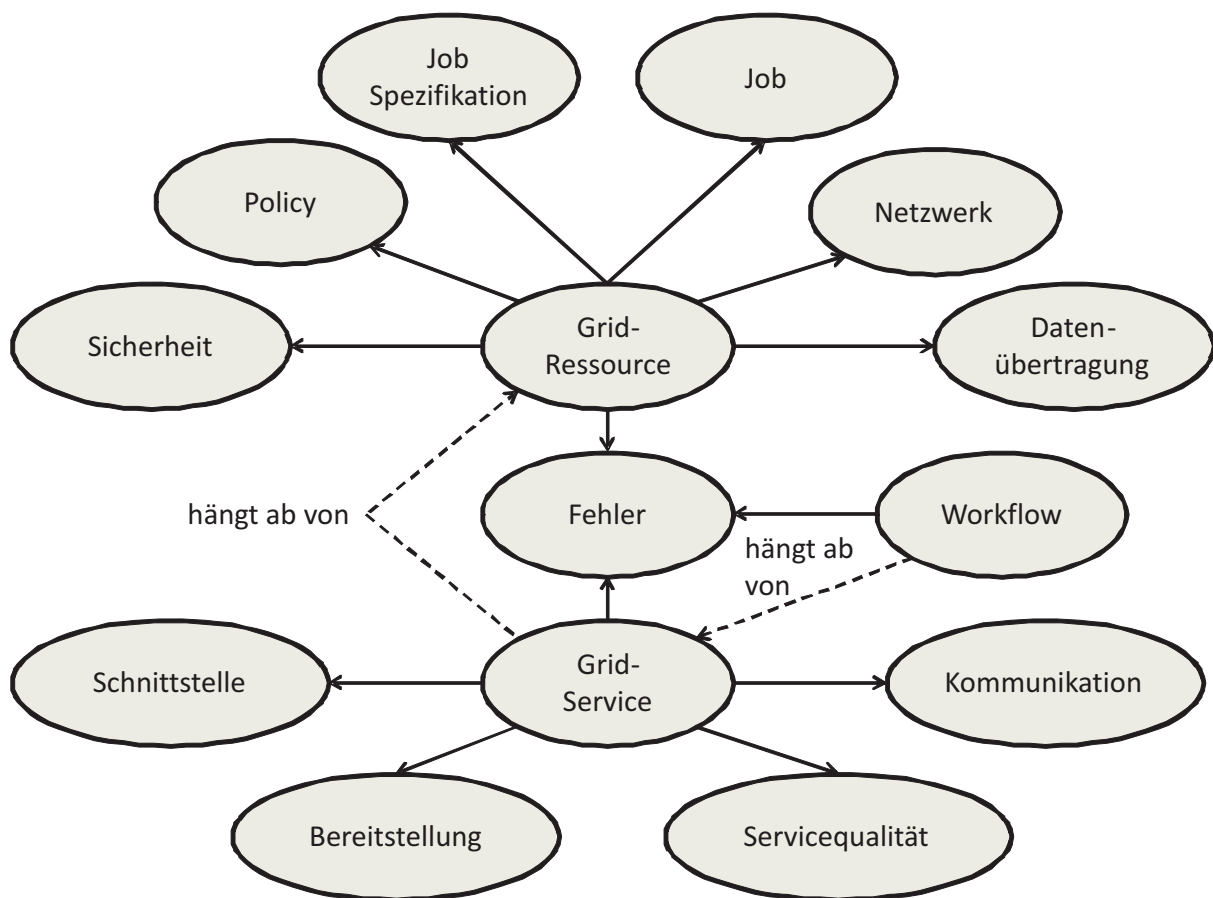


Abbildung 2.3: Fehlerontologie für Grid-Services und Grid-Ressourcen [HoFa08]

Ausgehend von der Fehlerontologie von Hofer et al. [HoFa08] werden im folgenden Abschnitt verschiedene Technologien und Standards der Open-Grid-Services-Architektur eingeführt. Diese Technologien und Standards bilden die technische Grundlagen zur Behandlung der oben genannten Fehlerquellen. Sie können im Idealfall einzelne Fehler verringern bzw. eliminieren und damit die Zuverlässigkeit der Grid-Systeme erhöhen.



## 2.2 Die Open-Grid-Services-Architektur

Die Entwicklung der Open-Grid-Services-Architektur (OGSA) wurde getrieben durch die Anforderung nach einer Bereitstellung von nichttrivialer Servicequalität durch das Grid-System. Das Ziel besteht dabei darin, die Servicequalität auf Grundlage einer virtuellen Organisation zu erbringen und die Servicequalität der Ressourcen mit einem einheitlichen Satz an Metriken wie beispielsweise Verfügbarkeit zu messen. Als eine virtuelle Organisation wird im Kontext von Grid-Systemen eine Menge an Ressourcenkonsumenten und Anbietern bezeichnet, die sich über einheitliche Regeln zur Teilung von Ressourcen definieren [FoKT01]. Die einzelnen Individuen der virtuellen Organisation kollaborieren dabei hinsichtlich eines gemeinsamen Interesses mithilfe von Verträgen [BLRS98]. Die Open-Grid-Service-Architektur definiert Dienste, um die oben genannten Anforderungen an Grid-Systeme im Rahmen einer virtuellen Organisation bereitzustellen.

### 2.2.1 Der Begriff des Grid-Services

Allgemein wird der Begriff *Service* unterschiedlich angewandt und verstanden. Da er von zentraler Bedeutung für die folgenden Ausführungen ist, ist es unverzichtbar, ihn im Sinne der hier fortgeführten Argumentation richtig zu interpretieren. Daher soll er aus Sicht der serviceorientierten Architektur und der Grid-Systeme definiert werden.

Zentral hierbei ist die logische Trennung zwischen dem Bedarf und dem Mechanismus, der den Bedarf erfüllt. Der Aufrufer eines Dienstes ist nur am Ergebnis interessiert und hat keine Kontrolle über den Mechanismus, mit dem das Ergebnis erreicht wird. Dies steht im Einklang mit der Autonomie des lokalen Ressourcenmanagements in Grid-Systemen, das dem Benutzer ein Ergebnis seiner Berechnung liefert, ohne dass der Benutzer Kenntnis über die interne Abarbeitung hat.

Im Folgenden werden die für die Arbeit wichtigen Begriffe *Service* aus Sicht der Softwareentwicklung sowie zwei speziellen Ausprägungen – der *Web-Service* als zentraler Bestandteil der serviceorientierten Architektur und der *Grid-Service* als wichtige Einheit der Open-Grid-Services-Architektur – definiert:

- *Service*: Ein *Service* ist definiert als jegliches Handeln oder jegliche Erfüllung einer Leistung, die eine Partei einer anderen anbieten kann und die hauptsächlich immateriell ist und nicht im Besitz von irgendetwas resultiert [Elfa07]. Ein *Service* aus Sicht der Softwareent-



wicklung ist eine grobkörnige, feststellbare Entität, die als einzelne Instanz existiert und mit Anwendungen und anderen Services interagieren kann [Elfa07]. Die Lieferung des Service erfolgt über ein Computernetzwerk [BHHK08].

- *Web-Service* [BHMN<sup>+</sup>04]: Ein *Web-Service* wird definiert als ein Softwaresystem, das entworfen wurde, um Kompatibilität für Interaktionen zwischen Rechnern über ein Netzwerk zu unterstützen. Ein *Web-Service* lässt sich wie folgt charakterisieren [KoLe04]:
  - Kapselung: Ein *Web-Service* ist eine unabhängige, in sich geschlossene bzw. gekapselte Anwendung, die eine genau definierte Aufgabe erfüllt.
  - Lose Kopplung: Die Systeme des *Web-Service*-Konsumenten und des *Web-Service*-Anbieters bleiben autonom und kommunizieren über Nachrichten miteinander.
  - Virtualisierung: Durch Virtualisierung wird eine Austauschbarkeit der Hardware und Software erreicht [FKNT02a]. Die Interaktion zwischen dem *Web-Service*-Konsumenten und dem *Web-Service*-Anbieter ist dynamisch: Ein Konsument A agiert mit Anbieter B, wenn Anbieter B die aktuellen Anforderungen des Konsumenten A erfüllen kann. Der Konsument als Dienstleistungsnehmer wird dabei in den Erstellungsprozess integriert [BHHK08].
  - Einheitliche Konventionen: *Web-Services* unterstützen viele unterschiedliche Datenformate, Protokolle und Mechanismen zur Zusicherung von Qualitätseigenschaften.
  - Selbstbeschreibend: Ein *Web-Service* wird begleitet von Metadaten, die während der Laufzeit von weiteren *Web-Services* ausgewertet werden können. Name, Beschreibung, Version und Dienstgüte sind typische Beschreibungsmerkmale.
  - Komposition: *Web-Services* können entweder in weitere *Web-Services* zerlegt werden oder mehrere wiederverwendbare Basis-*Web-Services* können zu einem neuen *Web-Service* zusammengestellt werden.
- *Grid-Service* [FKNT02b] [FoKe03, S. 59]: Ein *Grid-Service* ist ein *Web-Service*, der standardisierte Schnittstellen, Verhaltensweisen und Konventionen definiert, die es Services ermöglichen, transient<sup>1</sup> und zustandsbehaftet<sup>2</sup> zu sein. Die Schnittstellen behandeln die Servicesuche, dynamische Serviceerstellung und Serviceverwaltung. Dies schließt auch Services zur Überwachung und Abrechnung mit ein. Die Konventionen umfassen eine einheitliche Namensgebung und die Möglichkeit der Versionierung.

---

<sup>1</sup>Transient bedeutet, dass neue Services erzeugt und alte Services gelöscht werden können.

<sup>2</sup>Bei einem zustandsbehafteten Service lässt sich eine Serviceinstanz von einer anderen unterscheiden.

Grid-Services weisen dabei Prozesscharakter auf [Sinz08]. Dies bedeutet, dass die Grid-Dienste in Workflows<sup>3</sup> eingebettet sind, die beispielsweise einen Geschäftsprozess realisieren.

### 2.2.2 Grundlegende Grid-Dienste

Wurde in den vorangegangenen Abschnitten Begriff des Grid-Service abstrakt gesehen, so werden in diesem Abschnitt die wichtigsten Grid-Services der Open-Grid-Services-Architektur vorgestellt. Diese Dienste sind auf der Kollektiv- und Ressourcenschicht des Schichtenmodells der Grid-Protokollarchitektur anzuordnen und stehen dadurch Softwareanwendungen wie der Grid-Versicherung zur Verfügung. Diese grundlegenden Grid-Services werden bei dem Entwurf der Versicherung als gegeben angenommen.

Je nach Literaturquelle und Implementierung der Open-Grid-Services-Architektur können sich die angebotenen Grid-Services hinsichtlich Umfang und Ausprägung ändern. Übereinstimmend werden folgende Grid-Services als wichtige Grid-Services aus Sicht der Grid-Architektur genannt [FKSB<sup>+</sup>06] [FoGa01]:

- Sicherheitsdienste zur Authentifizierung, Autorisierung bei nur einmaliger Anmeldung: Diese Grid-Services erlauben es dem Benutzer, Anwendungen auf jeder verfügbaren Grid-Ressource unter Verwendung eines standardisierten Authentifizierungszertifikat auszuführen. Das ermöglicht die Kommunikation zwischen den Anwendungen untereinander und den Zugriff auf Ressourcen, sobald der Benutzer autorisiert wurde.
- Einen standardisierten, im gesamten Grid gültigen Namensraum zur eindeutigen Identifikation von Dateien, Grid-Diensten und anderen Ressourcen: Das World Wide Web verwendet die universell einsetzbaren Uniform-Resource-Identifiers (URIs) als weltweiten Namensraum für die Bereitstellung von Inhalten zwischen Webservern und Webbrowsern. Diese Technologie wird auch in Grids verwendet, um Ressourcen weltweit eindeutig zu identifizieren.
- Resource-Management-Services, welche die folgenden Ansammlungen von Services umfassen:
  - Ressourcenregistrierung und Ressourcensuche: Ressourcen werden in Verzeichnissen verwaltet, auf die bei der Ressourcensuche zugegriffen werden kann.

---

<sup>3</sup>Ein Workflow beschreibt ein Netz von Arbeitsschritten als Lösungsverfahren für betriebliche Aufgaben [Sinz08].

- Ressourcen-Accounting: Zur Abrechnung des Ressourcenverbrauchs werden Resource Accounting Services eingesetzt.
- Ressourcen-Scheduling und Co-Scheduling: Unter Ressourcen-Scheduling versteht man den Prozess des automatisierten Zuweisens von Grid-Ressourcen zu Benutzeranfragen. Co-Scheduling beinhaltet die Allokation der Ressourcen von verschiedenen Anbietern, die zusammen die benötigten Ressourcen bereitstellen.

Die Services des Ressourcenmanagements haben nicht nur die Aufgabe, eine Zuweisung der Ressourcen durchzuführen, sondern auch eine Erhöhung der Zuverlässigkeit des Grid-Systems zu unterstützen [Dabr08]. Dazu werden fehlertolerante Verfahren zur Ressourcensuche [JAKP05] und zum Ressourcen-Scheduling [MaMP06] eingesetzt, um das technische Risiko zu verringern.

- Execution-Management-Services und Information-Services zur Ausführungsüberwachung und Leistungsmessung: Ressourcenkonsumenten, Ressourcenanbieter, und Anwendungen müssen in der Lage sein, zu jeder Zeit auf Informationen zum Zustand des Grid-Systems zugreifen können. Diese Grid-Services können Informationen über den aktuellen Ausführungsfortschritt des Dienstes und die aktuellen Leistungskriterien wie Netzwerkbandbreite oder aktuelle Systemlast zur Verfügung stellen. Eine wichtige Aufgabe dieser Dienste ist das Erkennen von auftretenden Fehlern bei der Ausführung eines Rechenjobs. Zusätzlich wird die Einhaltung der Servicequalität überwacht. Die Verbrauchsmessung der Informationsdienste bildet die Grundlage für die Abrechnung des Ressourcenverbrauchs.

Für die Umsetzung dieser Grid-Dienste werden in der Open-Grid-Services-Architektur Web-Service-Technologien eingesetzt, die im nächsten Abschnitt vorgestellt werden.

### 2.2.3 Serviceorientierte Architekturen und Web-Services

Die Open-Grid-Services-Architektur baut auf den Prinzipien einer serviceorientierten Architektur und Web-Services (WS) auf, die auf Basis von offenen Standards Schnittstellen bereitstellen [Tali02] [FKSB<sup>+</sup>06]. Eine serviceorientierte Architektur ist ein spezieller Typ eines verteilten Systems, in dem alle Instanzen Dienste sind, die wohl definierte Operationen ausführen [JoEF04] [FKSB<sup>+</sup>06]. Diese Dienste müssen eine netzwerkadressierbare Schnittstelle bereitstellen und mit Standardprotokollen und standardisierten Datenformaten kommunizieren. Der Kommunikationsprozess zwischen den beteiligten Instanzen basiert auf einem wohl definierten Nachrichtenaustausch, unabhängig von der dem Dienst zu Grunde liegenden Plattform und anbieterspezifischen

Implementierungsdetails.

Der allgemeine Vorteil serviceorientierter Architekturen besteht in der Senkung der Transaktionskosten durch die Standardisierung der Schnittstellen und Protokolle [Rein07]. Grid-Dienste profitieren von der technischen Spezifizierung, was hinsichtlich der Verwendung von Schnittstellen und Protokollen zu einer störungsfreien Nutzung eines Grid-Service führt. Die Konsumenten passen ihre Anwendungen an die standardisierten Schnittstellen und Protokolle an. Die Anbieter können ihre Dienste mithilfe dieser Standards veröffentlichen und den Konsumenten zur Nutzung anbieten.

Durch die Standardisierung erhoffen sich Unternehmen, dass sich mithilfe von Web-Services nicht nur interne Applikationen leichter integrieren lassen, sondern dass auch Geschäftsprozess-Outsourcing künftig standardisiert und damit in kürzester Zeit bzw. mit geringerem Kostenaufwand realisiert werden kann [FrWK02].

Die SOA ebnet einen Weg zur Restrukturierung isolierter, monolithischer Softwarekomponenten und -infrastrukturen in eine flexible Komposition von interaktiven Diensten [Kaye03]. Die Anwendung einer SOA unterstützt die Wiederverwendbarkeit von existierenden Anwendungskomponenten. Serviceorientierte Architekturen adressieren grundlegende Herausforderungen von offenen Systemen, die sich effizient betreiben lassen und eine Autonomie in heterogenen Umgebungen erreichen [HuSi05]. Dieser Ansatz wirkt besonders dann viel versprechend, wenn mehrere Anwendungen auf unterschiedlichen Technologien und Plattformen miteinander kommunizieren sollen. Dies ermöglicht es Unternehmen erstmals, Dienste mit einem größtmöglichen Freiheitsgrad zu verbinden und anzupassen, um Geschäftsprozesse mit einem minimalen Programmieraufwand abzubilden [PaGe03] [Papa07] [FrWK02].

Web-Services spielen eine fundamentale Rolle bei der Umsetzung und Implementierung einer serviceorientierten Architektur [Papa07]. Das Web-Service-Paradigma [BHMN<sup>+</sup>04] definiert dabei ein verteiltes, Service-orientiertes Modell, in dem Dienste über den Austausch von XML-Dokumenten<sup>4</sup> interagieren [WoWe02] [KoLe04]. Eine SOA verspricht insbesondere offene Standards für Web-Services und damit eine Unabhängigkeit vom Anbieter dieser Dienste, wiederholte Nutzbarkeit und Kombination von Diensten, um eine schnelle Reaktion auf geänderte Geschäftsanforderungen zu gewährleisten, geringere Wechselkosten durch das dynamische Einbeziehen von austauschbaren Diensten, die Konzentration auf Prozesse und den Schutz von getätigten Investitionen durch Kapselung existierender Funktionen [Rein07]. Die Anwendungs-

---

<sup>4</sup>Die eXtended Markup Language (XML) erlaubt eine semantische Auszeichnung von strukturierten Inhalten und bietet die Möglichkeit, eigene Elemente zu definieren [BPSMM<sup>+</sup>06].

Integration durch Web-Services führt zu flexiblen, lose gekoppelten Prozessen in virtuellen Organisationen [GGKS02]. Eine SOA beschreibt aber nicht nur die Dienste selbst, sondern auch die Interaktionen von Dienstanfragern und -anbietern. Die Beteiligten werden in Serviceanbieter, Servicekonsument und Serviceregistrierungsdatenbank klassifiziert.

In einer serviceorientierten Architektur können Serviceanbieter ihre Dienste in einer zentralen Registrierungsdatenbank veröffentlichen. Anschließend werden diese Dienste dynamisch gefunden und zur Laufzeit von Servicenachfragern benutzt. Beispielsweise kann ein Dienstanbieter einen Speicherdienst in einer solchen Registrierungsdatenbank veröffentlichen. Eine Datenanalyseanwendung, die diesen Speicherplatz benötigt, um ihre Daten abzuspeichern, kann diesen Speicherdienst suchen und über standardisierte Schnittstellen aufrufen und verwenden. Die Anwendung kann die Daten ohne Wissen darüber, wie die Daten z. B. unter Verwendung eines verteilten Dateisystems abgespeichert werden, ablegen.

Web-Services sind eingebettet in einen Web-Service-Technologiestapel, der eine große Menge an Spezifikationen für die Kommunikation, die Dienstbeschreibung, die Dienstsuche und weitere Funktionalitäten beschreibt. Im Kontext von SOAs sind die folgenden drei Standards am wichtigsten [CDKN<sup>+</sup>02]:

- das Simple-Object-Access-Protocol (SOAP) für die Kommunikation zwischen Web-Services,
- die Web-Service-Description-Language (WSDL) für die Beschreibung der Schnittstelle eines Web-Service und
- die Universal Description, Discovery, and Integration (UDDI) Sprache zur Beschreibung eines Verzeichnisses für die Veröffentlichung und Suche von Web-Services.

Weitere Web-Service-Standards und -Technologien finden sich bei [Kreg03] [GGKS02] [CKMT<sup>+</sup>03] [CDKN<sup>+</sup>02]. Für die kommerzielle Nutzung von Web- bzw. Grid-Services sind Service-Level-Agreements essenziell. Das allgemeine Konzept und der hierfür definierte Standard werden im folgenden Abschnitt näher erläutert.

## 2.3 Service-Level-Agreements (SLAs)

Service-Level-Agreements werden allgemein in Unternehmen eingesetzt, um auf individueller Basis Rechte und Pflichten zwischen einem Dienstleistungsnehmer und einem Dienstleistungsgeber zu definieren [Bern03]. Eine Service-Level-Agreement ist ein Grid-Service-Vertrag, der

die minimalen Erwartungen und Verpflichtungen zwischen dem Anbieter und dem Konsumenten eines Grid-Dienstes definiert [BhSC01] [BCLW<sup>+</sup>04]. Die Vereinbarung von Leistungsmessgrößen wie beispielsweise die Zuverlässigkeit der Grid-Ressourcen des Anbieters helfen dabei Qualitätsunsicherheiten abzusichern, da die Qualität der Leistung im Vorhinein kaum bewertet werden kann [TaGü05, S. 187]. SLAs bilden die Grundlage für die kommerzielle Nutzung von Grid-Diensten und Grid-Ressourcen [LeRD03].

In der vorliegenden Arbeit dienen die vereinbarten Leistungsmessgrößen eines SLA dazu, das Risiko einer Verletzung der zugesicherten Leistung zu bestimmen und die Kosten des Risikotransfers auf eine Versicherung festzulegen. Die Feststellung eines Schadens erfolgt durch die Überwachung der Serviceausführung auf Grundlage der Leistungsmessgrößen des SLA.

### 2.3.1 Aufbau von Service-Level-Agreements

Um eine standardisierte Bereitstellung von Grid-Diensten und Grid-Ressourcen zu erreichen, wurde ein einheitlicher Standard zur Beschreibung von Service-Level-Agreements vom Open-Grid-Forum verabschiedet [ACDK<sup>+</sup>06]. Dieser Standard ermöglicht sowohl Anbietern als auch Nutzern eine automatisierte Verarbeitung von SLA unter Verwendung von Web-Service-Technologien. Dies schließt die automatisierte Definition, Verhandlung, Überwachung, Bereitstellung und Sanktionierung der SLAs mit ein [SGMM03] [BCLW<sup>+</sup>04] [DuRF05] [LKAC<sup>+</sup>08].

Zwingend erforderlich ist bei der Gestaltung von SLAs, dass die zu erbringenden Leistungen beschrieben und damit dokumentiert werden. Weiterhin muss das für die Leistung relevante Niveau mit der Definition von Schwellenwerten abgegrenzt werden. Da es sich hierbei um einen der wichtigsten Inhalte für die Vertragsbeziehung handelt, müssen die geregelten Leistungen genau beschrieben werden. Der Aufbau eines Service-Level-Agreements nach dem Web-Service-Agreement (WS-Agreement)-Standard zeigt Abbildung 2.4.

Im Einzelnen können folgenden Eigenschaften festgelegt werden [ACDK<sup>+</sup>06]:

- Name: Dieser Platzhalter gibt die Möglichkeit, einen Namen für das SLA festzulegen.
- Kontext: Der Kontext sollte die Vertragspartner und weitere, für das gesamte SLA gültige Parameter wie beispielsweise die Gültigkeit des SLA festlegen. In den in dieser Arbeit betrachteten Szenarien ist die Gültigkeit für eine Dienstauführung bzw. die einmalige Ressourcennutzung festgelegt. Bei erneuter Nutzung eines Grid-Dienstes muss ein neues SLA vereinbart werden.

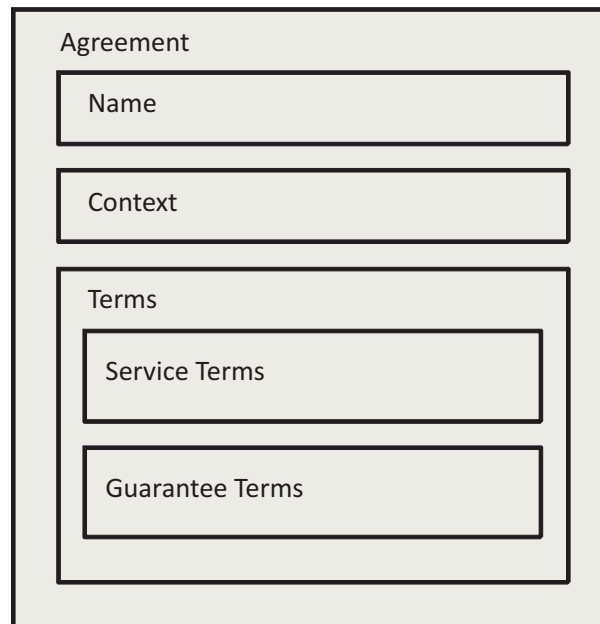


Abbildung 2.4: Aufbau eines Service-Level-Agreements nach dem WS-Agreement-Standard [ACDK<sup>+</sup>06]

- **Agreement Terms:** Der Hauptteil der WS-Agreement-Dokumente besteht aus zwei Blöcken, den Service-Terms und Guarantee-Terms. Die Service-Terms stellen Informationen über den Leistungsgegenstand und seine Verwendung bereit. Sie dienen der technischen Beschreibung des Dienstes und bilden die Grundlage für die zu erwarteten Garantien, die in den Guarantee-Terms spezifiziert werden.

Allgemein legen die Guarantee-Terms die Dienstgüte für den im Service-Term definierten Dienst fest, auf den sich die beteiligten Parteien geeinigt haben. In einem Grid-System kann das beispielsweise die Festlegung einer minimalen Verfügbarkeit eines Grid-Ressourcenbündels oder die Festlegung einer maximalen Ausführungszeit oder Antwortzeit für einen Grid-Dienst bedeuten. Zusätzlich lassen sich diese technischen Metriken mit ökonomischen Werten verknüpfen. Als Beispiele lassen sich hier aus Konsumentensicht die Festlegung der Bedeutung der Ausführung oder Strafzahlungen bei Abweichung nennen und aus Anbietersicht die Festlegung einer Wahrscheinlichkeit für die Erfüllung der vereinbarten Dienstgüte. Diese letztgenannten Werte sind gleichzeitig wichtige Bestandteile des Risikomanagements, das im folgenden Kapitel näher analysiert wird.

Es ist abhängig von den einzelnen Grid-Betreibern, eine konkrete Ausprägung dieser allgemeinen Vorlage zu erstellen, wie sie der WS-Agreement-Standard vorgibt. Wichtig dabei ist die



Messbarkeit der im SLA spezifizierten Dienstgüteparameter, die, eingebunden in ein Framework, zur automatisierten Überwachung von SLAs die Grundlage des Risikomanagements bilden können [BBCF<sup>+</sup>04].

### 2.3.2 Einsatz von Service-Level-Agreements in Grid-Transaktionen

Der Einsatz von Service-Level-Agreements umfasst den vollständigen Gütertauschprozess einer digitalen Transaktion, wie sie beispielsweise im E-Commerce anzutreffen ist. Analog dazu lässt sich eine Grid-Transaktion in fünf Phasen gliedern [MüEK03].

- **Anbahnungsphase:** In der Anbahnungsphase stellt der Grid-Serviceanbieter Informationen über sein Produkt in Form von Service-Terms in einem SLA-Template bereit. Dies stellt eine Vorlage dar, nach der der Konsument suchen kann, um einen Dienst zu finden, der seinen Vorgaben entspricht.
- **Aushandlungsphase:** Die Aushandlungsphase dient der Verhandlung eines Service-Level-Agreements auf Basis des vorgegebenen SLA-Template, das in der Anbahnungsphase zur Dienstsuche genutzt wurde. Verhandlungsgegenstand sind die Agreement-Terms des SLA-Dokuments, um mithilfe eines Verhandlungsprotokolls – Auktion oder bilaterale Verhandlung – den Preis und optional weitere Bestandteile wie Vertragsstrafen festzulegen. Die definierten Kennzahlen der Dienstgüte werden so gewählt, dass sie von beiden Parteien zweifelsfrei verstanden, einer Partei zugeordnet und gemessen werden können.
- **Abwicklungsphase:** Die Abwicklungsphase beschreibt die Durchführung der Grid-Transaktion. Der Serviceanbieter stellt die benötigten Ressourcen zur Dienstauführung bereit und startet die Dienstauführung. Der Servicekonsument bezahlt den Dienst in einer virtuellen oder realen Währung. Wichtig ist in dieser Phase die Messung der vereinbarten Dienstgüte, um eine mögliche Verletzung des SLA feststellen zu können und Sanktionen gegenüber dem Anbieter oder Schäden gegenüber einer Versicherung in der Durchsetzungsphase behandeln zu können.
- **Anpassungsphase:** Die Anpassungsphase dient der dynamischen Anpassung der Grid-Ressourcen an die Dienstauführung, um die vereinbarte Dienstgüte einzuhalten. Weichen zur Laufzeit gemessene Werte von der vereinbarten Performanz ab, so lassen sich dynamisch Gegenmaßnahmen zur Bereitstellung weiterer Rechenleistung oder Bandbreite definieren, die automatisch eingeleitet werden.



- **Durchsetzungsphase:** In der Durchsetzungsphase werden schließlich Abweichungen von der im SLA vereinbarten Dienstgüte behandelt. Dies können Strafzahlungen sein, die Meldung eines Schadens an eine Versicherung oder die Abgabe einer Bewertung über den Anbieter, die in seine Reputation einfließen.

Um die kommerzielle Nutzung der Grid-Systeme im vollen Umfang zu nutzen, und dabei die Transaktionskosten möglichst gering zu halten, wurden Systeme entwickelt, die ein automatisiertes Management von SLAs ermöglichen. Je nach Rahmenwerk werden unterschiedliche Transaktionsphasen unterstützt. Da diese Rahmenwerke teilweise sehr komplex aufgebaut sind, wird an dieser Stelle auf eine nähere Beschreibung verzichtet und nur ein kurzer Überblick über bestehende Systeme gegeben.

- [BCLW<sup>+</sup>04]: Dieses Rahmenwerk konzentriert sich auf die Umsetzung von Geschäftszielen in Service-Level-Agreements. Hierfür werden neben den SLAs interne Ziele zur Umsetzung der SLAs definiert, die der Optimierung der SLAs auf den bestehenden Ressourcen des Anbieters dienen. Das System konzentriert sich dabei auf die Unterstützung der Abwicklungs-, Anpassungs- und Durchsetzungsphase.
- [DDKK<sup>+</sup>04]: Der Fokus dieses Rahmenwerks liegt auf der Integration verschiedener Dienste zur Unterstützung des Zugriffs auf ausgelagerte Web-Service-Dienste. Bis auf die Aushandlungsphase werden alle Transaktionsphasen unterstützt. Aufbauend auf einem Dienst zur Messung des Verbrauchs wird die vereinbarte Dienstgüte evaluiert und mögliche Änderungen seitens des Konsumenten oder des Anbieters werden veranlasst. Ein Anbieter hat in diesem Rahmenwerk die Möglichkeit, durch Abschluss weiterer Verträge Teile der Leistung wiederum an weitere Anbieter auszulagern.
- [LKAC<sup>+</sup>08]: Diese Arbeit verwendet die Dienstgüteattribute, die in SLAs beschrieben sind, um in einer OGSA-basierten Umgebung die Einhaltung der Dienstgüte während der Ausführung eines Grid-Service sicherzustellen. Hierfür wird ein Execution-Management-Service realisiert, der während der Ausführung einer Anwendung dynamisch verschiedene Parameter des nachgefragten Service mit dem Ziel anpassen kann, die zugesicherte Qualität des Service einzuhalten.

Die oben vorgestellten Rahmenwerke, Architekturen und Standards können in verschiedenen Anwendungsfeldern und Organisationsformen eingesetzt werden, von denen weitere technische Risiken ausgehen. Diese werden anhand der am häufigsten anzutreffenden Ausprägungen in den folgenden Kapiteln näher erläutert.

## 2.4 Anwendungsfelder von Grid-Systemen

Diese Sektion gibt einen Überblick über die Evolution der Grid-Systeme bezüglich ihrer Einsatzgebiete. Computational-Grids bieten Rechenleistungen in Form von CPU-Zyklen an, während bei Data-Grids der Zugriff auf und die Verteilung von großen Datenmengen im Vordergrund stehen.

In Utility-Grids werden nicht nur die grundlegenden Ressourcen wie CPU oder Festplattenplatz bereitgestellt, sondern auch eine Anwendungslogik, die über eine Schnittstelle beschrieben wird und so dem Nutzer als Utility zugänglich gemacht wird. Sie bieten dem Benutzer standardisierte Güterbündel an. Dies können beispielsweise Teile von Geschäftsprozessen sein, die wiederum aus mehreren Teilprozessen bestehen.

Allgemein nimmt dabei die IT-Komplexität in den verschiedenen Anwendungsfeldern zu. Steht bei Computational-Grids noch der Zugriff auf die grundlegenden Rechenressourcen im Vordergrund, muss in Utility-Grids schon eine Anwendungslogik verwaltet werden, was die technischen Risiken durch die zunehmende Komplexität erhöht.

### 2.4.1 Computational-Grid

Verteiltes Problemlösen mithilfe von Rechnern wird häufig für die Modellierung und Simulation von komplexen wissenschaftlichen und technischen Problemen wie Wettervoraussage<sup>5</sup>, Portfolioverwaltung [MVNVAC<sup>+</sup>07] oder medizinische Nutzung [BBCD<sup>+</sup>03] eingesetzt. Jedoch ist die durchschnittliche Kapazität eines Rechners unzureichend für solche anspruchsvollen Einsatzgebiete. Aus diesem Grund wurden erste Computational-Grid-Systeme dafür entwickelt, Super-Computing Rechenzentren zu vernetzen [RBS03], um eine Ausführung von Anwendungen mit einer hohen Leistungsanforderung zu ermöglichen.

Eine Computational-Grid wird definiert als eine Hardware- und Softwareinfrastruktur, die zuverlässigen, konsistenten und kostengünstigen Zugriff zu High-End-Rechenleistungen bereitstellt [FoKe99].

Abbildung 2.5 zeigt ein typisches Grid-Szenario. Unabhängige Benutzer greifen auf eine Menge an Ressourcen über Grid-Services zu. Die Daten werden über eine Internet-Infrastruktur aus-

---

<sup>5</sup>Weitere Informationen finden sich auf den Seiten des European Centre for Medium-Range Weather Forecasts (ECMWF): <http://www.ecmwf.int/>

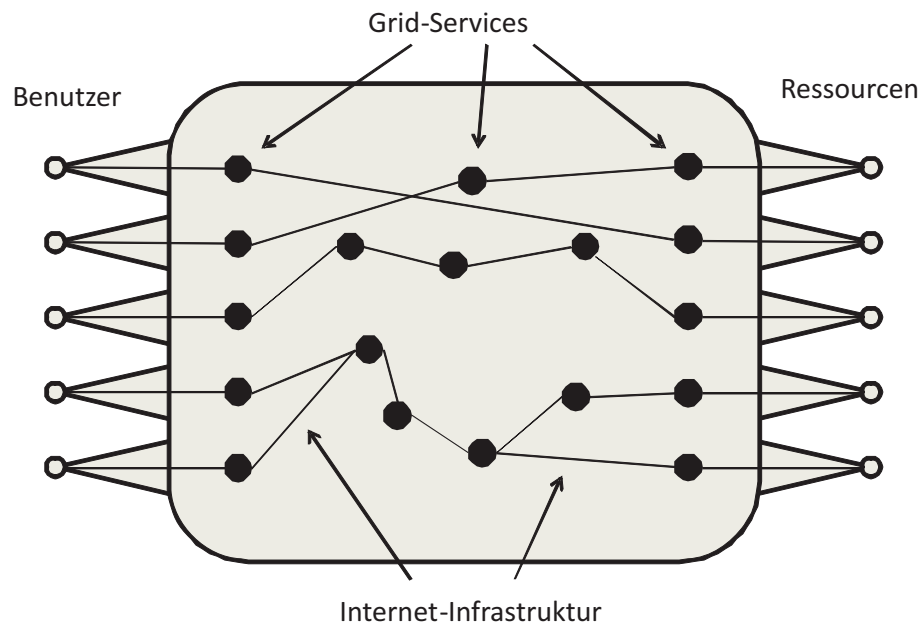


Abbildung 2.5: Abstraktes Modell eines Grid-Systems [FoGa01]

getauscht. Die Ressourcen werden in einem Computational-Grid dem Benutzer transparent gemacht und durch das Ressourcenmanagement automatisch zur Verfügung gestellt.

Die oben genannten Vorteile motivieren die Nutzung eines Computational-Grid aus Sicht eines wissenschaftlichen oder kommerziellen Nutzers. Das Zusammenschalten von verschiedenen Rechenzentren zu einem Computation-Grid stellt eine erhöhte Parallelität bereit, die der Nutzer als ein Mehr an Rechenleistung verwenden kann. Jedoch vernachlässigt diese Sichtweise die Anreize, an einem Computational-Grid teilzunehmen bzw. Rechenleistung für ein Computational-Grid bereitzustellen. Veranlassungen für die Teilnahme an einem Computational-Grid sind zum einen Kostengründe und zum anderen Gründe, mit seiner Rechenleistung einen Beitrag für ein Projekt auf freiwilliger Basis leisten zu wollen [Skil02].

Eine Zusammenschaltung von bestehenden Rechenkapazitäten zur Steigerung der Rechenleistung ist in vielen Fällen kostengünstiger als die Neuanschaffung eines größeren, einzelnen Systems, das mehr Rechenleistung besitzt. Dies geht einher mit der Handelbarkeit von Rechenleistung auf Basis von Einheiten wie CPU-Zyklen oder CPU-Stunden. Verschiedenen Anbieter können damit genutzte Rechenleistung verrechnen und somit dynamisch Kapazitäten bei Spitzenbedarf hinzunehmen und diese wieder freigeben. Insgesamt wird die Flexibilität der bestehenden Rechenleistung erhöht.

Ein Beispiel für die Zusammenschaltung von Grid-Ressourcen zu einem großen Computational-

Grid-System ist das Enabling Grids for E-science (EGEE)-Projekt [GJGB<sup>+</sup>05]. Das Ziel des EGEE-Projekts ist es, nationale Grid-Initiativen zu vereinen, um europäischen Forschern an Universitäten und in der Industrie einen gemeinsam genutzte Menge an Rechenressourcen zur Verfügung zu stellen, die unabhängig vom geografischen Ort ist und sich durch hohe Verfügbarkeit auszeichnet, wobei die oben genannten Vorteile aus Benutzer- und Anbietersicht realisiert werden sollen. Aktuell sind auf dieser Plattform 41000 CPUs<sup>6</sup> verfügbar, die von der Industrie wie auch von Forschungsinstitutionen zur Verfügung gestellt und genutzt werden.

Als zweites Beispiel für ein Computational-Grid wird an dieser Stelle die Berkeley Open Infrastructure for Network Computing (BOINC) vorgestellt. Die BOINC ist eine Software, die es erlaubt, Projekte mit global verteilten Grid-Ressourcen zu erstellen und zu betreiben [Ande04]. Dabei wird nicht die gebündelte Rechenleistung von Supercomputern in Rechenzentren verwendet, sondern auf die Rechenleistung von mehreren Millionen Desktoprechnern und Spielekonsolen zurückgegriffen. Häufig wird in diesem Zusammenhang auch von *Volunteer Computing* oder *Public Resource Computing* gesprochen, um darauf hinzuweisen, dass die Teilnahme am Grid-System auf freiwilliger Basis passiert und die Ressourcen meist von Privatpersonen öffentlich zur Verfügung gestellt werden.

Das bekannteste Projekt der BOINC-Plattform ist das Search for Extraterrestrial Intelligence at Home (SETI@Home)-Projekt [ACKL<sup>+</sup>02], das 1999 gestartet wurde und noch immer etwa 50% der gesamten Rechenleistung der Plattform von ca. 900 Teraflop konsumiert<sup>7</sup>. Zum Vergleich dazu leistet der aktuell schnellste Supercomputer etwa 600 Teraflop<sup>8</sup>. SETI@Home ist ein wissenschaftliches Projekt, das auf der Suche nach außerirdischer Intelligenz, empfangene und aufgezeichnete Radiosignale des Arecibo-Observatoriums analysiert und auswertet [ACKL<sup>+</sup>02]. Durch seinen Erfolg ist es zum Vorbild für weitere Projekte aus medizinischen und wissenschaftlichen Bereichen geworden wie zum Beispiel Projekten der Molekülfaltung<sup>9</sup> oder der Klimavorhersage<sup>10</sup>.

Das Ziel der BOINC-Plattform besteht in der besseren Auslastung bestehender Rechenleistung [Ande04]. Jedoch gibt es entscheidende Unterschiede zum herkömmlichen Grid-Computing. Im herkömmlichen Grid-Computing bestehen die Ressourcen aus Supercomputern, Clustern oder

---

<sup>6</sup>Dies ist der Stand im Februar 2008. Aktuelle Werte lassen sich auf der Webseite des Projekts finden: <http://www.eu-egee.org/> (Letzter Abruf: 12. Oktober 2008)

<sup>7</sup>Stand Februar 2008. Aktuelle Werte sind auf den Webseiten des BOINC Projekts zu finden: [http://boincstats.com/stats/project\\_graph.php?pr=bo](http://boincstats.com/stats/project_graph.php?pr=bo) (Letzter Abruf: 12. Oktober 2008)

<sup>8</sup><http://www.top500.org/list/2007/11/100> (Letzter Abruf: 12. Oktober 2008)

<sup>9</sup><http://folding.stanford.edu/> (Letzter Abruf: 12. Oktober 2008)

<sup>10</sup><http://www.climateprediction.net/> (Letzter Abruf: 12. Oktober 2008)

Desktoprechnern, die Forschungsinstituten gehören und die meiste Zeit angeschaltet sind und über eine hohe Netzwerkbandbreite verfügen. Tritt ein Fehler in einem Rechner auf, so wird dieser aus dem Grid System entfernt und der Fehler kann außerhalb des Grid Systems behoben werden [AnMc07]. Im Unterschied dazu sind die Ressourcen der BOINC-Plattform über eine Netzwerkverbindung geringerer Bandbreite wie beispielsweise DSL verbunden und werden häufig ausgeschaltet und damit vom Internet getrennt. Die Projekte der BOINC-Plattform haben keine Kontrolle über die Rechner der Teilnehmer. Aus diesem Grund kann auch Fehlverhalten der einzelnen Rechner nicht ausgeschlossen werden.

Um die Risiken einer fehlerhaften Berechnung zu begegnen, unterstützt die BOINC-Plattform redundantes Berechnen der Ergebnisse, um fehlerhafte Ergebnisse zu erkennen und abzulehnen. Ein Projekt spezifiziert hierzu die Anzahl der zu berechnenden Ergebnisse für ein Arbeitspaket [KoAV07]. Sobald eine bestimmte Anzahl an Ergebnissen verfügbar ist, werden diese miteinander verglichen. Werden keine übereinstimmenden Ergebnisse gefunden, so wird das Arbeitspaket an weitere Rechner zur Berechnung versandt, bis eine vorher festgelegte absolute Obergrenze oder eine Zeitobergrenze erreicht worden ist.

Treten technische Fehler wie beispielsweise Rechner- oder Netzwerkausfall bei der Kommunikation zwischen der zentralen Projektsteuerung und den Teilnehmern auf, so wird die Kommunikation exponentiell verzögert. Dies verhindert einen übermäßigen Zugriff auf die Steuerungsrechner nach erneuter Inbetriebnahme.

In beiden Beispielen ist die Zuverlässigkeit ein zentraler Bestandteil des Entwurfsprinzips. Wird eine hohe Verfügbarkeit und Zuverlässigkeit im EGEE hauptsächlich mithilfe redundanter Hardware erreicht, so wird dies auf der BOINC-Plattform durch die mehrfache Vergabe der gleichen Rechenaufgabe mithilfe von Software gelöst. Neben der Redundanz als wichtigem Merkmal zur Erhöhung der Zuverlässigkeit kommen in Computational-Grid-Systemen noch zwei weitere Verfahren zum Einsatz [Dabr08]:

- **Checkpoint und Recovery:** Bei einem Checkpoint- und Recovery-Verfahren wird zu einem bestimmten Zeitpunkt ein Zwischenergebnis der Ausführung gespeichert. Dies ermöglicht im Falle eines Fehlers die Fortsetzung der Ausführung ab dem letzten bekannten Zwischenergebnis. Die Abspeicherung von Zwischenergebnissen und ihrer meist aufwendige Wiederherstellung im Fehlerfall ist bei diesem Verfahren jedoch mit einem erheblichen Mehraufwand verbunden, der sich in Form zusätzlicher Ressourcen und erhöhtem Implementierungsaufwand niederschlägt [Dabr08].
- **Rescheduling:** Rescheduling bezeichnet das Neustarten des Rechenjobs auf neuen Res-

sourcen nach Auftreten eines Fehlers. Da in diesem Fall neue Ressourcen bereitgestellt werden müssen, kann es zu einer Verzögerung der erneuten Ausführung durch die Allokation neuer Ressourcen kommen [Dabr08].

Neben der Bereitstellung von Rechenleistung werden Grid-Systeme häufig auch zum Speichern und Verteilen großer Datenmengen eingesetzt, die im folgenden Abschnitt hinsichtlich ihrer Risiken und möglicher Verfahren zum Umgang mit den technischen Risiken betrachtet werden.

## 2.4.2 Data-Grid

Data-Grids bieten Dienste an, die dem Benutzer helfen große in verschiedenen Speichersystemen gespeicherte Datensätze zu durchsuchen, zu übertragen und zu verändern und auch Kopien dieser Datensätze zu erstellen und zu verwalten [VeBR06]. Dient die erste Aufgabe hauptsächlich dem Zugriff auf große Datenbestände, so liegt der Fokus der zweiten Aufgabe von Data-Grids auf der schnellen Verteilung von großen Datenmengen an eine Vielzahl von Empfängern. Um diese beiden Aufgaben zu erfüllen, stellen Data-Grids mindestens diese beiden Funktionalitäten zur Verfügung: einen leistungsfähigen und zuverlässigen Datentransfermechanismus über ein Hochgeschwindigkeitsnetzwerk und einen skalierbaren Mechanismus zur Replica<sup>11</sup>-Suche und -Verwaltung [CFKS<sup>+</sup>00]. Weitere Data-Grid-Dienste sind abhängig von den Anforderungen der jeweiligen Anwendung. Beispiele hierfür sind Dienste zur Überwachung der Konsistenz der Daten, zur Datenfilterung oder zur Verwaltung von Metadaten<sup>12</sup>. Ändern sich Datensätze und existieren weiteren Kopien dieser Datensätze, ist es die Aufgabe eines Data-Grids, diese zu aktualisieren. Dabei muss die Persistenz der Daten gewährleistet werden [MoRW05].

Anwendungsgebiete von Data-Grids sind wissenschaftliche Disziplinen und kommerzielle Anwendungsfelder, in denen große Datenmengen einer Vielzahl an Forschern zur Verfügung gestellt werden müssen und Content-Delivery-Networks (CDNs) [BuPV08], in denen Daten beispielsweise von Webseiten vielen Benutzern weltweit zur Verfügung gestellt werden müssen [SSPS04].

Wissenschaftliche Anwendungsgebiete umfassen Projekte aus der Hochenergiephysik wie den Large-Hadron-Collider (LHC) am CERN in Genf [Lebr00], virtuelle Observatorien von Astronomieprojekten [GrSz02] oder Projekte zur Proteinsimulation wie beispielsweise BioGrid [BSRB<sup>+</sup>08],

---

<sup>11</sup>Unter einer Replica versteht man im Zusammenhang mit Data-Grids eine lokale Kopie eines Datensatzes [VeBR06].

<sup>12</sup>Metadaten sind Daten über Daten, die weitere Informationen wie beispielsweise Dateiname, Erstellungsdatum, Änderungsdatum oder Größe der Datei umfassen können.



wo Daten in der Größenordnung von Petabytes gespeichert und analysiert werden müssen.

Gemeinsame Charakteristika dieser Projekte können dahingehend zusammengefasst werden, dass Daten von einem Instrument, Experiment oder Netzwerk an Sensoren erzeugt und meist vor Ort gespeichert werden. Diese Daten werden zu anderen Speicherorten mithilfe eines Replica-Mechanismus übertragen, der den besten Speicherort für die Daten bestimmt. Benutzer fragen ihre lokales Replica-Verzeichnis nach dem Datensatz ab, den sie benötigen. Ist der Datensatz lokal verfügbar, so wird er dem Benutzer sofort zur Verfügung gestellt, andernfalls muss der Datensatz von einem entfernten Speicherort auf das lokale System kopiert werden und wird dann erst dem Benutzer zur Verfügung gestellt. Anschließend können die Daten beispielsweise von einem Computational-Grid verarbeitet werden. In diesem Beispiel sind die Daten in heterogenen Systemen gespeichert und werden über Organisationsgrenzen hinweg verteilt. Dabei werden der Status der Daten verwaltet und Ergebnisse der Datenanalyse mit den Partnern geteilt. Benutzer eines Data-Grids, die an gemeinsamen Daten arbeiten, bilden eine virtuelle Organisation.

Für die Umsetzung der Data-Grid-Funktionalität werden die bestehenden Grid-Services erweitert. Die Ressourcenschicht muss zwei grundlegende Dienste bereitstellen: einen Dienst für den Zugriff auf die Daten und einen weiteren Dienst zum Zugriff auf die Metadaten. Der Datenzugriffsdienst stellt Mechanismen für den Zugriff, die Verwaltung und den Transfer von Daten zur Verfügung, die auf den Speichersystemen gespeichert sind. Der Metadatenzugriffsdienst stellt Mechanismen für den Zugriff und die Verwaltungsinformationen über die Daten, die in dem Speichersystem gespeichert sind [CFKS<sup>+</sup>00]. Es wird eine Trennung zwischen den eigentlichen Nutzdaten und den Informationen über die Nutzdaten vorgenommen. Diese können getrennt behandelt werden, was die Flexibilität erhöht und das Data-Grid weniger fehleranfällig gegen Ausfälle macht. Beide Dienste dienen zur Abstraktion der zu Grunde liegenden Speichersysteme.

Auf der Ebene der Kollektivschicht werden Dienste zur Replica-Verwaltung hinzugefügt [CFKS<sup>+</sup>00]. Die Aufgabe eines Replica-Managers ist es, Kopien der Dateien bzw. Replicas dieser Dateien im spezifizierten Speichersystem zu erzeugen oder zu löschen. Häufig werden Replicas erzeugt, weil der neue Speicherort eine bessere Performanz oder Verfügbarkeit für den Zugriff auf einen bestimmten Ort oder von einem bestimmten Ort aus besitzt. Eine Replica kann auch wieder gelöscht werden, wenn der Speicherplatz für einen anderen Zweck benötigt wird. Die Replicas in einer Datei werden in einem Replica-Verzeichnis gespeichert, dessen Einträge einer Datei auf einem Dateisystem oder Speichersystem entsprechen.

Für die Verwaltung der Replicas wird das Ressourcenmanagement, das ebenfalls auf der Kollektivschicht angesiedelt ist, genutzt. Mithilfe des Ressourcenmanagements wird der Speicher

den Dateien und ihren Replicas zugewiesen und wieder freigegeben. Ein weiterer Dienst, den das Data-Grid auf der Kollektivschicht bereitstellt, ist die Replica-Selektion. Replica-Selektion ist der Prozess der Auswahl einer Replica, die einer Anwendung zur Verfügung gestellt wird. Hier werden verschiedene Optimierungskriterien eingesetzt: absolute Performance, Kosten oder Sicherheit [CFKS<sup>+</sup>00].

Risiken in Data-Grid sind neben dem obligatorischen Ausfall einer Grid-Komponente die Nichtverfügbarkeit eines Datensatzes als lokaler Replica. Dadurch wird ein entfernter Zugriff auf eine Datei notwendig, was eine Verletzung der zugesicherten Qualität bedeuten kann. Weitere technischen Risiken treten ein, wenn verschiedene Replica nicht synchron sind und dadurch unterschiedliche Inhalte aufweisen [Dabr08]. Während für das erste technische Risiko verschiedene Verfahren zur Replica-Selektion bereits existieren und eingesetzt werden, sind im Fall des zweiten technischen Risikos nur wenige Verfahren bekannt [Dabr08].

Neben dem wissenschaftlichen Anwendungsbereich finden sich auch Data-Grid im Bereich von Content-Delivery-Networks im Einsatz [PaBu07]. Das Hauptziel von Content-Delivery-Networks ist das Zwischenspeichern von Daten, um einen schnelleren Zugriff auf Webseiten oder Multimediainhalte Endnutzern anbieten zu können. Im Gegensatz zu den Data-Grid-Netzwerken in der Wissenschaft, die meist offen sind und Dienste der Open-Grid-Services-Architektur verwenden, werden Content-Delivery-Networks überwiegend kommerziell unter Verwendung einer meist proprietären Architektur betrieben. Unternehmen lagern hierbei die Verteilung ihrer Daten an CDN-Anbieter aus, die ihre Inhalte weltweit verteilen und nach benutzter Bandbreite abrechnen [PaVa06].

### 2.4.3 Utility-Grid

Unter dem Begriff Utility-Computing versteht man Technologien und Geschäftsmodelle mit denen ein Serviceanbieter seinen Kunden IT-Dienstleistungen in Form von standardisierten Services zur Verfügung stellt und diese nach Verbrauch abrechnet [WiMS04]. Die IT-Services werden wie homogene Güter behandelt [Davi04]. In Analogie zum englischen Begriff *Utility*, der Dienstleistungen von Versorgungsbetrieben wie beispielsweise Strom, Telefon, Wasser oder Gas beschreibt, sollen Grid-Ressourcen und Grid-Services als ein weiteres Utility aufgefasst werden. So sollen für Utility-Grid-Services die gleichen Anforderungen wie für die vier oben genannten Utilities gelten [RoWe04]:

- Die Benutzer betrachten Grid-Services als Notwendigkeit.



- Hohe Verfügbarkeit des Service ist entscheidend.
- Einfache Verwendbarkeit ist ein signifikanter Faktor.
- Die volle Nutzung der Kapazität ist begrenzt.
- Services sind skalierbar, was zu Skaleneffekten führt.
- Exklusive Rechte werden für die Benutzung eines Service innerhalb abgegrenzter Ressourcen gewährt.

Diese Faktoren bilden auch die Grundlage für das Geschäftsmodell eines Utility-Grids. Ein Utility-Grid verwendet Grid-Technologien der Open-Grid-Services-Architektur, um das Paradigma des Utility-Computing zu realisieren. Neben den Technologien der Open-Grid-Services-Architektur werden in Utility-Grids auch Autonomic-Computing-Technologien [KeCh03] eingesetzt, um eine höhere Zuverlässigkeit durch autonome Selbstverwaltung der Grid-Systeme zu erreichen.

Für Kunden sind die Kosten variabel und basieren auf der aktuellen Kapazität, die sie nutzen – im Gegenteil zu fixen Kosten für eine Kapazität, die sie nur zu Spitzenzeiten benutzen [RoWe04]. Sie können die Kapazität jederzeit nachfragen, ohne Geld für eigene neue Ressourcen ausgeben zu müssen. Hersteller bieten neben grundlegenden Ressourcen von Computational- und Data-Grids auch Anwendungen und Geschäftsprozesse als Utility an.

Um Utility-Grids zum dynamischen Bezug von IT-Services einsetzen zu können, müssen verschiedene Anforderungen erfüllt sein. Eine Anforderung ist, dass sowohl Anbieter als auch Konsument ihre aktuell genutzten IT-Systeme so umorganisieren, dass sie in das Utility-Computing einbezogen werden können [YBAY<sup>+</sup>07]. Anbieter müssen die spezifischen Service-Bedürfnisse und Anforderungen der Konsumenten verstehen, um passende Policies für die Kunden zu erstellen. Mit der wechselnden Nachfrage nach Diensten von Seiten der Benutzer müssen Anbieter die Möglichkeit haben, die dynamische Fluktuation von Spitzen- und Nichtspitzen-Nachfrage nach Services zu erfüllen. Zur Festlegung der Dienstgüte werden von Anbietern Service-Level-Agreements verwendet, um die den Benutzern zugesicherte Dienstqualität zu erbringen. Wenn die erwartete Dienstqualität nicht eingehalten wird, sind die Anbieter verantwortlich, Kompensationsleistungen zu erbringen. Aus diesem Grund versuchen Anbieter, die Kundenzufriedenheit zu maximieren, um die Servicebedürfnisse zu befriedigen und das Risiko von SLA-Verletzungen zu minimieren [BCLW<sup>+</sup>04]. Eine serviceorientierte Architektur und autonome Kontrollmechanismen helfen, dieses Ziel zu erreichen [KeCh03] [Murc04].

Neben den technologischen Aspekten der Bereitstellung von Grid-Services als Utility, müssen sich die Anbieter auch die finanziellen Aspekte der Bereitstellung überlegen. Finanzielles Risikomanagement für Utility-Computing besteht aus zwei Faktoren [KeCh05]: Risiken bei der Lieferung des Dienstes und Preisrisiken. Die Risikofaktoren bei der Lieferung sind vor allem auf die technischen Risiken Verfügbarkeit und Zuverlässigkeit zurückzuführen, die in Kapitel 1 beschrieben wurden. Preisrisikofaktoren untersuchen die Risiken bei der Bepreisung des Dienstes mit Bezug auf die Verfügbarkeit der Ressourcen. Bei gegebenen Verträgen mit kurzer Laufzeit, niedrigen Wechselkosten und unsicherer Nachfrage der Konsumenten in Utility-Computing-Umgebungen ist es wichtig, dynamische und flexible Preisschemata zu haben, um Profite maximieren zu können und mögliche Verluste der Anbieter zu verringern [Pale04].

Im folgenden Abschnitt werden die technischen Risiken anhand der organisatorischen Ausprägung der Grid-Systeme untersucht. Sowohl Computational-Grids als auch Data-Grids und Utility-Grids können in einer der folgenden Organisationsformen realisiert werden, die mit ihren Eigenschaften technische Risiken betonen oder abschwächen können.

## 2.5 Organisationsformen von Grid-Systemen

Ein Grid-System kann im Allgemeinen über eine abgegrenzte Umgebung definiert werden. Die Abgrenzung eines Grid-Systems kann von einem kleinen Abteilungsnetzwerk bis hin zu einer großen Ansammlung an Ressourcen und Services reichen, die an verschiedenen Orten betrieben werden, über die Welt verteilt sind und vielen verschiedenen Organisationen wie Unternehmen oder Universitäten gehören. Die drei wichtigsten Organisationsformen, Enterprise-Grids, Inter-Grids und Open-Grids, werden in den folgenden Kapiteln detailliert vorgestellt. Diese drei Organisationsformen haben gemeinsame Aufgaben. Sie umfassen die Erzeugung, die Verwaltung und den Einsatz einer dynamischen Menge an Services und Ressourcen, die eine virtuelle Organisation bilden [FoKT01].

### 2.5.1 Enterprise-Grid

Der Begriff *Enterprise-Grid* – in der Literatur auch *Intra-Grid* genannt – definiert ein Grid-System, das von einer einzigen Organisations- oder Geschäftseinheit verwaltet wird [JoEF04]. Das Abgrenzungskriterium ist hier die verantwortliche Organisationseinheit und nicht der geografische Ort oder der Besitz der Ressourcen. Ein Enterprise-Grid kann sich nach dieser Defini-

tion durchaus auf verschiedenen Orte oder Rechenzentren verteilen.

Diese Grid-Organisationsform hat den Vorteil, dass einheitliche Regeln für die Zuweisung von Ressourcen an die einzelnen Benutzer und Anwendungen des Unternehmens definiert werden können, die sich eng an den Geschäftszielen des Unternehmens orientieren. Innerhalb des Unternehmens werden die Ressourcen automatisiert verwaltet, sodass Fehler einer manuellen Zuweisung der Ressourcen vermieden werden können [Stro05]. Interne Preise für die Nutzung des Enterprise-Grids ermöglichen die Abrechnung der Ressourcen mit den einzelnen Abteilungen des Unternehmens. Allerdings wird die Agilität des Enterprise-Grids durch seine enge Abgrenzung im Vergleich zu den anderen Grid-Organisationsformen eingeschränkt, da nur die Ressourcen der verwaltenden Organisationseinheit zur Verfügung stehen.

Typische Einsatzgebiete von Enterprise-Grids sind Rechenzentren und Desktoprechner von großen Unternehmen, die über ein lokales Netzwerk verbunden sind. So bestehen heutige Rechenzentren häufig aus einzelnen, in Silos organisierten Servern oder Clustern, die eine spezifische Aufgabe wie z. B. einen Datenbankdienst oder einen Web-Server-Dienst bereitstellen. Durch Integration dieser einzelnen Dienste auf ein Enterprise-Grid unter Verwendung einer serviceorientierten Architektur ermöglicht die Nutzung einer einheitlichen Plattform, die bestehende Kapazitäten besser auslastet und sich leichter verwalten lässt [Stro05]. Ungenutzte Kapazitäten von Desktoprechnern können durch den Einsatz von Grid-Technologie nutzbar gemacht werden und Teile rechenintensiver Aufgaben innerhalb eines Unternehmens zu übernehmen. Jedoch sind noch weitere Arbeiten notwendig, um vor allem lang andauernde Transaktionen in Grid-Systemen zu unterstützen, die für diese Art von Transaktionen eine zu geringe Verfügbarkeit aufweisen [JiPMK07].

Durch die Möglichkeit, innerhalb eines Enterprise-Grids die vollständige Kontrolle über die Ressourcen zu haben, kann die Servicequalität der Ausführung genau festgelegt werden [MeCa04] und dazu beitragen, die Unsicherheit bezüglich der Servicequalität zu verringern [Stro05]. Das technische Risiko wird auf das Enterprise-Grid begrenzt.

### **2.5.2 Partner-Grid**

Ein Partner-Grid – in der Literatur findet sich auch der Begriff Inter-Grid – ermöglicht das Teilen von Grid-Ressourcen mit anderen Organisationen über öffentliche Netzwerkinfrastrukturen wie das Internet [JoEF04]. Partner-Grids verfolgen die Idee einer On-Demand-Bereitstellung von Ressourcen als Utility, wobei zusätzliche Komplexität durch Service-Level-Anforderungen,

den Aufbau einer organisationsübergreifenden Sicherheitsplattform und den Einsatz von Grid-Diensten zur Integration der einzelnen Grid-Systeme hinzukommt [JoEF04]. Die Ressourcen der Teilnehmer eines Partner-Grid sind dabei vertrauenswürdig und in ihrer Kapazität und Funktionalität auf die Ressourcen der Partner beschränkt [PIJr06, S. 69ff.] [HwPa07].

Es wird davon ausgegangen, dass die Ressourcennutzung nicht frei ist, sondern Anreize geschaffen werden, die Ressourcen in der angeforderten Dienstgüte zu liefern. Werden in Enterprise-Grids meist die Leistungen nur intern verrechnet, so ist das Ziel bei Partner-Grids der Einkauf und Verkauf von Rechenkapazität oder Grid-Services der beteiligten Organisationen über die gemeinsame Grid-Plattform. Dies kann durch Sammeln von Guthabenpunkten geschehen, wie es häufig in wissenschaftlichen Projekten – ein Beispiel hierfür ist das PlanetLab Projekt<sup>13</sup> – anzutreffen ist oder durch monetäre Kompensation zwischen Geschäftspartnern.

Die Umsetzung eines Partner-Grids erfolgt meist durch Abschluss von langfristigen Rahmenverträgen mit den beteiligten Partnern zur Nutzung der Grid-Systeme. Innerhalb dieser Rahmenverträge wird die allgemeine Nutzung der Dienste und Ressourcen vereinbart. Bei jeder Nutzung wird meist ein Service-Level-Agreement für diese Transaktion getroffen, um eine bestimmte Dienstgüte zu garantieren. Diese Dienstgüte wird dabei überwacht und dient als Basis für die Nutzungsabrechnung. Im Vergleich zu Enterprise-Grids kommen bei Partner-Grids zwei neue Aspekte hinzu [AsBu06]:

- Einsatz einer öffentlichen Netzwerkinfrastruktur: Durch die Verwendung einer öffentlichen Netzwerkinfrastruktur erhöht sich das Risiko eines Ausfalls. Daten werden mithilfe der Internetprotokolle übertragen, die eine *best-effort* Policy verfolgen. Dies bedeutet, dass die Daten auf dem gerade bestmöglichen Weg übertragen werden und dadurch die Performanz des Netzwerks deutlich über wie auch deutlich unter dem erwarteten Qualitätsniveau liegen kann. Verteilte Angriffe auf die Netzwerkinfrastruktur sind weitere Risikoquellen, die bei Partner-Grids entstehen und die Verfügbarkeit einzelner Systeme beeinträchtigen können.
- Aufrechterhaltung der lokalen Autonomie: In Partner-Grids werden einzelne, lokal autonome Grid-Systeme miteinander verbunden. Jedes dieser Grid-Systeme kann wie ein Enterprise-Grid mit eigenen Regeln zur Zuweisung und Verteilung von Ressourcen aufgefasst werden. Diese lokale Autonomie einzelner Grid-Systeme bleibt in Partner-Grids aufrechterhalten. Messungen von Vázquez et al. [VHML07] haben ergeben, dass sich dadurch der zeitliche Mehraufwand einer Grid-Job-Ausführung um etwa 10% erhöht.

---

<sup>13</sup>Für nähere Hinweise wird auf die Web Seiten des PlanetLab Projekts unter <http://www.planet-lab.org/> verwiesen.

Partner-Grids sind die erste Ausprägung von Grids, die über die Grenzen einer einzelnen realen Organisation bzw. Unternehmung hinausgehen und die Ressourcen des Grids auch externen Nachfragern zur Verfügung stellen und somit die Bildung einer oder mehrerer virtueller Organisationen erforderlich machen. Besonderes Augenmerk ist hierbei auf die Sicherheitsanforderungen zu richten, da die Bedrohungen nun nicht mehr ausschließlich von innerhalb der Organisation kommen, sondern auch und vor allem Eindringlinge von außen abgewehrt werden müssen. Ohne sichere Kommunikation zwischen den beteiligten Parteien scheint es nämlich in den meisten Fällen nicht möglich, aus der Verbindung einen Nutzen zu ziehen.

Vorteile von Partner-Grids sind die Möglichkeit der Nutzung von Ressourcen der Partner und die Zusicherung einer bestimmten Dienstgüte. Nachteil ist die Limitierung der verfügbaren Ressourcen auf die Ressourcen der Partner, was zu Engpässen bei der Ressourcennutzung führen kann.

Allgemein bekannte Vertreter dieser Ausprägungen sind in diversen wissenschaftlichen Projekten zu finden. Ein Beispiel ist das vom Bundesministerium für Bildung und Forschung geförderte MediGRID [BBCD<sup>+</sup>03]. Aufgabenbereich des von 8 Partnern betriebenen MediGRID ist die Grundlagenforschung in vielen medizinischen Bereichen. Besonders von Partner-Grids können Forschungsgebiete profitieren, bei denen der Zugriff auf große Datenmengen und hohe Rechnerleistungen erforderlich ist und die eigenen Ressourcen hierfür nicht ausreichen.

### 2.5.3 Open-Grid

Open-Grids setzen eine vollständig virtualisierte Plattform voraus, die alle relevanten Ressourcen dynamisch zur Verfügung stellt. Das Schlagwort *open* weist auf eine offene, marktbasierende Koordination der Grid-Dienste unter Verwendung von SLAs hin. Hierbei muss sich der einzelne Nachfrager nicht mehr darum sorgen, selbst einen geeigneten Anbieter ausfindig zu machen, sondern kann die benötigte Leistung direkt aus dem Netz beziehen. Das Konzept der Open-Grids weist daher starke Ähnlichkeit mit der Definition der Utility-Grids auf, postulieren doch beide die Existenz eines weltweit verfügbaren Angebots an einfach zu beziehenden IT-Diensten. So handelt es sich bei den nachgefragten Grid-Services in der Regel um hochkomplexe Leistungen, die für gewöhnlich nicht ausschließlich aus der Komponente Rechenleistung bestehen, sondern stets auch andere Größen wie den benötigten Speicher in einem Ressourcenbündel umfassen. Weiterhin bestimmt sich der theoretisch gerechte Preis der Grid-Ressourcen in Abhängigkeit von der Gesamtnachfrage, was zu stark schwankenden Preisen bei stark schwankender Nachfrage führen kann.

Open-Grid-Systeme sind noch ein wichtiges Gebiet in der Forschung, dem in den letzten Jahren viel Beachtung geschenkt wurde. Vereinzelt befinden sich Open-Grid-Systemen in der Erprobungsphase. Erste Ansätze zur Schaffung einer offenen Grid-Infrastruktur sind beispielsweise die Cloud-Computing-Dienste von Amazon Web Services LLC <sup>14</sup>, die über Web-Service-Schnittstellen zur Verfügung gestellt werden. Diese Dienste können ohne vorherigen Rahmenvertrag von jedem Benutzer verwendet werden. Die Abrechnung erfolgt nach einem pay-per-use-Modell. Zwei Dienste stehen zur Verfügung <sup>15</sup>:

- Amazon Elastic-Compute-Cloud (EC2): Dieser Dienst stellt dynamisch Rechenleistung aus einer Rechenwolke im Sinne des Cloud-Computing bereit. Die Menge der Rechenleistung passt sich den Anforderungen der Anwendung an.
- Amazon Simple-Storage-Service (S3): Dieser Dienst bietet grundlegende Speicherfunktionalität an. Wie auch im ersten Dienst wird dynamisch Speicherplatz aus der Cloud an die nachfragende Anwendung zugeteilt und nur der tatsächliche Verbrauch abgerechnet.

Zusammen bilden beide Dienste die wichtigsten Funktionalitäten von IT-Ressourcen ab: Rechnen und Speichern. Darüber hinaus werden keine Mehrwertdienste angeboten<sup>16</sup>. Die Abrechnung erfolgt auf Basis von vorgegebenen Güterbündeln, die mit der Zeiteinheit CPU-Stunde gebucht werden können. Zusätzlich wird die benutzte Bandbreite für den Datentransfer hin und wieder zurück nach Bedarf berechnet.

Die technischen Risiken, wie sie zu Beginn dieses Kapitel eingeführt wurden, sind besonders stark in Open-Grid-Systemen ausgeprägt. Die Behandlung der Risiken wird im Amazon-Beispiel auf den Konsumenten der Ressourcen abgewälzt. Dies kann im schlimmsten Fall dazu führen, dass der Konsument für verbrauchte Ressourcen bezahlt, jedoch kein Ergebnis seiner Berechnung erhält. Zusätzlich werden von Amazon nur unzureichend Service-Level-Agreements unterstützt, die nur eine niedrige Verfügbarkeit garantieren und eine Reihe an Ausschlussklauseln beinhalten. Der Konsument ist in diesem Fall einer starken Unsicherheit hinsichtlich der Qualität ausgesetzt.

Nach der Betrachtung verschiedener Organisationsformen und Anwendungsfeldern von Grid-Systemen und deren technischen Risiken werden im folgenden Abschnitt verschiedenen Studien vorgestellt, die eine Quantifizierung der technischen Risiken ermöglichen.

---

<sup>14</sup>Amazon Web Services LLC: <http://www.amazon.com/gp/browse.html?node=3435361> (Letzter Abruf: 31.03.2008)

<sup>15</sup>Cloud Computing Dienste von Amazon: <http://www.amazon.com/gp/browse.html?node=201590011> (Letzter Abruf: 31.03.2008)

<sup>16</sup>Dies ist der Stand im März 2008.



## 2.6 Studien über technische Risiken und Risikomodelle

### 2.6.1 Studien zur Fehleranalyse in Computational-Grids

In den letzten Jahren wurde in verschiedenen Studien die Zuverlässigkeit in produktiv betriebenen Grid-Systemen analysiert. Die Analysen wurden an Computational-Grid-Systemen durchgeführt, die in der Wissenschaft zum Einsatz kommen. Wegen der Komplexität der Messung und der Vielzahl an Anfragen an diese Systeme war es bisher nur sehr schwierig, verlässliche Messungen über Zuverlässigkeit und Dienstgüte durchzuführen. Aus diesem Grund werden in den Studien nur grundlegende technische Metriken gemessen. Messdaten über kommerziell betriebene Utility-Grid-Netzwerke, die als Open-Grid organisiert sind, liegen bisher noch nicht vor, da diese Grid-Systeme bisher noch nicht in großem Umfang zum Einsatz kommen.

Die Ergebnisse der ersten Studie sind in Tabelle 2.1 dargestellt. In drei Zeitperioden wurden die auf dem LHC Computational-Grid ausgeführten Jobs analysiert. Das LHC Computational-Grid bietet Physikern die Möglichkeit, Daten der LHC-Experimente, die im Conseil Européen pour la Recherche Nucléaire (CERN) bei Genf erzeugt werden, zu speichern und weltweit zur weiteren Analyse verfügbar zu machen. Die Messergebnisse sind in 7 Kategorien eingeteilt. Für jede dieser Kategorien werden die absoluten Zahlen dargestellt und deren prozentualer Anteil an der gesamten Menge berechnet.

Die Kategorien setzen sich aus den folgenden Bestandteilen zusammen:

- **Ran:** Der Job wurde vollständig ausgeführt.
- **Success:** Das Ergebnis der Ausführung wurde korrekt empfangen.
- **Res (Resubmit):** Nach einem Fehler wird der gleiche Job erneut ausgeführt.
- **Aborted:** Der Job wurde vom Ressourcenanbieter mit einem Fehler abgebrochen. Der Abbruch kann sowohl während als auch nach der Ausführung aufgetreten sein.
- **Cancelled:** Der Job wurde vom Benutzer abgebrochen.

Eine Job in der Kategorie *Ran-Res-Success* bedeutet, dass nach der ersten Ausführung des Jobs ein Problem aufgetreten ist, der Job ein zweites Mal ausgeführt wurde und bei dieser zweiten Ausführung erfolgreich ein Ergebnis geliefert hat. Wie in Tabelle 2.1 aufgelistet, sind signifikante Fehler bei der Job-Ausführung im LHC-Grid zu beobachten. In den gemessenen Perioden sind bei 25% - 33% der ausgeführten Jobs Fehler aufgetreten, wobei immerhin noch 5% - 8%

Tabelle 2.1: Messergebnisse zur Zuverlässigkeit der Job-Ausführung im LHC-Grid für drei Messperioden aus dem Jahr 2006 [LGWT06]

Zeitperiode	20-30 Nov		7-10 Dec		19-30 Dec	
	#Jobs	Prozent	#Jobs	Prozent	#Jobs	Prozent
Ran-Success	169.590	64,86%	135.178	71,50%	212.684	72,43%
Ran-Res-Success	4.771	1,83%	6.207	3,28%	7.089	2,42%
Aborted	51.571	19,72%	25.453	13,46%	38.343	13,06%
Res-Aborted	12.348	4,72%	6.543	3,46%	10.498	3,58%
Ran-Aborted	1.078	0,41%	1.443	0,76%	1.197	0,41%
Ran-Res-Aborted	9.444	3,61%	12.483	6,60%	16.075	5,48%
Cancelled	12.677	4,85%	1.794	0,95%	7.746	2,65%
Fehler gesamt	87.118	33,31%	47.676	25,22%	73.859	25,15%

der Fehler nach der Berechnung auf dem Grid gemessen wurden. Eine erneute Ausführung der fehlerhaften Jobs führt in ca. 30% der Jobs zu einer erfolgreichen Ausführung.

Diese Messdaten werden in der Arbeit von Lie weiter ausgewertet und nach statistischen Zusammenhängen untersucht. Ihre Ergebnisse zeigen, dass das Auftreten von Fehlern in Schüben passiert und von einzelnen Knoten im System dominiert wird [LGWT06]. Eine differenzierte Betrachtung der Fehlerquellen findet in der Studie nicht statt, da die verfügbare Software dies nicht zuließ.

An dieser Stelle setzt die zweite Studie an. In dieser Studie werden sowohl die Ursachen für auftretende Fehler als auch Qualitätsschwankungen bei der Ausführung von Jobs auf Computational-Grid-Systemen gemessen und analysiert. Im Unterschied zur vorangegangenen Studie, die passiv Messungen von verschiedenen Benutzergruppen analysiert hat, wird in dieser Studie aktiv mit einem Satz an vordefinierten Test-Jobs die Untersuchung durchgeführt. Zum Einsatz kommen zwei produktive Grid-Systeme, TeraGrid und GEON. Die beiden Grid-Systeme unterscheiden sich in ihrem Aufbau. Ist die Struktur der Grid-Ressourcen im GEON-Grid weitgehend homogen, kommen im TeraGrid stark heterogene Grid-Ressourcen zum Einsatz. Der Versuchsaufbau bestand darin, über einen Zeitraum von 3 Monaten stündlich je einen Test-Job an die beiden Grid-Systeme zu schicken. Die Ergebnisse sind in Tabelle 2.2 dargestellt. Die Ausführung eines Jobs wurde in 4 Kategorien aufgeteilt: eingehender Datentransfer (Data Transfer IN) der zu berechnenden Daten, die Berechnung selbst (Compute) und ausgehender Datentransfer (Data



Tabelle 2.2: Messergebnisse in Sekunden bei der Ausführung eines Test-Jobs in zwei Computational-Grid-Systemen, TeraGrid und GEON-Grid. Dieser Job benötigt 100 MB Daten als Eingabe, die im Rechenschritt aggregiert werden und abschließend in einer 500MB Datei an den Aufrufer zurückgeschickt werden. Über einen Zeitraum von drei Monaten wurde stündlich dieser Test-Job an beide Systeme geschickt. [KHOS<sup>+</sup>06]

Statistik	Data Transfer IN		Compute		Data Transfer OUT		Runtime	
	TeraGrid	GEON	TeraGrid	GEON	TeraGrid	GEON	TeraGrid	GEON
Average	8,38	29,12	18,40	10,58	15,72	15,88	227,62	369,01
Min	5,28	21,02	17,99	10,15	13,38	14,31	145,37	324,80
Max	194,09	81,30	61,21	22,67	130,53	189,03	1142,66	720,24
STDEV	9,20	11,68	2,04	0,94	7,76	9,00	116,02	34,79

Transfer OUT) für die Übertragung der Ergebnisse und die Gesamtausführungszeit. Die Zeitnahme selbst erfolgt in Sekunden. Für die Erstellung der Tabelle wurden nur erfolgreiche Jobs verwendet; fehlerbehaftete Jobs wurden ausgeblendet.

Deutliche Abweichungen vom Mittelwert weisen die vom Netzwerk abhängigen Messwerte *Data Transfer*. Hier kommt zum Tragen, dass die Daten über ein öffentliches Netzwerk transportiert werden, das keine Zusicherungen hinsichtlich der zu erwarteten Qualität macht. Die Berechnung des Jobs selbst hingegen zeigt hier deutlich weniger Abweichung vom Mittelwert. Jedoch ist ein signifikanter Unterschied in der gemessenen Ausführungszeit zwischen den eher homogenen Ressourcen im GEON-Grid und den heterogenen Ressourcen im TeraGrid zu erkennen.

Die Qualitätsschwankungen vergrößern sich noch stark in der Gesamtausführungszeit. Diese wird dominiert von den Warteschlangen der Grid-Systeme, in die die Jobs vor ihrer Ausführung eingereiht werden. Sie sind für die Betrachtungen in dieser Arbeit weniger relevant, da nur Jobs betrachtet werden, die sofort ausgeführt werden und nicht vor ihrer Ausführung in eine Warteschlange eingereiht werden.

Ebenfalls wurde im Rahmen dieser Studie die Zuverlässigkeit der Grid-Systeme gemessen. Für die Test-Jobs lag die Erfolgsquote zwischen 55% und 80%, was in etwa den Messungen in der ersten Studie entspricht. Die häufigsten Fehler, die während der Studie auftraten, waren auf Benutzerfehler, Fehler der Middlewareumgebung und eine fehlerhafte Konfiguration von Netzwerkkomponenten zurückzuführen. Insgesamt war in den beiden untersuchten Grid-Systemen weniger die Hardware die Fehlerursache; die meisten Fehler haben ihren Ursprung in der eingesetzten Grid-Middleware.

Die dritte Studie konzentriert sich ebenfalls auf die Zuverlässigkeit von Grid-Systemen. Hierfür wurden Daten des französischen Grid5000 über ein Jahr erhoben und ausgewertet [IJSE07]. Die Zuverlässigkeit des Computational-Grid mit 15 Standorten und 1296 einzelnen Rechnern schwankte in dieser Zeit zwischen 35% und 98%. Die mittlere Zeit zwischen zwei Fehlern lag in diesem Zeitraum bei 12 Minuten. Die Messungen der Fehler bezogen sich in fast allen Fällen auf nur einen Standort, was die Annahme unterstützt, dass zwei Grid-Standorte als unabhängig von einander anzusehen sind.

Die Fehler im Grid5000 wurden wesentlich von Hardwareausfällen während der Ausführung und fehlerhaften Informationen der Zuweisungsalgorithmen über den Zustand der Rechner beeinflusst. Die Zuweisung der Jobs zu den Knoten im Grid-System erfolgt auf nicht mehr aktuellen Daten, was zu einem Abbruch des Jobs führte [IJSE07].

## 2.6.2 Das Modell von Dai et al. zur Modellierung von Grid-Risiken

Um das Risiko, das von den Fehlerquellen ausgeht, festzulegen, müssen Erwartungswerte über den Eintritt eines Fehlers ermittelt werden. Diese werden als Zufallszahlen in der Grid-Simulation verwendet. Grundlage für das Risikomodell in der Simulationsstudie bildet das Modell von Dai et al. [DaPZ07], das in verschiedenen Teilschritten zu einem umfassenden Modell erweitert wurde [DaXP02] [XiPD04] [LeDa07]. Es ist das bisher einzige Modell, das verschiedene Ebenen der Grid-Risiken betrachtet [Dabr08]. Darüber hinaus verwendet das Modell einfache zu messende Werte für die Bestimmung des Risikos, die auch in einer Simulationsumgebung umsetzbar sind. Im Brennpunkt des Modells steht die Zuverlässigkeit, mit der ein Grid-Dienst auf Grid-Ressourcen ausgeführt werden kann.

Als Kennzahl für die Bewertung der Grid-Performanz kommt die Ausführungszeit eines Grid-Dienstes zum Einsatz. Wird der Dienst innerhalb einer vom Benutzer vorgegebenen Zeit ausgeführt, so wird die Ausführung als erfolgreich angenommen, andernfalls ist ein Fehler bei der Dienstauführung aufgetreten.

Abbildung 2.6 zeigt risikobehaftete Komponenten bei der Ausführung auf Grid-Systemen und ordnet diese den Schichten der Grid-Protokollarchitektur zu, die in Sektion 2.1 vorgestellt wurde. Zusätzlich zu den bisherigen Betrachtungen werden in diesem Modell auch das Ressourcenmanagement und die Ressourcennachfrage als eigenständige risikobehaftete Komponenten mit aufgenommen. Die Schichten des Modells sind:

- *Ressourcennachfrage*: Die Ressourcennachfrage wird durch zwei Fehlerarten im Modell

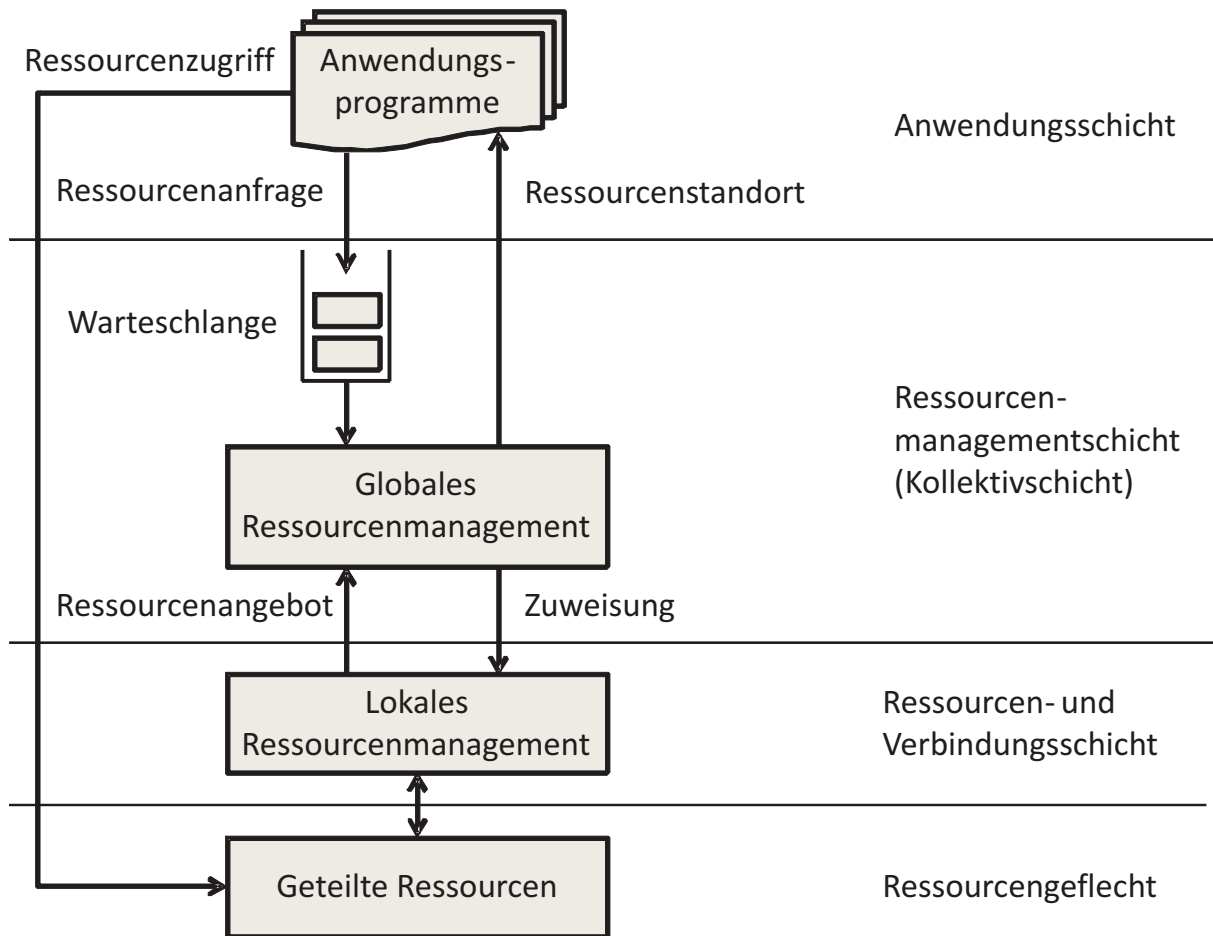


Abbildung 2.6: Risikobehaftete Komponenten von Grid-Systemen und ihre Zuweisung zum Schichtenmodell der Grid-Protokollarchitektur in Anlehnung an [DaPZ07]

von Dai [DaPZ07] beschrieben: Fehler durch Blockieren und Fehler durch Überschreitung von Zeitgrenzen (time-out-Fehler). Fehler durch Blockieren treten durch die meist in ihrer Kapazität beschränkte Eingangswarteschlange eines Grid-Systems auf, in der die Anfragen zwischengespeichert werden bevor sie vom Ressourcenmanagement abgearbeitet werden. Ist die Kapazität der Eingangswarteschlange erschöpft, werden alle neuen Anfragen von Benutzern abgewiesen. Wartet die Ressourcenanfrage zu lange in einer Warteschlange, so tritt ein time-out-Fehler auf.

- *Ressourcenmanagement*: Während der Zuweisung von Grid-Ressourcen an Nachfrager können verschiedene Fehler auftreten [DaPZ07]:
  - Fehlerhafte Beschreibung der Ressourcen: Durch eine fehlerhafte Beschreibung der Ressourcen kann es zu einer falschen Zuweisung kommen, die eine Ausführung eines Grid-Dienstes auf Grid-Ressourcen unmöglich macht.
  - Keine Zuweisung: Das Ressourcenmanagement konnte keine passenden Ressourcen im Grid-System finden und zuweisen. Gründe können beispielsweise zu wenig Ressourcen durch die Grid-Anbieter sein oder Fehler in den eingesetzten Zuweisungsalgorithmen.

Im Gegensatz zum Modell von Dai, der von konkreten Allokationsalgorithmen abstrahiert, wird im Rahmen dieser Arbeit eine Methode zur Allokation von Ressourcen mit in die Simulation einbezogen. Dies ermöglicht eine detailliertere Betrachtung auftretender Fehler. Jedoch werden diese Fehler nur am Rande betrachtet, da es bisher noch keine verlässlichen Messungen hinsichtlich des Einflusses dieser Risiken in produktiven Grid-Systemen gibt.

- *Netzwerk*: Das Grid-Netzwerk besteht aus Knoten und Kanten, die die Verbindungen zwischen den einzelnen Grid-Standorten darstellen. Die Verbindungen zwischen den Knoten werden als virtuelle Verbindungen angenommen, um von physikalischen Netzwerkverbindungen zu abstrahieren und Fehler auf logischer Ebene zu betrachten. Einzelne Knoten und Verbindungen zwischen den Knoten werden durch eine konstante Ausfallrate modelliert, die durch weitere dynamische Netzwerkparameter ergänzt werden [DaPZ07]. Die zusätzlichen Netzwerkparameter werden im Folgenden aufgeführt
  - Für virtuelle Verbindungen: verfügbare Bandbreite, Menge der übertragenen Daten  
Die verfügbare Bandbreite stellt die freie Kapazität der Netzwerkverbindung dar, die für eine Übertragung der Daten genutzt werden kann. Abhängig von der zur Verfügung stehenden Bandbreite  $B$  und der zu übertragenden Datenmenge  $D$  lässt sich die

Zeit  $T$  für die Datenübertragung über die Verbindung  $V$  zwischen dem Knoten  $i$  und dem Knoten  $j$  wie folgt berechnen [DaPZ07]:  $T(i, j) = D(i, j)/S(i, j)$ . Die Zuverlässigkeit der Datenübertragung  $R$  mit einer konstanten Ausfallwahrscheinlichkeit einer Verbindung  $\lambda_{i,j}$  ist damit:

$$R(i, j) = \exp^{-\lambda_{i,j} * T(i, j)}$$

- Für Knoten: Die gesamte Kommunikationszeit  $T$  mit einem Knoten  $j$  hängt von der Menge der Knoten  $i \in Q$  ab, die mit Knoten  $j$  kommuniziert:  $T(j) = \sum_{i \in Q_j} T(i, j)$ . Analog zur Zuverlässigkeit der virtuellen Verbindungen lässt sich die Zuverlässigkeit eines Knotens wie folgt modellieren [DaPZ07]:

$$R(j) = \exp^{-\lambda_j * T(j)}$$

- *Anwendung und Ressourcen*: Dieser Teil des Modells betrachtet die Ausführung einer Anwendung auf Grid-Ressourcen. Neben einer konstanten Ausfallrate für Grid-Ressourcen und Anwendungen beeinflusst die Länge der Ausführungszeit die Wahrscheinlichkeit einer erfolgreichen Ausführung. Wie bereits im Netzwerkmodell zur Anwendung gebracht, kommt auch hier eine Exponentialfunktion zum Einsatz. Im Detail ist die Zuverlässigkeit der Ausführung eines Programms  $i$  auf Ressourcen  $j$  mit Verfügbarkeit  $h$

$$R_{Res}(h, i, j) = \exp^{(-\lambda(h, j) * t(h, i, j))},$$

wobei  $\lambda(h, j)$  die Verfügbarkeit der Ressourcen  $h$  auf Knoten  $j$  und  $t(h, i, j)$  die Ausführungszeit der Anwendung  $i$  auf Knoten  $j$  unter Verwendung der Ressourcen  $h$  sind.

Für das Risikomanagement in dieser Arbeit stehen die Zuverlässigkeit des Netzwerks und die Ausführung einer Anwendung auf den Grid-Ressourcen im Vordergrund. Fehler, die von der Ressourcennachfrage und dem Ressourcenmanagement ausgehen werden aus Gründen der Komplexität und der fehlenden Messdaten von produktiven Grid-Systemen nur am Rande betrachtet.

Die Berechnung der Zuverlässigkeit eines Grid-Systems macht dabei einige problematische Annahmen hinsichtlich des Aufbaus des Grid-Systems:

- Es wird einen globaler Ressourcenmanager angenommen, der im Zentrum des virtuellen Netzwerks sitzt und die Zuweisung der Ressourcen vornimmt. Ein solches zentrales Ressourcenmanagementsystem (RMS) weist in großen Grid-Systemen Probleme hinsichtlich fehlender Skalierbarkeit und schlechterer Allokationsraten auf. Aus diesem Grund wird in dieser Arbeit ein dezentrales Ressourcenmanagementsystem verwendet, dessen Vorstellung in Kapitel 3 erfolgt.

- Eine Änderung der konstanten Ausfallrate von Ressourcen zur Laufzeit wird nicht in Betracht gezogen. Auch diese Annahme wird im Rahmen dieser Arbeit nicht gemacht. Das Risikomanagementverfahren wird hinsichtlich schubweise auftretender Risikoänderungen bei der Ausführung eines Grid-Dienstes untersucht.

## 2.7 Zusammenfassung

Die Betrachtungen dieses Kapitel zeigten, dass Grid-Systeme verschiedenen technischen Risiken unterworfen sind, die von einem Ressourcenmanagement berücksichtigt werden müssen. Diese Risiken können durch Fehler der Grid-Ressourcen und bei der Datenübertragung auftreten und Auswirkungen auf die Dienstgüte eines Grid-Service haben.

Maßnahmen wie redundantes Ausführen oder erneutes Ausführen bei eingetretenem Fehler werden aktuell eingesetzt, um die Zuverlässigkeit der Grid-Systeme zu erhöhen. Jedoch zeigen die Ergebnisse der vorgestellten Studien, dass dies nur bedingt zu einer Verbesserung der Zuverlässigkeit führt. Aus diesem Grund muss das Ressourcenmanagement die Möglichkeit bieten, weitere Strategien zur Behandlung von Risiken unterstützen.

Im folgenden Kapitel wird ein offenes Utility-Grid-System eingeführt, dessen Modell die technischen Risiken berücksichtigt und die Integration von Risikomanagementverfahren unterstützt. Zusätzlich ermöglicht das Grid-System durch den Einsatz ökonomischer Verfahren bei der Allokation der Ressourcen eine monetäre Bewertung der auftretenden Risiken.



# Kapitel 3

## Die CATNETS Grid-Ökonomie

*„Treating distributed computation as an economy turns out to be quite fruitful in the analysis, design, and control of such systems. In a sense, economics become a new programming methodology!“ [HuHo95]*

Ziel des Kapitels ist die detaillierte Beschreibung der CATNETS<sup>1</sup> Grid-Ökonomie [ERSR<sup>+</sup>05], die als Simulationsumgebung zur Beantwortung der Forschungsfragen dieser Dissertation herangezogen wird. In dieser Simulationsumgebung ist es wichtig, dass die in Kapitel 2 vorgestellten Risiken abgebildet werden können und die Grid-Versicherung eingebaut werden kann. Diese beiden Anforderungen sind essenziell wichtig für die Untersuchung der Grid-Versicherung im weiteren Verlauf dieser Arbeit.

In der CATNETS Grid-Ökonomie besitzt jeder handelnde Akteur eine Rolle, die er ausfüllt. Diese Rolle legt fest, ob der Akteur ein Konsument oder eine Anbieter in der Grid-Ökonomie ist und welche Produkte von ihm angeboten oder nachgefragt werden. Eine Rolle wird mithilfe von Softwareagenten umgesetzt, die als Repräsentanten menschlicher Entscheidungsträger autonom interagieren, um ihre Ziele zu erreichen [Maes94] [Jenn00].

Die ersten beiden Abschnitte stellen die CATNETS Grid-Ökonomie vor. Kapitel 3.1 beschreibt das abstrakte Utility-Grid-Modell, dessen allgemeinen Aufbau und die Rollen der Teilnehmer. Die Umsetzung dieses Modells präsentiert Abschnitt 3.2. Die verwendeten Algorithmen zur

---

<sup>1</sup>Das von der EU geförderte Forschungsprojekt Catallaxy paradigm for decentralized operation of dynamic application networks (CATNETS) beschäftigte sich zwischen 2004 und 2007 mit der effizienten Bereitstellung von Grid-Diensten durch skalierbare und dynamische Ressourcenallokationsmechanismen. Fokus des Projekts war die Evaluation des Katallaxie-Paradigmas für die dezentrale Steuerung von dynamischen Anwendungsschichtnetzwerken. Weitere Information und die Ergebnisse des Projekts finden sich auf dessen Webseite unter <http://www.catnets.eu> (Letzter Abruf: 02.04.2008).



Dienstsuche, Verhandlung und dynamischen Anpassung stehen hier im Fokus der Betrachtung. Abschnitt 3.3 fasst die Ergebnisse dieses Kapitels zusammen.

## 3.1 Das Utility-Grid-Modell

### 3.1.1 Service- und Ressourcenmarkt

Das Utility-Grid-Modell der CATNETS Grid-Ökonomie umfasst zwei voneinander abhängige Märkte, einen Service- und einen Ressourcenmarkt [ERSR<sup>+</sup>05] [ENRS<sup>+</sup>06], die in Abbildung 3.1 gezeigt werden. Die Wahl eines zweistufigen Markts orientiert sich dabei an den in Kapitel 1 eingeführten Geschäftsmodellen Software-as-a-Service und Infrastructure-as-a-Service, wobei Softwareprodukte auf dem Servicemarkt und Infrastrukturre Ressourcen auf dem Ressourcenmarkt abgebildet werden können. Die Aufteilung auf zwei Märkte lässt eine isolierte Betrachtung der Grid-Versicherung auf dem Ressourcenmarkt zu, ohne die Einflüsse des Servicemarkts zu vernachlässigen.

Auf beiden Märkten werden Grid-Services als Produkte gehandelt, die eine festgelegte Menge an Produktattributen und das Verhandlungsattribut *Preis* besitzen. Auf dem Ressourcenmarkt sind dies elementare Dienste eines Computational- oder Data-Grid zur Berechnung oder Speicherung von Daten, die im weiteren Verlauf als *Ressourcen-Services* bezeichnet werden. Obwohl die elementaren Dienste als standardisiert angenommen werden, werden sie durch verschiedene Attribute charakterisiert, die von Benutzern nachgefragt werden können. Auf dem Servicemarkt werden Anwendungsdienste gehandelt, die ein breites Spektrum von Anwendungen abdecken und in der Terminologie der CATNETS Grid-Ökonomie als *Basic-Services* bezeichnet werden. Beispiele hierfür sind Datenbankdienste oder Dienste zur Transformation von Daten. Anwendungsdienste verwenden dabei die elementaren Dienste zur Erfüllung ihrer Funktionalität. Es wird dabei angenommen, dass sich die Qualität der Anwendungsdienste wie die der elementaren Dienste standardisiert beschreiben lässt. Auf dem Servicemarkt werden die Anwendungsdienste von *Complex-Services* nachgefragt, die beispielsweise Workflows entsprechen können und einen oder mehrere *Basic-Services* benötigen.

Sowohl auf dem Servicemarkt als auch auf dem Ressourcenmarkt lassen sich Rollen für die Akteure identifizieren, die entweder der Gruppe der Anbieter oder die Gruppe der Nachfrager repräsentieren. Die Rollen des Utility-Grid-Modells der CATNETS Grid-Ökonomie sind [ENRS<sup>+</sup>06]:

- Servicekonsument: Ein Servicekonsument ist ein Nachfrager von *Basic-Services* auf dem

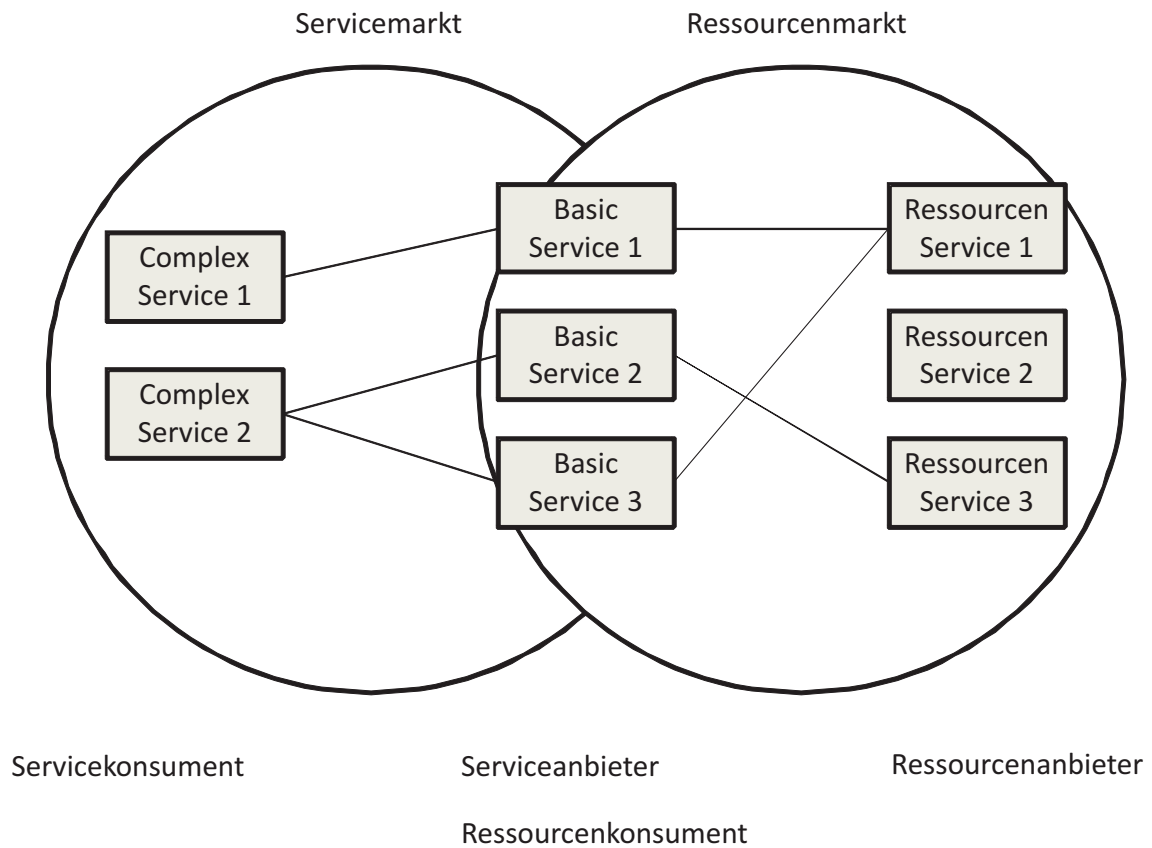


Abbildung 3.1: Das Utility-Grid-Modell der CATNETS Grid-Ökonomie: Aufteilung der Grid-Dienste in einen Servicemarkt, auf dem Anwendungsdienste (Basic-Services) gehandelt werden und einen Ressourcenmarkt, auf dem elementare Dienste (Ressourcen-Services) gehandelt werden. [ERSR<sup>+</sup>05]

Servicemarkt. Es arbeitet Anfragen der Grid-Benutzer ab, indem er einen Allokationsprozess startet, der mit der Dienstsuche beginnt und nach der Ausführung des Workflows mit Basic Services endet.

- **Serviceanbieter:** Ein Serviceanbieter ist ein Verkäufer eines *Basic-Service* auf dem Servicemarkt. Ein Serviceanbieter verwaltet die Anfragen, gibt Angebote ab und verhandelt mit den Servicekonsumenten. Dabei bedient er zu einem Zeitpunkt exklusiv einen Nachfrager.
- **Ressourcenkonsument:** Eng verbunden mit dem Serviceanbieter ist der Ressourcenkonsument, der bei erfolgreichem Abschluss durch seinen zugehörigen Serviceanbieter die entsprechende Menge an Ressourcen zur Dienstleistungserbringung auf dem Ressourcenmarkt nachfragt. Seine Aufgaben sind analog zum Servicekonsumenten die Suche nach Anbietern, ihre Auswahl und Verhandlung.
- **Ressourcenanbieter:** Ein Ressourcenanbieter ist ein Verkäufer eines *Ressourcen-Service*. Im Gegensatz zum Serviceanbieter kann ein Ressourcenanbieter mehrere Ressourcenkonsumenten zu einem Zeitpunkt bedienen.

Die oben beschriebenen Rollen werden durch drei verschiedene Softwareagenten umgesetzt, wobei der Softwareagent zur Realisierung eines *Basic-Service* die Rolle des Serviceanbieters und die Rolle des Ressourcenkonsumenten übernimmt. Erlöse der Serviceanbieterseite des Basic-Service dienen dabei der Ressourcennachfragerseite als Budget.

**Complex-Service-Agenten** Complex-Service-Agenten repräsentieren Workflows, die in Geschäftsprozessen eingebunden werden können. Sie sind der Zugangspunkt der Grid-Benutzer zum Utility-Grid und können *Complex-Services (CSs)* für Grid Benutzer ausführen. Dabei ist ein Complex-Service definiert als eine Menge an *Basic-Services (BSs)*. Complex-Service-Agenten sind nicht spezialisiert: Dies bedeutet, dass jeder Complex-Service Agent jede Anfrage akzeptiert und sich um die Ausführung der Basic-Services kümmert. Eine Anfrage eines Complex-Service durch einen Grid-Benutzer besteht aus der Festlegung des auszuführenden Workflows, einem Budget, das der Complex-Service-Agent zur Verfügung hat, und der maximalen Zeitdauer, die die Ausführung des Dienstes dauern darf. Die geforderte Dienstgüte setzt sich aus dem angeforderten Workflow und der maximalen Ausführungszeit zusammen.

**Basic-Service-Agenten** Basic-Service-Agenten stellen Complex-Service-Agenten Basic-Services bereit, die sie benötigen, um ihre Complex-Services dem Utility-Grid-Benutzer zu liefern. Basic-Services haben in der CATNETS Grid-Ökonomie zwei Attribute: *Name* und *Qualität*. Das Attribut *Name* ist ein eindeutiger Identifikator des Basic-Service, dessen Semantik (zum Beispiel die Art des bereitgestellten Service) allen Teilnehmern der Grid-Ökonomie bekannt ist. Das Attribut *Qualität* zeigt die zu erwartete Dienstgüte an. Die Werte der Dienstgüte werden in dem Modell als eine Menge diskreter Werte angenommen. Auch hier ist allen Marktteilnehmern bekannt, welches Qualitätsniveau sie erwarten können.

Gibt es in der Grid-Ökonomie beispielsweise einen Basic-Service mit dem Namen *PDFKonverterService* und den Qualitätsstufen  $\{bronze, silber, gold, platinum\}$  und einen weiteren Basic-Service mit dem Namen *DruckerService* und den Qualitätsstufen  $\{silber, gold\}$ , so lässt sich der Workflow eines Complex-Service aus diesen beiden Basic-Services wie folgt definieren:

$$CS_1 = \{\langle PDFKonverterService, gold \rangle, \langle DruckerService, silber \rangle\}$$

Im Gegensatz zum Complex-Service-Agenten, die jede Anfrage nach einem bestimmten Complex-Service bedienen können, bieten Basic-Service-Agenten grundsätzlich nur einen bestimmten Service mit einem bestimmten Qualitätsniveau an. Dies ist insofern keine einschränkende Modellannahme, da mehrere Basic-Service-Agenten nebeneinander im System existieren können. So ist es beispielsweise möglich, verschiedenen Qualitätsstufen eines bestimmten Basic-Service durch mehrere Basic-Service-Agenten zu modellieren oder mehrere Basic-Service-Agenten mit den gleichen Attributen im Grid-System zu platzieren. Dabei entspricht ein Basic-Service-Agent einem Anbieter eines Grid-Service im Sinne des Utility-Computing-Paradigmas. Er bietet einen standardisierten Dienst mit den gleichen Eigenschaften wie alle weiteren Anbieter an.

**Ressourcen** Ressourcen, die von einem Ressourcen-Agenten verwaltet werden, haben ein Attribut *Name*, zum Beispiel *Festplattenspeicher* oder *CPU*. Auch hier gilt die Annahme, dass das Attribut eindeutig ist und die Bedeutung allen Agenten bekannt ist. Zusätzlich kann für jede Ressource die Menge durch das Attribut *Menge* spezifiziert werden. Beispiele für Ressourcentupel sind: *Name = Festplattenspeicher* und *Menge = 50*, *Name = CPU* und *Menge = 100*. Eine weitere Annahme besteht darin, dass die Einheiten der Ressourcen zwischen den einzelnen Anbietern genormt sind. Die maximal verfügbare Menge einer Ressource eines bestimmten Ressourcenanbieters ist in der CATNETS Grid-Ökonomie limitiert.

**Ressourcenbündel** Ein Ressourcenbündel ist ein Ressourcentupel der Art

$$\langle \text{Ressourcenname}, \text{Ressourcenmenge} \rangle$$

Beispiele für Ressourcenbündel sind:

$$\text{Ressourcenbündel}_1 = \{ \langle \text{CPU}, 70 \rangle, \langle \text{Festplattenspeicher}, 40 \rangle \}$$

$$\text{Ressourcenbündel}_2 = \{ \langle \text{CPU}, 50 \rangle, \langle \text{Festplattenspeicher}, 50 \rangle, \langle \text{Hauptspeicher}, 75 \rangle \}$$

Jeder Basic-Service benötigt ein bestimmtes Ressourcenbündel für die Bereitstellung eines Dienstes. Sowohl die Anzahl der Einzelressourcen als auch deren Quantität werden vom Basic-Service-Anbieter festgelegt.

Eine Verbindung

$$\langle \text{PDF Konverter}, \text{gold} \rangle \longrightarrow \text{Ressourcenbündel}_1$$

spezifiziert, dass für die Ausführung einer PDF-Konvertierung mit der Qualität *gold* das Ressourcenbündel 1 benötigt wird, das 70 CPU Einheiten und 40 Festplattenspeichereinheiten umfasst.

**Ressourcenagenten** Ressourcenagenten bieten Ressourcen eines Ressourcenanbieters zum Verkauf an. Sie bilden die Schnittstelle zwischen den lokalen Ressourcen des Ressourcenanbieters auf der einen Seite und den Ressourcenkonsumenten auf der anderen Seite. Ihre Aufgabe ist es die Ressourcen zur Ausführung von Basic-Services eines Basic-Service-Agenten bereitzustellen. Jeder Ressourcenagent hat ein verfügbares Ressourcenbündel, das den maximalen Ressourcen des Ressourcenagenten entspricht. Beispiele für maximal verfügbare Ressourcenbündel (VRB) sind:

$$\text{VRB}_1 = \{ \langle \text{cpu}, 100 \rangle, \langle \text{storage}, 40 \rangle \}$$

$$\text{VRB}_2 = \{ \langle \text{cpu}, 100 \rangle, \langle \text{storage}, 40 \rangle, \langle \text{ram}, 150 \rangle \}$$

Auch hier gilt analog zu den Basic-Service-Agenten, dass mehrere Ressourcenagenten mit dem gleichen, maximal verfügbaren Ressourcenbündel nebeneinander in der CATNETS Grid-Ökonomie existieren können.

### 3.1.2 Netzwerkmodell

Das Netzwerkmodell beschreibt die Standorte des Grid-Systems und den Aufbau des Netzwerks zwischen den Standorten. Es bildet die wichtigsten Funktionen und Kennzahlen real existierender Netzwerke ab. Eine Modellierung des Netzwerks ist im Rahmen dieser Arbeit wichtig, um Risiken, wie sie von Netzwerken ausgehen, modellieren zu können. Wie die Risikostudien aus Kapitel 2 zeigen, geht ein Großteil der Risiken eines Grid-Systems vom Netzwerk aus. Die Knoten bzw. Standorte des Grid-Systems beheimaten die oben vorgestellten Services bzw. deren Agenten, die über das Netzwerk Nachrichten austauschen, um Anbieter zu suchen, Services und Ressourcen zu verhandeln und die Basic-Services mithilfe von Ressourcen-Services auszuführen.

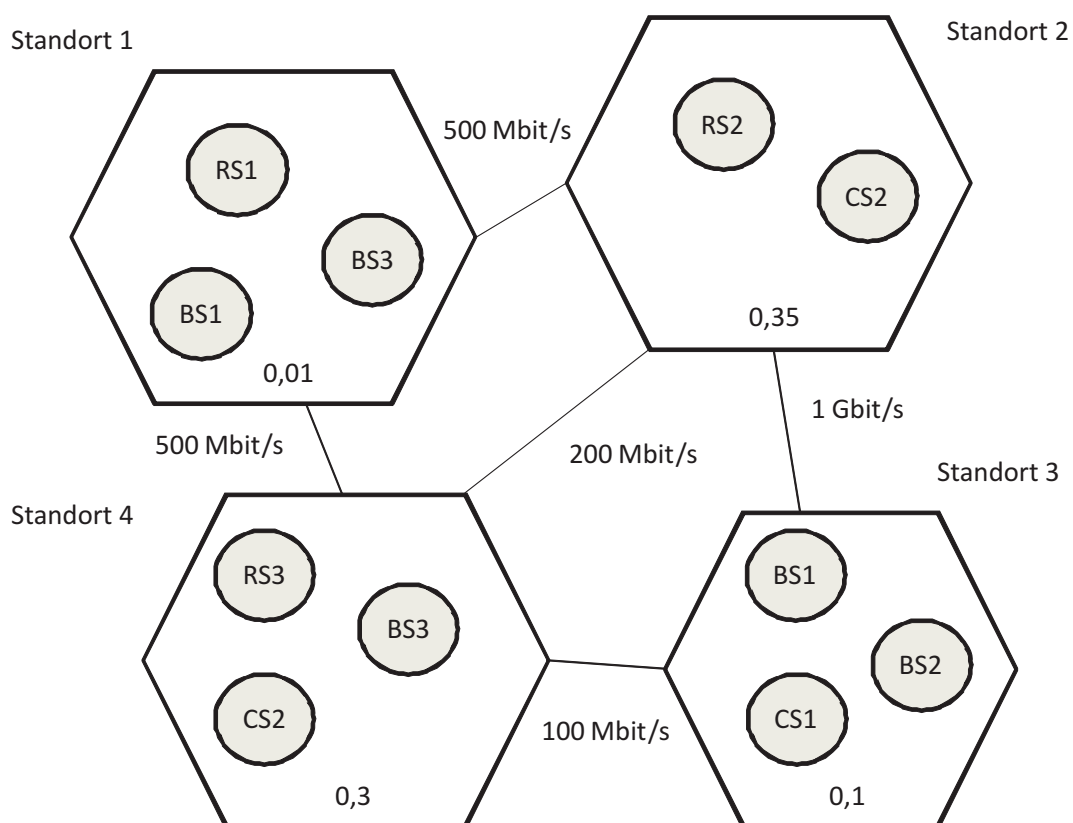


Abbildung 3.2: Netzwerkmodell zur Modellierung der Risiken, denen die Marktteilnehmer ausgesetzt sind [SHES<sup>+</sup>08]. Die Zahlenangabe innerhalb der Standorte legt die Ausfallwahrscheinlichkeit des Knotens fest.

Abbildung 3.2 zeigt den schematischen Aufbau des Grid-Netzwerks der CATNETS Grid-Ökonomie. Formal ist das Grid-Netzwerk als ungerichteter Graph definiert,

$$\text{GridNetzwerk} = \langle K, V \rangle$$

wobei  $K = \{1, \dots, n\}$  eine Liste an Knoten ist und  $V = \{\langle i_1, j_1 \rangle, \dots, \langle i_m, j_m \rangle\}$  die Kommunikationsverbindungen zwischen den Knoten definiert. Jeder Knoten  $i$  wird charakterisiert durch:

- eine *Ausfallwahrscheinlichkeit*, die die Zuverlässigkeit des Knotens festlegt,
- ein Tripel aus

$$\langle \text{CSA}_i, \text{BSA}_i, \text{RA}_i \rangle,$$

wobei  $\text{CSA}_i$  eine Menge an *Complex-Service-Agenten*,  $\text{BSA}_i$  eine Menge an *Basic-Service-Agenten*, und  $\text{RA}_i$  eine Menge an *Ressourcenagenten* beschreibt. Jeder Knoten im Netzwerk kann 0 oder mehr CSAs, BSAs, und RAs beherbergen:

$$|\text{CSA}_i| \geq 0, |\text{BSA}_i| \geq 0, |\text{RA}_i| \geq 0$$

Ein Knoten, der keine Agenten besitzt, ist als *Router* modelliert, d. h., dieser Knoten erbringt keine Grid-Dienstleistung in Form der drei definierten Services sondern leitet nur eingehende Nachrichten an alle erreichbaren Nachbarknoten weiter. Knoten ohne Agenten modellieren die in echten Grid-Systemen bestehenden Verzögerungen zwischen verschiedenen Grid-Standorten.

Jede Kommunikationsverbindungen zwischen zwei Knoten  $\langle i, j \rangle$  ist definiert durch eine Bandbreite  $bw_{i,j}$ , die die maximal verfügbare Menge an Informationen, die auf dem Kommunikationskanal übertragen werden kann, in der Einheit Bits/Sekunde festlegt.

Mithilfe dieses Netzwerkmodells lassen sich sowohl die Risiken eines Ausfalls eines Grid-Standorts als auch Risiken durch eine Überlastung des Netzwerks darstellen. Dies ermöglicht die Abbildung der Risiken der Ressourcen- und Verbindungsschicht, wie sie das Modell von Dai identifiziert [DaPZ07]. Ein Vergleich verschiedener realer Grid-Netzwerke findet sich in Kapitel 7. Dort wird aus drei ausgewählten Grid-Netzwerken ein realistisches Szenario für die Evaluation der Grid-Versicherung abgeleitet. Im folgenden Abschnitt wird die Implementierung der heuristischen Strategie der Softwareagenten vorgestellt, die in der CATNETS Grid-Ökonomie zum Einsatz kommt.

## 3.2 Implementierung der CATNETS Grid-Ökonomie

Die Implementierung der CATNETS Grid-Ökonomie realisiert den Handel von Basic-Services auf dem Servicemarkt und Ressourcenbündeln auf dem Ressourcenmarkt, die im vorangegangenen Abschnitt eingeführt wurden. Hierfür muss ein Prozess zum digitalen Gütertausch implementiert werden, der die in Abbildung 3.3 gezeigten Transaktionsphasen umfasst.

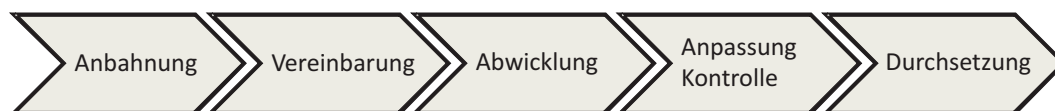


Abbildung 3.3: Die Transaktionsphasen einer elektronischen Markttransaktion [MüEK03, S. 297ff.]

In der Anbahnungsphase sucht der Konsument nach den Produkten, die er zur Erfüllung seines Grid-Service benötigt und von einem Anbieter zur Verfügung gestellt werden. Allgemein kann dies in Grid-Systemen durch zentral bzw. hierarchisch organisierte Suchverfahren oder durch dezentrale Suchverfahren erfolgen [Rein07]. Zentrale bzw. hierarchische Verfahren werden häufig mit Verzeichnissen realisiert, in denen Anbieter ihre Produktinformationen veröffentlichen und Konsumenten dieses Verzeichnis durchsuchen. Dies hat den Nachteil, dass die Informationen im Verzeichnis veraltet sein können, was zu Fehlern bei der Suche führen kann. Aus diesem Grund setzt die Implementierung der CATNETS Grid-Ökonomie das dezentrale Suchverfahren *Flooding* ein, das diesen Nachteil nicht besitzt und in Abschnitt 3.2.1 vorgestellt wird.

Die Verhandlungsphase dient den beteiligten Parteien dazu, einen Kontrakt auszuhandeln [MüEK03, S. 297]. Die Implementierung der CATNETS Grid-Ökonomie verwendet in dieser Phase eine bilaterale Verhandlung, die im Gegensatz zu Auktionen besser skaliert und somit auch in großen Grid-Systemen verwendet werden kann [BSEV08]. Die Details der Umsetzung präsentiert Abschnitt 3.2.2.

An die Verhandlungsphase schließt sich die Abwicklungsphase an, in der die Konsumenten die nachgefragten Grid-Services bezahlen und die Anbieter die Grid-Services bereitstellen und ausführen. Auf dem Servicemarkt stellt der Serviceanbieter seinen Basic-Service zur Verfügung und auf dem Ressourcenmarkt wird der Basic-Service durch den Ressourcenanbieter ausgeführt. Die Ausgestaltung der Abwicklungsphase wird in Abschnitt 3.2.3 vorgestellt. Ebenfalls in dieser Phase ist der Abschluss der Versicherung angesiedelt, dessen Implementierung in Kapitel 6 beschrieben wird.



In der nachgelagerten Anpassungs- und Durchsetzungsphase werden zum einen die Dienstauführung überwacht, um Abweichungen von der vereinbarten Qualität feststellen zu können und zum anderen die auftretenden Schäden durch die Versicherung kompensiert. Diese beiden Phasen sind nicht in der CATNETS Grid-Ökonomie implementiert und werden im Rahmen der Implementierung der Grid-Versicherung hinzugefügt. Details der Implementierung werden im Kapitel 6 präsentiert.

### 3.2.1 Flooding-basierte Dienstsuche

Die Aufgabe der Dienstsuche ist – wie eingangs dieses Kapitels erwähnt – das Auffinden von Serviceanbietern, die den nachgefragten Service anbieten. Da große, geografisch verteilte Grid-Systeme einige Charakteristika – wie unzuverlässige Services und periodische Verfügbarkeit – von aktuellen Peer-To-Peer (P2P)-Systemen bezüglich der Verfügbarkeit der Services zeigen [MKLN<sup>+</sup>02] [IaFo04], kommen für die Dienstsuche in realen Grid-Systemen Methoden von aktuellen Peer-To-Peer-Systemen zum Einsatz. Peer-To-Peer-Systeme stellen in diesem Zusammenhang eine Möglichkeit dar, Grid-Systeme zu implementieren [TaTr03].

Diese P2P-Suchverfahren können tausende Services verwalten und von tausenden Teilnehmern gemeinsam genutzt werden. Des Weiteren verfügen sie über einen dezentralen Aufbau, da nur durch einen dezentralen Aufbau die Gewährleistung gegeben ist, eine solch große Menge an Services zu verwalten. Für die Umsetzung skalierbarer Ansätze zur Dienstsuche gibt es eine Reihe von Ansätzen [Rein07], wobei hier näher auf das Flooding-Verfahren eingegangen wird [AdHu00].

Mit Flooding kontaktiert ein Knoten, der eine bestimmte Ressource oder einen bestimmten Service benötigt, alle Nachbarn im System, die wiederum ihre Nachbarn kontaktieren, bis ein Knoten, der den nachgefragten Service bereitstellt, erreicht wird [DiPi06]. Flooding führt die Suche ohne Verzeichnisse oder Wissen über die spezifische Topologie des Grid-Netzwerks aus und bietet einen attraktiven Ansatz für die Dienstsuche in dynamischen Utility-Grid-Systemen, die im Simulationsmodell der CATNETS Grid-Ökonomie nachgebaut werden.

Zur formalen Beschreibung des Algorithmus wird ein Grid-Netzwerk mit  $N$  Knoten angenommen, wobei jeder Knoten eine Anzahl an Services anbietet. Es gibt  $S$  verschiedene Services. Jeder Knoten kennt  $d$  andere Knoten des Netzwerks, die seine Nachbarn darstellen. Das System wird als gerichteter Graph  $G(K, V)$  modelliert, wobei jeder Knoten des Graphen einem Knoten des verteilten Systems entspricht und es eine Kante von  $A$  zu jedem von  $A$ s Nachbarn gibt. Da

jede Kante in  $G$  nicht unbedingt einer physikalischen Verbindung entspricht, wird der Graph  $G$  auch *Overlay-Netzwerk* genannt [ABKM01].

Des Weiteren wird ein Overlay-Netzwerk angenommen, in dem jeder Knoten  $d$  Nachbarn hat und vorherige Suchanfragen nicht speichert. Wenn ein Knoten  $A$  einen bestimmten Service  $x$  benötigt, flutet er immer das Netzwerk mit seiner Suchnachricht. Knoten  $A$  sendet eine Nachricht aus, die alle oder einen Teil seiner Nachbarn empfangen, verarbeiten und die Nachricht an ihre Nachbarn weiterleiten.

Um eine Überlastung des Netzwerk mit Suchanfragen zu vermeiden, wird eine Suche auf eine maximale Anzahl an Schritten  $t$  begrenzt. Gleichzeitig nimmt das Risiko einer fehlerhaften oder stark verzögerten Übertragung mit zunehmender Distanz zu, da mehr potenziell ausfallbehaftete Knoten durchlaufen werden müssen. Dieser Parameter ist ähnlich dem Time-To-Live (TTL)-Parameter, der in vielen Netzwerk-Protokollen verwendet wird. Im Detail beinhaltet die Suchanfrage ein Zählerfeld, das am Anfang mit  $t$  initialisiert wird. Jeder Zwischenknoten, der die Nachricht empfängt verringert diesen Zähler um 1. Solange der Wert nicht 0 ist, fährt der Knoten wie oben beschrieben fort; andernfalls hört der Knoten auf, seine Nachbarn zu kontaktieren. Ein Knoten sendet nur eine positive Antwort an den suchenden Knoten zurück, wenn er genügend freie Kapazität hat, den Service zu liefern.

Am Ende einer Suche hat der nachfragende Knoten  $A$  entweder die Kontaktinformationen für den gesuchten Service  $x$  oder keine Antwort erhalten, wenn gerade alle Services verwendet werden oder wegen technischer Fehler nicht verfügbar sind. Im letzteren Fall nimmt der Knoten  $A$  an, dass ein Knoten mit dem gewünschten Service nicht gefunden werden kann. Technische Fehler können der Ausfall eines Grid-Standorts sein oder die Überlastung des Netzwerks, was eine Übertragung der Nachricht nur stark verzögert zulässt und im vorliegenden Fall wie ein Ausfall einer Verbindung interpretiert wird.

Die implementierte Suchstrategie der CATNETS Grid-Ökonomie entspricht der oben vorgestellten Suchstrategie mit *Flooding*. Ein Knoten  $A$ , der nach Service  $x$  sucht, überprüft zuerst seine lokalen Ressourcen und kontaktiert anschließend alle seine Nachbarn bis zu einer maximalen Ausbreitung von  $t$  Schritten. Simulationen mit verschiedenen Simulationsszenarien, die im Rahmen des CATNETS Projektes durchgeführt wurden, haben ergeben, dass die Reichweite der Suche einen signifikanten Einfluss auf die Anzahl der erfolgreichen Verhandlungen im System hat [BCCC<sup>+</sup>07]. Der Einsatz des *Flooding*-Suchverfahrens liefert ein aktuelles Abbild der verfügbaren Ressourcen, jedoch werden hier exponentiell viele Nachrichten verschickt. Die Untersuchungen von Buss et al. [BCCC<sup>+</sup>07] haben ergeben, dass eine Suchtiefe von 2-3 Knoten

in Netzwerken mit einer hohen Ressourcendichte einen guten Wertbereich darstellt, während in Szenarien Ressourcenknappheit die Suchtiefe bis hin zur maximalen Suchtiefe eingestellt werden sollte. Ein weiterer Einflussparameter ist die verfügbare Netzwerkbandbreite, die eine Verzögerung auf den Transport der Nachrichten ausübt. Bei zu geringer Netzwerkbandbreite kann eine Erhöhung der Suchtiefe zu gegenläufigen Ergebnissen führen, da das Netzwerk überlastet ist und Nachrichten nur mit starker Verzögerung zustellen kann.

## 3.2.2 Verhandlung der Grid-Services

### 3.2.2.1 Allgemeine Prinzipien

In der Verhandlungsphase der Grid-Transaktion legen der Nachfrager und Anbieter den Preis für den Grid-Service fest. Es wird dabei angenommen, dass auf den Verhandlungsprozess technische Ausfälle wenig Einfluss haben, da technische Lösungen das Risiko eines Verhandlungsabbruchs durch Ausfälle vermindern können. In der Verhandlungsphase steht das Verhandlungsergebnis im Vordergrund. Kann ein Anbieter mehr als seine Grenzkosten erlösen und bezahlt ein Nachfrager weniger als sein Budget für einen Grid-Service, so steht diese Differenz für die Kosten des Risikomanagements zur Verfügung. Im konkreten Fall dieser Arbeit kann von dieser Differenz die Prämie der Versicherung bezahlt werden. Aus diesem Grund werden an dieser Stelle die Prinzipien des Verhandlungsverfahrens vorgestellt.

Die CATNETS Grid-Ökonomie setzt den dezentralen Marktmechanismus AVALANCHE ein [Eyma00], der an Grid-Netzwerke angepasst wurde. Dieser Ansatz realisiert einen Softwareagenten-basierten elektronischen Markt, in dem Agenten direkt miteinander kommunizieren und autonom Entscheidungen treffen, ohne den Zustand des Gesamtsystems zu kennen und in Betracht zu ziehen. Die Agenten handeln als ökonomische Subjekte miteinander, wenn sie einen möglichen Nutzen-gewinn durch die Verhandlung erreichen können.

Als theoretisches Fundament für Konzepte eines solchen dynamischen Marktprozesses dient Friedrich August von Hayeks Katallaxie-Paradigma [Haye45] [EySM03]. Die Katallaxie beschreibt einen Zustand der spontanen Ordnung, der dadurch zustande kommt, dass die Teilnehmer untereinander kommunizieren und ein gemeinsames Ziel erreichen, welches ein Einzelner nicht erreichen kann. Autonome Softwareagenten verhandeln dabei miteinander durch wechselseitigen Austausch von Angeboten [ZIRo96], treffen Entscheidungen unter unvollständiger Information und passen ihre Verhandlungsstrategien durch evolutionäre Algorithmen an [MiFo04] [Berg07].

### 3.2.2.2 Variablen der Strategie

Abbildung 3.4 zeigt den schematischen Aufbau der bilateralen Verhandlung. Durch iterativen Austausch von Angeboten nähern sie sich Nachfrager und Anbieter mit ihren Preisen an und erreichen schließlich einen Preis, der von beiden akzeptiert wird und den Kaufpreis des Service darstellt.

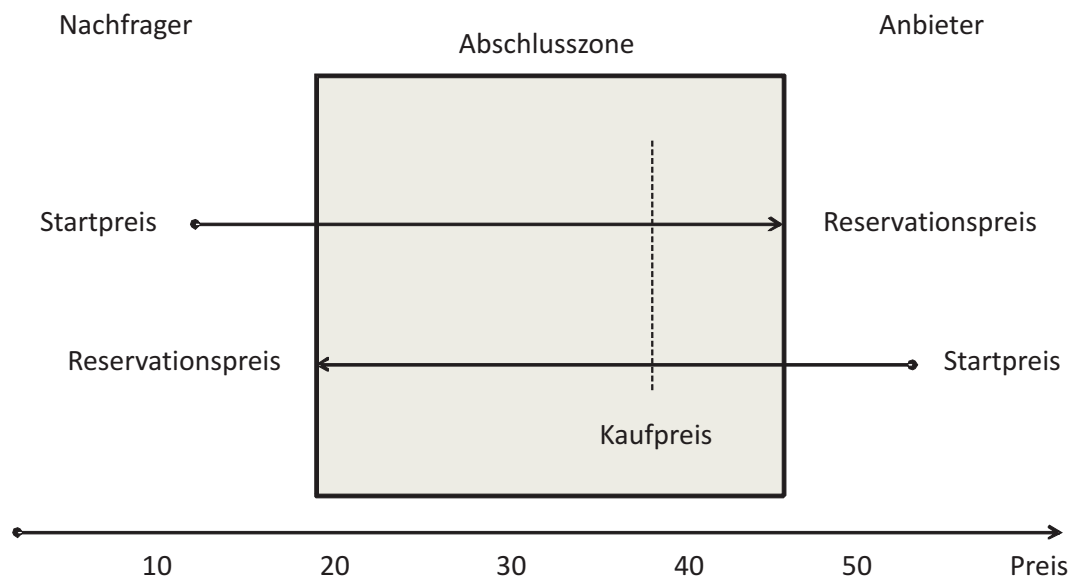


Abbildung 3.4: Aufbau der bilateralen Verhandlung: Nachfrager und Anbieter haben eine subjektive Vorstellung davon, in welchem Preisintervall ein Abschluss möglich ist. Die Grenze stellt bei beiden Verhandlungspartnern ihr individueller Reservationspreis dar.

Die heuristische Strategie, die beide Verhandlungspartner verfolgen, wird durch 5 Parameter beschrieben [Eyma00]. Sie beschreiben einen Prozess zum Austausch von Zugeständnissen zwischen den Verhandlungspartnern [Prui81, S. 91]. Die Parameter sind [Eyma00]:

- *Acquisitiveness*: Der Parameter *Acquisitiveness* definiert die Wahrscheinlichkeit, ein unilaterales Zugeständnis im folgenden Angebot zu machen. Die Entscheidung über ein Zugeständnis wird durch Ziehen einer stochastischen Probe gefällt. Ein Wert von 70% bedeutet, dass der Agent eine kompetitive Strategie verfolgt, die in nur 30% der Verhandlungsschritte zu einem Zugeständnis führt. Im Falle eines Zugeständnisses wird ein Käufer sein Angebot erhöhen und ein Verkäufer den Preis seines Angebots senken. Andernfalls wird der Preis unverändert an den Verhandlungspartner als neues Angebot zurückgeschickt.
- *Satisfaction*: Dieser Parameter repräsentiert die Zufriedenheit der Verhandlungspartner über den Fortschritt der Verhandlung. Ziel des Parameters ist es, einen endlichen Zeithorizont zu definieren.

izont festzulegen, dessen Erreichung einen Abbruch der Verhandlung zur Folge hat. Ein Zufriedenheitswert von 75% würde bedeuten, dass in 75% der Fälle der Verhandlungsprozess fortgeführt wird.

- *Price Step*: Die Höhe des Zugeständnisses wird durch den Parameter *Price Step* definiert. Dieser Wert legt den prozentualen Anteil am Verhandlungsintervall fest, der im Falle einer Zugeständnisentscheidung reduziert wird.
- *Price Next*: Dieser Parameter ist verantwortlich für die Anpassung des Verhandlungsintervalls nach einer erfolgreichen Verhandlung. Hiermit versuchen die Agenten, einen höheren Nutzen aus der Verhandlung zu ziehen, da ein gleich bleibendes Intervall keine Nutzensteigerung mit sich bringt. So versucht der Käufer, in der nächsten Verhandlung weniger zu bezahlen und der Verkäufer ist bemüht, seinen Service zu einem höheren Preis zu verkaufen.
- *Weight Memory*: *Weight Memory* legt fest, wie stark historische Preisinformationen in die aktuelle Marktsituation mit einfließen sollen. Damit lassen sich kurzfristige Preisänderungen im Markt wie eine ausfallbedingte Verknappung von Services abbilden. In diesem Fall sind historische Preisinformationen nicht mehr aktuell und haben durch eine Gewichtungsänderung weniger Einfluss auf die Einschätzung der Agenten.

Die Auswahl, mit deren Hilfe eine Entscheidung nach diesen Parametern vorgenommen wird, kann regelbasierte, spieltheoretische oder adaptive Ansätze verfolgen [Prei98]. In regelbasierten Ansätzen ist die Strategie im Agenten festgeschrieben. Ein Satz vordefinierter Regeln legt fest, wie der Agent in einer bestimmten Situation zu handeln hat [BGIP05]. Spieltheoretische Ansätze leiten die Verhandlungsstrategie aus spieltheoretischen Problemen ab [BiVu99]. Adaptive Ansätze schließlich passen ihre Strategie an, indem sie von Agenten in ihrem Umfeld lernen. In diesem Lernschritt setzen sie ihre eigene Leistung in Beziehung zu den Leistungen anderer Agenten. Aufgrund der dynamischen Natur der Grid-Systeme und der daraus resultierenden Änderung der Topologie bietet sich eine adaptive Strategie an. Die Agenten haben in diesem Ansatz die Möglichkeit, sich einer ändernden Umgebung anzupassen. Dies ist sowohl in der Verhandlung als auch bei der Bewertung der Risiken durch die Agenten von Vorteil (siehe Kapitel 7).

Die Implementierung der Strategie unter Verwendung eines adaptiven Ansatzes erfolgt durch die Verwendung einer Lernstrategie mit Rückkopplungseffekt aus der Gruppe der evolutionären Algorithmen. Die oben genannten Parameter werden hierfür zu einem Genotyp<sup>2</sup> zusammengefasst.

---

<sup>2</sup>Ein Genotyp ist ein aus der Biologie entliehener Begriff zur Beschreibung der genetischen Konstitution eines Einzelnen. Im Zusammenhang mit dem hier vorgestellten Verhandlungsprotokoll legt der Genotyp das Verhalten

Die Bewertung des Genotyps wird durch die Berechnung der Fitness eines Agenten vorgenommen. Diese wird aus dem Verhandlungsergebnis und seiner persönlichen Zielvorgabe abgeleitet, eine möglichst hohe Konsumenten- bzw. Produzentenrente zu erreichen [EySH07]. Die Fitness stellt einen wichtigen Wert hinsichtlich des Erfolgs der zu evaluierenden Risikomanagementstrategie dar. Im folgenden Abschnitt wird dieser Lernalgorithmus, dessen Ziel eine Verbesserung der Fitness eines Agenten ist, detailliert beschrieben.

### 3.2.2.3 Der Lernalgorithmus

Zwischen zwei Verhandlungen kann sich der Genotyp eines Agenten verändern. Ob und inwieweit das geschieht, entscheidet der Lernalgorithmus, der im Folgenden erklärt wird. Dieser Algorithmus gehört zur Kategorie co-evolutionären Algorithmen. Diese Algorithmen werden eingesetzt, um eine nichtlineare Dynamik abzubilden und komplexes Verhalten über mehrere Generationen von Genotypen hinweg zu erlernen. Im konkreten Fall der bilateralen Verhandlungsstrategie wird dieses Verhalten mit 5 Parametern ausgedrückt, für die Werte durch ein Trial-and-Error-Verfahren gesucht werden, um mit einer erfolgreichen Strategie gegenüber anderen Agenten erfolgreich zu sein [MiFo04]. Der Algorithmus wurde von Smith und Taylor [SmTa98] entworfen und implementiert, und für die CATNETS Grid-Ökonomie optimiert. Es besteht aus zwei Teilschritten: Versenden eigener Erfahrungswerte an andere Agenten und Lernen von empfangenen Erfahrungswerten auf Grundlagen der Lerninformationen.

**Verschicken eigener Lerninformationen** In der Ausgangssituation wird angenommen, dass der Agent  $n$  die Möglichkeit hat,  $s_1, \dots, s_d$  Services zu handeln. Weiterhin hat der Agent  $n$  seine  $i_n$ -te Verhandlung erfolgreich mit einem geschätzten Profit von  $\Pi_n^{i_n}(g)$  für den Service  $s \in \{s_1, \dots, s_d\}$  abgeschlossen. Einen Lernschritt hat der Agent  $n$  zum letzten Mal am Ende seiner  $j_k$ -ten Verhandlung durchgeführt.

Falls der Agent  $n$  schon mindestens  $k$  Verhandlungen für einen Service erfolgreich abgeschlossen hat, verschickt er Lerninformationen der Art

$$(G_n^{i_n}, F_n^{i_n})$$

an andere Agenten seines Typs, wobei  $G_n^{i_n}$  den Genotyp und  $F_n^{i_n}$  die Fitness des Agenten  $n$  in Verhandlung  $i$  darstellt. Dabei ist  $F_n^{i_n}$  seine aktualisierte *Fitness*, die in diesem Fall wie folgt berechnet wird:

---

des Softwareagenten während der Verhandlung fest.

(a) Für jeden Service  $s_j \in \{s_1, \dots, s_d\}$  wird zunächst der Wert  $\Pi(g_j)$  bestimmt:

Seien dazu

$$\Pi_1(g_j), \dots, \Pi_k(g_j)$$

die geschätzten Profite der letzten  $k$  erfolgreichen Verhandlungen von Agent  $n$  für den Service  $s_j$ . Dann ist

$$F_n^{i_n}(s_j) = \frac{1}{k} \left( \Pi_1(s_j) + \dots + \Pi_k(s_j) \right).$$

(b) Die aktualisierte Fitness  $F_n^{i_n}$  ist schließlich

$$F_n^{i_n} = \left( \Pi(s_1) + \dots + \Pi(s_d) \right).$$

**Lernen empfangener Lerninformationen** Zunächst werden die bei Agent  $n$  gespeicherten Lerninformationen mit der höchsten Fitness ausgewählt. Der dazugehörige Genotyp ist

$$G_f = (G_{f,1}, \dots, G_{f,5})^\tau = (a_f, s_f, t_f, b_f, w_f)^\tau.$$

Im ersten Schritt kommt es zu einem *Crossover*. Dabei entsteht ein neuer Genotyp  $\tilde{G}_n^{i_n+1}$ , der aus einer zufällig bestimmten Mischung von Genen der Genotypen  $G_n^{i_n}$  und  $G_f$  besteht. Danach folgt die *Mutation*: Aus dem Genotyp  $\tilde{G}_n^{i_n+1}$  wird der Genotyp  $G_n^{i_n+1}$ , indem einzelne Gene zufällig leicht verändert werden.

a) Crossover

Es sind  $C_1, \dots, C_5$  stochastisch unabhängige Zufallsvariablen mit folgender Binomialverteilung:

$$C_j = \begin{cases} 1 & \text{mit Wahrscheinlichkeit } c \\ 0 & \text{mit Wahrscheinlichkeit } c \end{cases} \quad \forall j \in \{1, \dots, 5\}$$

Dann gilt

$$\tilde{G}_{n,j}^{i_n+1} = (1 - C_j) \cdot G_{n,j}^{i_n} + C_j \cdot G_{f,j} \quad \forall j \in \{1, \dots, 5\}.$$

b) Mutation

Es sind  $M_1, \dots, M_5, X_1, \dots, X_5$  stochastisch unabhängige Zufallsvariablen mit folgenden Verteilungen:

$$M_j = \begin{cases} 1 & \text{mit Wahrscheinlichkeit } m \\ 0 & \text{mit Wahrscheinlichkeit } m \end{cases} \quad \forall j \in \{1, \dots, 5\}$$

$$X_j \sim \mathcal{N}(0, 1) \quad \forall j \in \{1, \dots, 5\}$$



Das heißt,  $X_j$  ist  $\forall j \in \{1, \dots, 5\}$  standardnormalverteilt.

Dann gilt

$$G_{n,j}^{i_n+1} = \max \left\{ 0; \min \left\{ \tilde{G}_{n,j}^{i_n+1} + M_j \cdot \left( \left( \frac{1}{k} X_j \right) \bmod(1) \right); 1 \right\} \right\}$$

$\forall j \in \{1, \dots, 5\}$ .

Mit dem Lernschritt endet die Verhandlungsphase. Nachfrager und Anbieter haben sich auf den Preis und die Dienstgüte eines SLA für einen Grid-Dienst geeinigt. Der Anbieter bereitet die Umsetzung des SLAs vor, und der Konsument startet die Nutzung des Ressourcen-Service durch der Ausführung des Basic-Service, was im folgenden Abschnitt präsentiert wird.

### 3.2.3 Ausführung eines Service auf Grid-Ressourcen

Die Ausführung eines Basic-Service auf Grid-Ressourcen wird in der CATNETS Grid-Ökonomie nur rudimentär unterstützt, da der Fokus auf der Dienstsuche und Verhandlung liegt. Für das Risikomanagement ist jedoch die Ausführungsphase essenziell wichtig, da die technischen Risiken, die in dieser Arbeit betrachtet werden, während der Ausführung eines Basic-Service auftreten können. Diese Risiken stehen im Mittelpunkt des Versicherungsmodells, das in den folgenden Kapiteln eingeführt wird.

Aus diesem Grund muss das bestehende Ausführungsmodell erweitert werden. Tabelle 3.1 zeigt einen Vergleich des Ausführungsmodells der CATNETS Grid-Ökonomie und das erweiterte Modell dieser Arbeit. Wird die Ausführung eines Basic-Service in der CATNETS Grid-Ökonomie immer als zuverlässig angenommen, so ist die Zuverlässigkeit der Ausführung im Modell dieser Arbeit abhängig von den Produkteigenschaften des Basic-Service. Welcher Basic-Service auf welcher Ressource ausgeführt wird, hängt dabei von der oben beschriebenen Strategie zur Dienstsuche und Verhandlung ab.

Zusätzlich wird das Ausführungsmodell um den Datentransfer vom Benutzer zum Ausführungs-ort des Basic-Service und um den Transport der Ergebnisse zum Benutzer zurück erweitert. Dies ermöglicht einen Einbezug des Datentransfers in das integrierte Risikomanagement, der einen großen Einfluss (siehe Kapitel 2) auf die Gesamtausführungszeit eines Grid-Service haben kann.



Tabelle 3.1: Gegenüberstellung des Ausführungsmodells von CATNETS und das erweiterte Modell für diese Arbeit

Kriterien	CATNETS	Modell dieser Arbeit
Eingehender Datentransfer	nicht unterstützt	individuell auf einen Basic-Service bezogen
Risikomodell	nicht vorhanden	abhängig von Datenmenge und verfügbaren Bandbreite
Ausführung des Service	eine global definierte Ausführungszeit für alle Basic-Services	individuelle Ausführungszeit für jeden Basic Service
Risikomodell	Fehler während der Serviceausführung nicht betrachtet	Ausfall eines Dienstes abhängig von den Produkteigenschaften des Basic-Service
Ausgehender Datentransfer	nicht unterstützt	individuell auf einen Basic-Service bezogen
Risikomodell	nicht vorhanden	abhängig von Datenmenge und verfügbaren Bandbreite

### 3.3 Zusammenfassung

Grid-Ökonomien sind ein wichtiger Schritt hin zu einer Kommerzialisierung von Grid-Ressourcen. Die CATNETS Grid-Ökonomie unterstützt durch die Aufspaltung in die Teilmärkte Service- und Ressourcenmarkt verschiedene Geschäftsmodelle des Utility-Computing wie Software-as-a-Service und Infrastructure-as-a-Service. Die dieser Arbeit zu Grunde liegende Implementierung unterstützt dabei alle Transaktionsphasen eines elektronischen Markts von der Anbahnungsphase bis hin zur Durchsetzungsphase. Durch die Erweiterung der Abwicklungsphase der CATNETS Grid-Ökonomie ist es möglich, Risiken bei der Ausführung von Basic-Services auf Grid-Ressourcen zu modellieren und dadurch ein breites Spektrum von Ausfällen, wie sie in Kapitel 2 identifiziert wurden, zu simulieren. Dies ermöglicht eine genauere Abbildung der Risiken realer Grid-Systeme.

Der Einsatz eines adaptiven Ansatzes zur Implementierung der Verhandlungsstrategie ermöglicht eine Simulation der Dynamik realer Grid-Systeme. Die Teilnehmer müssen sich kontinuierlich an eine veränderte Umgebung anpassen, die von verschiedenen, technischen Risiken beeinflusst wird.

Im folgenden Kapitel wird die Grid-Versicherung eingeführt und mit alternativen Risikobehandlungsstrategien verglichen, bevor auf die Realisierung und Integration der Grid-Versicherung in die CATNETS Grid-Ökonomie eingegangen wird.



## Kapitel 4

# Risikomanagement im Grid-Computing

*„Above all, it is vital to recognise that completely guaranteed behaviour is impossible and that there are inherent risks in relying on computer systems in critical environments. The unforeseen consequences are often the most disastrous.“ [Neum86]*

Risikomanagement im Grid-Computing ist neues Forschungsgebiet, das bisher noch sehr wenig Beachtung gefunden hat [DGPB<sup>+</sup>06]. Modelle bisheriger verteilter Systeme wie Cluster oder Client-Server-Anwendungen zeichnen sich durch ein weitaus niedrigeres Risiko aus, so dass diese nicht im Grid-Computing anwendbar sind. Im Vergleich zum herkömmlichen Risikomanagement, wie es beispielsweise in der Finanzwirtschaft eingesetzt wird, müssen Risiken im Grid-Computing vollständig automatisiert eingeschätzt sowie ein passendes Risikomanagementverfahren ausgewählt und angewandt werden [Voss07]. Allgemein bedeutet die Bewältigung von Risiken eine Reduzierung negativer Einflüsse der Unsicherheit, die zu Verlusten oder ungünstigen Einflüssen auf den Geschäftsbetrieb führen können. Ziel dieses Kapitels ist die Einführung von Prozessen und Methoden zum Umgang mit Grid-Risiken, wobei der Fokus auf dem Risikotransfer mithilfe von Versicherungen liegt.

Als Grundlage dienen Service-Level-Agreements, die in Kapitel 2 eingeführt wurden. Akzeptiert ein Ressourcenanbieter ein Service-Level-Agreement über die Ausführung eines Basic-Service auf seinen Ressourcen, garantiert er damit, die Ausführung des Basic-Service zur vereinbarten Dienstgüte zu erbringen. Dabei können die in Kapitel 2 besprochenen technischen Risiken auftreten. Diese Risiken können die Ausführung eines Basic-Service scheitern lassen und müssen vom Ressourcenkonsumenten behandelt werden, da Anbieter es häufig ablehnen, bei Ausfällen Vertragsstrafen zu bezahlen oder Ersatzleistungen zu liefern [HoKV06]. Des Weiteren sind vollständige Garantien in Computersystemen aufgrund ihrer Komplexität nahezu unmöglich [Neum86].

Zur Lösung dieses Problems bietet sich ein in das Grid-System integrierte Risikomanagement an, das Fehler auf ökonomischer Ebene behandeln kann [TeSh03] [HoKV06]. Die ökonomische Betrachtung der Risiken führt dazu, dass die Verfahren zur Behandlung der Risiken eine einfachere Umsetzung in eine Grid-Ökonomie erwarten lassen, als dies mit rein technischen Verfahren möglich wäre.

Kapitel 4.1 führt einen Risikomanagementzyklus ein, der einen strukturierten Prozess zur Einschätzung und Behandlung von Grid-Risiken beschreibt. Strategien zur Behandlung von Risiken und ihre Anwendung in Grid-Systemen beschreibt Abschnitt 4.2, wobei das Konzept des Risikotransfers mit Versicherungen von alternativen Strategien abgegrenzt wird. Abschnitt 4.3 fasst die Ergebnisse des Kapitels zusammen.

## 4.1 Der Grid-Risikomanagementzyklus

Das technische Risiko leitet sich aus der in einem SLA vereinbarten Dienstgüte ab. Nach dem Modell von Dai et al. [DaPZ07] sind die nachgefragte Ressourcenmenge und die Zeit, die diese Ressourcenmenge benötigt wird, Merkmale, die das technische Risiko entscheidend beeinflussen. Serviceanbieter können mithilfe von Erfahrungswerten eine Abschätzung über den Erfolg einer Serviceausführung dem SLA hinzufügen, die es Konsumenten ermöglicht, den Serviceanbieter auszuwählen, der die Anforderungen hinsichtlich des zu erwartenden Risikos erfüllt. Zusätzlich können diese Einschätzungen durch eigene Erfahrungswerte des Konsumenten verbessert werden.

Jedoch sind sich Grid-Benutzer bewusst, dass sie ein Risiko eingehen, da die Serviceanbieter keine Garantie einer fehlerfreien SLA Erfüllung abgeben können. Für die Beschreibung eines systematischen Umgangs mit Grid-Risiken wird im Folgenden ein Risikomanagementzyklus eingeführt und hinsichtlich der Anwendung auf Utility-Grid-Systeme untersucht. Dieser Risikomanagementzyklus wird von verschiedenen Best-Practice-Ansätzen abgeleitet, die in der Literatur beschrieben werden [GoLS03] [StGF02] [GrSh05] [WoRu03]. Der Risikomanagementzyklus baut auf dem Risikomanagement-Standard der Federation of European Risk Management Associations (FERMA) [Fede02] auf und verwendet die Risikobegriffe nach ISO/IEC Guide 73<sup>1</sup>. Abbildung 4.1 zeigt den an Grid-Systeme angepassten Risikomanagementzyklus. Im Vergleich zum Originalmodell wurden zwei Phasen zur Berichterstellung weggelassen, da in Grid-

---

<sup>1</sup>Der ISO/IEC Guide 73 aus dem Jahr 2002 legt einheitliche Begriffe aus dem Risikomanagement fest, wie sie in Standards verwendet werden sollen.

Systemen dieser Prozess automatisiert abläuft und keine schriftlichen Zwischenberichte erzeugt werden.

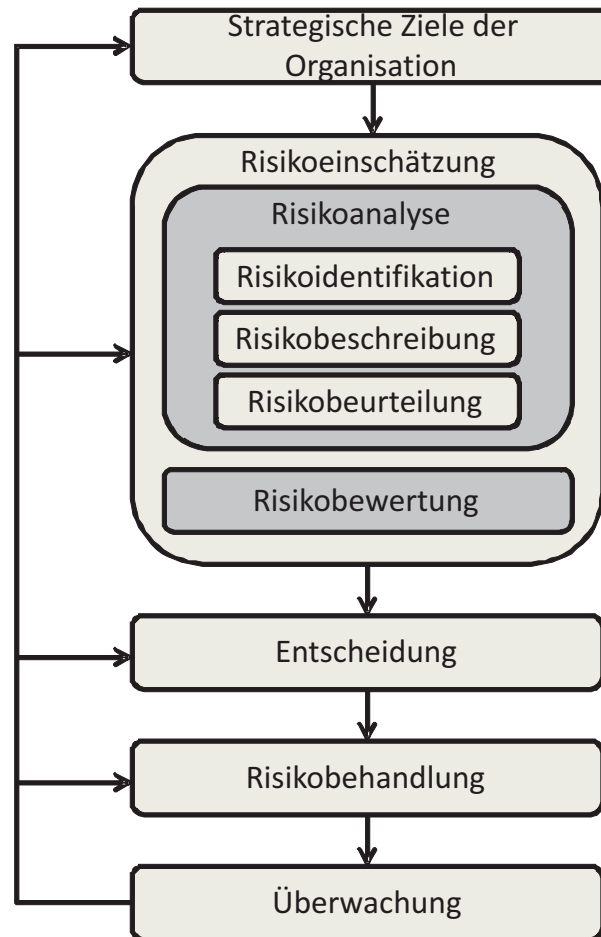


Abbildung 4.1: An Grid-Systeme angepasster Risikomanagementzyklus der FERMA [Fede02].

- **Strategische Ziele der Organisation:** Die Ziele des Softwareagenten leiten sich aus der Strategie der Organisation ab. Sowohl ein Servicekäufer als auch ein Serviceverkäufer können zur Erfüllung der Ziele der Organisation entscheiden, ob sie eine Kooperation eingehen oder ablehnen. Die Softwareagenten verfolgen dabei die in Kapitel 3 eingeführte Geschäftsstrategie, nur dann eine Kooperation einzugehen, wenn ein bestimmter Nutzen erreicht werden kann.

Für ein einfaches SLA ohne Vertragsstrafen sieht der Entscheidungsbaum in der CAT-NETS Grid-Ökonomie so aus, wie in Abbildung 4.2 gezeigt. Sobald eine Entscheidung über eine Kooperation getroffen wird, unternimmt der Grid-Konsument keine Anstrengung mehr, das Risiko des SLA zu verändern. Im Falle keiner Transaktion haben Konsument und

Anbieter einen Nutzen von 0. Bei Zustimmung zum SLA und fehlerhafter Ausführung hat der Konsument einen Nutzen von 0 und der Anbieter einen Nutzen von  $u(x)$ . Falls der Grid-Dienst erfolgreich ausgeführt wurde, haben sowohl Konsument als auch Anbieter einen Nutzen von  $u(x)$ .

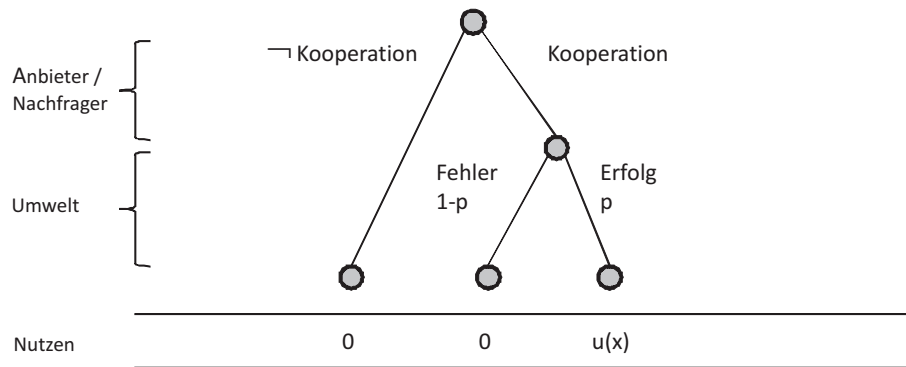


Abbildung 4.2: Entscheidungsbaum mit Auszahlungsmatrix (Nutzen) für Servicekonsument und Serviceanbieter bei Akzeptanz des Risikos

- **Risikoeinschätzung:** Die Risikoeinschätzung umfasst die Risikoanalyse und die Risikobewertung. Am Ende der Risikoeinschätzung steht eine Entscheidung über eine Akzeptanz des Risikos oder die Anwendung eines Verfahrens zur Behandlung des Risikos.
  - **Risikoanalyse:** Die Identifikation eines Risikos stellt den Ausgangspunkt dar. In diesem Schritt werden die Risiken durch Unsicherheit über die tatsächliche Dienstgüte identifiziert. Wie die historische Analyse in den Risikostudien und die Szenario-basierte Analyse der Ontologie in Kapitel 2 gezeigt haben, gehen signifikante Risiken vom Netzwerk, den Grid-Services selbst und vom Ressourcenmanagement aus. Die Beschreibung der Risiken wurde an einem exemplarischen Ausführungsablauf gezeigt und Messdaten über die Häufigkeit der auftretenden Fehler wurden vorgestellt. Dies ermöglicht eine Beurteilung über die Bedeutung der Fehler und des Risikos, das mit dem Fehler einhergeht.
  - **Risikobewertung:** Nach Abschluss des Risikoanalyseschritts werden die Risiken hinsichtlich der ökonomischen Auswirkungen untersucht. Die Grundlage hierfür bilden quantitative Kennzahlen, die sowohl die Varianz als Streuungsmaß der zu untersuchenden Größen als auch das Ausfallrisiko als Maß des Verlustes bei Nichterfüllung eines SLA mit einbeziehen [BrFr03, S. 9ff.] [Shah05]. Das Ausfallrisiko berücksichtigt dabei die Tatsache, dass die Teilnehmer der Grid-Ökonomie risikoavers sind. Dadurch kommt dem Ausfallrisiko gegenüber der Varianz eine größere Bedeutung

zu.

Die Risikobewertung erfolgt durch den Grid-Konsumenten. Dieser trifft abhängig von den Risikokennzahlen eine Entscheidung über die Behandlung der Risiken. Dabei wird die Annahme getroffen, dass die Kennzahlen wahrheitsgemäß vom Anbieter übermittelt werden. Das Zahlungsrisiko, das ein Grid-Anbieter trägt, wird in dieser Arbeit dadurch ausgeschlossen, dass der Ressourcenkonsument immer vorab die Ressourcen für einen festgelegten Zeitraum bezahlt.

- **Risikobehandlung:** Eine Risikobehandlung betrachtet den Prozess zur Auswahl und Durchführung von Maßnahmen, die zu einer Risikoänderung führen. Zu den 4 wichtigsten Verfahren zur Risikobehandlung gehören Risikoakzeptanz, Risikotransfer, Risikovermeidung, und Risikoreduktion [Wolk07] [MCTH<sup>+</sup>06]. Eine Abgrenzung der Verfahren und ein beispielhaftes Anwendungsszenario in Grid-Ökonomien sind Inhalte des folgenden Kapitels 4.2.
- **Überwachung:** Eng mit der Risikobehandlung verknüpft ist die Überwachung der vereinbarten Dienstgüte. Hier wird auf grundlegende Grid-Dienste zur Messung von zeit-, mengen-, und nutzenbasierte Werte zurückgegriffen und mit leistungsfähigen SLA-Frameworks verknüpft [SMSM<sup>+</sup>02] [KeLu03]. Häufig wird dem Grid-Konsumenten ein Portal zur Verfügung gestellt, das den Status der SLAs grafisch aufbereitet und die Funktion einer Berichterstellung übernimmt.

## 4.2 Verfahren zur Risikobehandlung in Grid-Ökonomien

Eine Übersicht über häufige Einsatzgebiete der 4 Risikobehandlungsstrategien gibt Abbildung 4.3. Weniger häufig auftretende Schäden mit einer geringen Schadenshöhe werden meist akzeptiert und durch Bildung von Rücklagen aufgefangen. Häufig auftretende Schäden mit geringer bis mittelgroßer Schadenshöhe sind Kandidaten für die Strategie der Risikoreduktion bzw. Risikoverminderung. Ein klassischer Risikotransfer adressiert seltene Schäden mit einer meist hohen Schadenshöhe [Bank04]. Treten diese Schäden zusätzlich noch häufig auf, bietet sich die Strategie der Risikovermeidung an.



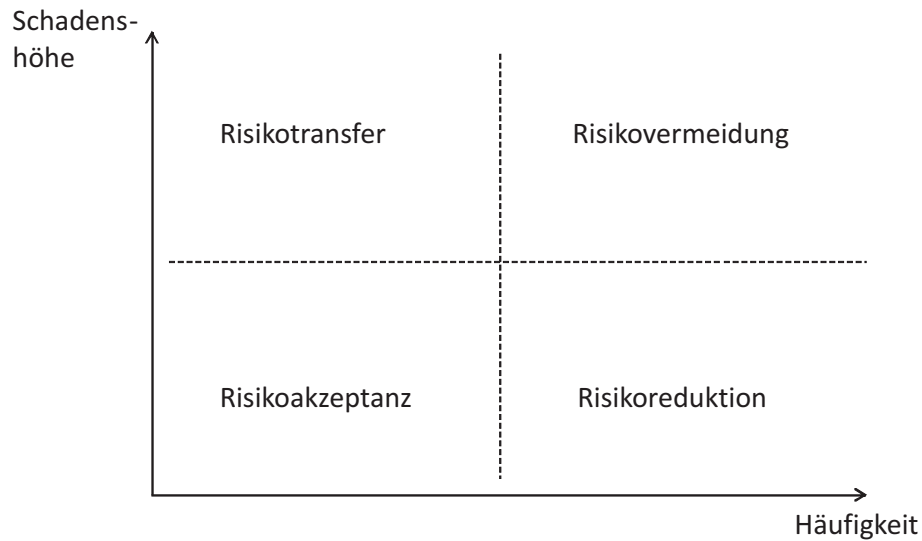


Abbildung 4.3: Einordnung der Verfahren zur Risikobehandlung in ihren charakteristischen Einsatzbereich [Bank04]

### 4.2.1 Risikoakzeptanz

Risikoakzeptanz ist eine Risikomanagementstrategie, die berücksichtigt, einen möglichen Verlust selbst tragen zu müssen [Wolk07]. Risikoakzeptanz bedeutet die Planung zukünftiger Risiken mit Blick auf die heutige und zukünftig notwendige Risikotragfähigkeit durch Eigenkapital der Organisation, in deren Auftrag die Softwareagenten handeln. Eine Erhöhung des Eigenkapitals bedeutet eine Erhöhung der Risikotragfähigkeit, um eventuell neue Risiken einzugehen oder bewusst eingegangene Risiken vorsorglich besser abzudecken (siehe Abbildung 4.3).

Risikoakzeptanz ist eine praktikable Strategie für kleine Risiken, deren Kosten, das Risiko zu versichern, über die Zeit größer sind als der mögliche Gesamtverlust. Alle Risiken, die nicht durch ein anderes Risikomanagementverfahren behandelt werden, werden implizit akzeptiert und sind somit diesem Risikomanagementverfahren zugeordnet. Dieses Vorgehen umfasst auch Risiken, die so groß oder katastrophal sind, dass sie nicht mehr versichert werden können oder Risiken, die beispielsweise über den Versicherungsschutz hinausgehen. Die Strategie der Risikoakzeptanz dient in der vorliegenden Arbeit als Vergleichsszenario zum Risikotransfer auf eine Versicherung.

Die in Kapitel 3 beschriebenen Softwareagenten der Grid-Ökonomie haben ein Budget und können auf eine bestimmte Menge des Eigenkapital der besitzenden Organisation zurückgreifen. Softwareagenten, deren dazugehörige Organisation mehr Eigenkapital hat, können dadurch mehr Risiken eingehen als Agenten mit Zugriff auf ein geringeres Eigenkapital und Geschäfte ab-

schließen, die mit mehr Risiko behaftet sind. Dies kann beispielsweise in einem Szenario von Vorteil sein, in dem es wenig zuverlässige Grid-Ressourcen gibt und die Entscheidung zwischen einer Dienstaufführung auf einer risikoreichen Ressource oder keiner Dienstaufführung getroffen werden muss und wegen der Bedeutung des Dienstes auf einer Ressource mit einem höheren Risiko ausgeführt wird.

Eine häufig anzutreffende Risikomanagementmethode zur Behandlung von akzeptierten Risiken ist die so genannte Selbstversicherung. Bei der Selbstversicherung wird ein errechneter Betrag beiseitegelegt, um einen möglichen zukünftigen Verlust zu kompensieren. Für die Kalkulation des Betrags wird auf statistische Verfahren der Versicherungswirtschaft zurückgegriffen, sodass der Betrag, der Ähnlichkeiten mit einer Versicherungsprämie aufweist, ausreichend ist, um zukünftige Verluste abzudecken. Selbstversicherung unterscheidet sich von einer Versicherung dadurch, dass die Prämien an einen unternehmenseigenen Versicherer bezahlt werden und nicht an einen herkömmlichen, externen Versicherer. Dabei gelten die gleichen Anforderungen bezüglich des Risikos wie für den Risikotransfer auf eine herkömmliche Versicherung.

Ist es in einem realen Grid-System nicht möglich, einen externen Versicherungsdienst zu betreiben, so lassen sich die Konzepte der Versicherung als Selbstversicherung unternehmensintern einsetzen. Dies ist insofern in Grid-Systemen möglich, da einzelne Grid-Standorte meist aus vielen einzelnen Rechnern bestehen. In diesem Szenario kann beispielsweise ein Ressourcenanbieter eine Selbstversicherung nutzen, um über dieses Konzept Ressourcen zur Absicherung aus einem Ressourcenpool zu beziehen. Dies würde dazu führen, dass jeder einzelne Standort des Grid-Systems eine eigene Risikovorsorge trifft und intern mit der Hilfe einer Selbstversicherung kleinere Schäden im Grid-System kompensiert.

Aus entscheidungstheoretischer Perspektive muss bei Risikoakzeptanz eine Entscheidung über die Akzeptanz des SLA und der damit verbundenen risikobehafteten Ausführung eines Grid-Service auf Grid-Ressourcen und einer Ablehnung des SLA getroffen werden. In einer Risikosituation ist diejenige Alternative zu wählen, die sich durch die beste Kombination aus negativen und positiven Konsequenzen auszeichnet und aus diesem Grund als „akzeptabel“ erscheint. Da die Teilnehmer der Grid-Ökonomie als risikoavers angenommen werden, werden sie das Angebot auswählen das am wenigsten Risiko aufweist.

### 4.2.2 Risikovermeidung

Die Risikovermeidungsstrategie hat das Ziel, keine Handlung durchzuführen, die mit einem Risiko behaftet ist [Wolk07]. Allgemein bietet Risikovermeidung die beste Strategie hinsichtlich der Elimination der Risiken. Allerdings wird durch Risikovermeidung auch ein möglicher Nutzengewinn verhindert, der mit der Strategie zur Risikoakzeptanz noch möglich gewesen wäre. Mit einer Risikovermeidungsstrategie würden keine Transaktion in einer Grid-Ökonomie mehr getätigt werden, da auf einem Grid-System von Natur aus immer das potenzielle Risiko besteht, dass ein Grid-Dienst nicht ausgeführt werden kann. Diese Strategie ist aus diesem Grund nicht in Utility-Grid-Systemen anwendbar, da ein Fremdbezug von risikobehafteten Grid-Dienstleistungen nicht mehr getätigt werden könnte.

### 4.2.3 Risikoreduktion

Risikoreduktion, auch Risikoverringern oder Risikoverminderung genannt, umfasst die Anwendung von verschiedenen Methoden, die die Schwere des Verlusts verringern. In diese Kategorie lassen sich die meisten und verwendeten Methoden zum Risikomanagement in Grid-Systemen einordnen. Auf Grundlage von SLAs lässt sich durch verbesserte Planung und durch Einsatz von Fehlertoleranzmechanismen eine Risikoreduktion erreichen [MCTH<sup>+</sup>06] [YuBu05]. Zusätzlich kommen Algorithmen zur Abschätzung der Ausführungszeit zum Einsatz [CJSN03], die auf Grundlage des Zustands des Grid-Systems eine Abschätzung vornehmen. Neben diesen rein technischen Möglichkeiten kommen in Grid-Systemen auch Verfahren aus der Finanzwirtschaft zum Einsatz, die Diversifikationseffekte zur Risikoreduktion ausnutzen.

Diversifikationseffekte werden erzielt, wenn sich die Risiken von mindestens zwei mit Risiken behaftete Positionen gegenseitig kompensieren und dadurch das Risiko der zusammengefassten Positionen kleiner ist als die Summe der Einzelrisiken [Wolk07]. Die theoretische Grundlage für die Diversifikation liefert die Portfoliotheorie [Mark52]. Zwei Ziele werden aus Grid-Sicht damit verfolgt:

- Verringerung der Wartezeit bis zur Jobausführung: Redundante Anfragen eines Grid-Jobs nach Grid-Ressourcen führt zu einer verringerten Wartezeit des Jobs in der Warteschlange [Casa07]. Dadurch können technische Risiken, wie sie nach dem Modell von Dai [DaPZ07] zwischen Grid-Benutzer und Ressourcenmanagement entstehen, reduziert werden.

- Erhöhung der Wahrscheinlichkeit einer erfolgreichen Ausführung: Wie anhand des Beispiels der BOINC-Plattform in Kapitel 2 gezeigt wurde, werden die Grid-Jobs mehrfach an verschiedene, geografisch voneinander getrennte Ressourcenanbieter vergeben, damit wird ein Portfolio von verschiedenen Anbietern gebildet. Es wird das Ergebnis eines Ressourcenanbieters ausgewählt und für richtig erklärt, das mithilfe einer Mehrheitsentscheidung ermittelt wurde.

Allgemein lassen sich Diversifikationseffekte durch die Standardisierung der Grid-Dienste und Ressourcen in Utility-Grid-Systemen erzielen [RoWe04]. Dies reduziert die Abhängigkeit von einem Grid-Anbieter und reduziert somit das Risiko, dass beispielsweise alle Jobs eines Workflows oder eines Geschäftsprozesses von einem Ausfall betroffen sind.

Weitere eingesetzte Methoden zur Risikoreduktion sind die Optimierung der Allokationsalgorithmen und die Analyse von kritischen Pfaden und Engpässen bei der Ausführung eines Workflows [MCTH<sup>+</sup>06]. Hinsichtlich der Allokationsalgorithmen können Risiken durch mehr bzw. effizientere Zuweisung von Ressourcen des Grid-Systems und die Unterstützung von Co-Allokation der Ressourcen von verschiedenen Ressourcenanbietern reduziert werden [CzFK99]. Engpässe im Hinblick auf verfügbare Grid-Dienste können durch dynamisches Bereitstellen von mehr Serviceinstanzen behoben werden. Damit können die Risiken signifikant reduziert werden.

Nicht nur bei der Ressourcenoptimierung vor einer Ausführung eines Grid-Service lassen sich Risiken reduzieren, sondern auch während der Ausführung. SLA-Management-Systeme überwachen dabei kontinuierlich die Ausführung eines Grid-Service und ergreifen bei einer erwarteten SLA-Verletzung Gegenmaßnahmen wie beispielsweise die Bereitstellung weiterer Ressourcen [DDKK<sup>+</sup>04] [PaDD05].

#### 4.2.4 Risikotransfer

Beim Risikotransfer bleiben die Risiken bestehen und es wird durch Abschluss eines Vertrages lediglich ein potenzieller zukünftiger Verlust transferiert. Für den Transfer muss eine Gegenleistung in der Regel in Form einer Prämie gezahlt werden. Der Abschluss von Versicherungen ist das wichtigste und am häufigsten angewendete Steuerungsinstrument zum Risikotransfer [Wolk07]. Der Fokus der Arbeit liegt auf der Untersuchung einer Versicherung für Grid-Transaktionen. Weitere Verfahren des Risikotransfers mit Optionen oder Futures werden in dieser Arbeit nicht betrachtet, da diese in einem Szenario einer sofortigen Bezahlung und Nutzung

der Grid-Services und der Grid-Ressourcen nicht angewendet werden können<sup>2</sup>.

Grundlage für den Risikotransfer stellt wiederum ein SLA zwischen Konsument und Anbieter in einer Grid-Ökonomie dar. Aufgrund der Komplexität des Grid-Systems und des Einflusses verschiedener Organisationen auf die Dienstleistungserbringung ist es nicht möglich, alle möglichen Szenarien eines Ausfalls in einem SLA festzulegen. In diesem Fall bieten sich Versicherungen an, die nicht abgedeckte Schäden eines SLA-Vertrags an eine Versicherung zu transferieren.

Versicherungen werden bereits im E-Commerce zur Behandlung von Sicherheitsrisiken oder von Risiken durch kriminelle Aktivitäten, die durch das Internet entstehen, eingesetzt [GoLS03] [Grze02a]. Sie bieten ein breites Spektrum an Einsatzmöglichkeiten bei der Behandlung von Risiken an, wie Abbildung 4.4 illustriert:

- **Versicherung, Selbstversicherung:** Der Bereich der Versicherung, der in dieser Arbeit betrachtet wird, umfasst Risiken, die häufig auftreten und nur einen kleinen Schaden verursachen. Hier stehen der Einsatz einer Selbstversicherung zum internen und einer Versicherung zum externen Risikotransfer zur Verfügung. Selbstversicherung als Risikoakzeptanzstrategie wird im Folgenden nicht weiter verfolgt, der Fokus liegt auf dem externen Transfer des Risikos. Die Charakteristik der Schäden einer Grid-Transaktion ist ein häufiger Schadenseintritt bei gleichzeitig geringer Verlusthöhe.
- **Begrenzte Risikoversicherung:** Die begrenzte Risikoversicherung deckt Schäden nur bis zu einem vorher festgelegten Maximalwert ab. Ist die Schadenshäufigkeit ähnlich hoch wie im ersten Fall bei gleichzeitig größerer Schadenshöhe, so bieten Versicherungen keine Produkte mehr zur vollständigen Schadensübernahme an, sondern nur bis zu einer Obergrenze.
- **Konventionelle Versicherung:** Konventionelle Versicherungen kommen im Bereich seltener Schäden mit hoher Schadenshöhe zum Einsatz. Diese finden sich im Bereich von Infrastrukturversicherungen, die eine Absicherung bei Hardware- und Softwareschäden oder Datenlöschungen anbieten [Grze02a].
- **Risikoverbriefung:** Die Risikoverbriefung wird zur Absicherung von katastrophalen Schäden mithilfe des Kapitalmarkts durchgeführt. Da die zu erwartende Schadenssumme, die durch die Ausführung eines Grid-Service auftritt, im Bereich einiger weniger Euro liegt, ist die Verbriefung von technischen Risiken in Grid-Systemen nicht relevant.

Risiken, die über die Risikoverbriefung hinausgehen, sind durch Versicherungen nicht mehr be-

---

<sup>2</sup>Weitere Informationen zum Einsatz von Optionen und Futures zur Allokation von Rechenressourcen finden sich bei Meinel [Mein08], Sutherland [Suth68] und Kenyon et al. [KeCh05].

handelbar, sodass andere Risikostrategien wie eine Risikovermeidung angewandt werden müssen.

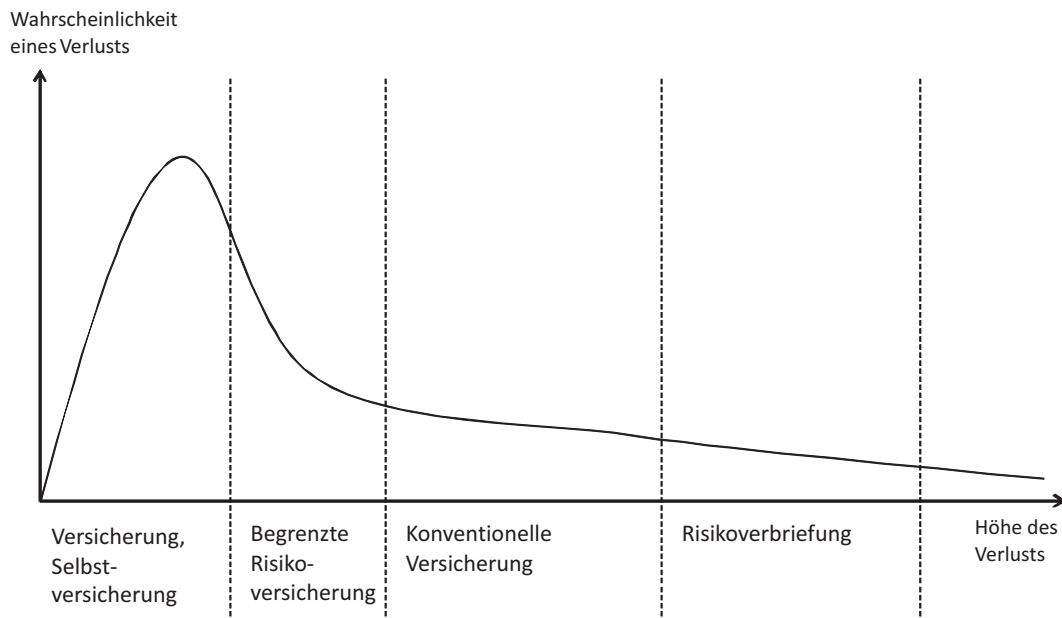


Abbildung 4.4: Einsatzszenarien verschiedener Versicherungsarten zum Risikotransfer [GrSh05]

Anhand des Ressourcenmarkts der CATNETS Grid-Ökonomie werden möglich Szenarien für den Einsatz einer Versicherung abgeleitet:

- Szenario 1: Ein Ressourcenkonsument schließt einen Vertrag mit einer Versicherung über die Ausführung eines Jobs oder eines Teilworkflows ab. Die Versicherung ist in diesem Fall ein Ressourcenanbieter, der bei Feststellung einer Verletzung des SLA die Ausführung übernimmt.
- Szenario 2: Ein Ressourcenkonsument schließt einen Vertrag mit einer Versicherung ab. Die Kompensationsleistung erfolgt jedoch nicht wie im ersten Szenario durch Lieferung von Ressourcen, sondern durch monetäre Kompensation, was dem in der realen Versicherungswirtschaft vorherrschenden Szenario entspricht.
- Szenario 3: Ein Ressourcenanbieter schließt einen Untervertrag mit einem anderen Ressourcenanbieter über die Ausführung eines Jobs oder eines Workflows. Wie im ersten Szenario erfolgt die Kompensation durch die Lieferung von Ressourcen.
- Szenario 4: Ein Ressourcenanbieter schließt einen Vertrag mit einer Versicherung über die monetäre Kompensation von entgangenen Einnahmen durch Hard- und Softwareausfälle.

Dieses Szenario entspricht einer Infrastrukturversicherung, wie sie aktuell von Versicherungsunternehmen angeboten werden [MüEK03] [Grze02a]. Der Fokus liegt nicht auf der einzelnen Grid-Transaktion, sondern auf der Grid-Infrastruktur im Allgemeinen.

Im Folgenden wird der Einsatz einer Versicherung aus Sicht eines Ressourcenkonsumenten (Szenario 2) näher erläutert. Dieses Szenario steht auch im Mittelpunkt der folgenden Ausführungen der Arbeit. Szenario 2 ermöglicht eine einfache Umsetzung im Vergleich mit Szenario 1, da in diesem Szenario die Versicherung keine Allokation von Ressourcen zur Kompensation durchführen muss. Die Konzentration auf den Ressourcenkonsumenten ist darin begründet, dass bei aktuellen Services von beispielsweise Amazon<sup>3</sup> der Kunde das Risiko der Nichtausführung eines Jobs zu tragen hat. Aus diesem Grund werden die beiden Szenarien aus Anbietersicht nicht weiter verfolgt. Das Szenario einer Grid-Versicherung mit dem Ressourcenkonsumenten als Versicherungsnehmer lässt sich jedoch auch auf das Szenario mit dem Ressourcenanbieter als Versicherungsnehmer übertragen.

Das Verhalten der Marktteilnehmer mit einer Versicherungsstrategie zur Risikobehandlung unterscheidet sich grundlegend von der Risikoakzeptanzstrategie, in der die Teilnehmer eher ein Angebot ablehnen würden als ein zu hohes Risiko einzugehen. Abbildung 4.5 zeigt den erweiterten Entscheidungsbaum unter Verwendung einer Versicherung zum Risikotransfer für das zweite der oben beschriebenen Szenarien. Die Versicherung wird von einem Konsumenten abgeschlossen, der Risiken eines SLA transferieren möchte. Es wird weiter davon ausgegangen, dass keine Vertragsstrafen definiert sind und die Versicherungsleistung als monetäre Kompensation erbracht wird. Die Wahrscheinlichkeit  $p$  entspricht dem Ausfallrisiko durch die Grid-Infrastruktur;  $q$  repräsentiert das Ausfallrisiko, das von den Grid-Diensten ausgeht. Beide haben zur Folge, dass ein SLA nicht eingehalten werden kann.

Die Konsequenzen aus der Nichterfüllung eines SLA durch den Ressourcenanbieter sind ohne Versicherung ein Nutzen von 0 und mit Versicherung ein um die Kosten der Versicherungsprämie  $p1$  reduzierter Nutzen  $p2$ , der im Fall einer vollständigen Absicherung  $u(x)$  entspricht. Ist die Ausführung erfolgreich gewesen, so ist der Nutzen ohne Versicherung  $u(x)$  und mit Versicherung  $u(x) - p1$ . Der Konsument kann abhängig von der Höhe der Prämie und seiner Risikoneigung entscheiden, ob er eine Versicherung abschließt oder nicht. Schematisch zeigt Abbildung 4.6 das Ziel, welches der risikoaverse Ressourcenkonsument bzw. seine Implementierung als Softwareagent zu erreichen versucht. Vor Abschluss einer Versicherung rechnet der Agent mit einem erwarteten Nutzen, der dem Nutzen einer erfolgreichen Ausführung seines Grid-Service ent-

---

<sup>3</sup><http://aws.amazon.com/>

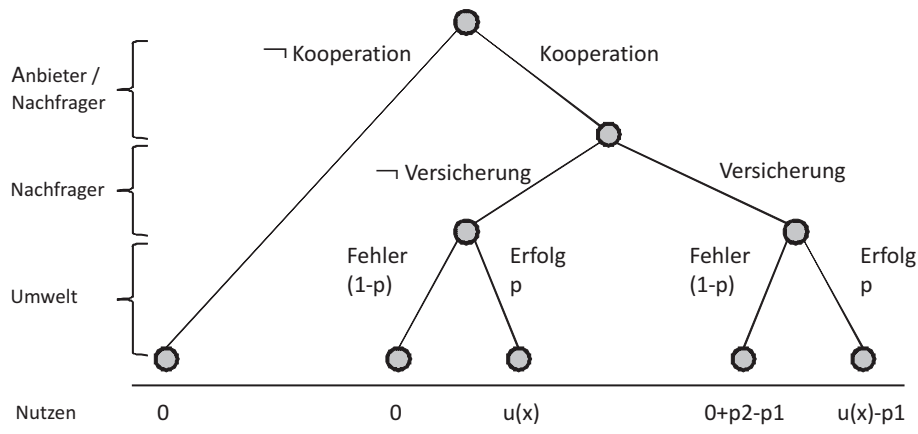


Abbildung 4.5: Entscheidungsbaum bei Verwendung einer Versicherung zum Risikotransfer aus Sicht eines Ressourcenkonsumenten

spricht. Durch Abschluss einer Versicherungsprämie kann er seinen Erwartungsnutzen erhöhen und in Richtung seiner risikoaversen Nutzenkurve anheben. Solange die Prämie kleiner als der Nutzenverlust ist, wird der Ressourcenkonsument eine Versicherung abschließen. Eine zusätzliche Einschränkung kommt in der hier betrachteten Grid-Ökonomie dazu; der Agent kann nur so viel Geld für eine Prämie ausgeben, wie es sein Budget gestattet. Obergrenzen können sein Budget oder der mögliche Nutzenverlust sein, je nachdem, welche Grenze zuerst erreicht wird.

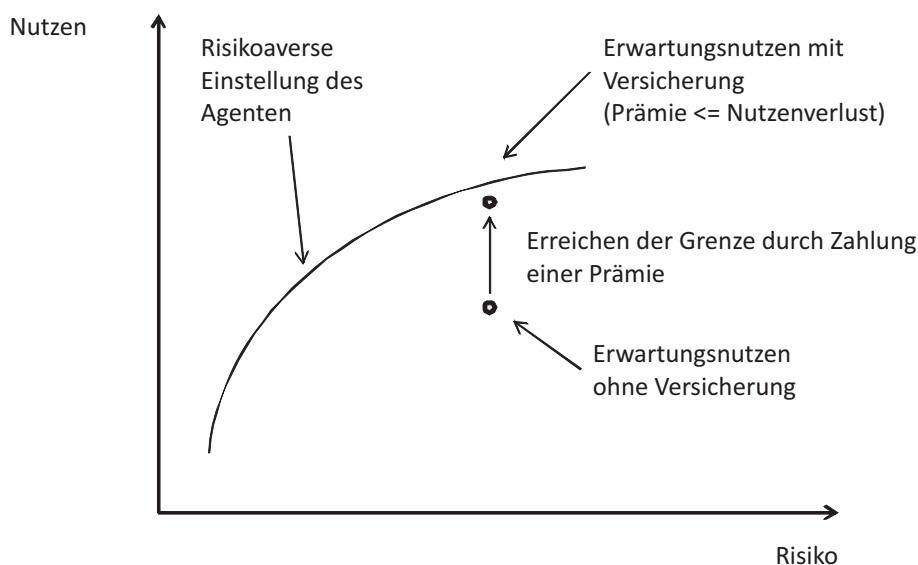


Abbildung 4.6: Risikoaverse Einstellung der Ressourcenkonsumenten in Anlehnung an [Bank04]



## 4.3 Zusammenfassung

Grundlage des Risikomanagements in Grid-Systemen stellt die zwischen Konsument und Anbieter vereinbarte Dienstgüte dar. Zur besseren Bewertung kann das SLA um Schätzungen des technischen Risikos seitens des Anbieters bzw. des Konsumenten erweitert werden. Der in diesem Kapitel eingeführte Risikomanagementzyklus beschreibt – ausgehend von den strategischen Zielen der Organisation, in deren Auftrag die Agenten handeln – verschiedene Verfahren zur Behandlung der auftretenden technischen Grid-Risiken. Wichtig für Grid-Systeme ist hierbei die Möglichkeit der automatisierten Bewertung und Behandlung der Risiken.

Die Versicherung als wichtiges Verfahren zum Risikotransfer wurde anhand verschiedener Szenarien untersucht und schließlich kam man zu einem Szenario, in dem der Ressourcenkonsument eine Versicherung abschließt, um das Risiko einer Verletzung der zugesicherten Dienstgüte zu transferieren. Die Kompensation des Schadens erfolgt monetär. Im Fokus des folgenden Kapitels steht die Modellierung eines Versicherungsmodells für das ausgewählte Szenario und dessen Umsetzung in die CATNETS Grid-Ökonomie.

## Kapitel 5

# Realisierung einer Versicherung in einer Grid-Ökonomie

*„Technology virtualises the value chain [of insurances] and lowers barriers to entry – The rising tide of technology will enable an increasing number of niche service providers from inside and outside of the traditional value chain. The 15-year time frame [as from 2005] will inevitably produce a greater number of partially and even totally virtual insurance companies to meet the needs of consumers and businesses.“ [Bisk06]*

In den vorangegangenen Kapiteln wurden technische Risiken beschrieben und die Grid-Versicherung eingeführt. Dieses Kapitel stellt die Realisierung der Grid-Versicherung in den Fokus. Dieses Vorgehen umfasst sowohl die Realisierung der einzelnen Versicherungsbestandteile als auch die Integration in das Ressourcenmanagement der CATNETS Grid-Ökonomie.

Die Grid-Versicherung ist der erste Risikotransfermechanismus, der in Grid-Ökonomien zum Einsatz kommt. Dabei werden die Konzepte der Versicherungswirtschaft unter Einbezug der Anforderungen aus den einführenden Kapiteln 2, 3 und 4 an Grid-Systeme angepasst. Die Grid-Versicherung bietet einen voll automatisierten Risikotransfer als Teil des Ressourcenmanagements an und realisiert eine vollständig virtualisierte Versicherung zur Steuerung der technischen Risiken in Grid-Ökonomien.

Funktionen und Aufgaben von Versicherungen in Grid-Ökonomien beschreibt Kapitel 5.1. Bekannte Aufgaben von realen Versicherungen werden hinsichtlich ihrer Übertragbarkeit auf Grid-System analysiert. Der folgende Abschnitt 5.2 gibt einen Überblick über die 4 wichtigsten Bestandteile einer Versicherung.

Eine wichtige Voraussetzung für die effiziente Anwendung von Versicherungen stellt Kapitel

5.3 vor. In diesem Kapitel wird die grundlegende Versicherbarkeit der technischen Grid-Risiken untersucht. Mithilfe des entscheidungstheoretischen Ansatzes von Karten [Kart72] werden die Bedingungen festgelegt, die für einen effizienten Einsatz einer Versicherung gelten müssen.

Hauptteil des Kapitels stellt Abschnitt 5.4 dar. Hier werden die einzelnen Bestandteile der Grid-Versicherung unter Zuhilfenahme von Verfahren der Schadens- und Haftpflichtversicherungen modelliert. Ausgehend von der Versicherungsperiode werden Szenarien zur Erstattung im Schadensfall betrachtet und ein Verfahren zur Prämienberechnung eingeführt, das zur Prognose der technischen Grid-Risiken zum Einsatz kommt. Von besonderer Bedeutung ist das versicherungstechnische Risiko einer Versicherung, das über die Ruinwahrscheinlichkeit der Versicherung entscheidet und das ebenfalls in diesem Abschnitt betrachtet wird.

Szenarien zur Einbettung der Grid-Versicherung in die CATNETS Grid-Ökonomie stellt Kapitel 5.5 vor. Verschiedene Möglichkeiten der Umsetzung werden diskutiert und es wird ein Szenario ausgewählt, das für die Simulationsstudie implementiert wird. Abschnitt 5.6 fasst die Ergebnisse des Kapitels zusammen.

## 5.1 Funktionen und Aufgaben von Versicherungen

Versicherungen besitzen eine Reihe mikro- und makroökonomischer Funktionen und Aufgaben, die von der Verbesserung der Risikoallokation bis zur Entlastung des Staates reichen [ZwEi07, S. 15ff.]. Nicht alle dieser Aufgaben und Funktionen sind in einer Grid-Ökonomie umsetzbar und von Bedeutung. Aus diesem Grund werden nur die Wirkungsweisen ausgewählt, die sowohl in einer Grid-Ökonomie umgesetzt als auch in der Simulationsstudie ausgewertet werden können. Der Beitrag einer Versicherung fördert dabei die wirtschaftliche Effizienz in der Grid-Ökonomie auf verschiedenen Wegen [ZwEi07, S. 15ff.]:

- *Verbesserung der Risikoallokation*: Die effiziente Allokation von Risiken verringert die Transaktionskosten und reduziert die Schäden durch prompte und automatisierte Schadensregulierung. Beides steigert die Effizienz der Dienstauführung auf Grid-Ressourcen.
- *Vermögenssicherung*: Der Schutz bestehenden Vermögens ist eine bedeutende Funktion der Versicherung, weil damit den Wirtschaftssubjekten – Ressourcenkonsument und -anbieter – eine verbesserte Grundlage für ihre Planung bereitgestellt wird. Dies ermöglicht dem Ressourcenkonsument eine gesicherte Grundlage, risikobehaftete, aber gewinnträchtige Geschäfte in Angriff zu nehmen. Ein Konsument auf dem Grid-Markt hat durch die Versi-

cherung die Möglichkeit, den maximalen Schaden seiner Transaktion zu versichern. Dies ermöglicht es ihm risikobehaftete Ressourcenanbieter auszuwählen und bei fehlerhafter Dienstauführung seinen entstandenen Schaden erstattet zu bekommen.

- *Kontrolle des Verhaltens der Ressourcenanbieter*: Eine Versicherung dient auch der Kontrolle des Verhaltens der Ressourcenanbieter. Denn üblicherweise werden Versicherungen risikogerecht kalkuliert, d. h., die Prämie ist umso höher, je größer das Risiko (Schadensbetrag und/oder Schadenswahrscheinlichkeit) ist. Dieses Verursacherprinzip schafft Anreize für die Ressourcenanbieter, die von ihnen ausgehenden Risiken zu reduzieren. Durch die Kalkulation von risikogerechten Prämien werden Transaktionen mit unzuverlässigen Ressourcenanbietern auf Grid-Märkten für den Ressourcenkonsumenten durch die hohen Versicherungsprämien teurer. Als Konsequenz daraus wird ein risikoreicher Transaktionspartner weniger häufig bevorzugt, was zu einer schlechteren Auslastung seiner Ressourcen führt.

Ziel des Grid-Versicherungsmodells ist es, diese Funktionen und Aufgaben unter Einbeziehung der gridspezifischen Eigenschaften umzusetzen. Bevor auf das Versicherungsmodell selbst eingegangen wird, werden die Bestandteile einer Versicherung im folgenden Abschnitt dargelegt.

## 5.2 Bestandteile einer Versicherung

Eine Versicherung besteht aus 4 Hauptbestandteilen [ZwEi07], die eine eigenständige Aufgabe realisieren und deren Ergebnis die jeweils folgende Komponente als Eingangsdaten benötigt. Die in Abbildung 5.1 gezeigten 4 Hauptbestandteile einer Versicherung sind eingebunden in einen Prozess zur Abwicklung eines Schadens. Die technische Kontrolle überwacht den versicherten Service und aktualisiert bei eintretenden Schäden die Schadensstatistik, die wiederum für die Neuberechnung der Versicherungsprämien verwendet wird. In einem letzten Schritt dieses Prozesses wird der Schaden reguliert. Dieser Prozess wird im Folgenden detailliert erläutert und auf Grid-Systeme übertragen:

- *Technische Kontrolle*: Die technische Kontrolle umfasst Verfahren zur Überwachung und Messung von Merkmalen des versicherten Service. Sie muss nicht durch die Versicherung selbst erbracht werden; Eine Überwachung durch einen externen Dienstleister ist ebenfalls möglich [ZwEi07]. Eine objektive, technische Kontrolle ist für die Versicherung wichtig, um einen Schaden beurteilen zu können.

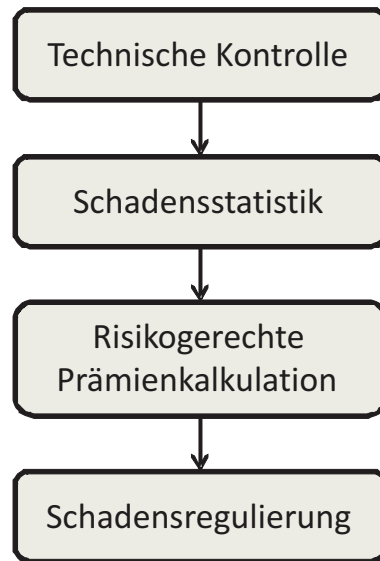


Abbildung 5.1: Bestandteile einer Versicherung [ZwEi07]

Übertragen auf Grid-Systeme bedeutet dies, dass die Parameter der vereinbarten Dienstgüte überwacht werden müssen, um den Eintritt eines Schadens feststellen zu können. Ist die Dienstgüte durch ein standardisiertes SLA-Dokument beschrieben, bieten sich bestehende Systeme zum Monitoring von Service-Level-Agreements an, wie sie in Kapitel 2 kurz vorgestellt wurden. Ein solches System zur Messung der versicherten Dienstgüte und Erkennung von Schäden wird als Voraussetzung für die Arbeit angenommen. Die Feststellung und Meldung eines Schadens an die Versicherung erfolgt in der Simulationsstudie dieser Arbeit durch den Konsumenten. Der Konsument bezieht für die Ermittlung eines Schadens auch Messdaten des Anbieters mit ein, der alle ihn betreffenden Parameter misst und ihm zur Verfügung stellt.

- **Schadensstatistik:** Die Schadensstatistik ist der zentrale Datenspeicher für die Bewertung der Risiken durch die Versicherung. Hier werden alle schadensrelevanten Werte gespeichert und aggregiert. Die Aktualisierung der Schadensstatistik erfolgt durch die Meldung eines Schadens im Ergebnis der technischen Kontrolle. Welche Daten in der Schadensstatistik mindestens gespeichert werden müssen, richtet sich nach dem Verfahren zur Prämienberechnung, das auf Grundlage der Schadensstatistik die Versicherungsprämien aktualisiert. Details hierzu werden in Kapitel 5.4 vorgestellt.
- **Risikogerechte Prämienkalkulation:** Die Kalkulation einer risikogerechten bzw. fairen Prämie stellt die zentrale Aufgabe der Versicherung dar. Diese Aufgabe entscheidet über den Erfolg der Versicherung hinsichtlich der Einschätzung der zu erwarteten Schäden. Die

Realisierung der Prämienkalkulation für technische Grid-Risiken wird in Kapitel 5.4 eingeführt.

Wichtig für die Bewertung des Erfolgs einer Versicherung sind der Prämieinnahmeprozess, der die Summe der Prämieinnahmen über die Zeit beschreibt, und der Schadensprozess, der die Ausgaben der Versicherung zur Kompensation der Schäden repräsentiert [ZwEi07, S. 231]. Idealerweise nimmt der Prämieinnahmeprozess der Versicherung mehr ein, als Schäden im Schadensprozess kompensiert werden müssen. Übersteigen die Schäden die Prämieinnahmen, wird die Versicherung insolvent und sie kann die Schäden der Versicherungsnehmer nicht mehr kompensieren. Aus diesem Grund muss die Versicherung Maßnahmen ergreifen, dies zu verhindern. Diese Maßnahmen werden ebenfalls in Kapitel 5.4 vorgestellt.

- **Schadensregulierung:** Die Schadensregulierung umfasst die Ausbezahlung der Versicherungssumme nach Schadensmeldung des Versicherungsnehmers. Die Schadensmeldung wird in herkömmlichen Versicherungen meist manuell von einem Sachbearbeiter aufgenommen, der die Schadensregulierung veranlasst.

Im Gegensatz hierzu erfolgt die Kompensation des Schadens in Grid-Systemen automatisiert. Eine weitere Besonderheit in Grid-Systemen stellt die Möglichkeit dar, den Schaden über Ersatzleistung der versicherten Grid-Ressourcen zu kompensieren. Diese Besonderheiten der Schadensregulierung in Grid-Systemen werden in Kapitel 5.4 diskutiert.

Bevor auf die Umsetzung der Versicherungsbestandteile in Grid-Systemen näher eingegangen wird, wird die Versicherbarkeit von Grid-Risiken im folgenden Abschnitt näher untersucht. Ziel ist es, technische Grid-Risiken auszuwählen, die von einer Grid-Versicherung versichert werden können.

### **5.3 Versicherbarkeit von Grid-Risiken**

Grundsätzlich ist nicht jedes Risiko jederzeit versicherbar [Nguy07, S. 85]. In diesem Zusammenhang stellt sich die Frage, welche Kriterien erfüllt sein müssen, damit ein technisches Grid-Risiko versichert werden kann. Dieser Abschnitt führt allgemeine Kriterien ein, die auf die Versicherbarkeit eines Risikos schließen lassen. Die Grenzen der Versicherbarkeit, die durch diese Kriterien festgelegt werden, sind jedoch nicht als objektiv zu interpretieren, da der Abschluss einer Versicherung auf einem subjektiven Entscheidungsproblem zwischen Versicherer und Ver-

sicherungsnehmer beruht [Berl82, S. 12] [Kart72, S. 283]. Diese allgemeinen Kriterien dienen im weiteren Verlauf des Kapitels als Grundlage für die Untersuchung der Versicherbarkeit der technischen Grid-Risiken, die in Kapitel 2 beschrieben wurden.

Für die Untersuchung stehen verschiedenen Modelle zur Auswahl, mit denen eine Kategorisierung in versicherbare und nicht versicherbare Risiken vorgenommen werden kann. Ist ein Risiko nicht versicherbar, so stellt die Versicherung keinen effizienten Ansatz zur Behandlung der Risiken dar. In diesem Fall sollte ein anderes Verfahren zur Risikobehandlung wie beispielsweise Risikovermeidung oder Risikoreduktion angewandt werden.

Die technischen Grid-Risiken, denen ein Grid-Service während seiner Ausführung ausgesetzt ist, werden im Folgenden nach dem Ansatz von Karten [Kart72] hinsichtlich ihrer Versicherbarkeit überprüft. Von den 4 Fehlerquellen, die von Hofer et al. [HoFa08] identifiziert und in Abbildung 2.3 gezeigt wurden, sind Servicequalität und Kommunikation Fehlerquellen, die die Ausführung des Grid-Service beeinflussen. Die beiden anderen Fehlerquellen Schnittstelle und Bereitstellung sind vor der Ausführung des Grid-Service angesiedelt und werden aus diesem Grund nicht weiter betrachtet.

Der Ansatz von Karten ist ein entscheidungstheoretischer Ansatz, der sehr häufig in der Versicherungsliteratur zitiert wird [Grze02b]. Die Entscheidung hängt dabei vor allem von den risikotheorietischen Eigenschaften ab, die unabhängig von anwendungsspezifischen Zusammenhängen angewandt werden können, um spezifische Risiken zu beurteilen. Zur Beurteilung der gridspezifischen Risiken werden nach Karten die Kriterien *Zufälligkeit*, *Eindeutigkeit*, *Schätzbarkeit*, *Unabhängigkeit* und *Größe* herangezogen [Kart72]:

- *Zufälligkeit*: Die Erfüllung des Kriteriums der Zufälligkeit fordert, dass ein den Versicherungsfall auslösendes Ereignis bei Vertragsabschluss ungewiss und unbeeinflussbar sein muss [Kart72, S. 287]. Dieses Kriterium schließt auch alle Schadensfälle aus, die ein Versicherungsnehmer selbst verursacht [TrHS04, S. 98].

Bei den oben genannten technischen Grid-Risiken ist die Versicherbarkeit auf den ersten Blick gegeben. Weder der Versicherungsnehmer noch der Versicherungsgeber können das Netzwerk und die Servicequalität der Ausführung des Grid-Service beeinflussen. Dies gilt allerdings nur, wenn das Netzwerk oder das Grid-System weder einem Versicherungsnehmer noch einem Versicherungsgeber gehört bzw. von ihnen verändert werden kann. Ist beispielsweise das Netzwerk, über das eine versicherte Kommunikation stattfindet, im Besitz des Versicherungsnehmers, kann dies moralische Risiken zur Folge haben. Moralischen Risiken können ebenfalls auftreten, wenn eine Serviceausführung auf Ressourcen eines



Grid-Systems versichert wird und dieses Grid-System dem Versicherungsnehmer gehört.

Im Rahmen dieser Arbeit werden diese absichtlichen Handlungen jedoch ausgeschlossen, sodass eine Ungewissheit und Unbeeinflussbarkeit für die zu betrachteten Risiken gegeben ist.

- *Eindeutigkeit*: Das Kriterium der Eindeutigkeit verlangt, dass sowohl das Ereignis – der Eintritt des Schadensfalls – als auch die Schadenshöhe in objektiver Weise festgelegt werden können [Mugl80] [Kart72, S. 289].

Die eindeutige Bestimmung des Schadensfalls hängt eng mit der Möglichkeit einer technischen Kontrolle des versicherten Grid-Service zusammen. Sowohl die Servicequalität als auch die Kommunikation zwischen Ressourcenkonsument und Ressourcenanbieter lassen sich überwachen, wie verschiedene Rahmenwerke zum Monitoring von Service-Level-Agreements [BCLW<sup>+</sup>04] [DDKK<sup>+</sup>04] [LKAC<sup>+</sup>08] und von Grid-Netzwerken [BuKB02] [LSWK<sup>+</sup>99] [ZaSa05] zeigen. Der Eintritt des Schadens kann aus diesem Grund als objektiv bestimmbar in den betrachteten Grid-Systemen angenommen werden.

Die Schadenshöhe eines versicherten Grid-Service umfasst zum einen die Kosten für die Ausführung des Service und zum anderen die Kosten für die Übertragung der Eingangsdaten und des Ergebnisses. Die Ausführung eines Service wird durch die Kosten der Ressourcen und eventuellen Lizenzkosten (siehe einführendes Beispiel in Kapitel 2) bestimmt und die Datenübertragung durch die Menge der transportierten Daten. Diese Größen sind messbar und aus diesem Grund in objektiver Weise bestimmbar, sodass auch die Schadenshöhe eindeutig in Grid-Systemen festgelegt werden kann.

- *Schätzbarkeit*: Das Kriterium der Schätzbarkeit repräsentiert das unzureichende Wissen über einen möglichen Schadenseintritt [Knig21]. Nach Arrow [Arro51, S. 424] ist dieses Merkmal ein unbedingtes Kriterium für die Versicherbarkeit. Die Schätzbarkeit bedeutet, dass es sich um Ereignisse handeln muss, bei denen der Eintritt oder Nichteintritt des Schadens und die Höhe des Schadens aufgrund von statistischen Wahrscheinlichkeiten vorherbestimmbar sind. Um eine risikogerechte Prämie zu bestimmen, müssen die Wahrscheinlichkeiten für den Versicherungsgeber quantifizierbar sein [Nguy07, S. 108].

Bei Schäden durch Grid-Risiken liegen teilweise statistische Daten vor, die durch Messungen von Serviceausführungen auf Grid-Systemen gewonnen wurden [LGWT06] [KHOS<sup>+</sup>06] [IJSE07]. Zusätzlich zur Messbarkeit der Schäden treten diese sehr häufig ein [LGWT06], was die Schätzbarkeit der technischen Risiken erleichtert. Ein Versicherer kann auf die



Vergangenheitsdaten für die Anpassung der Versicherungsprämie zurückgreifen.

Die Schadenshöhe ist in Grid-Systemen durch technische Kontrolle der Ausführung eines Grid-Service festgelegt, sodass eine Grid-Versicherung nur die Häufigkeit der Schäden schätzen muss. Wie das Modell von Dai et al. [DaPZ07] zeigt, kann die Zuverlässigkeit der technischen Grid-Risiken und damit die Häufigkeit der Schäden für die Risiken Servicequalität und Kommunikation durch Verteilungen beschrieben werden. Aus diesem Grund wird das Kriterium der Schätzbarkeit als erfüllt angenommen.

- *Unabhängigkeit*: Das Kriterium der Unabhängigkeit verlangt, dass die Risiken nicht positiv korreliert sein dürfen [Nguy07, S. 89]. Dies muss ausgeschlossen werden, um einen Zufallsprozess der versicherten Schadensereignisse im Versichertenbestand zu gewährleisten. Hinreichende stochastische Unabhängigkeit der einzelnen versicherten Risiken vermindern das Risiko des gesamten Bestands und sind zentrale Voraussetzung für den Effekt des Risikoausgleichs im Kollektiv [Berl82, S. 49] [Grze02b, S. 56].

Erleidet im Schadensfall ein großer Teil der Versicherten gleichzeitig einen Schaden, dann stellen die individuellen Schadensfälle keine unabhängigen Ereignisse mehr dar. Dies wird auch als Kumulrisiko oder Katastrophenrisiko bezeichnet [Herb92, S. 2ff.]. Eine solche Anhäufung gefährdet die Zahlungsfähigkeit des Versicherungsgebers, wenn die Schadenssumme die Kapazität des Versicherers übersteigt.

Messungen im französischen Grid<sup>5000</sup> [BCCD<sup>+</sup>06] haben ergeben, dass nur eine kleine positive Korrelation zwischen den Ausfällen in Grid-Systemen existieren; die meisten Risiken – etwa 80% – beziehen sich meist nur auf einen Standort des Grid-Systems [IJSE07]. Dies folgt dem Aufbau der CATNETS Grid-Ökonomie, ein Grid-System als eine Menge autonom agierender Einheiten zu betrachten, die, bedingt durch eine virtualisierte Infrastruktur, auf gemeinsamen Ressourcen koexistieren können.

Durch die starke Vernetzung der Grid-Ressourcen kann es zu gehäuften Schadensereignissen kommen, wie der vollständige Ausfall des Amazon S3-Dienstes im Juli 2008 zeigte<sup>1</sup>. Der Bericht zeigt, dass sich der Ausfall des Dienstes über mehrere Stunden hinweg durch einen Anstieg der Fehler im System angekündigt hat. Eine möglichst hohe Anzahl Versicherungsnehmer vorausgesetzt, wird eine Grid-Versicherung ebenfalls das erhöhte Risiko wahrnehmen und darauf reagieren können.

---

<sup>1</sup>Bericht über den vollständigen Ausfalls des Amazon S3-Dienstes am 20.07.2008: <http://status.aws.amazon.com/s3-20080720.html> (Letzter Abruf: 17.09.2008)

In der Simulationsstudie dieser Arbeit werden Kumulrisiken jedoch nicht simuliert, da es aktuell noch keine wissenschaftlichen Studien über die Häufigkeit des Auftretens und die Auswirkungen dieser Risiken gibt. Weitere Analysen über Kumulrisiken in produktiven Grid-Systemen werden benötigt, um diese Risiken zu quantifizieren und detailliert auf die Grenzen ihre Versicherbarkeit hin untersuchen zu können. Aus diesem Grund wird im Rahmen dieser Arbeit eine weitgehende Unabhängigkeit der Risiken angenommen, die die Servicequalität und die Kommunikation eines Grid-Service beeinträchtigen.

- *Größe*: Das Kriterium der Größe beschreibt den höchstmöglichen Schaden eines Einzelrisikos. Der Höchstschaten darf bestimmte, unternehmensspezifische Grenzen nicht überschreiten, damit das Risiko als versicherbar gilt [Kart00].

Die Größe des höchstmöglichen Schadens ist ein gut quantifizierbares Kriterium, da die Größe durch die Verhandlung der Ressourcen individuell festgelegt wird und sich am Marktpreis der Ressource orientiert. Sowohl für das technische Risiko der Servicequalität als auch das der Datenübertragung kann der höchstmögliche Schaden angegeben werden.

Da es sehr schwierig ist, Teilschäden bei der Ausführung einer Grid-Transaktion festzustellen, werden immer 100% der Schadenssumme versichert. Folgeschäden, die eintreten können, wenn ein Grid-Service in einen größeren Prozess eingebunden ist, werden nicht versichert. Diese können ein Vielfaches des Schadens einer Ausführung eines Grid-Service umfassen und hätten dadurch einen deutlich höheren Schaden eines Einzelrisikos zur Folge.

Dieses Kriterium, das zugleich das letzte Kriterium der Überprüfung der Versicherbarkeit darstellt, ist ebenfalls in Grid-Systemen erfüllt.

Die Analyse der Kriterien zur Versicherbarkeit nach Karten zeigt, dass die technischen Grid-Risiken Servicequalität und Kommunikation versicherbar sind. Kritisch in Grid-Systemen ist dabei die Forderung der Unabhängigkeit der technischen Risiken, die bei einem Ausfall des gesamten Grid-Systems nicht mehr gültig ist. Eine abschließende Beurteilung der Unabhängigkeit muss an einem konkreten Grid-System empirisch ermittelt und bei der Realisierung der Grid-Versicherung berücksichtigt werden. Dies würde über den Rahmen dieser Arbeit hinausgehen und wird aus diesem Grund nicht weiter betrachtet.

Am konkreten Beispiel der Simulationsstudie dieser Arbeit sind alle Kriterien der Versicherbarkeit erfüllt, sodass die technischen Grid-Risiken Servicequalität und Kommunikation durch die Grid-Versicherung versichert werden können. Dies wird zum einen durch den Einsatz von

stochastischen Zufallsvariablen zur Modellierung der Zufälligkeit und zum anderen durch die Möglichkeit der vollständigen Kontrolle der Parameter während eines Simulationslaufs erreicht. Details hierzu werden im folgenden Abschnitt sowie in Kapitel 6 und Kapitel 7 näher erläutert.

## 5.4 Realisierung des Versicherungsmodells

### 5.4.1 Schadensstatistik

Die Schadensstatistik ist der zentrale Datenspeicher der Versicherung und Grundlage für die Risikoabschätzung der Versicherung. Das Modell der Schadensstatistik erfasst in dieser Arbeit jedes Risiko, das die Grid-Versicherung übernimmt, die Anzahl der Versicherungspolicen, die Schadenssumme, die diese Versicherungspolicen verursacht haben und den durchschnittlichen Schaden pro Versicherungspolice. Neue Versicherungspolicen und Schadensmeldungen werden auf die bestehenden Werte aufaddiert. Die Schadensstatistik ist in Zellen organisiert, wobei eine Zelle einem individuellen Risiko oder einer Gruppe von Risiken entspricht. Eine beispielhafte Zelle mit  $a = 20$  Versicherten, einer Schadenssumme von  $s = 100$  und einem durchschnittlichen Schaden in Höhe von  $\frac{s}{a} = 5$  wird in Abbildung 5.2 gezeigt.

Versicherte a: 20
Schaden s: 100
Durchschnittlicher Schaden $s/a$ : 5

Abbildung 5.2: Beispielhafte Zelle der Schadensstatistik

Während einer Aktualisierung der Versicherungsprämien sind keine Veränderungen in der Schadensstatistik möglich, um Inkonsistenzen der Datenstruktur zu vermeiden. Wann eine Aktualisierung der Versicherungsprämien stattfindet, legt die Versicherungsperiode fest, die im folgenden Abschnitt diskutiert wird.

### 5.4.2 Versicherungsperiode

Die Versicherungsperiode bezeichnet den Zeitraum, für den eine Prämie berechnet wird. Während einer Versicherungsperiode sind die Prämien eines Versicherten konstant; an ihrem Ende werden üblicherweise eine Neuberechnung der Prämie und gegebenenfalls eine Prämienanpassung vorgenommen, die dann für die folgende Versicherungsperiode für den Versicherten gültig

sind. In konventionellen Versicherungen ist die Periode normalerweise auf ein Jahr festgelegt. Dies ist jedoch nicht auf Grid-Systeme übertragbar, da Ausführungszeiten von Grid-Services je nach Anwendung im Bereich von Millisekunden bis hin zu einigen Stunden oder Tagen liegen können. Dieser große Unterschied in der zeitlichen Ausdehnung erfordert einen anderen Ansatz zur Festsetzung der Versicherungsperiode als in herkömmlichen Versicherungen.

Zwei grundlegende Ansätze, mit denen die Versicherungsperiode in Grid-Systemen festgelegt werden kann, werden deshalb an dieser Stelle eingeführt und diskutiert. Zum einen kann ein fester Zeitraum gewählt werden, analog beispielsweise zu Haftpflichtversicherungen, bei denen die Versicherungsperiode in der Regel ein Jahr beträgt. Dies soll im Folgenden als zeitbezogene Versicherungsperiode bezeichnet werden. Zum anderen kann die Versicherung für jede durchgeführte Transaktion separat abgeschlossen werden, was im Folgenden als transaktionsbezogene Versicherungsperiode bezeichnet wird. Abbildung 5.3 und Abbildung 5.4 stellen exemplarisch die beiden Möglichkeiten dar.

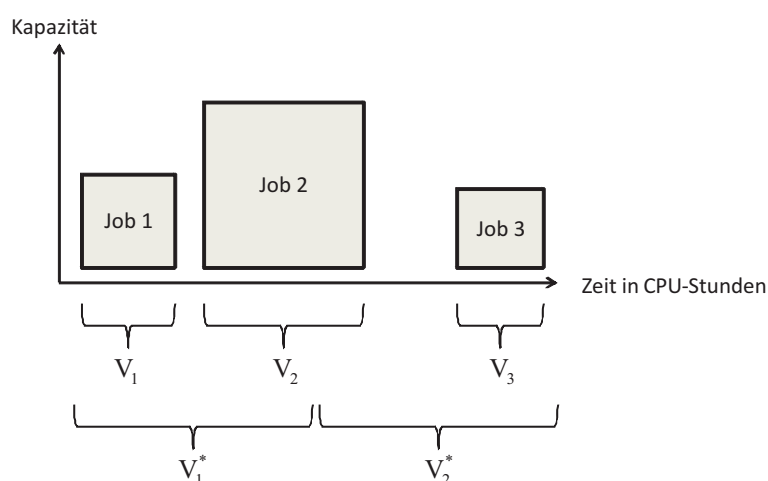


Abbildung 5.3: Zeitbezogene Versicherungsperiode

Die Strategie einer transaktionsbezogenen Versicherungsperiode ist für Versicherungen in Grid-Ökonomien geeigneter als die Strategie einer zeitbezogenen Versicherungsperiode. Dies legt der Charakter der zu versichernden Güter nahe: Während bei realen Versicherungen mit einer zeitbezogenen Versicherungsperiode die Gefahr eines Schadensereignisses kontinuierlich besteht, ist diese im Grid an die Durchführung von Transaktionen geknüpft. Falls ein Ressourcenkonsument nicht kontinuierlich Ressourcen nachfragt, bezahlt er bei zeitbezogenen Versicherungsperioden Versicherungsschutz für Teile der Periode, in denen kein Schaden entstehen kann, da er die Grid-Ressourcen nicht benutzt. Er würde die Leistung bezahlen, die er nicht in Anspruch nimmt. In Abbildung 5.3 wird das genannte Problem für das Zeitintervall zwischen Job 2 und Job 3 deut-

lich, da die Ressourcen in der Versicherungsperiode  $V_2^*$  nicht vollständig ausgelastet werden.

Überdies stellt sich bei zeitbezogenen Versicherungsperioden die Frage nach der Länge der gewählten Zeitintervalle. Werden lange Zeitintervalle gewählt, so verlangsamt dies den Aufbau von Erfahrungsdaten als Basis von Prognosen des Erwartungsschadens. Überdies wird Fehlverhalten nur verzögert sanktioniert und kann schlimmstenfalls nicht mehr zu einer Transaktion zugeordnet werden. Dieses Problem wird in Abbildung 5.3 für die Versicherungsperiode  $V_1^*$  deutlich: Scheitert Job 1, kann der Anbieter dennoch Job 2 zu den Konditionen von Job 1 versichern, da eine Prämienneuberechnung erst im Laufe von Job 2 erfolgt.

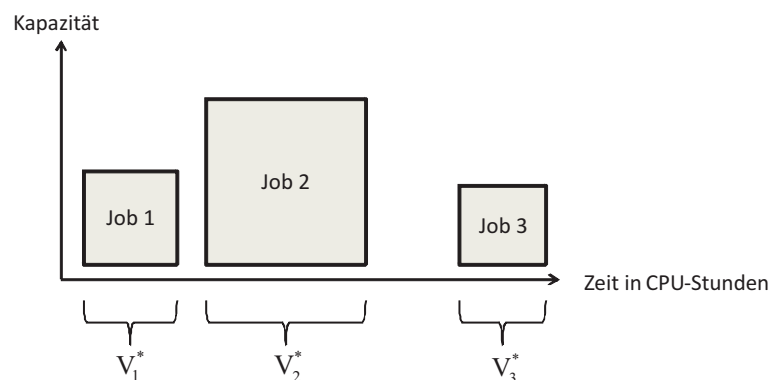


Abbildung 5.4: Transaktionsbezogene Versicherungsperiode

Bei einer transaktionsbezogenen Versicherungsperiode, wie sie Abbildung 5.4 zeigt, wird das Schadensereignis von Job 1 hingegen schon in die für Job 2 zu bezahlende Versicherungsprämie mit einbezogen. Diese Vorgehensweise bietet eine aktuelle Abbildung der Schadensstatistik auf die Versicherungsperioden. Jedoch führt dies zu einer ständigen Neuberechnung der Prämien nach jeder Transaktion. In Grid-Systemen mit überwiegend langen Jobs, die über mehrere Minuten oder Stunden ausgeführt werden, ist dies keine Einschränkung. Handelt es sich hingegen um ein Grid-System mit überwiegend interaktiven Jobs bzw. Grid-Services, die sich durch sehr kurze Ausführungszeiten bei gleichzeitig hoher Ausführungsfrequenz auszeichnen, würde es einen großen Rechenaufwand bedeuten, nach jeder Transaktion eine Neuberechnung der Versicherungsprämien durchzuführen.

Der methodische Vorteil aktuellerer Versicherungsprämien bei einer transaktionsbezogenen Versicherungsperiode wird durch den erhöhten Rechenaufwand bei einer hohen Anzahl an Transaktionen wieder aufgehoben. Aus diesem Grund wird im Versicherungsmodell dieser Arbeit eine Mischung beider Verfahren verwendet. Die transaktionsbasierte Versicherungsperiode wird mit einer zeitbezogenen Berechnung der Versicherungsprämien verknüpft, was die hohe Rechen-

intensität einer rein transaktionsbasierten Versicherungsperiode abmildert. Da die Transaktion länger als die Periode der Neuberechnung dauern kann, treten so genannte Spätschäden auf. Spätschäden sind Schäden, bei denen das Schadensereignis in eine der Entrichtung der Versicherungsprämie nachgelagerte Versicherungsperiode fällt. [Schm05, S. 269].

Dies erfordert ein Verfahren zur Prämienberechnung, das Spätschäden mit in die Prognose einbezieht. Ein solches Verfahren zur Prämienberechnung wird im folgenden Abschnitt eingeführt.

### 5.4.3 Prämienberechnung

Zentrale Herausforderungen bei der Prämienberechnung einer Grid-Versicherung sind zum einen die exakte Prognose des Erwartungsschadens für die Gesamtheit der Versicherten und zum anderen die Prognose individueller Erwartungsschäden. Die Prognose für die Gesamtheit der Versicherten ist Voraussetzung für die Zuverlässigkeit der Versicherung aus Sicht des Versicherungsnehmers, im Schadensfall die vereinbarte Versicherungssumme zu erhalten. Sie entscheidet darüber, ob die Schäden durch Prämieinnahmen der Versicherung ausreichend abgedeckt sind und die Versicherung keinen Ruin erleidet. Die Prognose individueller Erwartungsschäden einer Grid-Transaktion unterstützt die Vorhersage für das Gesamtkollektiv und ist eine wichtige Voraussetzung für die Sanktionierungswirkung der Versicherung.

Wird eine Transaktion aufgrund des aktuellen Zustands des Grid-Systems und der vereinbarten Qualitätskriterien als besonders risikoreich eingeschätzt, so muss dies auch in der Prämie ausgedrückt werden können. Die Schadenshöhe ist bei Abschluss der Versicherung bekannt; sie entspricht dem Preis der Ressourcen, der im Service-Level-Agreement festgelegt ist. Aus diesem Grund reduziert sich die Prognose des individuellen Erwartungsschadens auf die Vorhersage der Eintrittswahrscheinlichkeit des Schadens.

#### 5.4.3.1 Prämiendifferenzierung

In der Grid-Versicherung liegen keine homogenen Gesamtbestände vor, vielmehr unterscheiden sich die Einzelrisiken - wie die Studien über die Zuverlässigkeit von Grid-Systemen in Kapitel 2 belegen - sehr stark voneinander. Aus diesem Grund bietet es sich an, die Prämie im Hinblick auf Merkmale zu differenzieren, die signifikant die technischen Grid-Risiken beeinflussen. Würde die Grid-Versicherung für alle einzelnen Risiken des Gesamtbestandes eine einheitliche Prämie  $p$  festlegen, so ist es erst möglich, ex-post die risikogerechte Prämie zu ermitteln. Differenziert die Grid-Versicherung hingegen die Prämien nach verschiedenen Risikogruppen, so ist

die Prämienfestsetzung unabhängig von der Bestandszusammensetzung für die einzelnen Gruppen risikogerecht [Meye05].

Für die im vorherigen Abschnitt als versicherbar identifizierten, technischen Grid-Risiken Servicequalität und Kommunikation müssen Merkmale ausgewählt werden, die einen signifikanten Einfluss auf das Risiko und damit auf die Versicherungsprämie haben. Diese ausgewählten Merkmale werden auch Tarifmerkmale genannt [Pitk75]. Sie ermöglichen es, jedes Risiko in die von den Tarifmerkmalen festgelegte Matrix einzuordnen [Pitk75].

Die Auswahl der Tarifmerkmale orientiert sich an dem Modell von Dai [DaPZ07], das in Kapitel 2 eingeführt wurde<sup>2</sup>. Die Merkmale, aufgegliedert nach den beiden technischen Risiken Servicequalität und Kommunikation, sind:

- Servicequalität: Im Fokus des technischen Risikos Servicequalität steht die Zuverlässigkeit der Serviceausführung. Diese wird nach dem Modell von Dai signifikant von den Merkmalen Ausführungszeit des Basic-Service und Größe des verwendeten Ressourcenbündels beeinflusst [DaPZ07].
- Kommunikation: Die Zuverlässigkeit einer virtuellen Netzwerkverbindung wird nach Dai von drei Merkmalen festgelegt [DaPZ07]: verfügbare Bandbreite, Menge der zu übertragenen Daten und Entfernung zwischen den Kommunikationspartnern.

Die Merkmale der technischen Grid-Risiken werden nun in einem zweiten Schritt in verschiedene Gruppen eingeteilt, für die unabhängig voneinander eine Prämie berechnet werden kann. Dadurch lässt sich eine genauere Abschätzung der Risiken erreichen. Ausführungen von Services, die besonders häufig eine bestimmte Schadenshöhe verursachen, lassen sich damit zu einer Gruppe zusammenfassen, für die eine individuelle Prämie berechnet wird. Die genaue Festlegung der Einteilung der Gruppen erfolgt individuell für ein Simulationsszenario und seine Risikokonfiguration.

Für die weitere Betrachtung in dieser Arbeit wird als Beispiel das technische Risiko der Servicequalität ausgewählt, dessen Umsetzung durch die geringere Anzahl an Tarifmerkmalen einfacher erscheint. Die Grid-Versicherung bietet in diesem Fall eine Versicherungspolice an, die das Risiko einer Verletzung der vereinbarten Servicequalität durch fehlerhafte Serviceausführung abdeckt. Die vorgestellten Verfahren lassen sich jedoch ohne Einschränkungen auch auf das technische Risiko der Kommunikation mit seinen drei Tarifmerkmalen übertragen.

---

<sup>2</sup>Ein allgemeiner Prozess zur Bestimmung von Tarifmerkmalen aus empirisch erhobenen Daten wird in der theoretischen Abhandlung zur Bildung eines Versicherungstarifs von Pitkänen [Pitk75] beschrieben.



### 5.4.3.2 Prognose auf Basis von Tarifmerkmalen

Für die im vorherigen Abschnitt eingeführten Tarifmerkmale des technischen Risikos Servicequalität muss für jede mögliche Kombination der Tarifmerkmalsausprägungen die Höhe der Prämie festgelegt werden. Abbildung 5.5 zeigt eine Schadensstatistik für 3 Merkmalsausprägungen des Tarifmerkmals Ausführungszeit und 2 Merkmalsausprägungen des Tarifmerkmals Ressourcenmenge<sup>3</sup>. Mit diesem Tarif lässt sich die Zuverlässigkeit der Serviceausführung in 6 Schadensgruppen untergliedern, für die jeweils eine Prämie berechnet werden kann.

	Ressourcenmenge > (4 CPU, 2GB RAM)	Ressourcenmenge < (4 CPU, 2GB RAM)
Ausführungszeit > 10h	Versicherte: 10 Schadenssumme: 1000 Durchschnittlicher Schaden: 100	Versicherte: 20 Schadenssumme: 800 Durchschnittlicher Schaden: 40
Ausführungszeit <= 10h und >5h	Versicherte: 40 Schadenssumme: 2000 Durchschnittlicher Schaden: 50	Versicherte: 15 Schadenssumme: 225 Durchschnittlicher Schaden: 15
Ausführungszeit <= 5h	Versicherte: 35 Schadenssumme: 420 Durchschnittlicher Schaden: 12	Versicherte: 25 Schadenssumme: 250 Durchschnittlicher Schaden: 10

Abbildung 5.5: Beispielhafte Schadensstatistik der Tarifmerkmale Ausführungszeit und Ressourcenmenge

Der Vorteil einer Prognose auf Grundlage von Tarifmerkmalen gegenüber einer rein individuellen Prognose besteht darin, dass der Erwartungsschaden eines versicherten Service-Level-Agreements, das die Servicequalität beschreibt, Vergangenheitsdaten anderer versicherter SLA für die Schätzung verwenden kann. Dies führt zu einer deutlich breiteren Datenbasis aufgrund der größeren Stichprobe und verringert dadurch ungenaue Schätzungen aufgrund fehlender oder lückenhafter Erfahrungsdaten, wie sie mit wachsender Größe von Grid-Systemen auftreten können. SLAs neuer Grid-Services werden hinsichtlich der bestehenden Tarifmerkmale eingruppiert. Dies hat den Vorteil, dass keine Erfahrungsdaten über das Risiko neuer Services aufgebaut werden müssen, sondern auf Erfahrungsdaten ähnlicher SLAs bzw. deren Tarifmerkmale zurückgegriffen werden kann.

Für die Kalkulation der Prämie wird jeder Merkmalsausprägung ein Tariffaktor zugeordnet. Dieser Tariffaktor bestimmt den Einfluss der Merkmalsausprägung auf das Risiko, das mit diesem Merkmal verknüpft ist und damit auf die zu bezahlende Prämie.

<sup>3</sup>In realen Grid-Systemen kann diese Zahl der Merkmalsausprägungen deutlich höher liegen und auch mehr Merkmale umfassen. An dieser Stelle abstrahiert die Simulationsmodell von realen Grid-Systemen, in dem es die oben eingeführten Tarifmerkmale und ihre Ausprägungen betrachtet.



Tabelle 5.1: Beispielhafte Klassifizierung des Tarifmerkmals *Ausführungszeit*

Tarifmerkmal Ausführungszeit	
Ausführungszeit	Tariffaktor
Klasse 1: >10h	$x_1 = 1,20$
Klasse 2: $\leq 10h$ und $> 5h$	$x_2 = 1,00$
Klasse 3: $< 5h$	$x_3 = 0,80$

Die Berechnung der Prämie eines SLA mit den gegebenen Merkmalen erfolgt durch ein multiplikatives Modell, das Voraussetzung ist für die weitere Prämienberechnung mithilfe des Marginalsummenverfahrens [Schm05, S. 282]. Für die Merkmalsausprägung  $i$  des Tarifmerkmals  $x$  (Zuverlässigkeit der Serviceausführung) und die Merkmalsausprägung  $j$  des Tarifmerkmals  $y$  (Zuverlässigkeit der Ressourcenmenge) lässt sich die Prämie wie folgt bestimmen [Meye05]:

$$p_{ij} = p * x_i * y_j,$$

wobei  $p$  die Grundprämie der Versicherung ist. Die Tariffaktoren üben einen relativen Einfluss auf die Gesamtprämie aus. Ein Tariffaktor von  $< 1$  würde eine Reduzierung der Prämie bedeuten, während ein Tariffaktor von  $> 1$  eine Erhöhung der Prämie zur Folge hätte. Bei der Festlegung der Grundprämie gibt es keine eindeutige Vorgehensweise, sodass sie analog zu [Meye05] in den folgenden Ausführungen immer als 1 angenommen. Dadurch reduziert sich die Komplexität der Berechnung; die Prämie wird nur durch die Tariffaktoren bestimmt.

Tabelle 5.1 und Tabelle 5.2 zeigen eine beispielhafte Klassifizierung der beiden Tarifmerkmale mit ihren dazugehörigen Tariffaktoren. Das Tarifmerkmal *Ausführungszeit* wird in drei Klassen eingeteilt, in die alle Serviceausführungen hinsichtlich ihres Risikoprofils eingeordnet werden. Klasse 1 umfasst Ausführungszeiten größer 10 Stunden, Klasse 2 Ausführungszeiten zwischen 5 und 10 Stunden und Klasse 3 Ausführungszeiten, die unter 5 Stunden liegen. Das Tarifmerkmal Ressourcenmenge unterscheidet Ressourcenbündel der Klasse 1, die mehr als 4 CPUs und 2GB RAM haben und Ressourcenbündel der Klasse 2, die weniger Ressourcen umfassen.

Wird ein Basic-Service mit einer Ausführungszeit von 4 Stunden und mit einem Ressourcenbündel, das 2CPUs und 1GB Hauptspeicher umfasst, versichert, so wird die risikogerechte Prämie durch die Multiplikation der Tariffaktoren 0,80 für das Tarifmerkmal Ausführungszeit und 1,80 für das Tarifmerkmal Ressourcenmenge berechnet. Ein Versicherungsnehmer müsste für die Versicherung dieses Basic-Service 1,44 Geldeinheiten bezahlen.

Tabelle 5.2: Beispielhafte Klassifizierung des Tarifmerkmals *Ressourcenmenge*

Tarifmerkmal Ressourcenmenge	
Ressourcenmenge	Tariffaktor
Klasse 1: >(4 CPU, 2GB RAM)	$y_1 = 2,34$
Klasse 2: <=(4 CPU, 2GB RAM)	$y_2 = 1,80$

Aus verschiedenen Verfahren<sup>4</sup> zur Reservierung von Spätschäden wurde das Marginalsummenverfahren zur Bestimmung der Tariffaktoren ausgewählt. Das Marginalsummenverfahren ist ein sehr einfaches Verfahren, das zum einen iterativ lösbar ist und damit eine einfache Integration in ein Multi-Agenten-System zulässt und zum anderen gute Eigenschaften hinsichtlich schneller Konvergenz und geringe Empfindlichkeit gegen Ausreißer aufweist [Mack97]. In Deutschland wird das Verfahren vor allem in der KFZ-Haftpflichtversicherung eingesetzt [Meye05].

Es bezeichnet  $s_{ij}$  den in Zelle  $ij$  insgesamt beobachteten Schaden und  $a_{ij}$  die Anzahl der Versicherungsnehmer in Zelle  $ij$ . Im Marginalsummenverfahren werden die Tariffaktoren  $x_i$  und  $y_j$  der Tarifmerkmale  $x$  und  $y$  so festgesetzt, dass die Spaltensummen und Zeilensummen der geschätzten Schäden  $\hat{s}_{ij} = a_{ij} * x_i * y_j$  mit den entsprechenden Spalten- und Zeilensummen der beobachteten Schaden  $s_{ij}$  übereinstimmen [Meye05]. Es gilt:

$$\sum_j \hat{s}_{ij} = \sum_j s_{ij}, \forall i \in \{1, 2\}$$

$$\sum_i \hat{s}_{ij} = \sum_i s_{ij}, \forall j \in \{1, 2, 3\}$$

Das Marginalsummenverfahren stellt also darauf ab, dass hinsichtlich jeder Merkmalsausprägung das Teilkollektiv derjenigen Versicherungsnehmer, die diese Merkmalsausprägung aufweisen, gerade so viel Prämienaufkommen erbringt, wie dieses Teilkollektiv an Schaden erwarten lässt.

Setzt man

$$\hat{s}_{ij} = a_{ij} * x_i * y_j,$$

so lässt sich die Gleichung in eine iterativ berechenbare Form bringen:

<sup>4</sup>In [Schm05, S. 269ff.] und [Meye05] werden noch weitere Verfahren zur Berechnung von Tariffaktoren vorgestellt.

$$x_i = \frac{\sum_j s_{ij}}{\sum_j a_{ij} y_j}, \forall i \in \{1, 2\}$$

$$y_j = \frac{\sum_i s_{ij}}{\sum_i a_{ij} x_i}, \forall j \in \{1, 2, 3\}$$

Diese Form der Gleichungen wird zur iterativen Ermittlung der Tariffaktoren  $x_i$  und  $y_j$  verwendet und ist der zentrale Bestandteil des Algorithmus zur Bestimmung der Tariffaktoren zur Berechnung der Versicherungsprämie. Nach Ablauf einer Versicherungsperiode werden die Tariffaktoren aktualisiert und damit die Prämien für die neue Periode festgelegt. Die Prämien, die durch die Multiplikation der Tariffaktoren berechnet werden, sind faire Prämien, die den Erwartungsschaden eines Risikos abbilden. Weicht die Prognose der Versicherung vom tatsächlichen Risiko ab, so ist die Versicherung einem versicherungstechnischen Risiko ausgesetzt, das im folgenden Abschnitt näher betrachtet wird.

### 5.4.3.3 Versicherungstechnisches Risiko

Die Berechnung der fairen Prämie für eine Versicherungsperiode setzt sowohl für Individuen als auch für Kollektive die Kenntnis der entsprechenden Schadensverteilung voraus. Die exakte Berechnung der Prämie ist damit nur ex-post in Kenntnis der tatsächlichen Schadensdaten der Versicherungsnehmer möglich. Eine derartige nachträgliche Berechnung hat jedoch die Nachteile, dass sie einerseits einen erheblichen Vorfinanzierungsbedarf von Seiten der Versicherung erfordern würde und andererseits für die Versicherten lediglich eine Reduzierung, nicht aber eine Eliminierung des individuellen Risikos bedeuteten, da ihre Risikokosten während der aktuellen Periode unbekannt sind [Brae60, S. 17]. Aus diesem Grund wird das alternative Vorgehen einer Prämienberechnung und -zahlung vor Beginn der Versicherungsperiode durchgeführt, was jedoch die Notwendigkeit, den Erwartungswert des Schadens in der entsprechenden Periode vor Periodenbeginn prognostizieren zu müssen, erfordert [Brae60, S. 25]. Dies bringt zwangsläufig das Risiko einer Abweichung des tatsächlichen vom prognostizierten Schaden mit sich. Dies wird in der Literatur als versicherungstechnisches Risiko bezeichnet [Grze02b, S. 46].

Eine Klassifikation des versicherungstechnischen Risikos zeigt Abbildung 5.6. Analog zu [Farn06, S. 85ff.], [Albr92, S. 8] und [Wagn00, S. 139f.] lässt sich das Risiko der Abweichung von Erwartungsschaden und dem tatsächlich entstandenen Schaden differenzieren in Zufalls-, Änderungs-,

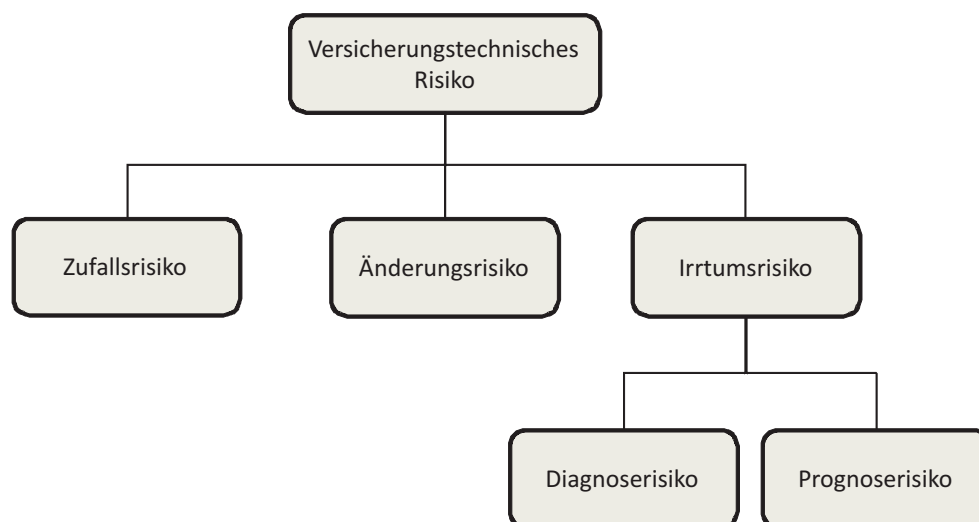


Abbildung 5.6: Klassifikation der versicherungstechnischen Risiken [Farn06, S. 85ff.] [Albr92, S. 8] [Wagn00, S. 139f.]

und Irrtumsrisiko. Das Zufallsrisiko ist hierbei das Risiko einer zufälligen Abweichung des tatsächlich entstandenen Schadens von seinem Erwartungswert zum Zeitpunkt der Prämienberechnung. Das Änderungsrisiko beschreibt die Abweichung der Wahrscheinlichkeitsverteilung der Schäden nach dem Zeitpunkt der Schätzung.

Das Irrtumsrisiko setzt sich aus dem Diagnoserisiko und dem Prognoserisiko zusammen. Das Diagnoserisiko ist das Risiko, die die Schadenshöhe beeinflussende Faktoren für die zum Prognosezeitpunkt bekannten Perioden in ihrer Wirkung falsch einzuschätzen. Das Prognoserisiko ist das Risiko, das die in der Vergangenheit richtig konstatierten Zusammenhänge für die prognostizierte Periode nicht zutreffen. Während das Zufallsrisiko eine echte Zufallsgröße ist und damit selbst bei vollständiger Information des Versicherungsunternehmens über die Schadensverteilungen seiner Versicherten weiterhin besteht, sind Änderungs- und Irrtumsrisiko das Ergebnis unvollständiger Information des Versicherungsunternehmens.

Eine Versicherung, die sich bei ihrer Prämienberechnung ausschließlich am versicherungstechnischen Äquivalenzprinzip orientiert, wird – selbst bei ausschließlicher Betrachtung des Zufallsrisikos, das als echte Zufallsgröße einfacher zu handhaben ist als das Änderungs- und Irrtumsrisiko – mit hoher Wahrscheinlichkeit bereits in der ersten Versicherungsperiode, im Falle einer symmetrischen Verteilung der Zufallsrisiken und ohne Berücksichtigung des Irrtumsrisikos mit einer Wahrscheinlichkeit von 50%, zahlungsunfähig werden [Albr92, S. 20] [ZwEi07, S. 244].

Um das Problem der ex-ante-Prognose von Erwartungsschäden  $E[S]$  zu lösen, ist es demgemäß

nötig, die Versicherung mit zusätzlichem Kapital auszustatten, das in Form eines Sicherheitszuschlags  $SK$  auf die faire Prämie  $p$  aufgeschlagen wird:

$$p = E[S] + SK$$

Der Sicherheitszuschlag dient dem Aufbau bzw. der Erhaltung eines Kapitalstocks  $K$  der Versicherung, der es dieser ermöglichen soll, Abweichungen des entstehenden Schadens von seinem Erwartungswert durch Zufalls- oder Prognosefehler aufzufangen. Dieser muss auch im Grid-Versicherungsmodell berücksichtigt werden, um auftretende versicherungstechnische Risiken zu behandeln.

Der Sicherheitszuschlag  $SK$  wird im Rahmen dieser Arbeit nach dem Erwartungswertsprinzip<sup>5</sup> als proportionaler Zuschlag modelliert:

$$p = E[S] + \lambda * E[S]$$

Der Parameter  $\lambda$  kann dabei beliebig gewählt werden. Er stellt die Risikoaversion der Versicherung dar. Solange das Sicherheitskapital  $SK$  kleiner als die Summe der individuellen Maximalschäden aller Versicherten ist, kann eine Zahlungsunfähigkeit nicht ausgeschlossen werden. Jedoch geht das versicherungstechnische Risiko im Sinne der Ruinwahrscheinlichkeit mit einer wachsenden Anzahl Versicherungsnehmer gegen null [ZwEi07, S. 243].

Die Grid-Versicherung muss für die Modellierung des versicherungstechnischen Risikos zwei Parameter abbilden:

- Kapitalstock  $K$ : Der Kapitalstock stellt zu Beginn der Simulation das Grundkapital der Versicherung dar, um auftretende Schäden in der ersten Periode zu bezahlen. In den weiteren Perioden muss die Versicherungsprämie den Kapitalstock erhalten, um einen Ruin der Versicherung zu vermeiden.
- Sicherheitszuschlag  $SK$ : Der Sicherheitszuschlag ist der *Preis für die Risikotragung* [Kart93, S. 38] durch die Versicherung. Der Sicherheitszuschlag kompensiert das Änderungs- und Irrtumsrisiko der Prämienkalkulation. Das Simulationsmodell folgt dem Erwartungswertsprinzip zur Prämienkalkulation. Wie oben eingeführt, bestimmt der Parameter  $\lambda$  den pro-

<sup>5</sup>Das Erwartungswertsprinzip ist ein einfaches funktionalparametrisches Prämienkalkulationsprinzip, das nur den Erwartungswert für die Prämienkalkulation heranzieht [ZwEi07, S. 244].

proportionalen Zuschlag des Erwartungswert als Sicherheitskapital. Dieser wird bei Start der Simulation abhängig vom simulierten Grid-Szenario festgelegt.

Mit der Betrachtung des versicherungstechnischen Risikos ist die Prämienberechnung als wichtige Funktion einer Versicherung abgeschlossen. Der folgende Abschnitt diskutiert Möglichkeiten der Erstattung des Schadens.

#### 5.4.4 Erstattung im Schadensfall

Erstattung im Schadensfall kann durch die bereits vorgestellten Szenarien einer monetären bzw. nichtmonetären Kompensation erfolgen. Welches Modell zum Einsatz kommt, hängt davon ab, wie im Fehlerfall mit den auftretenden Risiken umgegangen werden soll. Ist der Ressourcenkonsument bereit, eine verlängerte Dienstaufführung zu akzeptieren, so kann das Modell einer Erstattung durch Ressourcen angewandt werden. Hierfür müssen die entsprechenden Vorgaben im SLA spezifiziert werden.

Eine Entschädigung durch erneute Bereitstellung von Rechenleistung hat den Vorteil, dass der Geschädigte in jedem Fall sein geplantes Vorhaben – wenn auch mit Verspätung – durchführen kann, was im Fall monetärer Entschädigungen zumindest bei stark volatilen Preisen für die Nutzung von Ressourcen nicht zwangsläufig gegeben sein muss.

Nachteilig erscheint hingegen die relative Inflexibilität dieser Lösung, da nicht in jedem Fall die verabredete Dienstgüte überhaupt erbracht werden kann. Hier würde auch die zweite Ausführung des Grid-Service auf neuen Ressourcen scheitern. Überdies begründet diese Lösung die Notwendigkeit eines für die Versicherung verfügbaren Pools ungenutzter Ressourcen, sodass im Schadensfall entsprechende Leistungen bereitgestellt werden können. Entweder die Versicherung stellt diese selbst bereit oder muss sie über den Ressourcenmarkt beziehen.

**Szenario 1: Zeitunkritische Ausführung** Der Ressourcenkonsument macht eine Zeitvorgabe von  $x$  Sekunden; innerhalb dieser Zeitspanne muss der Grid-Service ausgeführt werden. Im Fall einer fehlerhaften Ausführung wird der Schaden an die Versicherung gemeldet und der Grid-Service wird ein weiteres Mal ausgeführt. Die Zeit bis zum Erhalt eines Ergebnisses verlängert sich dadurch um weitere  $x$  Sekunden.

**Szenario 2: Zeitkritische Ausführung** In einem zweiten Beispiel handelt es sich um einen zeitkritischen Job, der ausgeführt werden soll. Bei diesem ist der Zeitpunkt der Leistungser-

bringung ein für die Wertschätzung bedeutender Teil der Dienstgütevereinbarung. Entsprechend kann nicht zwangsläufig davon ausgegangen werden, dass ein Ressourcennachfrager nach dem Scheitern einer Job-Ausführung noch Bedarf an entsprechenden Ersatzressourcen hat.

Das erste Szenario verfolgt einen Ansatz, indem die Versicherung eine Art Sicherungsanbieter darstellt, der zu marktfähigen Preisen freie Kapazitäten vorhält. Dies ist mit hohen Kosten für die Versicherung verbunden, da eine Abweichung von den tatsächlichen Schäden zu den prognostizierten zu einer schlechten Auslastung führt. Im zweiten Szenario kompensiert die Versicherung das Risiko der Serviceausführung monetär. Die Versicherung muss keine Ressourcen vorhalten oder dem Ressourcenkonsumenten Ersatzressourcen vermitteln und kann dadurch einfacher realisiert und flexibler gestaltet werden. Die Entscheidung einer Beschaffung von Ersatzressourcen liegt in diesem Szenario beim Ressourcenkonsumenten.

Wegen der geringeren Komplexität wird Szenario 2 der Erstattung im Schadensfall in der Simulationsstudie verwendet. Weitere Optionen der Einbettung der Grid-Versicherung in die CATNETS Grid-Ökonomie werden im folgenden Abschnitt diskutiert.

## **5.5 Einbettung des Grid-Versicherungsmodells in die CATNETS Grid-Ökonomie**

### **5.5.1 Versicherungswettbewerb vs. Versicherungsmonopol**

Grundsätzlich besteht sowohl die Möglichkeit, Versicherungen in Grid-Ökonomien als Versicherungsmonopol als auch in Form mehrerer konkurrierender Versicherungen umzusetzen. Vom ökonomischen Standpunkt ist hierbei ein Versicherungswettbewerb sinnvoller, schließlich führt ein Wettbewerb zwischen verschiedenen Versicherungen zu einer detaillierteren Differenzierung der Prämien. Diesem Sachverhalt stehen eine weniger breite Datenbasis zur Schätzung des Risikos im Fall mehrerer Versicherungen und ein deutlich erhöhter Aufwand gegenüber, um eine Strategie zu entwerfen und umzusetzen, die mit mehreren Versicherungen umgehen kann. Zusätzlich wird die Grid-Versicherung im Rahmen dieser Arbeit als ein Teil des ökonomischen Ressourcenmanagements gesehen, sodass ein Versicherungsszenario mit mehreren Versicherungen als wenig wahrscheinlich anzunehmen ist.

Aus diesem Grund wird die Grid-Versicherung als Monopol in der CATNETS Grid-Ökonomie modelliert. Die Versicherungsprämie wird wie oben beschrieben als faire Prämie implementiert,



da andernfalls aufgrund der Monopolstellung mit überhöhten Preisen und damit einer unnötigen Erhöhung der Transaktionskosten zu rechnen ist.

### 5.5.2 Modellierung der Versicherungsnachfrage

Eine weitere Grundsatzfrage bei der Ausgestaltung von Versicherungen im Grid ist, ob ein Ressourcenkonsument gezwungen werden soll sich zu versichern. Gegen eine Versicherungspflicht spricht, dass diese in einer Grid-Ökonomie eine Beeinflussung der Ziele der Marktakteure und hier insbesondere der Ressourcenkonsumenten darstellt. Nur wenn sich eine Versicherung in einer Grid-Ökonomie auch ohne Versicherungszwang durchsetzt, kann sicher davon ausgegangen werden, dass ihre Existenz den Marktakteuren einen Nutzenzugewinn beschert. Aus diesem Grund wird eine freiwillige Versicherungsnachfrage durch den Ressourcenkonsumenten modelliert und realisiert.

Der Ressourcenkonsument muss dabei eine Entscheidung unter Unsicherheit treffen dessen Lösung unter Verwendung eines Entscheidungsbaums bereits im vorangegangenen Kapitel skizziert wurde. Als risikoaverses Individuum ist der Ressourcenkonsument bereit, einen Preis zu bezahlen, um eine risikoreichere Alternative zu vermeiden. Dies ist abhängig von seiner individuellen Risikoeinstellung, die sich über die Zeit ändern kann, um möglichst einen hohen Gewinn zu erzielen.

Dabei wird die Verhandlungsstrategie, wie sie in Kapitel 3 vorgestellt wurde, um zusätzliche Parameter erweitert. Diese Parameter modellieren die subjektive Risikoeinschätzung und die Zahlungsbereitschaft eines Agenten. Der Erwartungsnutzen ohne Versicherung wird mit dem Erwartungsnutzen mit Versicherung verglichen und diejenige Alternative ausgewählt, die für den Agenten als vorteilhafter erscheint.

Die Modellierung als zusätzliche Parameter hat den Vorteil, dass sie als Teil der Gesamtstrategie des Agenten durch den co-evolutionären Lernalgorithmus angepasst werden. In die Zielerreichung des Agenten fließt also sowohl das Verhandlungsergebnis als auch die Entscheidung der Risikobehandlung mit ein.

## 5.6 Zusammenfassung

Dieses Kapitel stellt das zentrale Modell der Arbeit, ein Versicherungsmodell für technische Grid-Risiken, und seine Realisierung vor und legt die Parameter fest, die in der Simulationsstudie



simuliert und konfiguriert werden müssen. Eine Bewertung der Risiken erfolgt durch Einordnung in Risikogruppen, die durch die Merkmalsausprägungen des SLA bzw. des Grid-Systemzustands festgelegt sind. Dabei erfüllt das Modell die Anforderungen, die in den vorangegangenen Kapiteln gestellt wurden:

- Voll automatisierter Risikotransfer auf eine Versicherung: Sowohl die Entscheidungen über die Versicherungsnachfrage wie auch das Angebot der Versicherungsleistung ist voll automatisiert darstellbar. Jedoch müssen einzelne Konfigurationsparameter der Versicherung speziell an ein Grid-System und dessen Risikokonfiguration angepasst werden.
- Effizientes Verfahren zur Berechnung: Durch die Auswahl des Marginalsummenverfahrens als iterative Berechnungsmethode ist eine effiziente Umsetzung in Algorithmen möglich.
- Abbildung wichtiger Risikofaktoren von Grid-Systemen: Durch die Prämiendifferenzierung mithilfe von Tarifmerkmalen, die wichtigen Risikomerkmale von Grid-Systemen entsprechen, kann eine Gruppierung der Risiken über ihre jeweilige Merkmalsausprägung vorgenommen werden.
- Einbettung in die CATNETS Grid-Ökonomie: Für die Modellierung der Versicherungsnachfrage wurde das bestehende Lernverfahren der CATNETS Grid-Ökonomie erweitert.

Die Implementierung des Modells in das Multi-Agenten-System des CATNETS Grid-Simulators wird im folgenden Kapitel vorgestellt. Dabei werden implementierungsspezifische Details des Modells detailliert besprochen und dokumentiert.

## Kapitel 6

# Simulation einer Grid-Versicherung in einer Grid-Ökonomie

*„ Nature is capable of building [...] systems whose complexity lies far beyond the reach of computers and supercomputers, present or prospective. [...] Modeling, then, calls for some basic principles to manage this complexity. We must separate what is essential from what is dispensable in order to capture in our models a simplified picture of reality. “ [Simo90, S. 7]*

Im vorangegangenen Kapitel wurde die Umsetzung des Grid-Versicherungsmodells vorgestellt und ein Modell zur Integration in die CATNETS Grid-Ökonomie konnte festgelegt werden. Dieses Kapitel analysiert die Möglichkeiten der Evaluierung des Grid-Versicherungsmodells, stellt die Implementierungsdetails vor und definiert die zentralen Fragestellungen der Evaluierung. Die Komplexität des Grid-Modells, die Abhängigkeiten zwischen einzelnen Parametern und das nichtlineare Verhalten machen es schwierig eine analytische Lösung anzugeben. Als Konsequenz daraus wird die Forschungsmethode der Simulation angewandt, die Techniken bietet, die Komplexität des Grid-Modells abzubilden und quantitativ zu evaluieren. Grid-Simulationen imitieren die Grid-Netzwerkinfrastruktur, Grid-Services und -Ressourcen und das Benutzerverhalten durch ein abstraktes Modell. Sie erlauben eine quantitative Analyse des Grid-Versicherungsmodells. Im Fokus des Simulationsmodells steht die Modellierung mithilfe von Softwareagenten, die eine Technologie darstellen, um komplexe betriebswirtschaftliche Zusammenhänge als System vieler interagierender Individuen abbilden zu können [NoMa07, S. 92].

Der Rest des Kapitels ist wie folgt aufgebaut: Abschnitt 6.1 betrachtet die grundlegenden Konzepte der Forschungsmethode Simulation. Ein strukturierter Prozess wird eingeführt, der einen generischen Ablauf einer Simulationsstudie beschreibt und bei der Verifikation und Validierung

der Ergebnisse zur Anwendung kommt. Einen Überblick über die CATNETS Grid-Simulationsumgebung gibt Abschnitt 6.2. Der Szenariogenerator, der Simulator und die Evaluationsumgebung werden vorgestellt und die wichtigsten Konfigurationsparameter beschrieben.

Die Implementierung des Grid-Versicherungsmodells in die CATNETS Grid-Ökonomie beschreibt Kapitel 6.3. Im Fokus stehen der Aufbau des Grid-Versicherungsagenten und die Integration der Grid-Versicherung. Der Simulationsentwurf folgt der Beschreibung der Implementierung in Abschnitt 6.4. Die zu untersuchenden Fragestellungen und Probleme werden definiert und es werden Simulationsszenarien entworfen, mit denen die Fragestellungen untersucht werden. Das Verfahren zur Evaluation der Szenarien ist zweigeteilt: Der erste Teil des Evaluierungsprozesses konzentriert sich auf einzelne Aspekte der Grid-Versicherung, während im zweiten Teil des Evaluierungsprozesses das Ressourcenmanagement des Grid-Systems als Ganzes betrachtet wird. Hierfür kommt eine Metrikenpyramide zum Einsatz, die einen Vergleich verschiedener Systemkonfigurationen mithilfe verschiedener Indikatoren und Indizes ermöglicht. Diese Metrikenpyramide wird in Abschnitt 6.5 detailliert beschrieben. Eine Zusammenfassung des Kapitels findet sich schließlich in Abschnitt 6.6.

## 6.1 Grundlagen von Simulationen

### 6.1.1 Grundlegende Konzepte

Die Simulationen in dieser Arbeit zielen darauf ab, Vorhersagen über das Verhalten des simulierten Modells in einer realen Grid-Umgebung zu machen. Beispielsweise ist es interessant, Effekte zu untersuchen, die sich durch Einführung einer Grid-Versicherung in einer Grid-Ökonomie ergeben. Auf das konkrete Modell der Grid-Versicherung bezogen, wird der Einfluss einer SLA-Verletzung durch Ausfall des Netzwerks auf die Grid-Versicherung und die Leistung des Grid-Systems untersucht. Daraus kann abgeleitet werden, ob der Einsatz der Grid-Versicherung einen positiven Effekt in einem realen Grid-System erwarten lässt. Da es sich im Rahmen einer Simulation nur um eine Abschätzung handelt, muss dieser Effekt durch Anwendung weiterer Forschungsmethoden an realen Systemen überprüft werden, was jedoch über den Rahmen dieser Arbeit hinausgehen würde.

Die Validität einer solchen Vorhersage hängt dabei stark von der Validität des Modells ab [Dool02]. Vorhersagen werden gemacht in Bezug auf bestimmte Änderungen in der Nachfrage nach Grid-Diensten, Netzwerkstruktur, Verteilung der Ressourcen oder Konfiguration, die das

Risikomanagement durch die Grid-Versicherung beeinflussen. Wenn die Vorhersagen wie erwartet eintreffen, ergibt sich daraus eine Motivation, die Grid-Versicherung auch in realen Grid-Systemen einzusetzen.

Für die Modellierung und Umsetzung von Simulationsmodellen gibt es eine Reihe unterschiedlicher Verfahren, die je nach Anwendungsgebiet ausgewählt werden müssen [GiTr05, S. 13]. Die Auswahl einer passenden Modellierungstechnik erfolgt daher auf Grundlage von 3 Charakteristika, mit denen Systeme hinsichtlich ihres Simulationsmodells eingeordnet werden können [Law07, S. 5ff.]:

- **Statisches vs. dynamisches Modell:** Ein statisches Modell repräsentiert ein System zu einem Zeitpunkt; Zeit spielt keine Rolle in diesem Modell. Ein System, das sich über die Zeit entwickelt, wird dynamisches System genannt. Da sowohl Zeitparameter wichtig bei der Beschreibung der Dienstqualität in Utility-Grid-Systemen als auch Risikokennzahlen zeitabhängig sind, wird das zu simulierende Grid-System als dynamisches System eingeordnet.
- **Deterministisches vs. stochastisches Modell:** Wenn ein Modell keine Zufallsprozesse enthält, ist es ein deterministisches System, andernfalls ein stochastisches System. Sowohl die Entscheidungsprozesse im zu untersuchenden Modell als auch die Zuverlässigkeit des Grid-Systems sind als stochastische Prozesse modelliert, sodass hinsichtlich dieses Kriteriums von einem stochastischen Modell ausgegangen wird.
- **Zeitkontinuierliches vs. zeitdiskretes Modell:** Von einem zeitdiskreten Modell spricht man, wenn die Ereignisse, die zu einer Änderung des Systemzustands führen, zu bestimmten Zeitpunkten auftreten. Können Ereignisse zu jeder beliebigen Zeit auftreten, so handelt es sich um ein zeitkontinuierliches Modell. Das Auftreten von Ereignissen ist im vorliegenden Grid-System an den Austausch von Nachrichten geknüpft, die ein zeitdiskretes Ereignis darstellen. Die Meldung eines Schadens an die Grid-Versicherung mithilfe einer Nachricht ist beispielsweise ein zeitdiskretes Ereignis, das eine Veränderung des Zustands der Grid-Versicherung bewirkt.

Diese Kriterien sind zugleich die allgemeinen Anforderungen an die Simulationsumgebung, die erfüllt sein müssen, um das Grid-Modell zu simulieren. Zusätzlich müssen weitere Entscheidungen getroffen werden, um einen passenden Modellierungsansatz auszuwählen, der häufig über die Modelleigenschaften ausgewählt wird. Nach [NoMa07, S. 11ff.] und [Gilb08] bietet sich agentenbasierte Modellierung als Ansatz an, um die Komplexität des Grid-Modells abzubilden. Dies wird anhand der folgenden Kriterien näher untersucht [GiTr05, S. 13]:

- Anzahl der zu simulierenden Ebenen: Die Simulation eines Grid-Systems besteht aus mehreren Ebenen, die in Kapitel 2 und Kapitel 3 vorgestellt wurden. Die Ebenen bestehen aus den grundlegenden Schichten eines Grid-Systems, die sich in Netzwerktopologie, Grid-Services und Grid-Ressourcen sowie Risikomodell untergliedern lassen und aus den Schichten der Grid-Ökonomie, auf denen die handelnden Akteure und die Grid-Versicherung einzuordnen sind. Zusammengefasst besitzt das Grid-Modell dieser Arbeit 2 logische Ebenen mit 5 Schichten, die simuliert werden müssen.
- Kommunikation zwischen den Agenten: Die Kommunikation zwischen den Agenten erfolgt unter Verwendung der Kommunikationsdienste des Grid-Systems, die als Peer-To-Peer-Netzwerk realisiert sind. Die Kommunikationsmöglichkeiten stellen einen essenziellen Bestandteil eines Grid-Systems dar und sind wichtig für die Koordination des Ressourcenmanagements. Aus diesem Grund ist es notwendig, einen Simulationsansatz auszuwählen, der komplexe Kommunikationsbeziehungen abbilden und verwalten kann.
- Komplexität der Agenten: Alle Softwareagenten auf der Ebene der Grid-Ökonomie – dies schließt auch die Grid-Versicherung mit ein – sind als komplex anzunehmen. Dies bedeutet, dass sie nichttriviales Verhalten aufweisen. Sie sind adaptiv und treffen autonom Entscheidungen und können somit zur Steuerung von dynamischen Grid-Systemen eingesetzt werden [FoJK04].
- Anzahl der Agenten: Die Anzahl der Agenten wird als hoch angenommen. Ziel ist es, Simulationen mit einigen hundert bis wenigen tausend Softwareagenten durchzuführen. Wegen der Komplexität des Modells und der beschränkten Rechenleistung ist es derzeit nicht möglich, größere Systeme zu simulieren [BCCC<sup>+</sup>07].

Unter Anwendung der Einordnung von Gilbert [GiTr05] erweist sich die Modellierung mit lernenden Softwareagenten als ein passender Ansatz für die Realisierung des Grid-Versicherungsmodells. Dieser Ansatz kann eine große Anzahl an Ebenen modellieren, komplexe und kommunizierende Agenten abbilden und eine große Anzahl an Agenten simulieren, was eine wichtige Größe für das Funktionieren einer Versicherung ist, da mit zunehmender Anzahl der Versicherungsnehmer das versicherungstechnische Risiko sinkt [ZwEi07, S. 240f.].

### 6.1.2 Prozess zur Durchführung von Simulationen

Die Durchführung einer Simulationsstudie umfasst einen Prozess, der in mehrere Schritte gegliedert ist und in Abbildung 6.1 dargestellt wird. Der strukturierte Ablaufplan unterstützt bei

der Planung, dem Entwurf, der Implementierung und der Analyse von Simulationen. Die folgende Prozessbeschreibung konzentriert sich auf die 6 wichtigsten Schritte [Law07, S. 67ff.]:

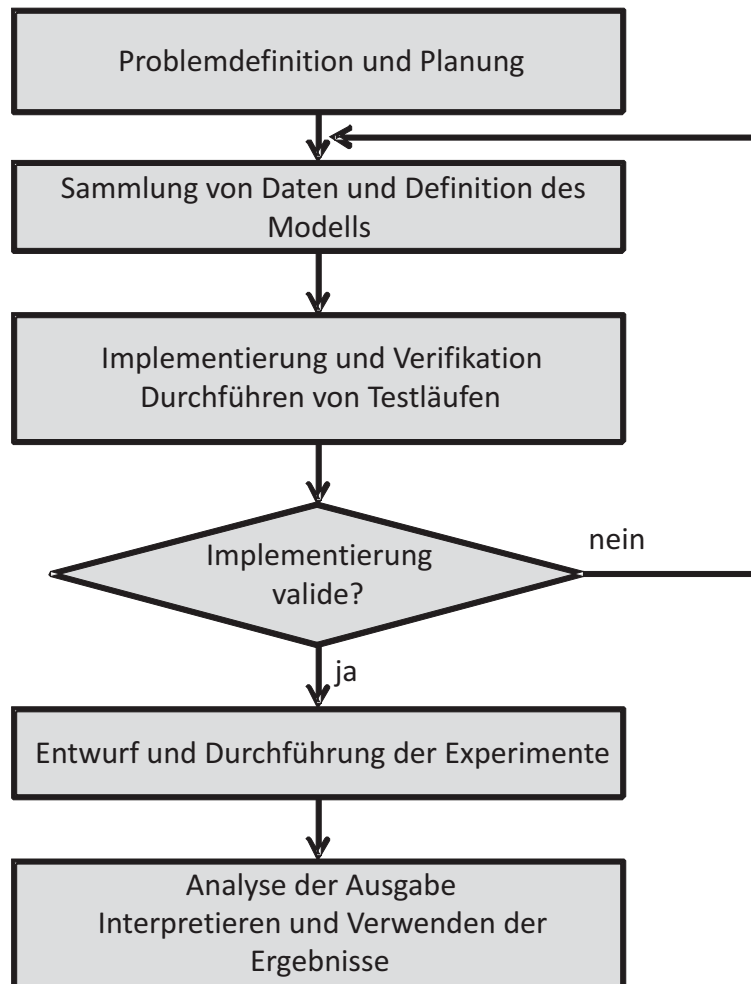


Abbildung 6.1: Allgemeiner Simulationsprozess zur Durchführung einer Simulationsstudie nach Law [Law07, S. 67].

- **Problemformulierung und Planung der Simulationsstudie:** Im ersten Schritt wird das Problem festgelegt, das von Interesse ist und allgemeine Ziele der Simulation definiert [Law07]. Das Problem ist die Modellierung einer Grid-Versicherung als integrierte Risikomanagementkomponente in einer Grid-Ökonomie wie sie in den vorangegangenen Kapiteln eingeführt wurde. Die Grid-Ökonomie eines Utility-Grid-Systems dient zugleich als Integrationsplattform des Versicherungsmodells und legt die Systemkonfiguration fest. In dieser Phase wird auch die Simulationssoftware festgelegt, die zum Einsatz kommen soll.

Nach eingehender Prüfung in den vorangegangenen Abschnitten eignet sich die Forschungs-

methode Simulation für das zu evaluierende Grid-Versicherungsmodell, sodass eine Auswahl der Simulationssoftware erfolgen kann. Aus einer Reihe von speziellen Grid-Simulatoren und allgemeinen Simulatoren für Multi-Agenten-Systeme wurde für die Simulationsstudie in dieser Arbeit der CATNETS Grid-Simulator ausgewählt [ERSZ<sup>+</sup>05], der in Sektion 6.2 vorgestellt wird. Dieser bietet eine Reihe zusätzlicher Softwareanwendungen zur Erzeugung von Szenarien und Evaluierung der Ergebnisse. Weiterhin zeigt er gute Skalierbarkeit und hohe Anpassungsfähigkeit an neue Anforderungen. Mithilfe dieser Simulationsumgebung werden verschiedenen Konfigurationen der Grid-Versicherung modelliert und die Effekte der Grid-Versicherung analysiert.

- **Sammlung von Daten des zu untersuchenden Systems und Modelldefinition:** Die Sammlung von Daten über das zu untersuchende System und die anschließende Modelldefinition ist der zweite Schritt im Simulationsprozess. Aus der Datensammlung werden die Eingangsverteilungen des Simulationsmodells festgelegt. Dies wurde durch Literaturrecherche zu dem technischen Risiko in Grid-Systemen durchgeführt. Wichtige Parameter wurden identifiziert und durch Verteilungen quantifiziert. Durch die Eingrenzung auf zwei wichtige Risikoquellen von Grid-Systemen wird eine Abstraktion vom realen Grid-System vorgenommen, um die Komplexität der Modelldefinition zu reduzieren.

Das Versicherungsmodell selbst definiert die wichtigen Eigenschaften einer realen Versicherung wie die Verwaltung einer Schadensstatistik, die Berechnung einer Prämie und die Abwicklung eines Schadensfalls. Komplexe Charakteristika einer Versicherungsökonomie können nicht abgebildet werden, sodass das Grid-Versicherungsmodell die Abstraktion einer realen Versicherung darstellt. Es wird von einem einfachen Modell ausgegangen, das bei Bedarf um weitere Funktionen erweitert werden kann, um die Komplexität der Simulation möglichst klein zu halten und dadurch die Skalierbarkeit der Simulationsumgebung zu erhalten. Durch verschiedene Simulationsszenarien werden unterschiedliche Aspekte des Modells betrachtet, wobei *ceteris paribus* alle anderen Parameter des Modells festgehalten werden.

- **Implementierung eines Programms und Verifikation:** Der dritte Schritt fokussiert auf die Implementierung des Modells in eine Simulationsumgebung und ihre Verifikation. Das Modell der Grid-Versicherung wird als Softwareagent implementiert und in den CATNETS Grid-Simulator als Simulationsplattform integriert. Details der Modellimplementierung werden in Sektion 6.3 vorgestellt und mithilfe von UML-Diagrammen dokumentiert [Obj07].



Wichtiger Bestandteil der Implementierung ist die Verifikation. Die Verifikation ist ein Prozess, der Techniken aus der Softwareentwicklung einsetzt, um die Korrektheit der Implementierung gegenüber der Modellspezifikation zu überprüfen [Law07, S. 248ff.] [NoMa07, S. 222]. Für die Verifikation im Rahmen dieser Arbeit kommen verschiedene Techniken zum Einsatz.

Zum Test der Integration des Modells wurde die Grid-Versicherung in einem ersten Implementierungsschritt als *Dummy* umgesetzt, der noch keine Algorithmen zur Berechnung enthält. Dies hat den Vorteil, dass Teile der Implementierung wie die Verarbeitung von Nachrichten getestet werden können, ohne bereits die Grid-Versicherung vollständig implementiert haben zu müssen. Dadurch können frühzeitig Fehler erkannt und beseitigt werden.

Ist das Modell vollständig umgesetzt, kommt die Technik des strukturierten Durchlaufs zum Einsatz. Der Quellcode wird Zeile für Zeile durchgegangen und es wird nach einfachen Semantikfehlern der Implementierung gesucht. Zusätzlich werden die Berechnung der Prämie und die Aktualisierung der Prämie nach Ablauf einer Versicherungsperiode als wichtige Funktionen der Versicherung einem Unit-Test [BeGa00] unterzogen, der die Korrektheit der Berechnung überprüft. Diesem Verfahren schließen sich Testläufe des vollständigen Simulationsmodells mit kleineren Szenarien an, die anhand ihres Ablaufplans analysiert werden. Hierfür werden an wichtigen Stellen im Quellcode der Zustand des Systems aufgezeichnet und die Messdaten manuell nach Ablauf der Simulation überprüft. Dieser Vorgang wird an verschiedenen Testszenarien durchgeführt. Am Ende dieser aufwändigen Untersuchung kann das Grid-Versicherungsmodell als verifiziert gelten, wobei eine vollständig korrekte Verifizierung nicht möglich ist, da es keinen zu 100% fehlerfreien Code geben kann [NoMa07, S. 222].

- Überprüfung der Validität des implementierten Modells: Die Validierung bezieht sich darauf, ob ein Modell das reale System korrekt repräsentiert und reproduziert [NoMa07, S. 226]. Der Idealfall einer Validierung bestünde darin, das Verhalten eines realen Grid-Systems mit dem simulierten Grid-System zu vergleichen. Jedoch ist dieser Ansatz nicht anwendbar, da es bisher noch kein Grid-System im Produktiveinsatz gibt, das eine Grid-Ökonomie für das Ressourcenmanagement einsetzt. Deshalb müssen alternative Verfahren herangezogen werden, um die Glaubwürdigkeit des Modells zu erhöhen.

Das grundlegende Modell der CATNETS Grid-Ökonomie kann als teilweise validiert gelten, da dieses zum einen mit Ergebnissen eines Prototypen überprüft wurde [CBFN<sup>+</sup>08]



[SEVC<sup>+</sup>07] und zum anderen die Ergebnisse mit einer zweiten Implementierung in einer weiteren Simulationsumgebung bestätigt wurde [ASEB07], das nach Axtell et al. [AAEC96] und Edmonds et al. [EdHa03] ein gültiges Validierungsverfahren darstellt.

Für die Validierung der Grid-Versicherung kommen zwei Verfahren zur Erhöhung der Glaubwürdigkeit zum Einsatz, die keine Validierung mit einem realen Grid-System durchführen, sondern nur eine Erhöhung der Glaubwürdigkeit zur Folge haben [NoMa07, S. 229]: Sensitivitätsanalyse und Vergleich von zwei oder mehr Modellen.

Für die Sensitivitätsanalyse werden diejenigen Werte identifiziert, die signifikanten Einfluss auf die Performanz des Systems haben und aus diesem Grund besonders sorgfältig modelliert werden müssen. Die Wertebereiche der Schlüsselwerte werden verändert und so interessante Konfigurationen des Grid-Systems und der Grid-Versicherung ermittelt. Die Vorgehensweise erfolgt durch Ausprobieren und Bewertung der Parameteränderungen. Kandidaten für die Sensitivitätsanalyse sind Parameter der Grid-Versicherung wie Versicherungsperiode oder die Festlegung der Tarifgruppen sowie Umgebungsparameter, die das Risiko signifikant beeinflussen. Ein Teil der Sensitivitätsanalyse ist die Untersuchung der Grid-Versicherung unter Extrembedingungen. Kommt es bei einer starken Erhöhung der Risiken zu dem erwarteten Ergebnis, so bestätigt sich die Glaubwürdigkeit des Grid-Versicherungsmodells. Kein Ressourcenkonsument sollte in diesem Fall mehr eine Versicherung abschließen können, da die Versicherungsprämien zu teuer sind.

Die Validierung durch den Vergleich von zwei oder mehreren Modellen, die gegenseitig evaluiert werden, erhöht ebenfalls die Glaubwürdigkeit [NoMa07, S. 229]. Nach Hales et al. [HaRE03] kann der Vergleich durch unterschiedliche Verfahren durchgeführt werden. In dieser Arbeit wird ein Vergleich verschiedener Modelle durchgeführt, um zu sehen, welches der Modelle hinsichtlich verschiedener Performanzkennzahlen bessere Ergebnisse produziert. Im konkreten Fall der Grid-Versicherung werden dieses Modell einer Risikotransferstrategie mit der Risikoakzeptanzstrategie verglichen und verschiedene Grid-Versicherungskonfigurationen untereinander.

- Entwurf der Experimente und Durchführung der (produktiven) Simulationsläufe: Nach der Verifikation und Validierung des Modells erfolgen der Entwurf und die Durchführung der Experimente. Beim Entwurf eines Simulationsszenarios müssen alle interessanten Werte der Systemkonfiguration eingestellt und dokumentiert werden. Dies umfasst auch die Länge der Simulation und die Anzahl der Simulationsläufe, die durchgeführt werden, um eine ausreichende Datenbasis für die Analyse zu erhalten. Dies ist deshalb wichtig, da die

Ergebnisdaten von stochastischen Prozessen – unter Verwendung von Zufallszahlen – beeinflusst werden. Die einzelnen Experimente orientieren sich an dem zu untersuchenden Teilausschnitt des Modells und werden im diesem und dem folgenden Kapitel vorgestellt.

- Analyse der Ausgabedaten: Sind die Simulationen erfolgreich durchgeführt worden, so müssen die Ausgabedaten analysiert werden. Dabei werden zwei allgemeine Ziele verfolgt. Zum einen wird die absolute Performanz des Systems einer bestimmten Systemkonfiguration bestimmt und zum anderen wird eine alternative Systemkonfiguration in relativer Beziehung zu einer weiteren Systemkonfiguration verglichen. Die Analyse erfolgt unter Anwendung statistischer Methoden wie der Berechnung von Mittelwerten und Varianzen und der Analyse von Zeitserien.

Die ersten beiden Prozessschritte wurden in den vorangegangenen Kapiteln behandelt. Alle weiteren Schritte werden in diesem und dem nächsten Kapitel durchgeführt. Konzentriert sich dieses Kapitel auf die Implementierung, Verifikation und Validierung, so stehen die Experimente und ihre Analyse im folgenden Kapitel im Vordergrund.

## 6.2 Der CATNETS Grid-Simulator

Das CATNETS Grid-Simulator ist ein Simulator von diskreten Ereignissen, der auf den Data-Grid-Simulator OptorSim aufsetzt [BCMC<sup>+</sup>03]. Das Data-Grid-Modell des OptorSim-Simulators [CCSMN<sup>+</sup>04] wurde im Rahmen des CATNETS Projekts<sup>1</sup> um das Grid-Ökonomiemodell aus Kapitel 3 erweitert. Als Besonderheit bietet OptorSim eine Beschleunigung der Simulationen mithilfe eines Zeitsprungmechanismus. Dieser prüft während einer Simulationsausführung, ob zum aktuellen Zeitpunkt eine Aktion ausgeführt werden soll oder nicht und springt, wenn keine Aktion ausgeführt werden soll, zum nächsten Zeitpunkt, bei dem ein Ereignis eintritt. Dies erhöht die Simulationsgeschwindigkeit um etwa den Faktor 10 und trägt damit signifikant zu einer Beschleunigung der Simulationsläufe bei. Neben dem eigentlichen Simulator selbst besteht die CATNETS Simulationsplattform aus weiteren Komponenten, die im folgenden Abschnitt eingeführt werden.

---

<sup>1</sup><http://www.catnets.eu> (Letzter Abruf: 04.09.2008)

### 6.2.1 Aufbau der Simulationsumgebung

Die CATNETS Simulationsumgebung besteht aus drei Komponenten, die Abbildung 6.2 zusammen mit den Eingabe- und Ausgabedaten der Komponenten zeigt. Der Szenariogenerator unterstützt bei der Planung eines Simulationsszenarios, indem er Grid-Systeme mit mehreren tausend Grid-Serviceanbietern und Grid-Ressourcenanbietern erzeugen kann. Ein solches Szenario wird vom Simulator eingelesen und simuliert. Dabei werden vordefinierte Metriken gemessen und gespeichert, um in einer ex-post-Analyse für die Evaluationskomponente verfügbar zu sein. Die Evaluationskomponente wendet auf den Messdaten statistische Verfahren an und bereitet die Daten grafisch auf, was die Analyse der Messdaten erleichtert.

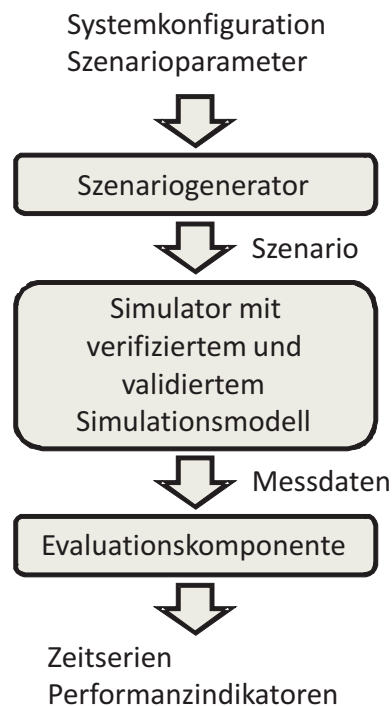


Abbildung 6.2: Die drei wichtigsten Komponenten der CATNETS Simulationsumgebung [CSSE<sup>+</sup>06]: der Szenariogenerator zur Erzeugung von Szenarien, der Simulator zur Simulation der Grid-Versicherung und die Evaluationskomponente zur Analyse der Messdaten.

**Szenariogenerator** Der Szenariogenerator hat die Aufgabe, Szenarien des in Kapitel 3 eingeführten Grid-Modells zu generieren. Da es sich um ein automatisiert erzeugtes Modell handelt, ist dieses per Definition nicht validiert. Jedoch kann durch die richtige Wahl der Parameter und Eingangsverteilungen verschiedene Charakteristika realer Grid-Systeme nachgebildet werden. Dadurch kann ein Modell erreicht werden, das Ähnlichkeiten mit realen Grid-Systemen

ausweist. Beispielsweise können die Größe des Grid-Systems, die Ressourcenmenge und deren Verteilung so gewählt werden, dass sie realen Grid-Systemen ähneln. Zusätzlich garantiert der Szenariogenerator, dass jeder Grid-Service die von ihm benötigte Ressource mindestens einmal im erzeugten Grid-System vorfindet.

Das automatisiert erzeugte Grid-Systemmodell umfasst die Netzwerktopologie und die verfügbaren Grid-Services und Grid-Ressourcen. Die Konfiguration der Grid-Ökonomie erfolgt manuell über Konfigurationsdateien. Im Einzelnen werden folgende Bestandteile automatisiert erzeugt [CSSE<sup>+</sup>06] [SEZS<sup>+</sup>07]:

- **Netzwerktopologie:** Im ersten Schritt werden eine vorgegebene Anzahl an Netzwerkknoten erzeugt und die Verbindungen zwischen den Netzwerkknoten festgelegt. Für die Erzeugung des Grid-Netzwerks kommt das Netzwerkmodell von Waxman [Waxm88] des BRITE<sup>2</sup> Netzwerkgenerators zum Einsatz [MLMB01a]. Jeder Knoten bekommt eine statische Ausfallwahrscheinlichkeit zugewiesen, die zwei Funktionen erfüllt: Zum einen bestimmt sie die Zuverlässigkeit des Grid-Knotens und zum anderen beeinflusst sie die Zuverlässigkeit der Kommunikation zwischen den Grid-Knoten. Sie wirkt im letzteren Fall wie ein Ausfall aller ausgehenden Netzwerkverbindungen. Den Verbindungen zwischen den Knoten wird eine Bandbreite zugewiesen, deren Kapazität begrenzt ist. Ist die Bandbreite einer Verbindung erschöpft, muss eine Nachricht unter Umständen einen längeren Weg bis zum Zielknoten im Netzwerk nehmen.
- **Erzeugung der Grid-Services und Grid-Ressourcen:** Parallel zur Erzeugung der Netzwerktopologie werden die Grid-Services und Grid-Ressourcen angelegt. In diesem Schritt werden die Anzahl und die Menge der Grid-Services und Grid-Ressourcen festgelegt, und der Workflow definiert, der eine statische Abbildungstabelle zwischen Grid-Services auf Grid-Ressourcen darstellt.
- **Zuweisung der Grid-Services und Grid-Ressourcen zu Grid-Knoten:** Abschließend werden die Grid-Service und Grid-Ressourcen auf die Netzwerkknoten verteilt. Erhält ein Netzwerkknoten einen Grid-Service zugewiesen, so wird er zum Grid-Standort, andernfalls ist der Knoten ein Router, der Nachrichten an benachbarte Knoten weiterleitet.

Die Grid-Versicherung wird nicht automatisch einem Knoten im Grid-Netzwerk zugewiesen. Sie muss manuell durch Editieren der Szenariokonfiguration hinzugefügt werden. Dabei wird ein

---

<sup>2</sup>BRITE steht für Boston University Representative Internet Topology Generator und ist ein flexibler Generator für Netzwerktopologien. Für weitere Informationen wird an dieser Stelle auf die Dokumentation von BRITE verwiesen [MLMB01b].

freier Knoten ausgewählt, der keine anderen Ressourcen enthält und als Versicherungsknoten deklariert.

**Simulator** Der CATNETS Grid-Simulator benutzt das automatisiert erzeugte Szenario des Szenariogenerators, um einen Simulationslauf zu erstellen. Die Erstellung eines Simulationslaufs erfolgt in zwei Schritten: Zuerst wird das Netzwerk aufgebaut und anschließend werden die Softwareagenten erstellt. Die Softwareagenten werden mit weiteren Parametern initialisiert, die zusätzlich für die Konfiguration des Simulators notwendig sind. Hierzu gehören die verschiedenen Einstellungen der verwendeten Algorithmen wie beispielsweise das Marginalsummenverfahren der Grid-Versicherung und weitere allgemeine Parameter, von denen die wichtigsten im Abschnitt zur Konfiguration des Simulator näher erläutert werden. Eine vollständige Übersicht der Parameter findet sich in der Dokumentation des Simulators [SEZS<sup>+</sup>07] [CSSE<sup>+</sup>06].

Die Architektur des Simulators zeigt Abbildung 6.3. Wie in Kapitel 3 detailliert beschrieben, benötigt ein Complex-Service (CS) einen oder mehrere Basic-Services (BS), die wiederum einen Resource-Service (RS) für die Ausführung ihres Service anfordern. Jede Anfrage nach Complex-Services wird von einem simulierten Grid-Benutzer gestartet, der in bestimmten Zeitintervallen Anfragen nach Complex-Services stellt. Werden die Zeitintervalle verändert, können unterschiedliche Lastzustände des Grid-Systems modelliert werden.

Für die Modellierung der Anfragen eines Grid-Benutzers stehen verschiedene Modelle zur Auswahl, die über einen Parameter des Simulators eingestellt werden können. Wie in Grid-Systemen üblich, werden die Benutzeranfragen nicht sofort an die entsprechenden Complex-Services der einzelnen Grid-Knoten weitergeleitet, sondern über einen Verteiler – hier Complex-Service-Dispatcher genannt – zugewiesen. Die Aufgabe des Complex-Service-Dispatchers besteht darin, einen Complex-Service für die Bearbeitung der Benutzeranfrage auszusuchen und die Anfrage an den ausgewählten Complex-Service weiterzuleiten. Hierfür verwendet der Complex-Service-Dispatcher Regeln, die meist die aktuelle Auslastung eines Complex-Service mit einbeziehen. Empfängt ein Complex-Service schließlich eine Benutzeranfrage, so startet er den in Kapitel 3 beschriebenen Allokationsprozess.

Bis auf den Grid-Benutzer, der aktiv neue Anfragen an das System stellt, sind alle anderen Teilnehmer reaktiv. Dies bedeutet, sie übernehmen nicht selbst die Initiative und bieten beispielsweise ihre Grid-Ressourcen aktiv an, sondern warten auf Anfragen, die als Nachrichten übermittelt werden. Auf diese Nachrichten reagieren die betreffenden Softwareagenten durch eine Antwortnachricht.

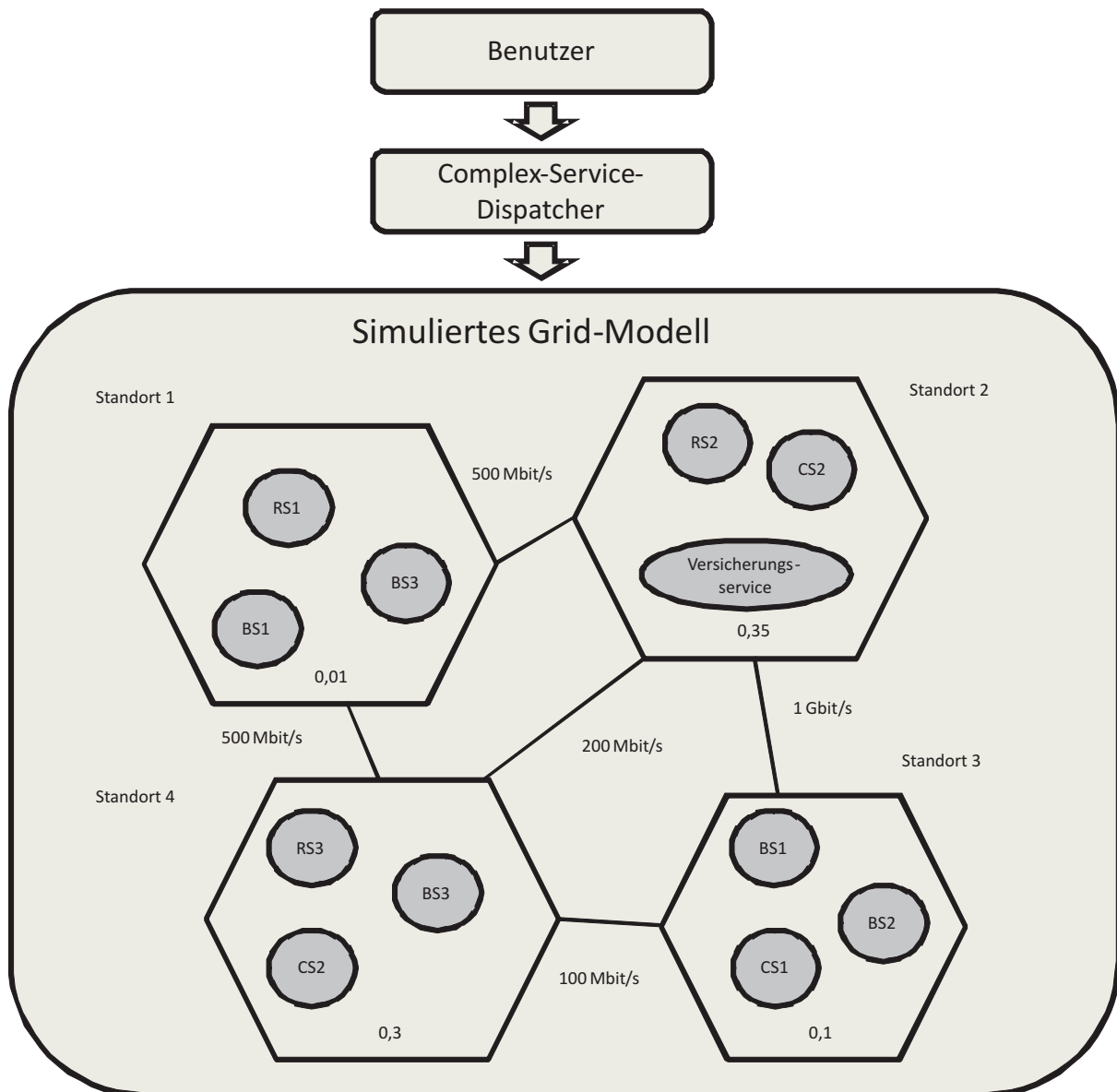


Abbildung 6.3: Überblick über die Architektur des CATNETS Grid-Simulators mit einem Grid-Benutzer und einer Grid-Versicherung [CSSE<sup>+</sup>06].

Die Messung der Metriken erfolgt in den entsprechenden Komponenten des Simulationsmodells. Für jede Metrik, die gemessen wird, wird eine Textdatei definiert, in die die gemessenen Daten während eines Simulationslaufs geschrieben werden. Für die Analyse von Zeitreihen werden einzelne Ereignisse protokolliert, die es ermöglichen, die betreffende Metrik als Funktion über die Zeit grafisch darzustellen. Die Analyse von Performanzindikatoren erfolgt auf aggregierten Werten, die während der Simulation berechnet werden und vor Abschluss der Simulation ebenfalls in Textdateien gespeichert werden.

**Evaluationskomponente** Die Evaluationskomponente besteht aus Analyseskripten, die eine ex-post-Analyse der Messdaten vornehmen. Als Evaluationssoftware kommt MATLAB<sup>3</sup> zum Einsatz, das eine Reihe von verschiedenen Möglichkeiten zur grafischen Analyse bietet.

Die gespeicherten Ergebnisse eines oder mehrerer Simulationsläufe werden durch die Evaluationskripte eingelesen, die die Daten automatisiert auswerten und grafisch aufbereiten. Insofern die Daten schon während des Simulationslaufs vorhanden sind, ist auch eine Auswertung während einer laufenden Simulation möglich. Dies hilft, fehlerhafte Einstellungen vorzeitig zu erkennen und den Simulationslauf gegebenenfalls abubrechen.

## 6.2.2 Konfiguration des Simulators

Dieser Abschnitt stellt eine Auswahl der Konfigurationsdateien und Konfigurationsparameter des Simulators vor. Die Auswahl der Konfigurationsparameter erfolgt nach der Bedeutung für den Simulationsentwurf in Sektion 6.4.

Folgende Konfigurationsdateien stehen im Simulator zur Verfügung:

- `parameters.conf`: Dies ist die zentrale Konfigurationsdatei des Simulators. Hier werden das zu simulierende Grid-Szenario festlegt und die Grid-Benutzereinstellungen vorgenommen. Die Parameter `number.complexservices`, der die Gesamtanzahl der Anfragen festlegt und damit die Simulationslänge definiert und `cs.delay`, der die Verzögerung zwischen zwei Benutzeranfragen in Millisekunden definiert, modellieren den Grid-Benutzer. Eine niedriger Wert von `cs.delay` hat eine hohe Systemlast zur Folge, während ein hoher Wert zu einer geringen Systemlast führt.

---

<sup>3</sup>MATLAB ist ein kommerzielles Softwarepaket zur Durchführung von interaktiven Berechnungen, Datenanalysen und Visualisierung von Daten. Weitere Informationen finden sich auf den Webseiten des Hersteller unter <http://www.mathworks.de/products/matlab/> (Letzter Abruf: 15.05.2008).



- `market.properties`: Diese Konfigurationsdatei definiert die Produkte, die auf dem Service- und Ressourcenmarkt gehandelt werden können und legt die Reservationspreise für die Produkte fest. Das Attribut `hard.lower.limit` setzt den Reservationspreis des Ressourcenanbieters fest und `hard.upper.limit` initialisiert den Reservationspreis bzw. das Budget des Ressourcenkonsumenten. Zusätzlich werden für einen Basic-Service die Ausführungszeit in Millisekunden und die Datenmenge, die für die Ausführung des Service übertragen werden muss, in Kilobyte definiert. Diese zusätzlichen Parameter stellen eine Erweiterung des ursprünglichen Modells dar und ermöglichen eine flexible Simulation der Ausführungsphase eines Basic-Service. Damit wird die Anforderung der Simulation einer Ausführungsphase, wie sie in Kapitel 3 definiert wurde, erfüllt.
- `learning.conf`: In dieser Datei wird die Konfiguration des Lernalgorithmus vorgenommen, der in Kapitel 3 eingeführt wurde. Ein wichtiger Wert ist die Wahrscheinlichkeit, dass ein Gen für einen Crossover herangezogen wird (Parameter `crossoverProbability`). Ein weiterer bedeutsamer Wert besteht in der Wahrscheinlichkeit, dass ein Gen einer Mutation unterzogen wird (Parameter `maturityThreshold`).
- `insurance.conf`: Alle zentralen Konfigurationsparameter der Grid-Versicherung werden in der Datei `insurance.conf` definiert. Die Konfigurationsparameter sind:
  - Versicherungsperiode: Dieser Konfigurationsparameter legt die Versicherungsperiode in Millisekunden fest.
  - Sicherheitszuschlag: Mit diesem Parameter kann man die Höhe des Sicherheitszuschlags festlegen. Dieser ist während eines Simulationslaufs konstant.
  - Anfangsbudget: Das Anfangsbudget stellt die initiale Kapitalausstattung der Versicherung dar. Es wird benötigt, um Schäden, die zu Beginn der Simulation auftreten, zu bezahlen. Damit wird ein Ruin der Versicherung zu Beginn der Simulation verhindert.

Die Initialisierung der Tariffaktoren wird aus Gründen der Vereinfachung direkt im Quellcode der Versicherung vorgenommen. Deshalb wird die Initialisierung der Grid-Versicherung im Rahmen der Dokumentation der Modellimplementierung im folgenden Abschnitt vorgestellt.

Für die Festlegung der Parameter in den ersten drei Konfigurationsdateien werden die Ergebnisse des CATNETS Projekts herangezogen, um ein stabiles und aussagekräftiges Grid-System festzulegen. An den entsprechenden Stellen wird auf die Arbeiten des CATNETS Projekts verwiesen



[BCCC<sup>+</sup>07].

## 6.3 Modellimplementierung

### 6.3.1 Implementierung des Grid-Versicherungsmodells

Die Implementierung des Grid-Versicherungsmodells in den CATNETS Grid-Simulator erfolgt als Softwareagent. Abbildung 6.4 zeigt die Implementierung der Grid-Versicherung als UML-Klassendiagramm<sup>4</sup>. Die Klasse `InsuranceAgent` ist die zentrale Klasse des Grid-Versicherungsagenten. Sie stellt eine Schnittstelle zur Kommunikation für Versicherungsnehmer bereit. Eine weitere wichtige Aufgabe des Versicherungsagenten besteht in der Überwachung der Versicherungsperiode, um bei Ablauf eine Neuberechnung der Versicherungsprämien zu veranlassen. Die Berechnungen des Grid-Versicherungsservice finden insgesamt in der Implementierung des konkreten Versicherungsmodells statt, die im Versicherungsagenten festgelegt wird.

Das Versicherungsmodell realisiert die Klasse `InsuranceModel`. Diese Klasse definiert die Tarifstruktur und speichert alle relevanten Daten, die zur Berechnung der Versicherungsprämien notwendig sind. Neben den Daten sind auch die Berechnungen der Versicherungsprämie in dieser Klasse zu finden. Die wichtigsten Methoden der Klasse sind:

- **Aufbau der Datenstruktur (`buildDataStructure`):** Diese Methode wird bei Erzeugung des Versicherungsagenten zu Beginn eines Simulationslauf aufgerufen und baut die Datenstruktur auf, um den in Kapitel 5 vorgestellten Tarif mit 2 Merkmalen und 3 bzw. 2 Tarifklassen verarbeiten zu können. Im Detail werden die Indexstruktur und die Wertebereiche der Tarifklassen festgelegt.
- **Initialisierung der Datenstruktur (`initDataStructure`):** Die Initialisierung der Datenstruktur erfolgt ebenfalls beim Erzeugen des Versicherungsagenten zu Beginn der Simulation. Hier werden die Konfigurationsparameter der Versicherung verarbeitet und die entsprechenden Variablen vorbelegt. Für eine korrekte erste Berechnung der Tariffaktoren durch das Marginalsummenverfahren werden die Werte der Tariffaktoren mit 1 initialisiert und jedes Feld der Schadensmatrix wird mit Versicherungsnehmern vorbelegt, die noch keinen Schaden erzeugt haben.

---

<sup>4</sup>Die Unified Modeling Language (UML) bietet verschiedenen Varianten zur Modellierung und Visualisierung von Zusammenhängen unabhängig von der jeweiligen Programmiersprache [Obj07].

- Aktualisierung der Tariffaktoren (`updateTarifFactors`): Die Methode zur Aktualisierung der Tariffaktoren implementiert das Marginalsummenverfahren, wie es in Kapitel 5 vorgestellt wurde. Es werden die Zeilen- und Spaltensummen der Schadensmatrix berechnet und die Tariffaktoren für die nächste Periode gesetzt.
- Bestimmung der Prämie (`getPremium`): Die Bestimmung der Prämie für gegebenes Risikoprofil implementiert die Methode `getPremium`. Für gegebenes Risiko eines Tarifmerkmals werden die Tarifklasse bestimmt und die Tariffaktoren multipliziert. Am Ende der Berechnung wird der Sicherheitszuschlag auf die Prämie addiert und das Ergebnis über den Versicherungsagenten an den Versicherungsnehmer gemeldet.
- Einzahlung der Prämie (`setPremium`): Diese Methode verbucht die Prämieeinnahme, indem sie die eingenommene Prämie auf die Gesamtprämien-summe aufaddiert. Es wird außerdem der Bezeichner des Agenten gespeichert, um einen Schaden einem Agenten zuzuordnen zu können.
- Ausbezahlung des Schadens (`getCompensationForClaim`): Die Ausbezahlung des Schadens erfolgt durch die Methode `getCompensationForClaim`. Von der Gesamtprämien-summe wird die Schadenssumme abgezogen und an den Versicherungsnehmer übermittelt.

Die letzte Klasse der Versicherungsagentenimplementierung (`InsuranceConstants`) enthält Konstanten, die zur internen Konfiguration und Umsetzung des Grid-Versicherungsmodells notwendig sind. Aus Gründen der Übersichtlichkeit werden diese Werte in einer eigenen Klasse gehalten.

**Nachrichten zur Kommunikation mit der Grid-Versicherung** Für die Kommunikation mit der Versicherung stehen 4 Nachrichtentypen zur Verfügung, die in Abbildung 6.5 gezeigt werden. Alle Nachrichten sind Unterklassen einer speziellen `InsurancePoint2PointMessage`, die eine Punkt-zu-Punkt-Kommunikation zwischen dem Versicherungsnehmer und der Versicherung implementiert. Drei Nachrichtentypen sind für die Kommunikation zwischen Versicherungsnehmer und Versicherung konzipiert. Ein Nachrichtentyp dient als Antwortnachricht der Grid-Versicherung. Ein Versicherungsnachfrager kann mithilfe einer `InsuranceRequestMessage` eine Prämienanfrage an die Versicherung schicken. Die Nachricht `InsurancePayMessage` übermittelt die Prämienbezahlung an die Versicherung und die Nachricht `InsuranceClaimMessage` übermittelt eine Schadensmeldung. Diese Nachrichten werden vom Versicherungsagenten mit seinen Methoden zur Verarbeitung von Nachrichten (z. B. `handleRequestMessage` für eine

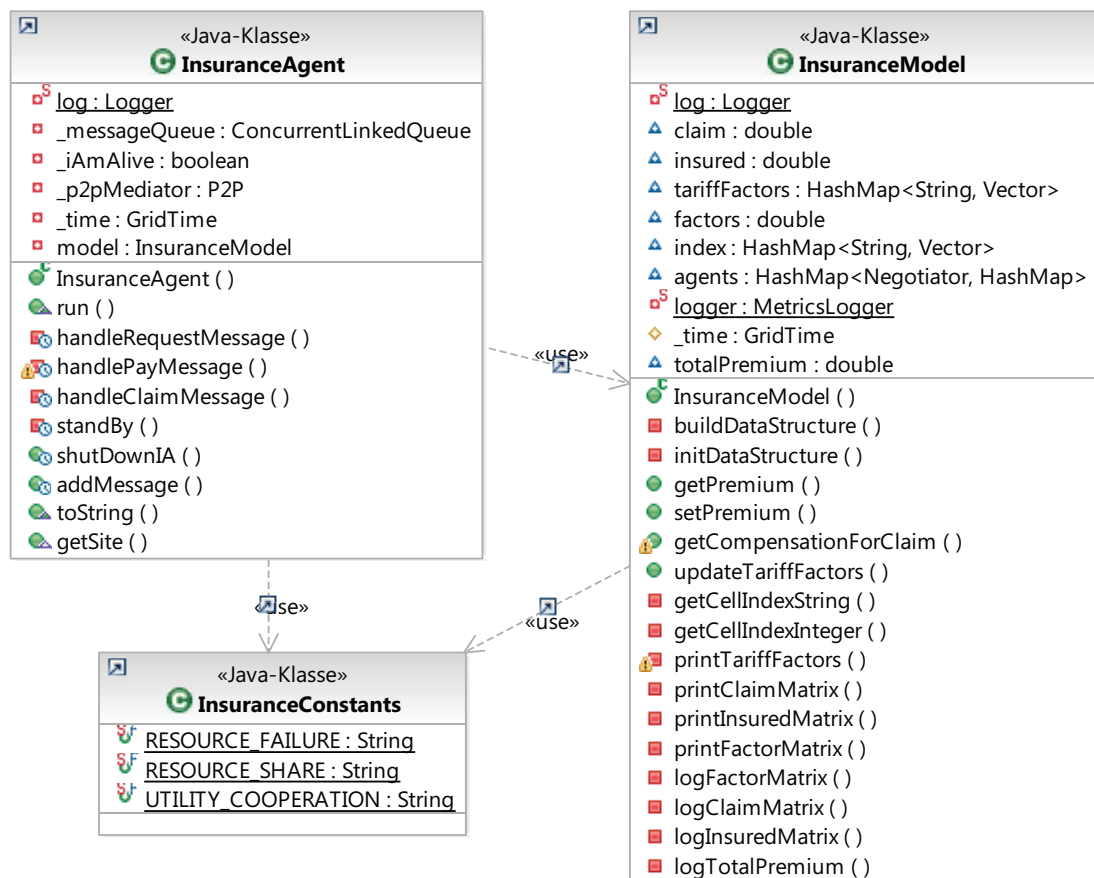


Abbildung 6.4: Implementierung des Grid-Versicherungsagenten als UML-Klassendiagramm

InsuranceRequestMessage) behandelt. Als Antwortnachricht empfängt ein Versicherungsnehmer immer eine InsuranceNotificationMessage.

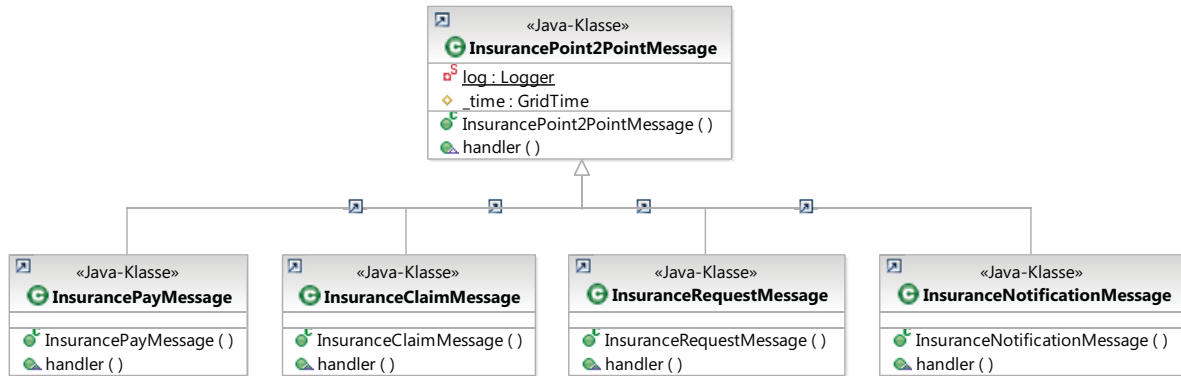


Abbildung 6.5: Verfügbare Nachrichten zur Kommunikation mit der Grid-Versicherung

Eine Besonderheit der Implementierung der Versicherungsnachfrage besteht darin, dass diese wie eine Verhandlung aufgefasst wird. Dies hängt von der Art des implementierten Simulationsmodells ab. Abbildung 6.6 zeigt die Versicherungsverhandlung, die sowohl die Kommunikation mit der Grid-Versicherung eines Versicherungsnehmers regelt als auch die Entscheidungen über die Annahme oder Ablehnung einer Versicherungsprämie trifft.

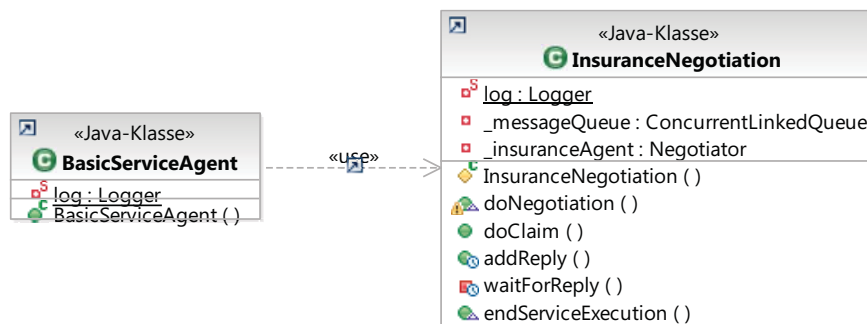


Abbildung 6.6: Die Klasse InsuranceNegotiation zur Kommunikation des Versicherungsnehmers (Basic-Service) mit der Grid-Versicherung

Die Methode doNegotiaton startet die Anfrage nach der Versicherungsprämie und trifft die Entscheidung über Annahme oder Ablehnung, während die Methode doClaim die Abwicklung der Schadenszahlung implementiert.

Der unten stehende Auszug aus der Implementierung der Methode doNegotiaton zeigt die Implementierung der Entscheidung über den Abschluss einer Versicherung. In dieser Implementie-

rung überprüft der Versicherungsnehmer zuerst, ob die zu bezahlende Prämie (`premium`) kleiner als der Nutzen aus der Transaktion (`utility`) ist. Ist die Versicherungsprämie bezahlbar, berechnet er den Erwartungsnutzen, den er bei Abschluss einer Versicherung hätte, unter Zuhilfenahme seiner persönlichen Risikoeinschätzung (`riskEstimation`). Andernfalls schließt er keine Versicherung ab, da die Versicherungsprämie den erwarteten Nutzen übersteigt.

```
//if the price of the insurance is cheaper than the utility
if (premium < utility) {
//insurance affordable
    withInsurance = (utility - premium) * riskEstimation +
        (utility - premium) * (1 - riskEstimation);
} else {
//insurance too expensive
}
```

Als Nächstes vergleicht der Agent die Erwartungsnutzen der beiden Alternativen, einmal mit Versicherung (`withInsurance`) und einmal ohne Versicherung (`withoutInsurance`). Wenn der Erwartungsnutzen mit Versicherung größer ist als ohne, erzeugt er eine Bezahlnachricht an die Versicherung (`InsurancePayMessage`) und übermittelt die Prämie in dieser Nachricht. Ist der Erwartungsnutzen ohne Versicherung größer als mit Versicherung, bezahlt der Agent keine Prämie an die Versicherung.

```
//compare expected utility of both alternatives,
//with and without insurance
if (withInsurance > withoutInsurance) {
    //pay premium for insurance
    InsurancePayMessage p2pMsg2 = new InsurancePayMessage(
        this, this._startNegotiator, this._insuranceAgent,
        (HashMap)_negotiationObject,premium,null);
    myP2P.acceptMessage(p2pMsg2);
} else {
//don't pay premium for insurance
}
```

Dieses Grundmodell der Versicherungsnachfrage wird durch weitere Varianten ergänzt, um die Entscheidung des Versicherungsnehmers zu verfeinern. Details dazu finden sich im Simulationsentwurf zur Versicherungsnachfrage in Kapitel 6.4.

### 6.3.2 Sequenzdiagramm eines beispielhaften Ablaufprozesses

Die Integration der Grid-Versicherung in die CATNETS Grid-Ökonomie zeigt das Sequenzdiagramm in Abbildung 6.7. Das Ablaufdiagramm startet mit dem Ende der Verhandlung auf dem Servicemarkt (1), dem eine erfolgreiche Verhandlung von Ressourcen auf dem Ressourcenmarkt folgt (1.1). Anschließend fragt der Basic-Service bei der Versicherung nach einem Prämienangebot für das ausgehandelte Ressourcenbündel. Hierfür schickt der Basic-Service die Risikokennzahlen an die Versicherung (1.2), die ein Prämienangebot zur Übernahme der Risiken zurückschickt (1.3), das in diesem Beispiel der Basic-Service bezahlt (1.4). Es folgen eine Bestätigung der Ressourcen an den Ressourcenanbieter (1.5 und 1.6) und eine Bestätigung des Basic-Service an den Complex-Service (2). Die Bestätigungsnachricht an den Complex-Service stellt zugleich den Startzeitpunkt für die Ausführung des Basic-Service auf den Grid-Ressourcen dar.

Nach Ende der maximalen Ausführungszeit gibt der Complex-Service das Signal zum Freigeben der Ressourcen (3 und 3.1). Mit dem Eintreffen der Freigabemessage wird bestimmt, ob ein Ausfall während der Ausführung eingetreten ist (3.1.1). Das Risiko eines Ressourcenausfalls wird durch Ziehen einer Zufallszahl unter Verwendung des Modells von Dai [DaPZ07] festgelegt (3.2). Ist ein Schaden entstanden, so wird dieser der Versicherung gemeldet (3.3) und die Kompensationsleistung kann empfangen werden (3.4). Das Gesamtergebnis der Ausführung wird in einem letzten Schritt an den Complex-Service gemeldet (4).

Dieser Ablaufplan stellt die Grundlage aller folgenden Untersuchungen dar. Die einzigen Änderungen des gezeigten Prozesses bilden die Kommunikationsnachrichten mit der Versicherung. Der Schritt 1.4 und die Schritte 3.3 und 3.4 können entfallen, wenn keine Versicherung abgeschlossen wird oder kein Schaden eingetreten ist. Zusätzlich wird der Servicemarkt als fehlerfrei angenommen; die Risiken treten nur auf dem Ressourcenmarkt auf, wo auch die Versicherung integriert ist. Auf welchem Knoten sich die Grid-Versicherung im Grid-Netzwerk befindet, ist allen Teilnehmern bekannt, sodass keine Suche nach einer Versicherung durchgeführt werden muss.

Im folgenden Kapitel werden der Entwurf der Simulationsszenarien vorgestellt und die Fragestellungen aufgeworfen, die anhand eines bestimmten Szenarios untersucht werden. Hierfür dient die Implementierung des Modells aus diesem Abschnitt als Grundlage.

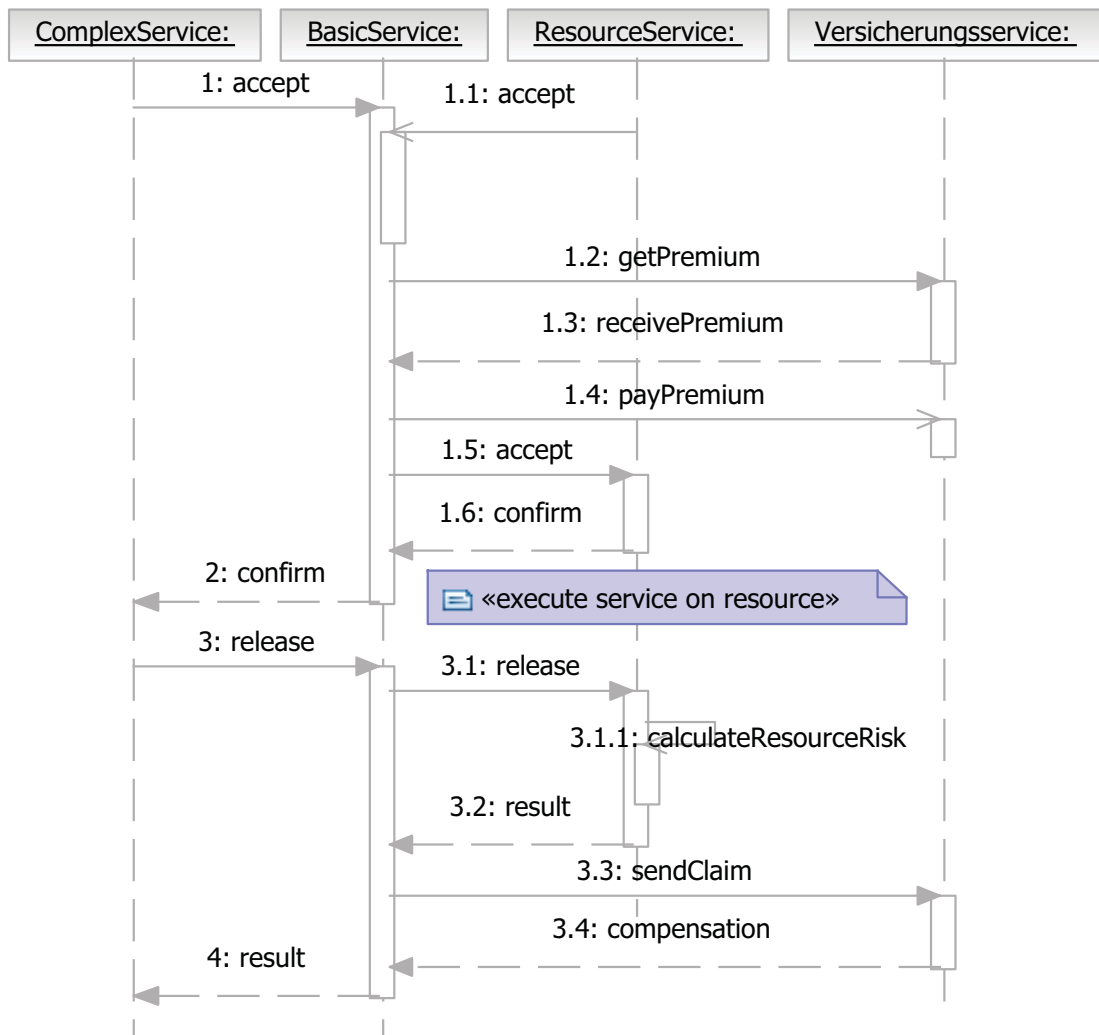


Abbildung 6.7: Sequenzdiagramm eines beispielhaften Ablaufprozesses einer Grid-Versicherung in der CATNETS Grid-Ökonomie

## 6.4 Simulationsentwurf

Der erste Teil des Simulationsentwurfs zielt darauf ab, die Dynamik der Grid-Versicherung in unterschiedlichen Konfigurationen zu untersuchen und miteinander vergleichen zu können. Dabei werden Parameter der Grid-Versicherung wie die Versicherungsperiode oder der Sicherheitszuschlag verändert, unterschiedliche Modelle der Versicherungsnachfrage betrachtet und verschiedene Grid-Systemkonfigurationen in ihrem Einfluss auf die Versicherungsprämien untersucht. Der Fokus liegt auf ausgewählten Metriken und ihrem Verhalten über die Zeit.

Der zweite Teil des Simulationsentwurfs beschreibt Szenarien, die den Einfluss der Grid-Versicherung auf die Performanz des Ressourcenmanagements anhand ausgewählter Indikatoren und Indizes analysieren lassen. Lag der Fokus des ersten Teils des Simulationsentwurfs auf einzelnen Kennzahlen der Grid-Versicherung und der Versicherungsnachfrage, so wird in diesem Abschnitt das Gesamtsystem im Brennpunkt stehen.

### 6.4.1 Dynamik der Grid-Versicherung

#### 6.4.1.1 Versicherungsperiode

Wie bereits bei der Modellbeschreibung der Grid-Versicherung in Kapitel 5 erörtert, kann die üblicherweise einjährige Versicherungsperiode realer Versicherungen nicht auf Grid-Systeme übertragen werden. Die Fragestellungen, die in diesem Simulationsszenario betrachtet werden, lauten:

1. Wie muss die Versicherungsperiode im Verhältnis zu den Ausführungszeiten der Grid-Services eingestellt werden, damit die Grid-Versicherung die Risiken zeitnah abbilden kann?
2. Wie entwickeln sich die Tariffaktoren bei unterschiedlichen Versicherungsperioden?

Anhand eines Szenarios werden Risikoprofile für Grid-Services mit verschiedenen Versicherungsperioden simuliert. Dabei werden die Tariffaktoren, der Prämieinnahme- und der Schadensprozess gemessen und analysiert.



### 6.4.1.2 Sicherheitszuschlag

Der Sicherheitszuschlag wird von der Grid-Versicherung erhoben, um die Versicherung vor dem Ruin zu bewahren [BuHW04] [SiMS03]. Das Simulationsszenario, das den Sicherheitszuschlag betrachtet, untersucht folgende Fragen:

1. Wie hoch muss der Sicherheitszuschlag in verschiedenen Grid-Systemkonfigurationen gewählt werden, um die Versicherung vor dem Ruin zu bewahren?
2. Welche Einflussgrößen wirken sich auf den Sicherheitszuschlag besonders stark aus?

Für die Analyse werden der Prämieeneinnahmeprozess und der Schadensprozess gemessen, wobei während eines Simulationslaufs immer gelten muss, dass die Prämieeneinnahmen größer als die Schadenssumme sind. Durch eine Sensitivitätsanalyse werden vorteilhafte Wertebereiche des Sicherheitszuschlags identifiziert und dokumentiert.

### 6.4.1.3 Versicherungsnachfrage

Die Untersuchung der Versicherungsnachfrage erfolgt in einem Simulationsszenario, in dem ein Ressourcenkonsument die Auswahl zwischen dem Abschluss einer Versicherung oder der Akzeptanz des Risikos hat. Im Fokus der Analyse steht die Strategie der Versicherungsnachfrage, die durch die beiden Parameter *Risikoeinschätzung* und *Zahlungsbereitschaft* festgelegt wird.

Der Parameter *Zahlungsbereitschaft* eines Ressourcenkonsumenten ist definiert als

$$\text{Zahlungsbereitschaft} = \text{Nutzen\_der\_Ressource}^{\text{Risikoeinschätzung}},$$

wobei der Nutzen der Ressource der verhandelte Transaktionspreis ist. Schätzt ein Ressourcenkonsument das Risiko eines Ausfalls der Ressource als hoch ein, so ist er bereit, mehr für den Risikotransfer zu bezahlen. Ist die Risikoeinschätzung eines Ressourcenkonsumenten nahe 0, sinkt auch seine Bereitschaft, Geld für die Übernahme des Risikos auszugeben. Die Versicherungsnachfrage wird in Bezug auf folgende Fragen untersucht:

1. Setzt sich die Strategie der Risikoakzeptanz oder die Strategie der Versicherung durch?
2. Wie entwickeln sich die Risikoeinschätzung und die Zahlungsbereitschaft im Fall einer freiwilligen Versicherung? Wie sieht eine erfolgreiche Strategie für die Versicherungsnachfrage aus?

3. Wie entwickelt sich der Ex-Post-Nutzen ausgewählter Versicherungsnehmer mit und ohne Grid-Versicherung?
4. Können Ressourcenkonsumenten mit einer Grid-Versicherung im Vergleich zu Ressourcenkonsumenten ohne Grid-Versicherung einen höheren Nutzen erzielen?

Als Metrik zur Analyse der Fragestellungen stehen die Zeitserien der beiden Parameter *Risiko einschätzung* und *Zahlungsbereitschaft* im Mittelpunkt. Zusätzlich wird der Ex-Post-Nutzen gemessen und ausgewertet. Er entspricht dem realisierten Nutzen nach Serviceausführung [LéMo07]. Die Betrachtung der Metriken erfolgt mit dem Ziel, erfolgreiche Strategien der Versicherungsnachfrage zu identifizieren. Die Analyse der Versicherungsprämien erfolgt wie schon in den vorangegangenen Simulationsszenarien auf Grundlage der Entwicklung der Tariffaktoren über die Zeit.

#### 6.4.2 Einfluss der Grid-Versicherung auf das Ressourcenmanagement

In Kapitel 4 wurden 4 Risikobehandlungsstrategien vorgestellt, von denen Risikoakzeptanz, Risikodiversifikation durch Redundanz und Risikotransfer verglichen werden. Es werden für die Untersuchung die gleichen Grid-Systemkonfigurationen verwendet, die auch schon in den letzten Abschnitten zum Einsatz kamen. Die zentralen Fragen dieser Untersuchung sind:

1. Welche Risikobehandlungsstrategie, Risikoakzeptanz, Risikodiversifikation oder Risikotransfer führt zu einer höheren Performanz des Ressourcenmanagements? Ist eine Effizienzsteigerung in allen untersuchten Szenarien nachweisbar?
2. Lassen sich die Wirkungsweisen der Grid-Versicherung in bestimmten Metriken nachweisen?
3. Welche Vor- oder Nachteile hat die Grid-Versicherung in Bezug auf die Risikoakzeptanz- und Redundanzstrategie? Wie hoch ist der Overhead, den die Grid-Versicherung mit sich bringt?

Die Analyse dieser Fragen erfolgt auf verschiedenen Aggregationsstufen, die in den folgenden Abschnitten definiert werden.

## 6.5 Definition der Metrikpyramide zur Evaluierung der Systemperformanz

Für die Analyse der Systemperformanz bietet sich eine einzige Kennzahl an, die es ermöglicht, verschiedene Systemkonfigurationen miteinander zu vergleichen und es zulässt, Auswirkungen des Risikomanagements – insbesondere der Grid-Versicherung – zu quantifizieren. Die Grundlage der Berechnung bilden die Messdaten, die während eines Simulationslaufs von den Teilnehmern der Grid-Ökonomie, der Grid-Versicherung und dem simulierten Grid-System erhoben werden.

### 6.5.1 Allgemeine Methode

Die allgemeine Methodik folgt der in Abbildung 6.8 gezeigten logischen Struktur der Daten und Indizes. Um die ökonomische Performanz des Risikomanagements zu messen, ist es eine wichtige Voraussetzung, eine möglichst breite Datenbasis durch Messung wichtiger Kennzahlen zu erhalten, da mit jedem Aggregationsschritt ein Verlust an Informationen verbunden ist. Das wird durch die Form einer Pyramide in Abbildung 6.8 visualisiert.

In einem ersten Schritt werden die diejenigen Variablen ausgewählt, die von Interesse und signifikant für die Evaluierung der Grid-Versicherung sind. An dieser Stelle werden wichtige Größen der Grid-Ökonomie, der Versicherung und des Grid-Systems ausgewählt, soweit sie im Simulationsmodell erhoben werden können. Diese werden als nichtaggregierte Rohdaten gesammelt, die in einem zweiten Schritt in eine Datenbank integriert werden. Die Integration bietet den Vorteil, die Daten filtern zu können, um Artefakte und nicht benötigte Metadaten auszublenden, die Kompatibilität der Daten sicherzustellen und mehrere Simulationsläufe für die weitere Analyse an einem zentralen Ort bereitzustellen. Aus den integrierten Daten werden Indikatoren ausgewählt, die charakteristisch für ein System sind und helfen, ein System zu verbessern. Ein Indikator ist definiert als Wert auf einer Werteskala, der von einer Reihe an beobachteten Fakten abhängt und relative Fakten als Funktion der Zeit anzeigt [Boss99]. Indikatoren geben Rückmeldung über das Systemverhalten, erlauben die Analyse der Performanz und lassen Vorhersagen über die zu erwartende Performanz des Systems zu.

Die nichtaggregierten Indikatoren werden zu aggregierten Indikatoren zusammengefasst, um eine gemeinsame Evaluierung der Einzelindikatoren zu ermöglichen. Für die Zusammenfassung der Einzelindikatoren ist es wichtig, diese vergleichbar zu machen, da die Messdaten unter-

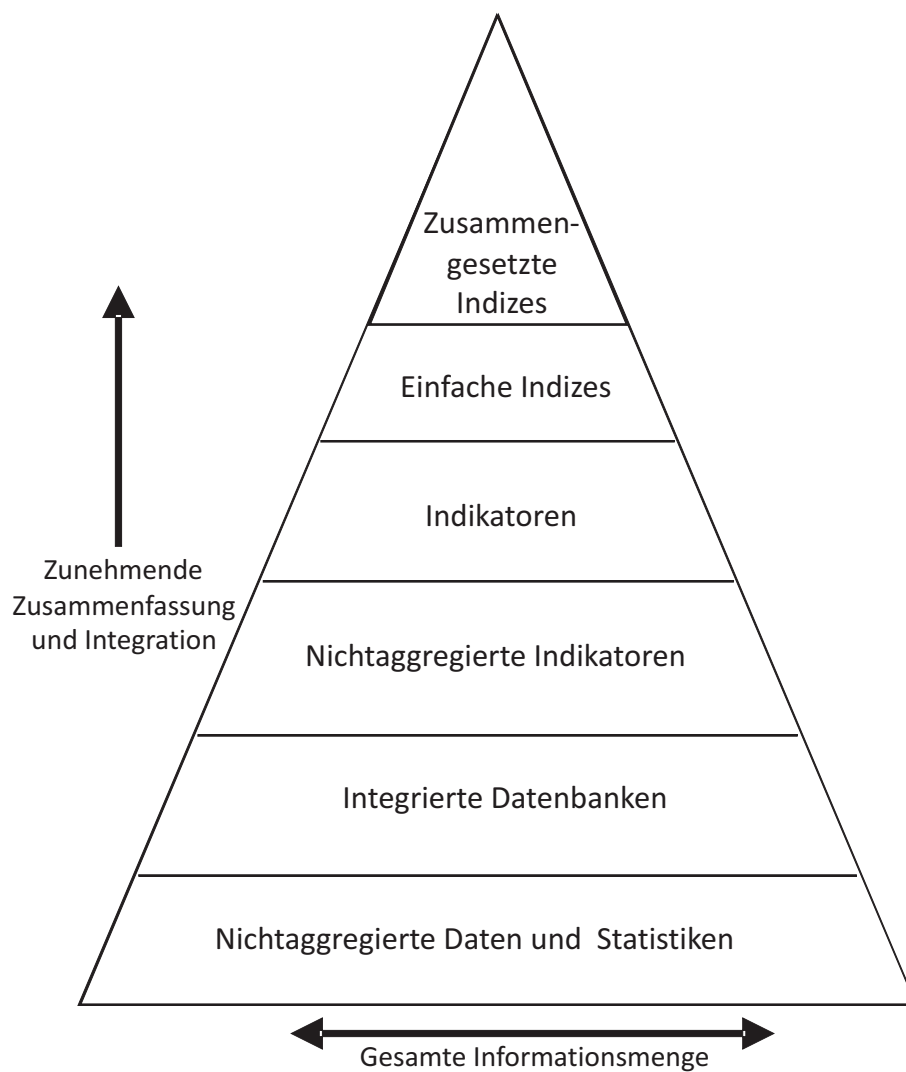


Abbildung 6.8: Allgemeine Methodik zum Erstellen eines zusammengesetzten Index [PiZC00, S. 42] [Boss99]

schiedliche Dimensionen und Messeinheiten aufweisen. Hierfür werden verschiedene Funktionen zur Normalisierung verwendet, die die Heterogenität der Einzelindikatoren eliminieren. Der einfachste Weg der Normalisierung, der zum Einsatz kommt, ist die Normalisierung in einem Intervall, wobei die Indikatoren als Prozentangabe des Intervalls angegeben werden.

Die letzten beiden Schritte der Aggregation berechnen aus den aggregierten Indikatoren einfache und zusammengesetzte Indizes. Diese Indizes drücken Informationen aus, die direkt für den Entscheidungsprozess relevant sind. Sie helfen, die Auswertung und die Beurteilung zu verbessern. Dies ist analog zu verschiedenen Indizes in den Sozial- und Wirtschaftswissenschaften [Watk07] [Horn93] [Eich88] sowie in der Informatik [PoFl00] zu sehen.

Eine Anwendung dieses Evaluierungsprozesses ermöglicht es, zwei Systemkonfigurationen mit nur einem einzigen Index bzw. einer Zahl zu vergleichen. Ein solcher Index repräsentiert ein aggregiertes Verhalten einer Systemkonfiguration in Bezug auf eine ausgewählte Anzahl an Eigenschaften. Die Anwendung von Statistik ist notwendig, um einen Index zu erhalten. Jedoch kann er nur im Zusammenhang mit einem detaillierten Rahmenwerk an Messdaten und Indikatoren seine Relevanz unter Beweis stellen. Erst eine detaillierte Betrachtung der Messergebnisse und Indikatoren lässt Rückschlüsse auf das Systemverhalten zu und verbessert dadurch die Anpassung der Grid-Versicherung an Grid-Systeme.

Obwohl stark aggregierte Indizes wegen ihrer Einfachheit attraktiv sind, beinhalten sie auch Risiken. Vor allem neigen sie dazu, die Komplexität und Details des realen Systems zu verbergen. Stark aggregierte Indizes sind wichtig, um einen Überblick zu geben, jedoch sollten sie leicht in ihre Bestandteile zerlegt werden können, sodass die Gründe für eine Erhöhung und Absenkung des Index analysiert und interessante Fragen auf niedrigeren Schichten beantwortet werden können, um eine umfassende Bewertung eines Systems zu ermöglichen. Die Pyramide mit ihren verschiedenen Schichten unterstützt diesen Prozess: zum einen ermöglicht sie die Analyse auf unteren Schichten und zum anderen einen Vergleich zweier Systemkonfigurationen mithilfe der stark aggregierten Indizes.

## 6.5.2 Die Metrikenpyramide

### 6.5.2.1 Aufbau

Die allgemeine Methodik wird auf das zu untersuchende Grid-Modell übertragen und die einzelnen Schichten werden auf dem Grid-System abgebildet. Abbildung 6.9 zeigt den Aufbau der

Pyramide. Diese wurde im Rahmen des CATNETS Projekts entwickelt und für das zu untersuchende Grid-Versicherungsmodell in dieser Arbeit angepasst.

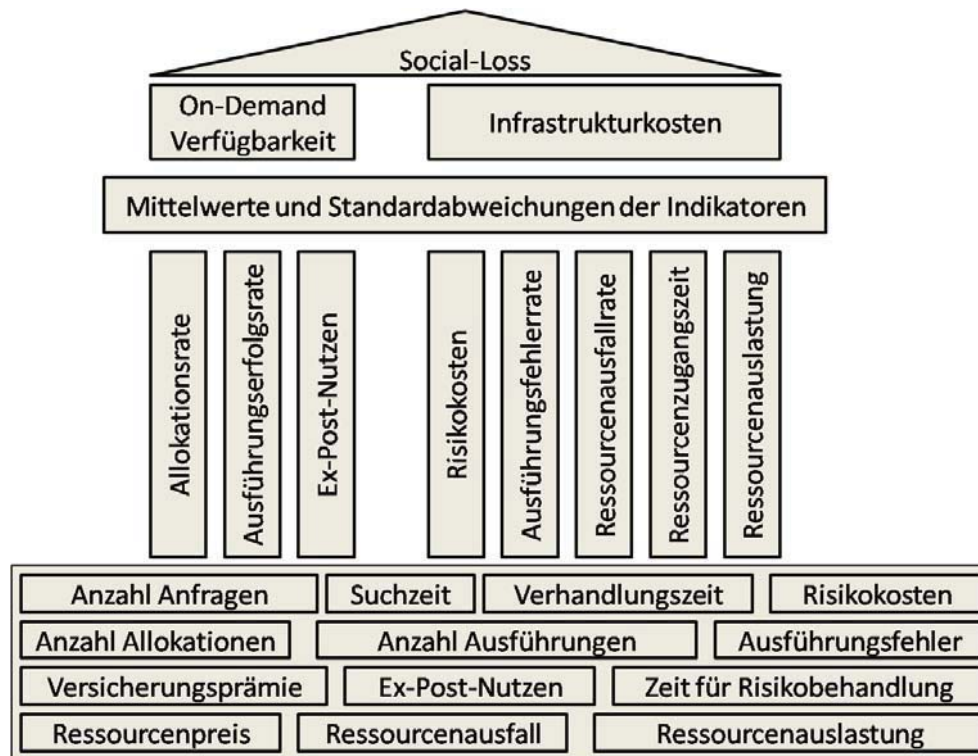


Abbildung 6.9: Metrikenpyramide zur Evaluierung der Gesamtperformanz des Ressourcenmanagements unter Einbezug der Grid-Versicherung in Anlehnung an [SEVC<sup>+</sup>07]

Die unterste Schicht beschreibt die Metriken, die während eines Simulationslaufs gemessen werden. Für jede Transaktion wird ein vollständiger Datensatz aller Messdaten erstellt und gespeichert. Der Fokus liegt hier auf dem Ressourcenmarkt der CATNETS Grid-Ökonomie und dem Einfluss der Grid-Versicherung. Aus diesem Grund werden Messgrößen des Servicemarkts nicht weiter betrachtet. Die Messdaten werden zur Berechnung der Indikatoren der nächsthöheren Schicht einer ersten Aggregation unterzogen. Es werden die Mittelwerte und die Standardabweichungen für jeden Indikator berechnet. Zugleich werden die Indikatoren normalisiert, damit sie zur Bildung der beiden einfachen Indizes, On-Demand-Verfügbarkeit und Infrastrukturkosten, im zweiten Aggregationsschritt verwendet werden können.

Im dritten und letzten Aggregationsschritt werden die beiden einfachen Indizes zum Social-Loss-Index zusammengefasst. Dieser Index ist charakteristisch für die Performanz eines Grid-Systems und ermöglicht einen Vergleich verschiedener Systemkonfigurationen [SEVC<sup>+</sup>07] [BCCC<sup>+</sup>07] [BSEV08]. In den folgenden Abschnitten werden die gemessenen Metriken detailliert erläutert

und die Berechnungen der drei Aggregationsschritte vorgestellt.

### 6.5.2.2 Technische Schicht der Messdaten

Die technische Schicht mit ihren Messdaten bildet die Grundlage der Evaluierung. Die Schicht stellt einfach zu messende Parameter bereit, die für die Berechnung der Indikatoren benötigt werden. Alle Messwerte werden für jede einzelne Transaktion gemessen und gespeichert. In chronologischer Reihenfolge werden folgende Messgrößen erhoben:

- **Anzahl Anfragen:** Diese Messgröße misst die Anzahl der Anfragen, die ein Basic-Service während eines Simulationslaufs verarbeitet. Die Messung der Metrik erfolgt zu Beginn der Ressourcenallokation auf dem Ressourcenmarkt und nach Abschluss der Allokation auf dem Servicemarkt. Für jede Anfrage  $x$ , die ein Basic-Service  $j$  verarbeitet, wird die Metrik inkrementiert:

$$\text{Anfragen}_x^j = \sum_{x=1}^n 1,$$

wobei  $n$  die Anzahl der gestarteten Transaktionen auf dem Ressourcenmarkt darstellt.

- **Suchzeit:** Die Suchzeit ist die Zeit, die ein Basic-Service  $j$  in Transaktion  $x$  benötigt, um die Ressourcenanbieter zu kontaktieren und einen Anbieter auszuwählen, mit dem er eine Verhandlung beginnt. Die Suchzeit wird maßgeblich von der Wartezeit auf Angebote von Anbietern bestimmt. In der Implementierung dieser Arbeit wartet ein Basic-Service maximal 5-mal einen zufallsgesteuerten *timeout* zwischen 500 und 1000 Millisekunden ab, bevor er die Allokation abbricht. Die Suchzeit einer Transaktion  $x$  berechnet sich dadurch wie folgt:

$$\text{Suchzeit}_x^j = \sum_{l=1}^n \text{timeout} + \text{Ufo}([0; \text{timeout}]),$$

wobei  $n$  die Anzahl der Wartezeitintervalle und *Ufo* eine Zufallszahl, die einer Gleichverteilung folgt, sind.

- **Ressourcenausfall:** Die Messgröße Ressourcenausfall misst, wie häufig ein Ressourcenanbieter  $k$  nicht verfügbar war. Ein Ressourcenanbieter wird in dieser Arbeit als nicht verfügbar angenommen, wenn er auf eine Ressourcensuche nicht mit einem Angebot antwortet. Die Wahrscheinlichkeit eines Ausfalls  $p_{res}$  entspricht der Ausfallwahrscheinlichkeit eines

Grid-Standorts. Wenn ein Ausfall eingetreten ist, wird die Messgröße inkrementiert. Ist kein Ausfall eingetreten, bleibt der Wert unverändert.

$$\text{Ressourcenausfall}_x^k = \sum_{i=1}^x a_{res},$$

wobei

$$a_{res} = \begin{cases} 1 & \text{mit Wahrscheinlichkeit } p_{res} \\ 0 & \text{mit Wahrscheinlichkeit } 1 - p_{res} \end{cases}$$

- **Verhandlungszeit:** Die Verhandlungszeit eines Basic-Service  $j$  in Transaktion  $x$  ist die Differenz zwischen der Start- und Endzeit der Verhandlung. Sie wird am Ende jeder Verhandlung berechnet. Die Startzeit  $t_{startVerhandlung}$  wird nach Auswahl des Ressourcenanbieters gemessen; Die Endzeit  $t_{endeVerhandlung}$  wird sowohl bei einer erfolgreichen als auch bei einer abgebrochenen Verhandlung ermittelt.

$$\text{Verhandlungszeit}_x^j = t_{endeVerhandlung} - t_{startVerhandlung}$$

- **Anzahl erfolgreicher Allokationen:** Die Metrik *Anzahl erfolgreicher Allokationen* misst, wie viele Ressourcenbündel einem Basic-Service  $j$  durch das Ressourcenmanagement erfolgreich zugewiesen wurden. Nach jeder erfolgreichen Verhandlung  $x_e$  wird der Wert der Metrik inkrementiert.

$$\text{Allokationen}_{x_e}^j = \sum_{i=1}^{x_e} 1$$

- **Ressourcenpreis:** Diese Messgröße erfasst den Preis der Ressource, den Basic-Service  $j$  an Ressourcenanbieter  $k$  in der Transaktion  $x$  bezahlt. Der Ressourcenpreis wird durch die in Kapitel 3 eingeführte bilaterale Verhandlung bestimmt.

$$\text{Preis}_x^j = \text{Preis}_x^{jk}$$

- **Zeit für Risikobehandlung:** Die Zeit für die Risikobehandlung schließt sich an die Verhandlung an und ist die Zeit, die Basic-Service  $j$  in Transaktion  $x$  benötigt, um auftretende Risiken zu behandeln. Im Fall eines Risikotransfers auf die Grid-Versicherung ist dies die Zeit, die benötigt wird, um eine Versicherung abzuschließen und die Prämie zu bezahlen.



Analog zur Berechnung der Verhandlungszeit werden Start- und Endzeit gemessen und deren Differenz gebildet.

$$\text{Risikobehandlungszeit}_x^j = t_{\text{endeRisikobehandlung}} - t_{\text{startRisikobehandlung}}$$

- **Versicherungsprämie:** Die Versicherungsprämie ist der Geldbetrag, den ein Basic-Service  $x$  in Transaktion  $x$  an die Versicherung  $v$  bezahlt. Trifft der Basic-Service die Entscheidung, die Versicherung nicht abzuschließen, wird die Versicherungsprämie mit 0 angenommen.

$$\text{Praemie}_x^j = \text{Praemie}_x^{jv}$$

- **Anzahl erfolgreicher Ausführungen:** Diese Metrik misst die Anzahl erfolgreicher Ausführungen eines Basic-Service  $j$ . Eine erfolgreiche Ausführung ist dann eingetreten, wenn je nach Risikobehandlungsstrategie mindestens eine ausgeführte Basic-Service-Instanz ohne Fehler ein Ergebnis zurückliefert. Wählt beispielsweise ein Basic-Service-Agent Redundanz als Risikobehandlungsstrategie, wird ein Basic-Service doppelt ausgeführt. Wenn eine Ausführung der redundanten Basic-Service-Instanzen erfolgreich ist, wird die Ausführung des Basic-Service als erfolgreich angesehen und die Metrik inkrementiert. Der Fokus der Metrik liegt nicht auf den einzelnen ausgeführten Basic-Service-Instanzen, sondern auf dem aggregierten Ergebnis aller ausgeführten Basic-Service-Instanzen. Die Anzahl der Ausführung ist wie folgt definiert:

$$\text{Ausfuehrungen}_m^j = \sum_{i=1}^m 1,$$

wobei  $m$  die Anzahl der erfolgreichen Ausführungen eines Basic-Service  $j$  darstellt. Im Fall einer gescheiterten Ausführung bleibt der Wert der Metrik unverändert.

- **Ausführungsfehler:** Die Metrik Ausführungsfehler misst alle Fehler, die bei der Ausführung eines Basic-Service aufgetreten sind. Für die Bestimmung eines Ausführungsfehlers wird das Modell von Dai et al. angewandt [DaPZ07]. Die in dieser Metrik erfassten Fehler müssen nicht zu einem Scheitern der Ausführung eines Basic-Service führen. Wenn ein Fehler eintritt und der Basic-Service redundant ausgeführt wird oder versichert ist, führt dies zu einer Erhöhung der Ausführungsfehlermetrik. Eine erfolgreiche Ausführung des

Basic-Service kann jedoch durch die gewählte Risikobehandlungsstrategie weiterhin möglich sein. Für einen Basic-Service  $j$  ist bei Eintritt eines Fehlers in Transaktion  $o$  die Metrik wie folgt definiert:

$$\text{Ausfuehrungsfehler}_o^j = \sum_{i=1}^o 1$$

Im Fall einer fehlerfreien Ausführung bleibt der Wert der Metrik unverändert.

- **Risikokosten:** Die Risikokosten sind die Kosten, die die gewählte Risikostrategie verursacht und ein Basic-Service  $j$  in Transaktion  $x$  tragen muss. Die Messung der Risikokosten wird im Folgenden für den Risikotransfer auf eine Versicherung, die Risikoakzeptanz und die Redundanz näher erläutert.

Im Fall eines Risikotransfers entsprechen die Risikokosten den Kosten zur Übernahme des Risikos durch die Versicherung. Die Prämie wird für jeder Transaktion  $x$  zwischen Basic-Service  $j$  und Versicherung  $v$  neu festgelegt.

$$\text{Risikokosten}_x^j = \text{Praemie}_x^{jv}$$

Die Risikokosten einer Risikoakzeptanzstrategie entsprechen dem Preis des Ressourcenbündels, der in Transaktion  $x$  zwischen Basic-Service  $j$  und Ressourcenanbieter  $k$  ausgehandelt wurde. Die Kosten fallen jedoch nur an, wenn die Ausführung des Basic-Service mit einem Fehler scheitert.

$$\text{Risikokosten}_x^j = \begin{cases} \text{Preis}_x^{jk} & \text{mit Wahrscheinlichkeit } p_{\text{Fehler}} \\ 0 & \text{mit Wahrscheinlichkeit } 1 - p_{\text{Fehler}} \end{cases}$$

Bei redundanter Ausführung eines Basic-Service fallen Kosten in Höhe der Kosten für das zweite Ressourcenbündel an. Dieser Preis wird in jeder Transaktion  $x$  zwischen Basic-Service  $j$  und Ressourcenanbieter  $k$  neu verhandelt.

$$\text{Risikokosten}_x^j = \text{Preis}_x^{jk}$$

- **Ressourcenauslastung:** Diese Messgröße protokolliert die Menge der konsumierten Grid-Ressourcen eines Ressourcenanbieters  $k$ . Für jede Transaktion  $x_k$ , in der der Ressourcenanbieter  $k$  beteiligt ist, wird die Auslastung der Ressourcen des Anbieters neu berechnet. Die

Auslastung ist definiert als der prozentuale Anteil der aktuell verbrauchten Ressourcen an den gesamt verfügbaren Ressourcen eines Anbieters  $k$ . Die Ressourcenauslastung wird sowohl zu Beginn als auch nach Ende einer Basic-Service-Ausführung gemessen.

$$\text{Ressourcenauslastung}_{x_k}^k = \frac{\text{verbrauchte Ressourcen}^k}{\text{Gesamtressourcen}^k}$$

- Ex-Post-Nutzen: Die Größe Ex-Post-Nutzen misst den realisierten Nutzen eines Basic-Service  $j$  in Transaktion  $x$ . Die Metrik ist für einen Ressourcenkonsumenten definiert als

$$\text{Ex-postNutzen}_x^j = \frac{\text{Ex-Ante-Nutzen}_x^j}{\text{Ressourcenpreis}_x^j}$$

im Fall einer Versicherung<sup>5</sup>,

$$\text{Ex-postNutzen}_x^j = 0$$

im Fall keiner Versicherung und einer gescheiterten Ausführung und

$$\text{Ex-postNutzen}_x^j = 1$$

im Fall keiner Versicherung und erfolgreicher Ausführung.

Der Ex-Ante-Nutzen eines Basic-Service  $j$  in Transaktion  $x$  entspricht dem Erwartungsnutzen eines Basic-Service vor der Ausführung und wird wie folgt berechnet:

$$\begin{aligned} \text{Ex-Ante-Nutzen}_x^j &= (\text{Ressourcenpreis}_x^j - \text{Praemie}_x^j) * \text{Risikoeinschaetzung}_x^j + \\ &(\text{Transaktionspreis}_x^j - \text{Praemie}_x^j) * (1 - \text{Risikoeinschaetzung}_x^j) \end{aligned}$$

### 6.5.2.3 Indikatoren - Erste Aggregation

Ziel dieser Schicht ist es, einen Satz an unabhängigen Metriken zu definieren, die zwischen 0 und 1 normalisiert sind. Die Annahme der Unabhängigkeit der Metriken macht es einfacher, gültige Funktionen zur Integration der Indikatoren in den darüber liegenden Schichten zu finden. Vor allem die Berechnung der beiden Indizes On-Demand-Verfügbarkeit und Infrastrukturkosten wird dadurch erleichtert.

<sup>5</sup>Die Division durch den Ressourcenpreis dient der Normalisierung des Ex-Ante-Nutzen auf das Intervall zwischen 0 und 1.

Zur Berechnung der Indikatoren werden die Messgrößen aus dem vorangegangenen Abschnitt herangezogen und im ersten Aggregationsschritt zu Indikatoren integriert. Die technischen Metriken werden so kombiniert, um ein Rahmenwerk zu erhalten, das eine Evaluierung verschiedener Systemkonfiguration und Risikobehandlungsstrategien zulässt. Die Auswahl der Indikatoren erfolgt in vier verschiedenen Kategorien, zu denen jeder Indikator zuzuordnen ist [SEVC<sup>+</sup>07]:

1. Effizienzmaße: Allokationsrate, Ausführungserfolgsrate, Risikokosten
2. Nutzenmaße: Ex-Post-Nutzen
3. Zeitmetriken: Ressourcenzugangszeit
4. Systemgrößen: Ausführungsfehlerrate, Ressourcenausfallrate, Ressourcenauslastung

Die Indikatoren für ein zu untersuchendes Grid-System sind:

- Allokationsrate: Die Allokationsrate ist ein Indikator für die Effizienz des Allokationsprozesses und wird als Verhältnis zwischen der Anzahl der erfolgreichen Anfragen und der Gesamtzahl an Anfragen berechnet [SEVC<sup>+</sup>07]. Als Grundlage dient die Zuordnungshistorie, die während eines Simulationslaufs aufgezeichnet wird. Ziel des Indikators ist es, die Performanz des Allokationsverfahrens in einer einfachen Größe abzubilden. Durch Analyse dieses Merkmales kann festgestellt werden, mit welcher Wahrscheinlichkeit die Allokation von Grid-Ressourcen erfolgreich war. Die Allokationsrate eines Basic-Service ist wie folgt definiert:

$$\text{Allokationsrate} = \frac{\text{Allokationen}}{\text{Anfragen}}.$$

- Ausführungserfolgsrate: Die Ausführungserfolgsrate ist ein Indikator, der anzeigt, mit welcher Wahrscheinlichkeit ein Basic-Service erfolgreich ausgeführt wird. Der Indikator berücksichtigt dabei den Einfluss einer Risikobehandlungsstrategie, die zu einer Erhöhung der Ausführungserfolgsrate beitragen kann. Die Metrik ist definiert als das Verhältnis zwischen den Ausführungen und den Allokationen eines Basic-Service:

$$\text{Ausführungserfolgsrate} = \frac{\text{Ausführungen}}{\text{Allokationen}}.$$

- Ex-Post-Nutzen: Der Indikator Ex-Post-Nutzen repräsentiert den durchschnittlich realisierten Nutzen eines Basic-Service. Er misst den durchschnittlichen Erfolg eines Basic-Service-Agenten im Hinblick auf seine Entscheidungen. Ein erfolgreicher Basic-Service,

der einen durchschnittlichen hohen Ex-Post-Nutzen erzielt hat, hat sowohl in der Verhandlung einen niedrigen Preis erzielt als auch eine gute Entscheidung hinsichtlich der gewählten Risikobehandlungsstrategie getroffen.

- **Ausführungsfehlerrate:** Die Ausführungsfehlerrate ist eine Kennzahl, die angibt, wie viele Fehler bei der Ausführung eines Basic-Service bei einer bestimmten Anzahl an Allokationen aufgetreten sind. Sie ist definiert als

$$\text{Ausführungsfehlerrate} = \frac{\text{Ausführungsfehler}}{\text{Allokationen}}$$

- **Ressourcenausfallrate:** Die Ressourcenausfallrate ist eine Kennzahl für die Verfügbarkeit eines Ressourcenanbieters. Der Indikator gibt an, wie viele empfangene Anfragen ein Ressourcenanbieter beantwortet hat. Der Anteil der nicht beantworteten Anfragen gibt Auskunft über die Verfügbarkeit der Ressourcen des Ressourcenanbieters, die Schwankungen durch Wartungsarbeiten und Fehlkonfiguration oder Ausfällen unterworfen ist. Sie ist definiert als

$$\text{Ressourcenausfallrate} = \frac{\text{Ressourcenausfall}}{\text{Anfragen}}$$

- **Ressourcenzugangszeit:** Dieser Indikator drückt die zusätzlich benötigte Zeit aus, die für die Ressourcensuche, die Verhandlung und die Risikobehandlung benötigt wird. Die Ressourcenzugangszeit repräsentiert den Overhead, der mit der Ressourcenauswahl und dem Risikomanagement einhergeht. Der Indikator ist die Summe aus folgenden Zeitmetriken:

$$\text{Ressourcenzugangszeit} = \text{Suchzeit} + \text{Verhandlungszeit} + \text{Risikobehandlung}$$

Die Normalisierung der Ressourcenzugangszeit erfolgt über eine negative Exponentialfunktion:

$$\exp^{-\text{Ressourcenzugangszeit} * 0,002}$$

- **Ressourcenauslastung:** Die Ressourcenauslastung zeigt die durchschnittlich verwendeten Ressourcen über einen Beobachtungszeitraum an, der in dieser Arbeit die Simulationszeit darstellt. Der Indikator stellt ein wichtiges Kriterium hinsichtlich der gleichmäßigen Belastung der Grid-Ressourcen dar.

$$\text{Ressourcenauslastung} = \frac{\text{Ausfuehrungszeit}}{\text{Simulationszeit}} * \text{Ressourcenmenge}.$$

Jede Ausführung eines Basic-Service auf den Ressourcen ist gekennzeichnet durch seine Zeitdauer und die anteilige Ressourcenmenge. Eine optimale Ressourcenauslastung würde bedeuten, dass ein Ressourcenanbieter immer 100% seiner Ressourcenmenge über die gesamte Simulationszeit auslasten konnte. Durch das Ressourcenmanagement und eintretende Ausfälle wird diese optimale Auslastung jedoch nicht erreicht, was sich auf eine geringere Ausführungszeit und/oder auf weniger stark ausgelastete Ressourcen niederschlägt.

- **Risikokosten:** Die Risikokosten geben die durchschnittlichen Risikokosten einer Risikobehandlungsstrategie an. Diese fließen im Fall eines Risikotransfers in Form einer Versicherungsprämie an die Versicherung, im Fall einer redundanten Ausführung an einen Ressourcenanbieter und müssen im Fall einer Risikoakzeptanz selbst getragen werden.

Für die weitere Verarbeitung der Indikatoren auf den höheren Aggregationsebenen müssen die Messwerte aggregiert und die Mittelwerte und Varianzen der Indikatoren berechnet werden. Das allgemeine Schema wird im Folgenden exemplarisch anhand des Indikators Allokationsrate näher erläutert. Ausgehend von den Messdaten erfolgt zuerst die Normalisierung der Daten für jeden Basic-Service  $j$ . Daran schließt sich die Berechnung des Mittelwerts und der Varianz über die Basic-Service  $j$  an.

### 1. Daten

Die Daten der zur Berechnung der Allokationsrate liegen für jeden Basic-Service  $j$  vor und werden in jeder Transaktion  $x$  aktualisiert.

	Anfragen	Allokationen
Basic-Service 1	200	100
Basic-Service 2	100	100
⋮	⋮	⋮

### 2. Normalisierung

$$\text{Allokationsrate}^j = \frac{\text{Allokationen}^j}{\text{Anfragen}^j}$$

	Allokationsrate <sup>j</sup>
Basic-Service 1	0,5
Basic-Service 2	1
⋮	⋮

### 3. Berechnung des Mittelwerts und der Varianz

$$\mu_{\text{Allokationsrate}} = \frac{1}{m} \sum_{j=1}^m \frac{\text{Allokationen}^j}{\text{Anfragen}^j}$$

$$\sigma_{\text{Allokationsrate}}^2 = \frac{1}{m} \sum_{j=1}^m \left( \frac{\text{Allokationen}^j}{\text{Anfragen}^j} - \mu_{\text{Allokationsrate}} \right)^2,$$

wobei  $m$  die Anzahl der Basic-Services ist.

Alle weiteren Metriken werden analog dem oben vorgestellten Schema berechnet und dienen als Eingangsgrößen für die folgende Aggregation.

#### 6.5.2.4 Einfache Indizes - Zweite Aggregation

Die zusammengesetzten Indikatoren werden durch eine Aggregation der einfachen, normalisierten Indikatoren berechnet. Aus ökonomischer Perspektive betrachtet, messen die zusammengesetzten Indikatoren den Nutzen eines Systems durch den Indikator *On-Demand Verfügbarkeit (ODM)* und die Kosten, die das Grid-System, das Ressourcenmanagement und das Risikomanagement verursachen, mit dem Indikator *Infrastrukturkosten (IK)*. Ein optimales Ressourcenmanagement mit einem optimalen, integrierten Risikomanagement würde die *On-Demand-Verfügbarkeit* von 1 und die *Infrastrukturkosten* von 0 besitzen. Ein solches System würde kein Risikomanagement benötigen, da die Zuverlässigkeit immer bei 100% liegt.

Der Indikator *On-Demand-Verfügbarkeit* integriert die wichtigen Systemkennzahlen Allokationsrate und Ausführungserfolgsrate, die auf eine hohe Verfügbarkeit des Systems schließen lassen. Zusammen mit dem Ex-Post-Nutzen bildet der ODM-Indikator den Nutzen des Systems ab. Ein Grund für die Bedeutung der Verfügbarkeit liegt darin, dass mit der zunehmenden Größe eines Computersystems dessen Zuverlässigkeit sinkt [GrSi91]. Gleichzeitig ist jedoch eine hohe Verfügbarkeit entscheidend für den Nutzen, den ein Benutzer eines Grid-Systems realisieren kann [GWWF<sup>+</sup>94]. Die *On-Demand-Verfügbarkeit* ist ein zusammengesetzter Indikator, der als Mittelwert der einfachen Indikatoren berechnet wird und wie folgt definiert ist:

$$\mu_{\text{ODM}} = \frac{1}{3} (\mu_{\text{Allokationsrate}} + \mu_{\text{Ex-Post-Nutzen}} + \mu_{\text{Ausfuehrungserfolgsrate}})$$

Die Varianz des zusammengesetzte Indikators ODM ist:

$$\sigma_{ODM}^2 = \left(\frac{1}{3}\right)^2 (\sigma_{Allokationsrate}^2 + \sigma_{Ex-Post-Nutzen}^2 + \sigma_{Ausfuehrungserfolgsrate}^2),$$

wobei angenommen wird, dass die einfachen Indikatoren unabhängig sind.

Der Indikator Infrastrukturkosten aggregiert die einfachen Indikatoren, die für die Kosten der Infrastruktur verantwortlich sind. Hierzu gehören die Kosten des Risikomanagements, Fehler von Service-Ausführungen, Ausfälle der Ressourcen sowie die Ressourcenauslastung und der zeitliche Overhead bis zur Ressourcennutzung (Ressourcenzugangszeit). Der Mittelwert und die Standardabweichung des Indikators Infrastrukturkosten werden analog zum ODM-Indikator berechnet:

$$\begin{aligned} \mu_{IK} = & \frac{1}{5} (\mu_{Risikokosten} + \mu_{Ausfuehrungsfehlerrate} + \mu_{Ressourcenausfallrate} \\ & + \mu_{(1-Ressourcenauslastung)} + \mu_{Ressourcenzugangszeit}) \end{aligned}$$

$$\begin{aligned} \sigma_{IK}^2 = & \left(\frac{1}{5}\right)^2 (\sigma_{Risikokosten}^2 + \sigma_{Ausfuehrungsfehlerrate}^2 + \sigma_{Ressourcenausfallrate}^2 \\ & + \sigma_{(1-Ressourcenauslastung)}^2 + \sigma_{Ressourcenzugangszeit}^2) \end{aligned}$$

Die Varianz repräsentiert eine Messgröße für das allgemeine Risiko, das die Stabilität des Systems bezüglich der einfachen und zusammengesetzten Indikatoren anzeigt. Die beiden zusammengesetzten Indikatoren Infrastrukturkosten und On-Demand-Verfügbarkeit bilden die Grundlage der dritten Aggregation, die im folgenden Abschnitt beschrieben wird.

### 6.5.2.5 Der Social-Loss-Index - Dritte Aggregation

Das Ziel der dritten Aggregationsschicht besteht darin, die Vergleichbarkeit von verschiedenen Szenarien zu erreichen. Der zu berechnende Index sollte eine aggregierte Maßzahl aus den einfachen Indizes ODM und IK sein. Das grundlegende Problem dabei ist, dass es sehr viele verschiedene Möglichkeiten gibt, aus einfachen Indizes einen zusammengesetzten Index zu berechnen. Das im weiteren Verlauf dieses Abschnitts vorgestellte Verfahren verwendet dabei ökonomische Konzepte, um die einfachen Indizes ODM und IK zu aggregieren.



Um ökonomische Konzepte im Evaluierungsprozess verwenden zu können, wird ein virtueller Entscheidungsträger eingeführt, dessen Ziel die Optimierung des Risikomanagements im untersuchten System ist. Das Verhalten des Entscheidungsträgers wird mit einer Zielfunktion beschrieben, die unter bestimmten Nebenbedingungen minimiert werden soll. Der Entscheidungsträger hat Kenntnis von der Verteilung der Nutzen und Kosten bzw. der beiden einfachen Indizes, die diese repräsentieren und hat bestimmte Präferenzen über ODM und IK. Analogien eines solchen Entscheidungsträgers existieren in der makroökonomischen Theorie, wobei an dieser Stelle auf zwei ähnliche Modelle Bezug genommen wird und deren statistische Verfahren zur Bildung des Social-Loss-Index herangezogen werden: Das Modell von Barro und Gordon [BaGo83] und das Modell von Poole [Pool70].

Barro und Gordon [BaGo83] ist ein ökonomisches Modell, das einen Sozialplaner beschreibt, der über seine Policy entscheidet. Angewandt auf das Beispiel einer Zentralbank erhebt der Sozialplaner einfache Indizes und wendet eine Verlustfunktion darauf an, um eine Entscheidung zu treffen. Während das Modell von Barro und Gordon die Abläufe einer Ökonomie als bekannt und deterministisch annimmt, wird diese Annahme im Modell von Poole [Pool70] nicht getroffen. Das Verhalten der betrachteten Indizes wird durch eine statistische Verteilung mit Mittelwert und Varianz angegeben. Das Ziel des Entscheidungsträgers ist es, die Variabilität des untersuchten Systems zu verringern<sup>6</sup>.

Angewandt auf das hier vorliegende Problem der Aggregation der beiden einfachen Indizes ODM und IK, wird eine Verlustfunktion aufgestellt, die die einfachen Indizes ODM und IK als Argumente enthält. ODM und IK werden als stochastische Variable angesehen und die ersten und zweiten Momente ihrer Verteilung berücksichtigt.

Der Social-Loss-Index repräsentiert den Verlust, der angibt, wie viel ein gegebenes System von einem optimalen System abweicht, wobei ein optimales System durch IK von 0 und (1-ODM) von 0 definiert ist. Das Ziel ist, diejenige Konfiguration auszuwählen, die möglichst nahe an das optimale System herankommt.

Für die Bildung der Verlustfunktion kommt das Modell von Poole zum Einsatz, das eine Verbesserung des Nutzens der Gesellschaft bei gleichzeitiger Verringerung der Variabilität beschreibt. Überträgt man das Ziel, die Variabilität des untersuchten Systems zu verringern, auf Grid-Systeme, bedeutet dies, dass ein System besser als ein anderes System ist, wenn ein Grid-System möglichst stabil ist und keine große Abweichung vom Mittelwert aufweist. Ein virtueller Entschei-

---

<sup>6</sup>Für eine weiterführende Diskussion der beiden Modelle wird an dieser Stelle auf [CGSR<sup>+</sup>05] und [SEVC<sup>+</sup>07] verwiesen.

Träger ist daran interessiert, die Ungleichheit zwischen den Teilnehmern des Grid-Systems möglichst klein zu halten. Unter Anwendung des gewichteten Euklidischen Abstands ist die Verlustfunktion des Social-Loss-Index wie folgt definiert:

$$SocialLoss = \alpha * \mu_{IK}^2 + \beta * \mu_{(1-ODM)}^2 + \alpha * \sigma_{IK}^2 + \beta * \sigma_{(1-ODM)}^2,$$

wobei  $\alpha$  und  $\beta$  die Gewichtung der beiden Indizes IK und (1-ODM) sind. Als Standardkonfiguration werden beide Indizes gleich gewichtet, d. h.  $\alpha$  und  $\beta$  haben jeweils den Wert 0,5.

Ein Vergleich zweier Systemkonfigurationen erfolgt durch Berechnung des Social-Loss-Index zur Einordnung der untersuchten Systemkonfigurationen. Gleichzeitig erfolgt ein Vergleich für die beiden darunter liegenden Schichten der einfachen Indizes und der Indikatoren, um die Gründe für die Änderung des Index detailliert analysieren zu können.

## 6.6 Zusammenfassung

Dieses Kapitel führt in die Forschungsmethode Simulation ein und begründet, warum diese Methode für die Untersuchung im Rahmen dieser Arbeit geeignet ist. Dabei wird ein allgemeiner Prozess zur Durchführung von Simulationsstudien detailliert beschrieben. Anhand dieses Prozesses werden wichtige Aspekte der Verifikation und Validierung des Simulationsmodells diskutiert und Probleme aufgezeigt. Die Verifikation des Modells erfolgt durch Anwendung verschiedener Konzepte aus der Softwareentwicklung, während die Validierung mangels real existierender Grid-Ökonomien nur teilweise durchgeführt werden kann. Im Fokus der Validierung steht aus diesem Grund die Erhöhung der Glaubwürdigkeit, die durch eine Sensitivitätsanalyse und den Vergleich verschiedener Modelle und Systemkonfigurationen unterstützt wird.

Der Rest des Kapitels präsentiert die Schritte Implementierung und Simulationsentwurf des Simulationsprozesses. Es wird der CATNETS Grid-Simulator eingeführt und die Implementierung der Grid-Versicherung in den Simulator wird dokumentiert. Anschließend folgt der Simulationsentwurf und es kommt zur Definition einer Metrikpyramide. Der Simulationsentwurf gliedert sich in zwei Teile, von denen der erste Teilaspekt der Grid-Versicherung und der Versicherungsnachfrage mithilfe von Zeitserien untersucht und der zweite den Einfluss der Grid-Versicherung auf das Ressourcenmanagement analysiert. Für den letztgenannten Fall kommt eine Metrikpyramide zum Einsatz, die verschiedene Ebenen zur Analyse anbietet und die Messdaten zu einem einzigen Index aggregiert, der einen einfachen Vergleich verschiedener Systemkonfigurationen

zulässt. Zusätzlich wird die Performanz des Risikomanagements durch die Metrikpyramide untersucht.

# Kapitel 7

## Evaluation der Grid-Versicherung

*„Agent-based model results are in many ways similar to results from traditional models and simulations, but the agent focus of agent-based models adds some new dimensions and challenges. [...] The key issue in analyzing agent simulation results is to understand how the micro-level behavioral rules of the agents, which are selectable or controllable in part, influence the macro-level behaviour of the system as a whole.“ [NoMa07, S. 261]*

Im Mittelpunkt dieses Kapitel steht die Evaluierung der Grid-Versicherung anhand der in Kapitel 6 beschriebenen Fragestellungen. Verschiedene Parameter werden auf Mikroebene verändert und resultierende Veränderungen auf Makroebene mithilfe eines visuellen Ansatzes präsentiert und analysiert. Zur Durchführung der Simulationen werden der CATNETS Simulator und die verifizierte Implementierung der Grid-Versicherung verwendet. Alle Simulationsläufe werden auf einem Intel Core 2 Duo T7500 Prozessor mit 4 GB Hauptspeicher und dem Betriebssystem Windows Vista mit installiertem Java in der Version 6<sup>1</sup> ausgeführt.

Der Aufbau des Kapitels folgt dem in Kapitel 6 eingeführten Prozess zur Durchführung einer Simulationsstudie. Abschnitt 7.1 beschreibt den Entwurf eines realistischen Simulationsszenarios. Als Vorlage des konkreten Szenarios dienen bestehende Grid-Systeme sowie kommerzielle Anbieter von Cloud-Computing-Services. Für das Simulationsszenario werden zwei Risikoprofile definiert und der Grid-Versicherungstarif abgeleitet. Das Szenario stellt die Grundlage für alle folgenden Untersuchungen der Simulationsstudie dar.

Die Abschnitte 7.2 und 7.3 präsentieren die Simulationsergebnisse der Simulationsstudie. In Ab-

---

<sup>1</sup>Der CATNETS Simulator stellt hohe Speicheranforderungen an die Hardware. Aus diesem Grund muss manuell mehr Speicher der Java Umgebung zugewiesen werden, um die Durchführung der Simulationen zu ermöglichen. Mithilfe der Option `-Xmx1024M` werden 1024 MB Speicher für die Ausführung der Simulationsläufe reserviert.

schnitt 7.2 werden einzelne Parameter von Versicherungen im Kontext von Grid-Systemen untersucht. Im Fokus der Betrachtung stehen die Länge der Versicherungsperiode und die Strategien der Versicherung und der Versicherungsnachfrage. Eine integrierte Analyse der Gesamtperformance wird in Abschnitt 7.3 durchgeführt, um Auswirkungen der Versicherung auf das Ressourcenmanagement zu analysieren. Ein Vergleich mit den Risikobehandlungsstrategien Risikoakzeptanz und Risikodiversifikation mithilfe der in Kapitel 6 definierten Metrikpyramide steht im Zentrum dieses Abschnitts. Abschnitt 7.4 fasst die gewonnenen Ergebnisse zusammen.

## 7.1 Erzeugung eines realistischen Szenarios

Die Erzeugung eines realistischen Szenarios erhöht die Validität der Simulationsergebnisse und ist damit ein wichtiger Schritt zur Vorbereitung der produktiven Simulationsläufe. Aus dem Aufbau verschiedener realer Grid-Systeme werden Parameter und Konfigurationseinstellungen der Netzwerkinfrastruktur und der Ressourcenverteilung für das Simulationsszenario abgeleitet. Messdaten dieser Grid-Systeme werden verwendet, um das technische Risiko der Ressourcen zu quantifizieren. Als Vorlage für die Festlegung des Service- und Ressourcenmarkts dienen kommerzielle Angebote von Cloud-Computing-Anbietern.

Aus der Konfiguration des Simulationsszenarios wird schließlich ein Versicherungstarif für die Grid-Ressourcen abgeleitet, der die Gegebenheiten des Szenarios berücksichtigt. In den folgenden Abschnitten werden die Details des Simulationsszenarios vorgestellt. Dieses Szenario ist als Grundscenario anzusehen, das für die Untersuchungen einzelner Aspekte der Grid-Versicherung angepasst wird. Veränderungen der Konfiguration werden bei den jeweiligen Simulationsläufen dokumentiert.

### 7.1.1 Netzwerkinfrastruktur und Ressourcenverteilung

Das Netzwerk legt die Kommunikationsverbindungen zwischen den Standorten der Complex-Services, der Basic-Services, und den Ressourcen des Simulationsszenarios fest und definiert damit die Entfernung zwischen den Handelspartnern. Die Auswahl der grundlegenden Netzwerkparameter orientiert sich am Aufbau bestehender Grid-Systeme und den Möglichkeiten des CATNETS Grid-Simulators zur Umsetzung der Netzwerkinfrastruktur. Über die Netzwerkinfrastruktur und Ressourcenverteilung kommerzieller Anbieter von Grid- bzw. Cloud-Computing-Systemen gibt es zum aktuellen Zeitpunkt noch keine Informationen, die bei der Festlegung

Tabelle 7.1: Netzwerkaufbau und Ressourcenverteilung des Simulationsszenarios im Vergleich zu den Grid-Systemen Grid'5000, D-Grid und LHC-Computing-Grid.

Kriterium	Grid'5000	D-Grid	LHC-Computing-Grid	Simulationsmodell
Netzwerkknoten	ca. 20	ca. 60	mehrere hundert	200
Ressourcenanbieter	ca. 20	ca. 50	>200	50
Recheneinheiten	ca. 5000	> 3000	> 50.000	2.900
Grad der Netzwerkverbindung	>1	>1	>1	>1
Bandbreite	10 Giga-bit/s	0,5 - 10 Gigabit/s	0,5 - 10 Giga-bit/s	1 Gigabit/s

der Netzwerkinfrastruktur verwendet werden können, sodass bei der Datenerhebung auf Grid-Systeme aus der Forschung zurückgegriffen wird. Im Einzelnen werden das französische Grid'5000, das deutsche D-Grid und das weltweit verteilte LHC-Computing-Grid betrachtet, um grundlegende Netzwerkparameter und Daten über die Verteilung und Menge der Grid-Ressourcen festzulegen<sup>2</sup>. Tabelle 7.1 zeigt eine Übersicht der Kriterien, die für die Festlegung des Simulationsszenarios von Bedeutung sind und deren Werteausprägung: sowohl für die drei untersuchten Grid-Systeme<sup>3</sup> als auch für das Simulationsszenario.

Die Anzahl der Netzwerkknoten legt die Größe des zu simulierenden Grid-Systems fest. Die Zahl der Netzwerkknoten des Grid'5000 und des D-Grid wird durch die Größe der jeweiligen Forschungsnetzwerke bestimmt, die von den Grid-Systemen verwendet werden. Die exakte Anzahl der Netzwerkknoten des LHC-Computing-Grid ist nicht öffentlich zugänglich, sie liegt jedoch im Bereich von mehreren hundert Netzwerkknoten. Die Größe des simulierten Netzwerks orientiert sich zwischen dem D-Grid und der Größe des LHC-Computing-Grid; die Netzwerkgröße wird im Simulationsszenario auf 200 Knoten festgelegt.

Im Simulationsszenario bieten 50 Ressourcenanbieter 2.900 Recheneinheiten an. Dies entspricht in etwa der Ressourcendichte des D-Grid. Die Ressourcendichte des simulierten Grid-Netzwerks

<sup>2</sup>Als Quelle der Datenerhebung dienen die Webseiten der jeweiligen Projekte (Stand Juni 2008). Grid'5000 (<https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home>), D-Grid (<http://www.d-grid.de/?id=206>), und LHC-Computing-Grid (<http://gridmap.cern.ch/gm/>) (Letzter Abruf: 30.06.2008)

<sup>3</sup>Sind keine genauen Zahlen eines Kriteriums verfügbar, so wurden die ungefähren Angaben der Webseiten übernommen.

wurde so gewählt, dass Ressourcennachfrager ausreichend Ressourcen vorfinden. Analog zu realen Grid-Systemen weisen Ressourcenanbieter auch im Simulationsszenario eine stark schwankende Menge an verfügbaren Ressourcen auf. Dies wird durch die Modellierung verschiedener Ressourcenanbieter abgebildet. Die Kapazitäten der einzelnen Anbieter sind: 20, 40, 50, 80, und 100 Recheneinheiten. Je 10 dieser Anbieter sind im Grid-Netzwerk verfügbar, die sich zu der Gesamtkapazität von 2.900 Recheneinheiten addieren.

Die Konnektivität des Grid-Netzwerkes wird durch den Grad der Netzwerkverbindung festgelegt. Ein Grad von mehr als 1 stellt sicher, dass bei Ausfall oder Überlastung einer Netzwerkverbindung mindestens eine weitere Verbindung zur Verfügung steht, die die Kommunikation zwischen den Knoten gewährleistet. Jeder Knoten im Simulationsszenario hat mindestens einen Grad von 2 oder mehr, sodass Engpässe des Netzwerks ausgeschlossen werden können. Zusätzlich gilt im Simulationsszenario: Je größer der Grad der Netzwerkverbindung ist, desto mehr Ressourcen sind auf diesem Knoten zu finden. Dadurch wird sichergestellt, dass die Ressourcenanbieter gehäuft an diesen Knoten im Netzwerk zu finden sind. Eine solche Ansammlung von Ressourcenanbietern kann als Rechenzentrum interpretiert werden, das über eine meist sehr gute und mehrfach redundante Netzwerkverbindung verfügt. Die Bandbreite als letztes wichtiges Kriterium des Netzwerks ist im Simulationsszenario auf 1 Gigabit/s begrenzt. Gleichzeitig wird die Größe der zu übertragenden Nachrichten so gewählt, dass genügend Kapazität für die Übertragung von Nachrichten zur Verfügung steht.

Abbildung 7.1 zeigt das Netzwerk, das im Simulationsszenario zum Einsatz kommt. Das Netzwerk wurde mithilfe des automatisierten Szenariogenerators der CATNETS Simulationsumgebung unter Verwendung der oben erläuterten Konfigurationsparameter erzeugt. Deutlich sichtbar sind die stark vernetzten Knoten in der Mitte des Netzwerks, auf denen die Ressourcen der Ressourcenanbieter zu finden sind. Am Rand des Netzwerks nimmt der Grad der Netzwerkverbindung ab; jeder Knoten hat jedoch mindestens eine Verbindung zu zwei seiner Nachbarn.

### 7.1.2 Szenario des Ressourcenmarkts

Der Ressourcenmarkt hat die Aufgabe, die Ressourcen der Ressourcenanbieter als Ressourcen-Services handelbar zu machen, die von Basic-Services nachgefragt werden können. Ressourcen-Services werden durch Ressourcenbündel definiert, wie sie in Kapitel 3 eingeführt wurden. Als Vorbild für die Festlegung der Ressourcenbündel dienen die Amazon EC2 Cloud-Computing-Services, die eine konkrete Ausprägung von Ressourcen-Services darstellen. Diese kommen bereits in verschiedenen kommerziellen Projekten zum Einsatz, sodass deren Ressourcenbündel-



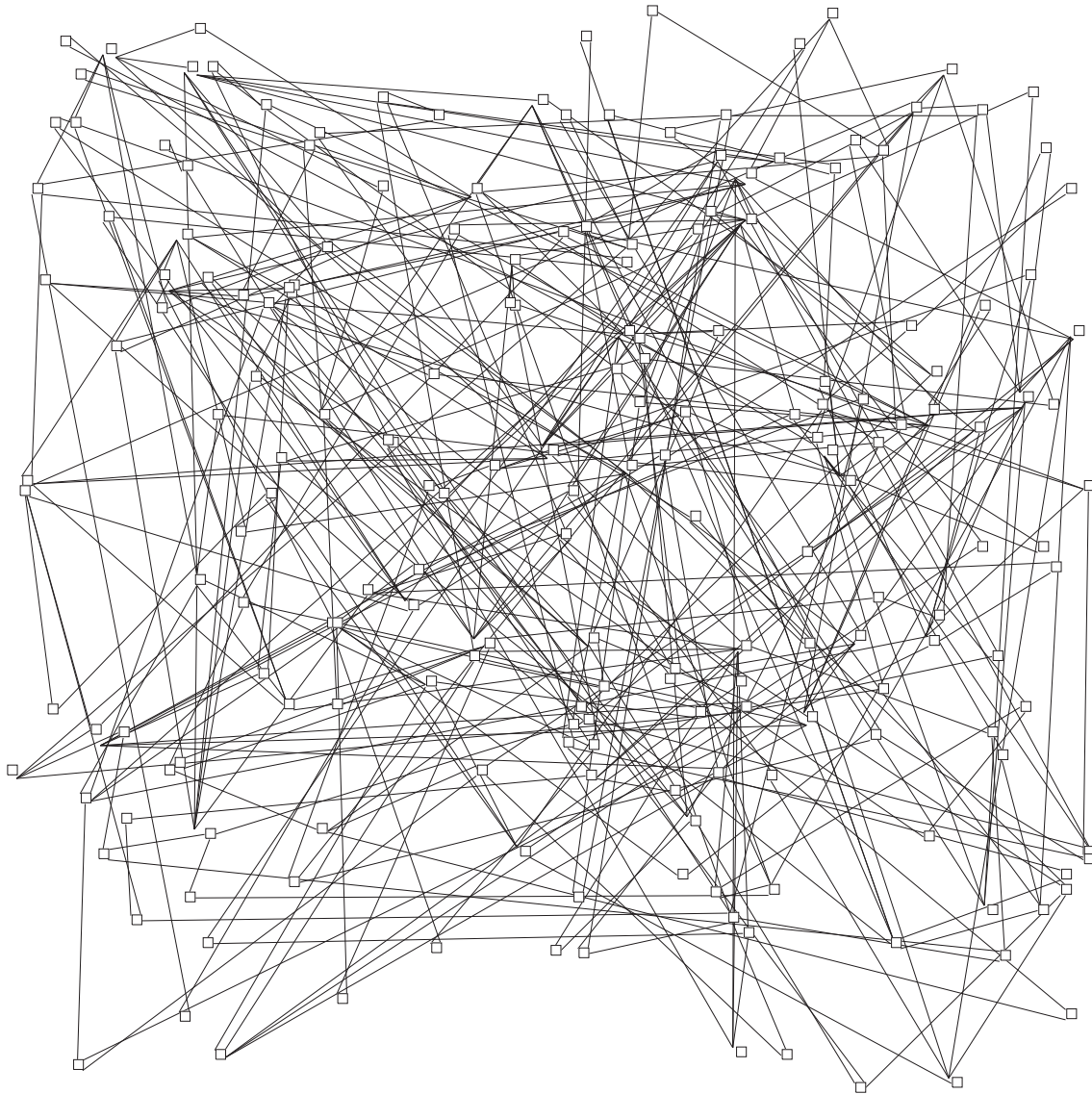


Abbildung 7.1: Netzwerk des Simulationsszenarios mit 200 Knoten und deren Netzwerkverbindung; das Netzwerk wurde mithilfe des CATNETS Szenariogenerators automatisiert erzeugt.



Tabelle 7.2: Konfiguration der Amazon Elastic-Cloud-2 (EC2) Services und die daraus abgeleitete Konfiguration des simulierten Ressourcenmarkts

Kriterium	Amazon EC2	Simulationsmodell
Anzahl Ressourcen-Services	5	5
Größe des Güterbündels	4 Ressourcen: Hauptspeicher, Festplattenspeicher, Recheneinheit, Prozessorarchitektur	1 Ressource: Recheneinheit
Geteilte Infrastruktur	ja	ja
minimale Nutzungsdauer	fest (1 Stunde)	keine
Nutzungsdauer	abhängig von der Nachfrage	abhängig vom Basic-Service
Preis verhandelbar	nein (Fixpreis)	ja (bilaterale Verhandlung)

konfiguration repräsentativ für ein breites Einsatzgebiet von Ressourcen-Services gelten kann.

Tabelle 7.2 zeigt die Konfiguration der Amazon EC2 Cloud-Computing-Services und das daraus abgeleitete Modell, das in der Simulationsstudie verwendet wird. Auf dem Ressourcenmarkt wird angenommen, dass es nicht nur einen Anbieter von Ressourcen-Services gibt, sondern die Konsumenten aus einer Reihe von Ressourcenanbietern auswählen können, die äquivalente Ressourcen-Services anbieten. Die Ressourcen-Services sind so weit standardisiert, dass sie zwischen verschiedenen Anbietern austauschbar sind.

Die Anzahl unterschiedlicher Ressourcen-Services, die ein Kunde auswählen kann, ist bei Amazon auf 5 festgelegt. Die gleiche Anzahl ist auch im Simulationsmodell verfügbar. Die Größe der Ressourcenbündel, die die Ressourcen-Services von Amazon beschreiben, ist durch 4 Einzelressourcen festgelegt. Hauptspeicher, Festplattenspeicher, Anzahl Recheneinheiten und die Prozessorarchitektur sind die Kenngrößen, die einen EC2-Service beschreiben. Das Simulationsmodell abstrahiert an dieser Stelle und betrachtet nur Ressourcenbündel, bestehend aus der Ressource Recheneinheiten. Es wird dabei angenommen, dass die Recheneinheiten die Hauptquelle des Risikos eines Ressourcenbündels darstellen und damit auch einen großen Einfluss auf das zu versichernde Risiko haben.

Sowohl im Amazon-EC2-Modell als auch im Simulationsmodell stellen die Ressourcenanbieter die Ressourcen-Services auf einer geteilten Infrastruktur bereit. Dies bedeutet, dass die Kapazität eines Ressourcenanbieters von mehreren Nachfragern genutzt werden kann, die unabhängig voneinander im System existieren können und als virtualisierte Ressourcen den Konsumenten zur Verfügung gestellt werden. Die Ressourcen des Anbieters bilden dabei eine *Cloud*

von Einzelressourcen, die dynamisch zu den benötigten Ressourcenbündelkonfigurationen der Ressourcen-Services gruppiert werden können.

Unterschiede zwischen dem Amazon-EC2-Modell und dem Simulationsmodell gibt es bezüglich der minimalen Nutzungsdauer der Ressourcen-Services. Alle Güterbündel von Amazon werden auf Stundenbasis angemietet, im Simulationsmodell gibt es keine minimale Nutzungsdauer. Da die Simulationsläufe in Normalzeit ablaufen, würde eine Nutzungsdauer von einer Stunde zu sehr langen Simulationszeiten führen. Wie lange die Nutzungsdauer eines bestimmten Ressourcen-Services ist, hängt sowohl bei Amazon wie auch im Simulationsmodell vom Konsumenten ab. EC2-Services können im Bereich von mehreren Stunden bis hin zu Monaten gemietet werden. Im Simulationsszenario werden die Zeiten des Ressourcenkonsums auf Millisekunden beschränkt, um eine möglichst große Anzahl an Transaktionen auf dem Ressourcenmarkt simulieren zu können. In allen Simulationsläufen schwankt die Nutzungsdauer der Ressourcen-Services zwischen 200 und 1000 Millisekunden.

Ein weiterer Unterschied zwischen dem Modell von Amazon-EC2 und dem Modell des Simulationsszenarios bezieht sich auf die Möglichkeit, die Preise der Ressourcen-Services zu verhandeln. Während bei Amazon die Preise für die Nutzung eines Ressourcenbündels feststehen, werden diese im Simulationsmodell durch eine bilaterale Verhandlung bestimmt. Für die Details der bilateralen Verhandlung wird auf das Kapitel 3 und die Dokumentation des CATNETS Projekts<sup>4</sup> verwiesen.

Die Nachfragerseite des Ressourcenmarkt umfasst 5 Basic-Service-Typen. Jeder dieser Basic-Service-Typen fragt ein bestimmtes Ressourcenbündel für eine bestimmte Zeit nach. Die Konfiguration der Basic-Services zeigt Tabelle 7.3. Eine Instanz eines Basic-Service vom Typ bs2 fordert beispielsweise immer ein Ressourcenbündel mit 4 standardisierten Recheneinheiten für 200 Millisekunden an. Der Reservationspreis des Nachfragers für dieses Ressourcenbündel beträgt 8 Geldeinheiten. Der Reservationspreis des Anbieters beträgt in diesem Fall 4 Geldeinheiten. Diese Preise stellen die Untergrenze bzw. die Obergrenze dar, zu der die Ressourcen-Service für bs2 gehandelt werden.

Die Gesamtanzahl der Basic-Services ist im Simulationsszenario auf 200 festgelegt, je 40 Basic-Services für jeden der in Tabelle 7.3 aufgeführten Basic-Service-Typen. Die Basic-Services befinden sich uniform auf dem Netzwerk verteilt. Die Anzahl der Nachfrager ist im Simulationsszenario 4-mal so groß wie die Anzahl der Ressourcenanbieter, d. h. im Durchschnitt bedient jeder Ressourcenanbieter 4 Basic-Services. Die Basic-Services nehmen dabei im Simulations-

---

<sup>4</sup><http://www.catnets.eu> (letzter Abruf: 17.07.2008)

Tabelle 7.3: Konfiguration der Nachfrage nach Ressourcen-Services durch Basic-Services und deren Reservationspreise

Basic-Service	Ressourcenmenge in [Recheneinheiten]	Ausführungszeit in [Millisekunden]	Reservationspreis (Anbieter) in [Geldeinheiten]	Reservationspreis (Nachfrager) in [Geldeinheiten]
bs1	1	1000	1	2
bs2	4	200	4	8
bs3	5	500	2	4
bs4	8	500	8	16
bs5	20	300	8	16

modell drei Rollen ein: Sie sind erstens Konsumenten auf dem Ressourcenmarkt (siehe oben), zweitens Anbieter auf der Servicemarkt, der im folgenden Abschnitt erläutert wird und drittens Nachfrager der Grid-Versicherung, die im darauf folgenden Abschnitt beschrieben wird.

### 7.1.3 Szenario des Servicemarkts

Der Servicemarkt dient im konkreten Szenario der Simulation der Nachfrage nach Basic-Services. Eine Evaluierung des Servicemarkts findet im Rahmen der Simulationsstudie nicht statt, da die zu evaluierende Grid-Versicherung nur auf dem Ressourcenmarkt agiert und keinen Versicherungstarif für den Servicemarkt anbietet.

Die Simulation der Nachfrage nach Basic-Services erfolgt über einen zweistufigen Prozess, der durch den Aufbau des Simulationsmodells vorgegeben ist:

- **Erster Schritt:** Ein simulierter Grid-Benutzer schickt mit einer zeitlichen Verzögerung von 1000 Millisekunden Anfragen nach Complex-Services an das simulierte Grid-System. Diese werden in die Warteschlange eines zufällig ausgewählten Complex-Service eingereiht. Wie viele Anfragen ein Grid-Benutzer in einem Simulationslauf an das System stellt, ist in den untersuchten Szenarien auf 20000 Anfragen festgelegt. Ein Simulationslauf mit 20000 Anfragen dauert mit der oben vorgestellten Basic-Service-Konfiguration zirka 5 Stunden und 30 Minuten.
- **Zweiter Schritt:** Ein Complex-Service sucht und allokiert einen freien Basic-Service eines bestimmten Typs (bs1, ..., bs5). Zur Vereinfachung der Allokation von Basic-Services akzeptiert der Complex-Service den Preis des Anbieters, ohne eine bilaterale Verhandlung zu

starten. Dieser Schritt verteilt die Anfragen der Grid-Benutzer über die Complex-Services auf freie Basic-Services.

Die Anzahl der Complex-Services im Netzwerk ist 200, die analog zu den 200 Basic-Services uniform im Netzwerk verteilt sind. Dies stellt sicher, dass jeder Basic-Service eine ähnliche Anzahl an Anfragen nach seinem Service erhält.

#### 7.1.4 Zuverlässigkeit der Basic-Service-Ausführung

Die Zuverlässigkeit der Basic-Service-Ausführung definiert die Häufigkeit einer erfolgreichen Ausführung eines bestimmten Basic-Services bei gegebener Konfiguration. Für die Modellierung der Zuverlässigkeit wird das Modell von Dai [DaPZ07] verwendet, das in Kapitel 2 eingeführt wurde.

Allgemein wird die Zuverlässigkeit  $p$  einer Basic-Service-Ausführung durch folgende Funktion bestimmt:

$$p = e^{-(n*t*k)},$$

wobei  $n$  die Anzahl der Recheneinheiten,  $t$  die Ausführungszeit des Basic-Service und  $k$  einen Skalierungsfaktor darstellt.  $k$  legt dabei die allgemeine Zuverlässigkeit des untersuchten Systems fest. In der Realität kann dieser Faktor beispielsweise durch eine unterschiedlich zuverlässige Hardware beeinflusst werden.

Es wird dabei angenommen, dass die Anzahl der Recheneinheiten und die Ausführungszeit signifikanten Einfluss auf die Zuverlässigkeit der Ressourcen haben. Belege hierfür finden sich in der empirischen Analyse des Grid'5000 von Iosup et al. [IJSE07]. Die Studie zeigt, dass die mittlere Zeit zwischen zwei Fehlern sowohl mit der Zeit als auch mit der Anzahl der Ressourcen zunimmt. Bei gegebenem konstanten Skalierungsfaktor wird die Zuverlässigkeit der Basic-Service-Ausführung durch die oben definierte Konfiguration der Basic-Services bs1 bis bs5 festgelegt. Jeder Basic-Service bekommt durch seine Konfiguration eine über die Zeit konstante Ausfallrate zugewiesen. Dies impliziert, dass jeder Ressourcenanbieter die Ausführung der Basic-Service mit der gleichen Zuverlässigkeit durchführt.

Durch Modifikation des Skalierungsfaktors werden zwei Risikoprofile für die Simulationsstudie festgelegt. Ein Risikoprofil zeigt eine höhere Zuverlässigkeit und damit ein geringeres Ausfallrisiko, während das zweite Risikoprofil eine deutlich geringere Zuverlässigkeit und damit ein

erhöhtes Ausfallrisiko aufweist. Die Simulationsläufe werden mit jeweils einem der beiden Risikoprofile ausgeführt und Auswirkungen der Risikoprofile auf die Ergebnisse analysiert. Es gilt ein homogenes Risikoprofil in einem Simulationslauf.

Die Abbildungen 7.2 und 7.3 zeigen den Verlauf der Risikoprofile für niedrige und hohe Zuverlässigkeit. Die Kreise stellen die Einordnung der Basic-Service-Konfigurationen auf das Risikoprofil dar. Das Profil mit niedrigem Risiko verwendet einen Skalierungsfaktor  $k = 5,0 * 10^{-5}$ . Die Zuverlässigkeit der Basic-Services schwankt hier zwischen 0,88 für bs5 und 0,98 für bs1. Durch die Verringerung des Skalierungsfaktors auf  $k = 2,0 * 10^{-5}$  reduziert sich die Zuverlässigkeit auf 0,97 für bs1 und 0,75 für bs5. Während sich die Zuverlässigkeit für bs1 zwischen den beiden Profilen nur gering unterscheidet, sinkt die Zuverlässigkeit für bs5 deutlich. Der Effekt der Veränderung des Skalierungsfaktors wirkt sich deutlich stärker auf die Basic-Services aus, die eine größere Menge an Recheneinheiten benötigen.

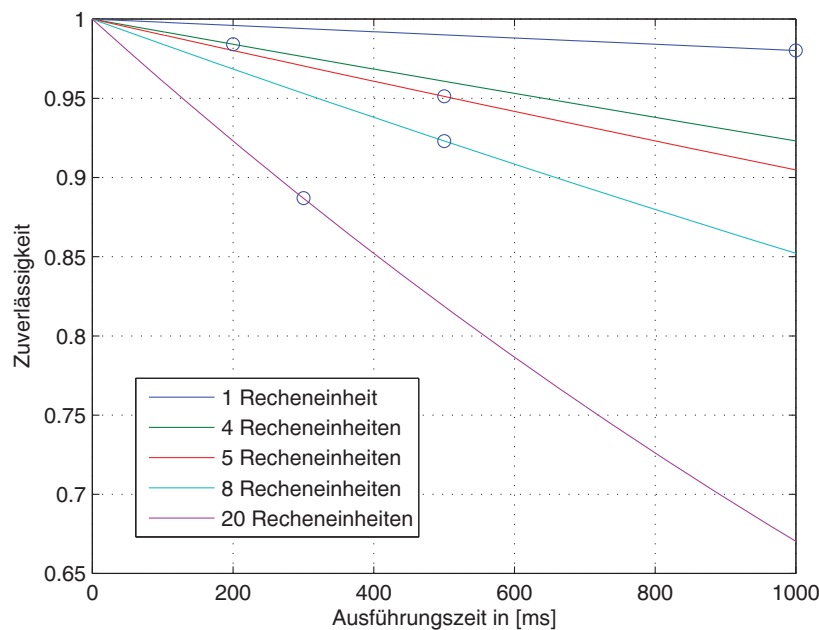


Abbildung 7.2: Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor  $k = 5,0 * 10^{-5}$  (niedriges Risikoprofil)

Die Definition der Risikoprofile erfolgt unter Verwendung der Studien zur Messung des technischen Risikos, die in Kapitel 2 vorgestellt wurden. Zukünftige Erkenntnisse können durch Anpassen des Skalierungsfaktors und/oder der Basic-Service-Konfiguration durchgeführt werden. Bisher sind diese Daten zu wenig detailliert und beziehen sich auf keine serviceorientierten

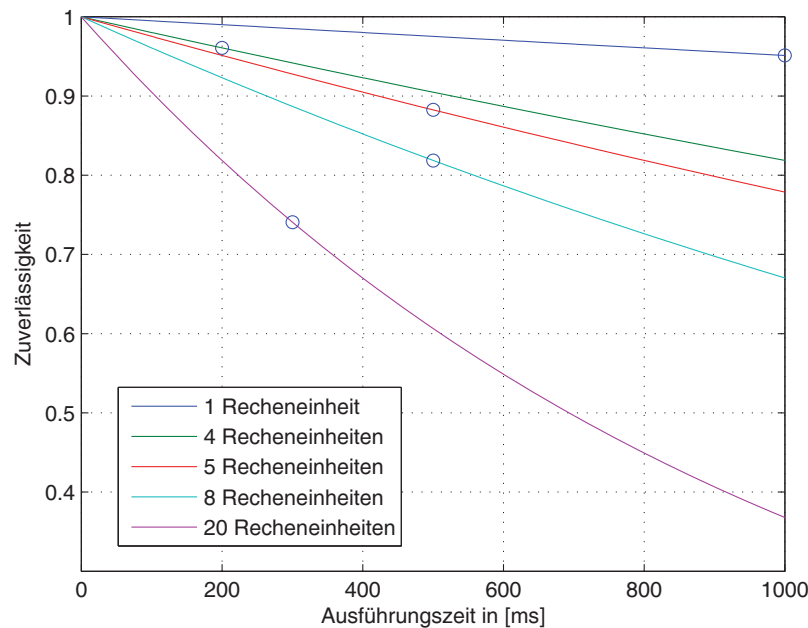


Abbildung 7.3: Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor  $k = 2,0 * 10^{-5}$  (hohes Risikoprofil)

Grid-Systeme wie sie in dieser Arbeit zu Grunde liegen. Ebenfalls liegen bisher keine Daten von kommerziellen Anbietern vor, sodass bei neueren Erkenntnissen eine Anpassung der Risikoprofile notwendig werden kann. Damit ergibt sich eine einfache Möglichkeit, neue Risikoprofile in das Simulationsmodell zu integrieren.

### 7.1.5 Versicherungstarif für die Basic-Service-Ausführung

Der implementierte Versicherungstarif des Simulationsszenarios versichert die Ausführung von Basic-Services auf Ressourcen, die durch einen Ressourcen-Service bereitgestellt werden. Die Auswahl der Tarifmerkmale orientiert sich an den beiden Merkmalen des Basic-Service, die auch für die Berechnung des Risikos verwendet werden. Dies sind zum einen die Anzahl der Recheneinheiten und zum anderen die Länge der Ausführungszeit. Der Versicherungstarif ist als zweifach zu klassifizierender Tarif implementiert. Für jedes Tarifmerkmal stehen 4 Klassen zur Verfügung, in die Basic-Services anhand ihrer Merkmalsausprägungen von der Versicherung eingeordnet werden. Der Versicherungstarif und die Einordnung der 5 Basic-Services in die Versicherungsklassen zeigt Abbildung 7.4.

Ausführungszeit	Recheneinheiten			
	Klasse 1: [1,3]	Klasse 2: [4]	Klasse 3: [5,7]	Klasse 4: [8,20]
Klasse 1: [100,210]		bs2 (4 RE, 200ms) 4-8		
Klasse 2: [220,310]				bs5 (20 RE, 300ms) 8-16
Klasse 3: [320,510]			bs3 (5 RE, 500ms) 2-4	bs4 (8 RE, 500ms) 8-16
Klasse 4: [520,1010]	bs1 (1 RE, 1000ms) 1-2			

Abbildung 7.4: Zweifach klassifizierender Versicherungstarif zur Versicherung der Basic-Service-Ausführung

Die Festlegung der Tarifklassen verwendet die Größe der erwarteten Schäden, die für die einzelnen Basic-Services auftreten können, um die Intervallgrenzen der Tarifklassen zu definieren. Diese werden in eckigen Klammern hinter den jeweiligen Tarifklassennamen in Abbildung 7.4 angegeben. Für das Tarifmerkmal *Ausführungszeit* gilt die Einheit Millisekunden, die Angaben für das Tarifmerkmal *Recheneinheiten* werden in Recheneinheiten quantifiziert. Durch die Festlegung auf zwei Tarifmerkmale und 4 Tarifklassen lassen sich maximal 16 verschiedene Basic-Service-Kombinationen versichern. Die 5 Basic-Services des Simulationsszenarios besetzen 5 dieser Felder, 11 Felder der Tarifstruktur bleiben frei<sup>5</sup>.

Die dünn besiedelte Matrix des Versicherungstarifs hat Auswirkung auf die Menge der Daten, die zur Schätzung der Tariffaktoren verwendet werden. Da es sich um einen multiplikativen Tarif handelt, bei dem die Tariffaktoren einer Zeile mit dem Tariffaktor einer Spalte multipliziert werden, lassen sich im vorliegenden Szenario mehrere Sonderfälle unterscheiden:

- bs1, bs2: Sowohl bs1 als auch bs2 haben in ihrer jeweiligen Spalte und Zeile keinen weiteren Basic-Service. Die Gruppe, die zur Ermittlung der Tariffaktoren herangezogen wird,

<sup>5</sup>Technisch bedingt lässt sich mit dem CATNETS Simulator keine größere Anzahl an Basic-Services simulieren. Eine Vergrößerung der Basic-Service-Anzahl führte in Testläufen zu instabilem Verhalten der Simulationsplattform. Durch die technische Limitierung müssen im Simulationsszenario 8 Tariffaktoren für 5 Basic-Services ermittelt werden. Ein Versicherungsansatz, der die Erwartungswerte der einzelnen Basic-Services schätzt, würde hier leichte Vorteile haben und nur 5 Werte ermitteln müssen. Bei steigender Anzahl an Basic-Services ist dieser Vorteil jedoch schnell wieder verloren.



besteht aus nur einem Element. In diesen beiden Fällen erfolgt die Schätzung der Tariffaktoren und damit der Versicherungsprämie unter Einbezug der Anzahl der Versicherten und des Schadens dieser Basic-Services.

- bs3, bs5: Die Basic-Services vom Typ bs3 und bs5 bilden entweder in der Zeile oder in der Spalte eine Gruppe mit einem weiteren Basic-Service eines anderen Typs. In beiden Fällen wird einer der beiden Tariffaktoren für eine Gruppe von Basic-Services berechnet. Der jeweils andere Tariffaktor wird durch einen einzelnen Basic-Service festgelegt.
- bs4: Die Tariffaktoren für Basic-Service 4 bilden sowohl in der Zeile als auch in der Spalte der Tarifmatrix eine Risikogruppe mit einem anderen Basic-Service-Typ.

Diese, durch die Konstruktion des Tarifs bedingte Gruppierung der Basic-Services dient als Vorlage für die Untersuchung auftretender Einflüsse des Versicherungstarifs auf die Versicherungsprämien für die einzelnen Basic-Services in Abschnitt 7.2.3. Die Frage, ob und welche Auswirkungen die Konstruktion des Versicherungstarifs auf die Prämien der Basic-Services hat, wird in diesem Abschnitt näher erläutert.

## 7.2 Simulationsergebnisse Teil 1: Dynamik der Versicherung

Der erste Teil der Simulationsstudie untersucht die Dynamik der Grid-Versicherung im oben eingeführten Simulationsszenario. Detailliert werden in den folgenden Abschnitten die Fragen, die während der Phase des Simulationsentwurfs in Kapitel 6 formuliert wurden, anhand verschiedener Simulationsläufe untersucht und die Ergebnisse sollen analysiert werden. Beginnend mit einzelnen Parametern des Grid-Versicherungsservice und seiner Entwicklung über die Zeit werden im weiteren Verlauf dieses Abschnitts die Versicherungsnachfrage auf Ebene der Gesamtpopulation und auf Ebene der Basic-Services untersucht.

### 7.2.1 Versicherungsperiode und Entwicklung der Tariffaktoren

Die Versicherungsperiode der Grid-Versicherung legt fest, wie häufig die Versicherungsprämie aktualisiert wird. Die Aktualisierung der Prämien erfolgt durch die Berechnung der Tariffaktoren nach dem Marginalsummenverfahren, das in Kapitel 5 eingeführt wurde. In diesem Zusammenhang stellt die Versicherungsperiode einen wichtigen Konfigurationsparameter dar, der die Anpassungsgeschwindigkeit der Prämien an den Schadensverlauf festlegt. Ziel der folgenden



Tabelle 7.4: Grundlegende Simulationsparameter des Szenarios

Simulationsparameter	Wert
Anzahl Benutzeranfragen	20.000
Wahrscheinlichkeit der Anfrage eines Basic-Service vom Typ $[bs1, \dots, bs5]$	bs1=0,2; bs2=0,2 bs3=0,2; bs4=0,2; bs5=0,2
Verzögerung zwischen zwei Benutzeranfragen	1.000 ms
Startwert der Tariffaktoren	0,5

Tabelle 7.5: Simulationsparameter zur Untersuchung der Versicherungsperiode unter verschiedenen Risikoprofilen

Simulationsparameter	Lauf 1	Lauf 2	Lauf 3	Lauf 4
Versicherungsperiode	8.000 ms	100.000 ms	8.000 ms	100.000 ms
Risikoprofil	$k = 5,0 * 10^{-5}$	$k = 5,0 * 10^{-5}$	$k = 2,0 * 10^{-5}$	$k = 2,0 * 10^{-5}$

Untersuchung ist die Analyse der Auswirkungen des Konfigurationsparameters *Versicherungsperiode* der Grid-Versicherung.

Tabelle 7.4 zeigt die grundlegende Konfiguration des Simulationsszenarios. Jeder Simulationslauf wird mit 20.000 Benutzeranfragen gestartet, wobei jeder der 5 Basic-Service-Typen mit einer Wahrscheinlichkeit von 20% nachgefragt wird. In Benutzeranfragen ausgedrückt, bedeutet dies etwa 4.000 Anfragen je Basic-Service-Typ. Zwischen den einzelnen Anfragen wird eine Verzögerung von 1.000 ms festgelegt, die verantwortlich für eine hohe Systemlast ist. Die Tariffaktoren der Grid-Versicherung werden zu Beginn der Simulation mit 0,5 angegeben. Die sehr niedrig gewählten Tariffaktoren führen zu sehr günstigen Prämien am Beginn der Simulation und ermöglichen es allen Basic-Services, eine Versicherung abzuschließen. Die Entwicklung der Tariffaktoren wird im Laufe dieses Kapitels unter verschiedenen Konfigurationen näher betrachtet.

Die Parameter der Simulationsläufe sind in Tabelle 7.5 aufgeführt. Neben dem Parameter *Versicherungsperiode* mit den beiden Wertausprägungen 8.000 ms und 100.000 ms werden die oben eingeführten Risikoprofile *niedriges* ( $k = 5,0 * 10^{-5}$ ) und *hohes Risiko* ( $k = 2,0 * 10^{-5}$ ) mit jeweils 20.000 Benutzeranfragen simuliert. Eine Versicherungsperiode mit 8.000 ms entspricht dem 8- bis 40-Fachen der Ausführungszeiten der Basic-Services; eine Versicherungsperiode mit 100.000 ms hat die 100- bis 500-fache Länge der Ausführungszeiten. Dies bedeutet, dass die längere Versicherungsperiode eine deutlich größere Menge an Schäden anhäuft, bevor eine Neuberechnung der Prämien stattfindet.

Die Untersuchung vergleicht Simulationsläufe sowohl für die beiden Versicherungsperioden als auch für die beiden Risikoprofile. Im Zentrum der Analyse stehen die Tariffaktoren der 4 Merkmalsausprägungen eines Tarifmerkmals, die die Schäden einer Risikogruppe abbilden und die Grundlage der Prämienberechnung darstellen. Abbildung 7.5 zeigt die Zeitserien der Tariffaktoren für eine Versicherungsperiode mit 8.000 Millisekunden für die Risikoprofile *hohes und niedriges Risiko*, gegliedert nach den beiden Tarifmerkmalen Ausführungszeit und Recheneinheiten.

Zu Beginn der Simulationen zeigen die Tariffaktoren in beiden Simulationsläufen starke Schwankungen. Die Datenbasis der Versicherung reicht hier noch nicht aus, um eine stabile Schätzung des Risikos durchzuführen. Die initial festgelegten Tariffaktoren erweisen sich als zu niedrig und werden abhängig von den gemeldeten Schäden deutlich erhöht. Mit zunehmender Simulationsdauer bildet sich ein Wertebereich heraus, der – je nach Tariffaktor – unterschiedlich starken Schwankungen unterworfen ist. Dieser Wertebereich bleibt für die einzelnen Tariffaktoren in Szenarien mit hohem Risiko stabil, während der Wertebereich in der Simulation mit dem niedrigem Risiko leicht abnimmt. Das oszillierende Verhalten der Tariffaktoren verschwindet jedoch nicht, sondern ist je nach Tariffaktor und Szenario unterschiedlich stark ausgeprägt. Ein Vergleich der Breite der Wertebereiche zeigt, dass diese im Szenario mit hohem Risiko stärker ausgeprägt sind. Mit der Abnahme des Risikos nimmt auch das oszillierende Verhalten der Tariffaktoren ab. Die Höhe der Tariffaktoren ist im Szenario mit hohem Risiko ebenfalls größer, was die Schlussfolgerung zulässt, dass die Versicherung auf das erhöhte Risiko einer Basic-Service-Ausführung mit einer erhöhten Prämie reagiert.

Ein ähnliches Verhalten zeigen die Simulationsläufe mit einer Versicherungsperiode von 100.000 Millisekunden, die in Abbildung 7.6 dargestellt sind. Zu Beginn der Simulation zeigen sich ebenfalls starke Schwankungen der Tariffaktoren, nach längerem Fortgang der Simulation bilden sich stabile Wertebereiche heraus, die den Bereich markieren, in dem die Tariffaktoren schwanken. Die Breite des Wertebereichs ist jedoch deutlich größer als der Wertebereich bei einer Versicherungsperiode mit 8.000 Millisekunden bei gleicher Risikokonfiguration. Dies deutet darauf hin, dass die Versicherung einer deutlich höheren Schwankung der Schäden ausgesetzt ist als die Versicherung mit kürzerer Versicherungsperiode. Ein Vergleich der Breite der Wertebereiche der Tariffaktoren gibt Hinweise darauf, dass auch die Versicherungsprämien im Maximum deutlich teurer sind als im Szenario mit einer Versicherungsperiode von 8.000 Millisekunden. Ähnlicher ist das Verhalten zwischen den beiden Konfigurationen im Szenario mit niedrigem Risiko. Hier zeigen die Zeitserien der Tariffaktoren Ähnlichkeiten in der Breite der Wertebereiche wie auch in der Höhe der Werte.

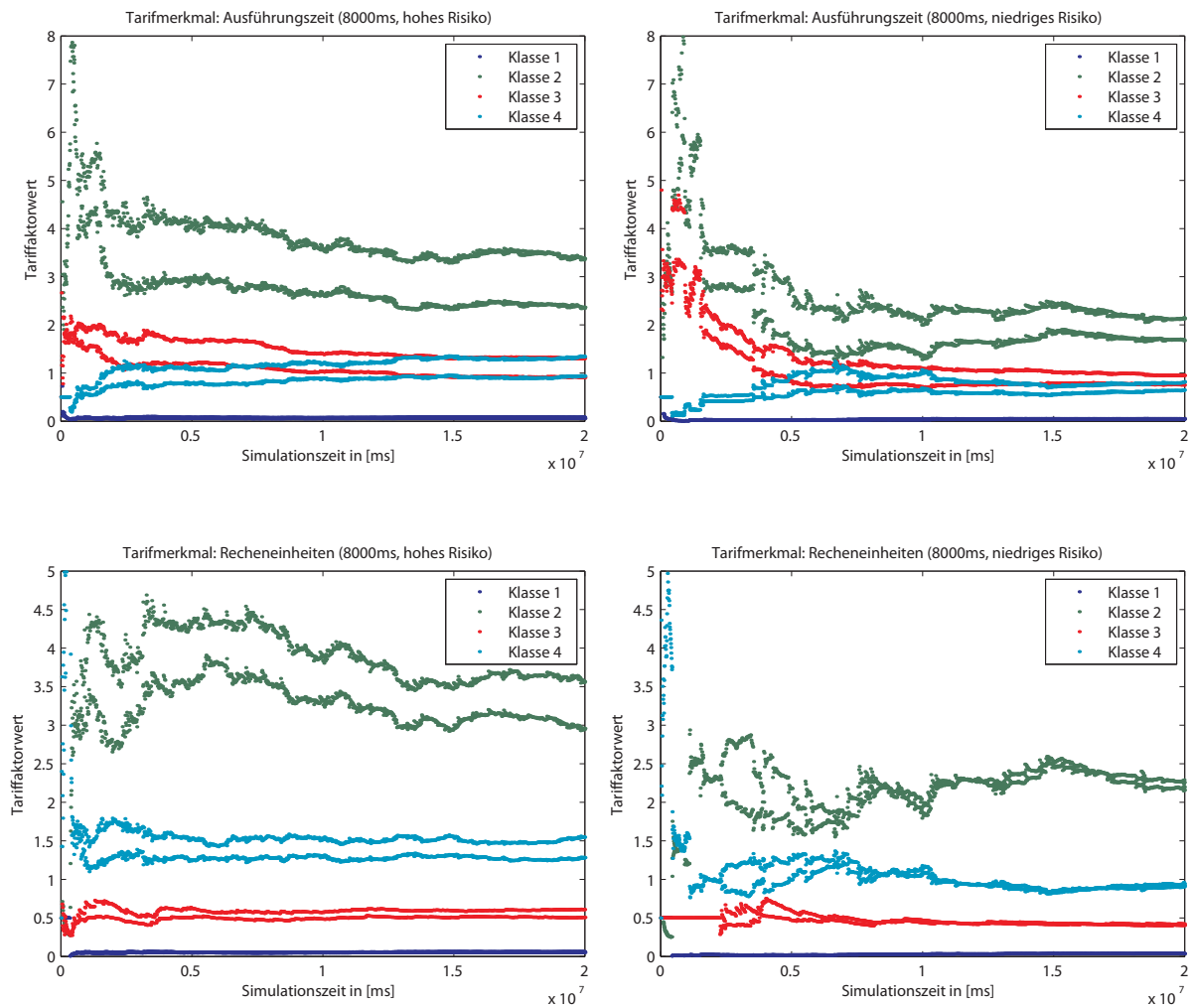


Abbildung 7.5: Zeitserie der Tariffaktoren für eine Versicherungsperiode mit 8.000 Millisekunden und die Risikoprofile *hohes* und *niedriges Risiko*

Alle untersuchten Simulationsläufe zeigen die Herausbildung eines Korridors, der abhängig von der Schadensstatistik ist. Der für alle Tariffaktoren initial gleiche Wert von 0,5 wird durch die Grid-Versicherung nach einer Einschwingphase mit teilweise starken Ausschlägen dem technischen Risiko angepasst.

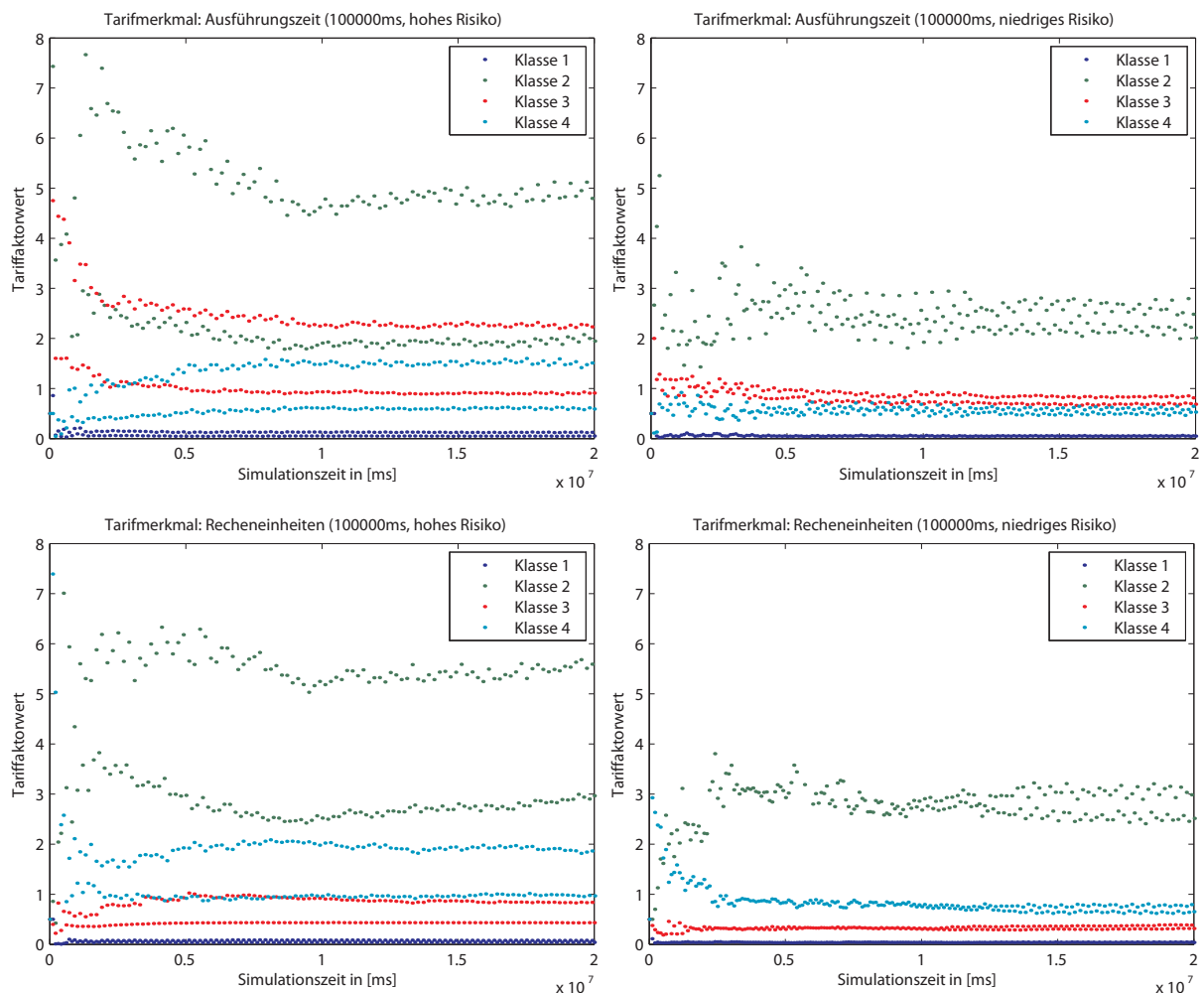


Abbildung 7.6: Zeitserie der Tariffaktoren für eine Versicherungsperiode von 100.000 Millisekunden und die Risikoprofile *hohes und niedriges Risiko*

Um Rückschlüsse aus den Zeitserien der Tariffaktoren auf das Verhalten des Systems zu bekommen, werden die Zeitserien der Schäden für die jeweiligen Szenarien detaillierter betrachtet. Allgemein lässt sich das oszillierende Verhalten der Tariffaktoren und damit der Versicherungsprämien auf zwei Einflussfaktoren zurückführen.

- Eine hohe Schwankung der versicherten Schäden zwischen den Perioden: Die Anfragen nach bestimmten Basic-Services werden zufällig erzeugt, sodass es zu einer gehäuften

Nachfrage nach bestimmten Basic-Services in einer Versicherungsperiode kommen kann. Eine Häufung in der Nachfrage nach bestimmten Grid-Services und Grid-Ressourcen lässt sich auch in realen Grid-Systemen beobachten, wie Messungen verschiedener Systeme ergeben haben [ILJA<sup>+</sup>07]. Eine erhöhte Nachfrage kann zu einem erhöhten Schaden in dieser Periode führen, dies unter der Voraussetzung, dass die Basic-Services ihre Serviceausführung versichert haben. Dies führt wiederum zu einer verringerten Schadenssumme in einer der folgenden Versicherungsperioden – wiederum unter der Voraussetzung, dass die Basic-Services ihre Serviceausführung versichert haben –, da alle Basic-Services zu einem gleichen Anteil nachgefragt werden und deshalb zu gleichen Anteilen auch Schäden produzieren. Dieser Effekt wird zusätzlich noch weiter dadurch verstärkt, dass die Anzahl der Versicherten durch die Anzahl an Basic-Services technisch bedingt mit 200 Agenten sehr klein ist.

- Das Auftreten von Spätschäden: Durch eine hohe Systemlast können Nachrichten, die eine Schadensmeldung an die Versicherung enthalten, verspätet von der Grid-Versicherung verarbeitet werden. Im Detail sind hier Verzögerungen beim Nachrichtentransport über das Netzwerk und die Wartezeit der Nachrichten in der Warteschlange der Versicherung zu nennen, die einen Spätschaden hervorrufen können. Ebenfalls kann die Analyse von Messdaten zur Bestimmung eines Schadenseintritts zu einer verzögerten Schadensmeldung führen. Verspätet bedeutet in diesem Zusammenhang, dass die Bezahlung der Versicherungsprämie und die Abwicklung des Schadens in verschiedenen Perioden stattfinden und die Versicherung die Prämien dadurch auf fehlerbehafteten Annahmen berechnet<sup>6</sup>.

Ein mögliches Spätschadenszenario könnte wie folgt aussehen: In der Periode  $t - 1$ , in der die Prämie bezahlt wird und kein Schaden gemeldet wurde, sind weniger Schäden angefallen und die Versicherung senkt für die folgenden Periode die Prämie. Jedoch fallen in der folgenden Periode  $t$  nun verstärkt Schäden an, sodass sie wiederum die Prämien erhöhen muss. Dieser Prozess kann ein oszillierendes Verhalten der Tariffaktoren hervorrufen.

Diese beiden Effekte werden im Folgenden hinsichtlich ihres Einflusses auf die beobachtete Entwicklung der Tariffaktoren untersucht. Grundlage der Analyse bilden die Schwankungen der Schadenssummen aufeinander folgender Perioden über die Zeit. Abbildung 7.7 zeigt die Zeitserie für eine Versicherungsperiode mit 100.000 ms und das Risikoprofil *hohes Risiko*. Auf der x-Achse ist die Anzahl der simulierten Versicherungsperioden abgetragen und auf der y-Achse

---

<sup>6</sup>In der Versicherungswirtschaft finden sich Beispiele für Spätschäden in speziellen Haftpflichtversicherungen, in denen eine Haftung für Schäden aus vergangenen Jahren anfallen kann, da der Schaden erst verspätet festgestellt wurde.

die beobachtete Schadenssumme einer Versicherungsperiode. Die Schadenssummen in Abbildung 7.7 entsprechen den Zeilensummen der Merkmalsausprägungen des Tarifmerkmals *Ausführungszeit*.

Der Verlauf der Schadenssummen zeigt eine hohe Volatilität zwischen den einzelnen Perioden. Einer Periode mit hohem Schadensaufkommen folgt sehr häufig eine Periode mit keinem Schaden in der entsprechenden Merkmalsausprägung. Die Merkmalsausprägungen Klasse 1 und 4 zeigen weniger starke Schwankungen, die bedingt sind durch ihre höhere Zuverlässigkeit und dem geringeren Preis. Die Merkmalsausprägungen Klasse 2 und Klasse 3 zeigen in diesem Szenario die größten Schwankungen. In diesen Klassen sind die Basic-Services vertreten, die zum einen die geringste Zuverlässigkeit haben und zu den höchsten Preise gehandelt werden. Ein Vergleich zwischen den Schwankungen der Schadenssummen und den dazugehörigen Tariffaktoren zeigt deutlich, dass die Schwankungen der Schadenssummen positiv mit den Schwankungen der entsprechenden Tariffaktoren korreliert sind. Die Breite des Korridors, in dem die Tariffaktoren schwanken, zeigt an, wie stark sich die Schadenssummen zwischen den Versicherungsperioden unterscheiden.

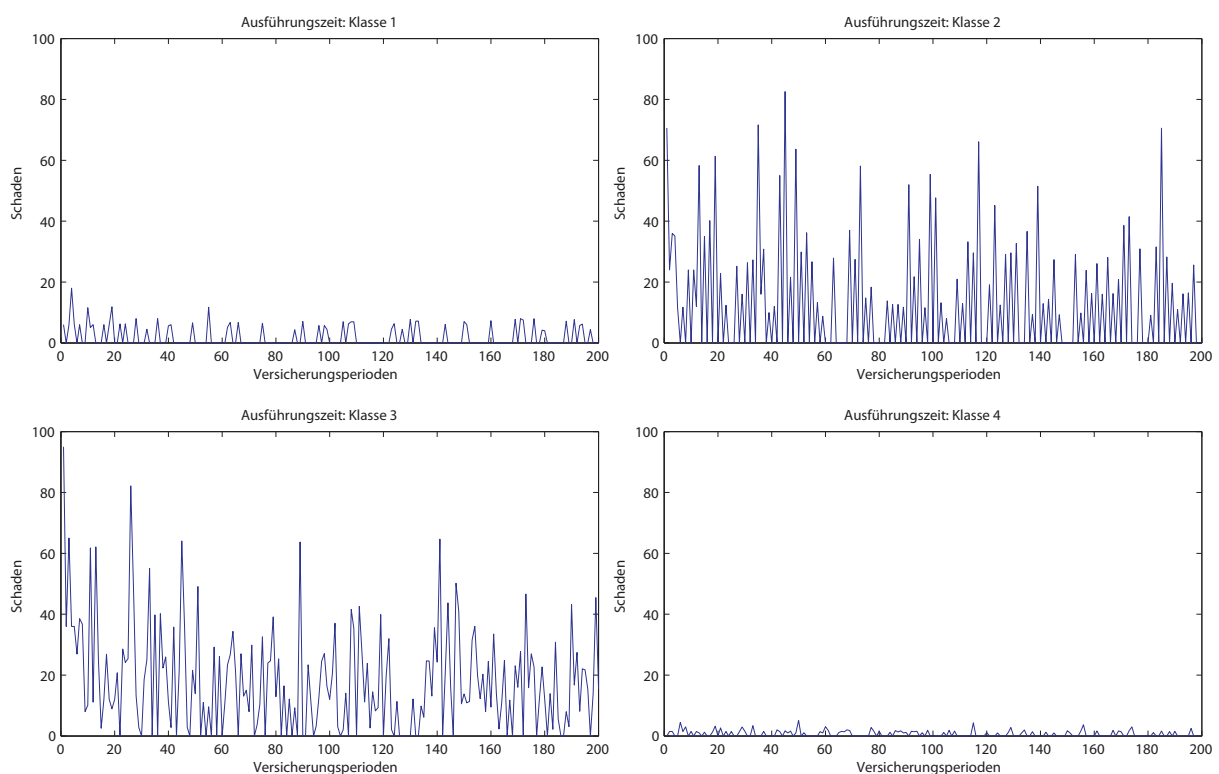


Abbildung 7.7: Zeitserie der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals *Ausführungszeit* mit Risikoprofil *hohes Risiko* und 100.000 ms Versicherungsperiode

Die Zeitserie der Schadenssummen für die Versicherungsperiode mit 8.000 ms zeigt Abbildung 7.8. Im Vergleich zu Abbildung 7.7 bleibt das Risikoprofil gleich, nur die Versicherungsperiode verkürzt sich. Da es sich bei beiden Simulationsläufen, um das gleiche Risikoprofil handelt, kann davon ausgehen werden, dass eine ähnlich hohe Schadenssumme in beiden Simulationsläufen auftritt. Sind bei einer Versicherungsperiode mit 100.000 ms Schadensaufkommen bis 95 Geldeinheiten zu beobachten, so verringern sich diese Ausschläge deutlich auf unter 40 Geldeinheiten. Perioden ohne Schäden lassen sich in beiden Szenarien beobachten. Die Höhe des Schadensaufkommens einer Periode kann durch die Verkürzung derselben deutlich verringert werden. Die Schadenssumme wird auf mehrere Perioden verteilt, sodass es zu weniger starken Ausschlägen kommt. Das weniger stark oszillierende Verhalten der Tariffaktoren bei verkürzter Versicherungsperiode verringert sich durch das geringere Schadensaufkommen in den Versicherungsperioden.

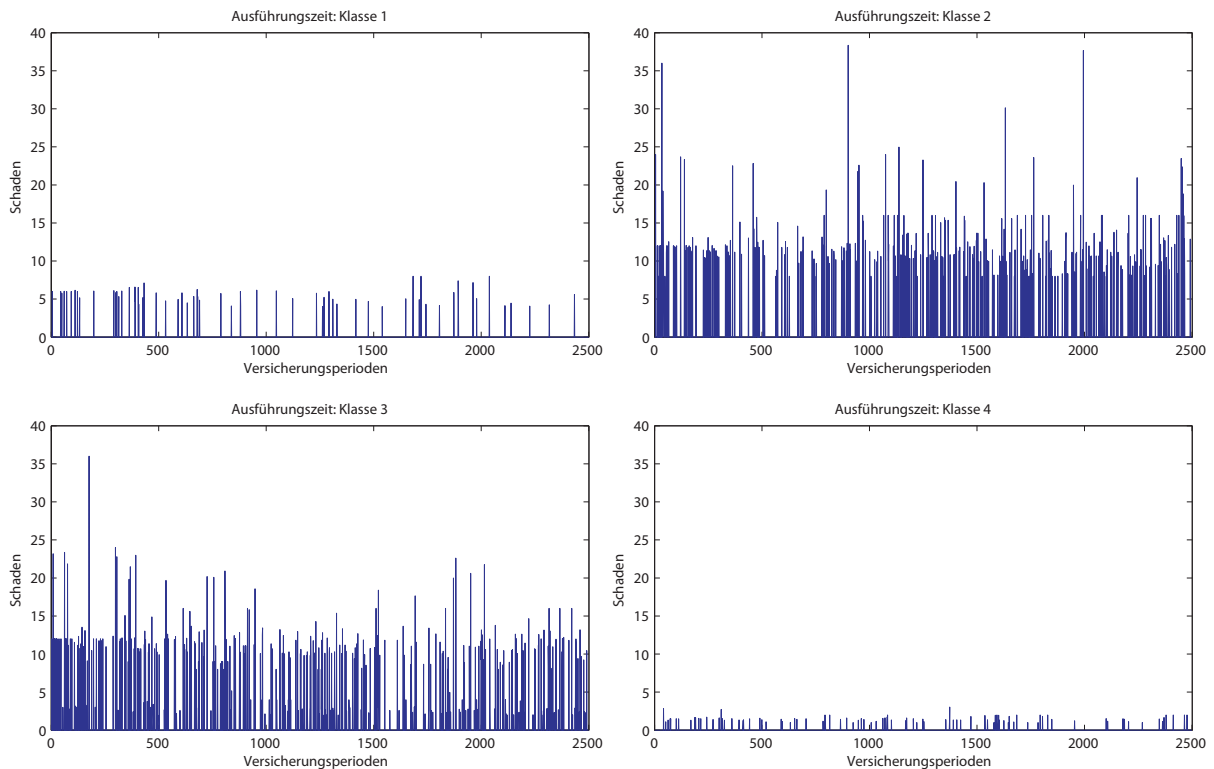


Abbildung 7.8: Zeitserie der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals *Ausführungszeit* mit Risikoprofil *hohes Risiko* und 8.000 ms Versicherungsperiode

Jedoch treten vermehrt Spätschäden bei verkürzter Versicherungsperiode auf wie Tabelle 7.6 zeigt. Liegt der Anteil der Spätschäden bei einer Versicherungsperiode von 100.000 ms deutlich unter einem Prozent, so steigt der Anteil der Spätschäden auf fast 10% in einem Szenario



Tabelle 7.6: Anteil Spätschäden für die untersuchten Simulationskonfigurationen

Simulationskonfiguration	Anteil Spätschäden
8000 ms, hohes Risiko	0.0943
8000 ms, niedriges Risiko	0.0400
100000 ms, hohes Risiko	0.0040
100000 ms, niedriges Risiko	0.0072

mit hohem Risiko und einer Versicherungsperiode von 8.000 ms. Alle gemessenen Spätschäden fallen in der folgenden Periode an; Spätschäden, die später auftreten, konnten nicht gemessen werden. Die erhöhte Anzahl an Spätschäden ist verantwortlich dafür, dass es in der Periode, in der die Spätschäden anfallen, zu Ausreißern bei der Schadenssumme kommt. Aus Sicht der Tariffaktoren spielen jedoch die Spätschäden eine untergeordnete Rolle, da es in der Zeitserie der Tariffaktoren keine starken Ausschläge gibt, die auf die Spätschäden zurückzuführen sind. Den dominanten Einfluss auf das oszillierende Verhalten des Tariffaktors übt das stark schwankende Verhalten der Schadenssummen aus. In allen weiteren Simulationsläufen wird daher eine Versicherungsperiode von 8.000 Millisekunden verwendet.

### 7.2.2 Sicherheitszuschlag

Der Sicherheitszuschlag zur Prämie bietet der Grid-Versicherung die Möglichkeit, sowohl (im Durchschnitt) Schäden abzudecken als auch die Preise für die Versicherungspolice zu steuern. Mithilfe des Sicherheitszuschlags wird die Strategie der Versicherung festgelegt. Aus diesem Grund ist der Sicherheitszuschlag neben der Versicherungsperiode der zweite wichtige Konfigurationsparameter der Grid-Versicherung. Der Preis einer Versicherungspolice  $p_{vp}$  setzt sich aus der Prämie  $p_{pr}$  und dem Sicherheitszuschlag  $sz$  wie folgt zusammen:

$$p_{vp} = p_{pr} * sz$$

Der Sicherheitszuschlag ist in der Grid-Versicherungsimplementierung als multiplikativer Faktor implementiert. Ein Wert von 1.0 entspricht einer fairen Prämie, die auf dem Erwartungswert des Schadens basiert. Werte  $\geq 1.0$  können beliebig – je nach Definition der Strategie der Versicherung – gewählt werden. Die Implementierung der Grid-Versicherung unterstützt zwei verschiedene Möglichkeiten, die Strategie einer Versicherung umzusetzen:



- **Statischer Sicherheitszuschlag:** Der Sicherheitszuschlag wird zu Beginn eines Simulationslaufs festgelegt und bleibt während eines Simulationslaufs konstant. Die Grid-Versicherung erhebt dadurch unabhängig von externen oder internen Einflüssen einen prozentualen Anteil an der Prämie, der dem Aufbau von Kapital oder dem Abdecken von Schäden dienen kann.
- **Dynamische Anpassung des Sicherheitszuschlags:** Diese Variante bietet die Möglichkeit, die Strategie der Versicherung in Abhängigkeit des Kapitals der Versicherung zu definieren. Für das Kapital der Versicherung werden Schwellenwerte definiert, bei deren Über- oder Unterschreiten der Sicherheitszuschlag angepasst wird. Fällt das Kapital der Versicherung, wird der Sicherheitszuschlag erhöht, um den Ruin der Versicherung zu vermeiden. Steigt das Kapital der Versicherung wieder über einen Schwellenwert, so wird der Sicherheitszuschlag reduziert, da der Ruin der Versicherung weniger akut ist. Dieser Prozess der zu- und abnehmenden Sicherheitszuschläge kann wie ein Bonus-Malus-System der Grid-Versicherung interpretiert werden. Die Versicherungsnehmer werden durch die Reduzierung des Sicherheitszuschlags und damit des Preises der Versicherungspolice von der Versicherung belohnt, wenn das Kapital der Versicherung stark zugenommen hat.

Eine Strategie der Versicherung mit einem statischen Sicherheitszuschlag weist eine Reihe von Problemen auf: Zum einen kann die Versicherung auf Schwankungen ihres Kapitals nicht reagieren und zum anderen muss der Sicherheitszuschlag für jedes Szenario neu eingestellt werden. Dies führt zu deutlich höherem Zeitaufwand, der benötigt wird, den Sicherheitszuschlag einzustellen. Der statische Sicherheitszuschlag dient in den folgenden Analysen als Referenzszenario der Strategie mit einem dynamischen Sicherheitszuschlag.

Der Strategie der Versicherung mithilfe eines dynamischen Sicherheitszuschlags zielt darauf ab, das zu Beginn der Simulation festgelegte Kapital der Versicherung über die Zeit zu halten und einen Ruin der Versicherung zu vermeiden. Es werden keine Gewinnvorgaben seitens der Grid-Versicherung modelliert, um die Komplexität möglichst gering zu halten. Tabelle 7.7 zeigt die Konfiguration der Schwellenwerte und die dazugehörigen Sicherheitszuschläge. Für ein gegebenes Startkapital  $K(0)$  von 400 Geldeinheiten werden 5 Schwellenwerte festgelegt. Die Überprüfung der Schwellenwerte ist an die Versicherungsperiode gekoppelt; sie findet nach der Anpassung der Versicherungsprämien statt. Der Grid-Versicherungsagent vergleicht das Kapital  $K$  in Periode  $t - 1$  mit den vorgegebenen Schwellenwerten und entscheidet, ob eine Anpassung des Sicherheitszuschlags vorgenommen werden muss. Fällt das Kapital unter 100 Geldeinheiten, so muss in der vorliegenden Konfiguration die doppelte Prämie bezahlt werden, steigt hingegen das

Tabelle 7.7: Definition von Schwellenwerten für die Anpassung des Sicherheitszuschlags.

Schwellenwert	Sicherheitszuschlag
$K(t-1) < 100$	2,0
$100 \leq K(t-1) < 300$	1,9
$300 \leq K(t-1) < 400$	1,7
$400 \leq K(t-1) < 500$	1,3
$500 \leq K(t-1)$	1,0

Kapital über 500 Geldeinheiten, so entfällt der Sicherheitszuschlag.

Abbildung 7.9 zeigt die Zeitserie des Prämieinnahme- und Schadensprozesses für drei ausgewählte Simulationsläufe. Der Simulationslauf mit dem Subskript `_n_1_5` verwendet einen statischen Sicherheitszuschlag über die Zeit, während die beiden anderen Simulationsläufe das in Tabelle 7.7 definierte Schema eines dynamischen Sicherheitszuschlags verwenden.

Ein Vergleich der Strategie mit statischem und dynamischen Sicherheitszuschlag zeigt, dass für einen ähnlichen Schadensverlauf beide Strategien den Ruin der Versicherung zu Beginn vermeiden können. Die Strategie eines dynamischen Sicherheitszuschlags kann jedoch deutlich schneller das Anfangskapital wieder ansammeln. Erst im letzten Viertel des Simulationslaufs erreicht die Strategie eines statischen Sicherheitszuschlags das Niveau der dynamischen Strategie. Dies zeigt, dass die dynamische Strategie einem statischen Sicherheitszuschlag deutlich überlegen ist hinsichtlich des vorgegebenen Ziels, das Startkapital der Grid-Versicherung zu halten. Des Weiteren zeigt die Strategie eines dynamischen Sicherheitszuschlags ein ähnliches Verhalten in beiden Risikoprofilen. Eine Strategie kann also auf mehrere Szenarien übertragen werden, während eine Strategie mit statischem Sicherheitszuschlag für jedes Szenario neu eingestellt werden muss.

Abbildung 7.10 zeigt die detaillierte Entwicklung des Kapitals der dynamischen Strategie für die Risikoprofile *hohes und niedriges Risiko*. Die waagrechten Linien entsprechen den Schwellenwerten der Strategie, die einen stationären Prozess darstellen, der sich über die Zeit nicht ändert.

Nach dem Start des Simulationslaufs verringert sich das Kapital sehr schnell, da die Versicherung das Risiko falsch einschätzt. Gleichzeitig werden die Sicherheitszuschläge bei Unterschreiten der Schwellenwerte erhöht, um den Ruin der Versicherung zu vermeiden. Im Simulationslauf mit hohem Risiko wird bei Unterschreiten von 100 Geldeinheiten des Kapitals der Versicherung die Abnahme des Kapitals kurzzeitig verlangsamt. Jedoch konnte durch die Erhöhung des

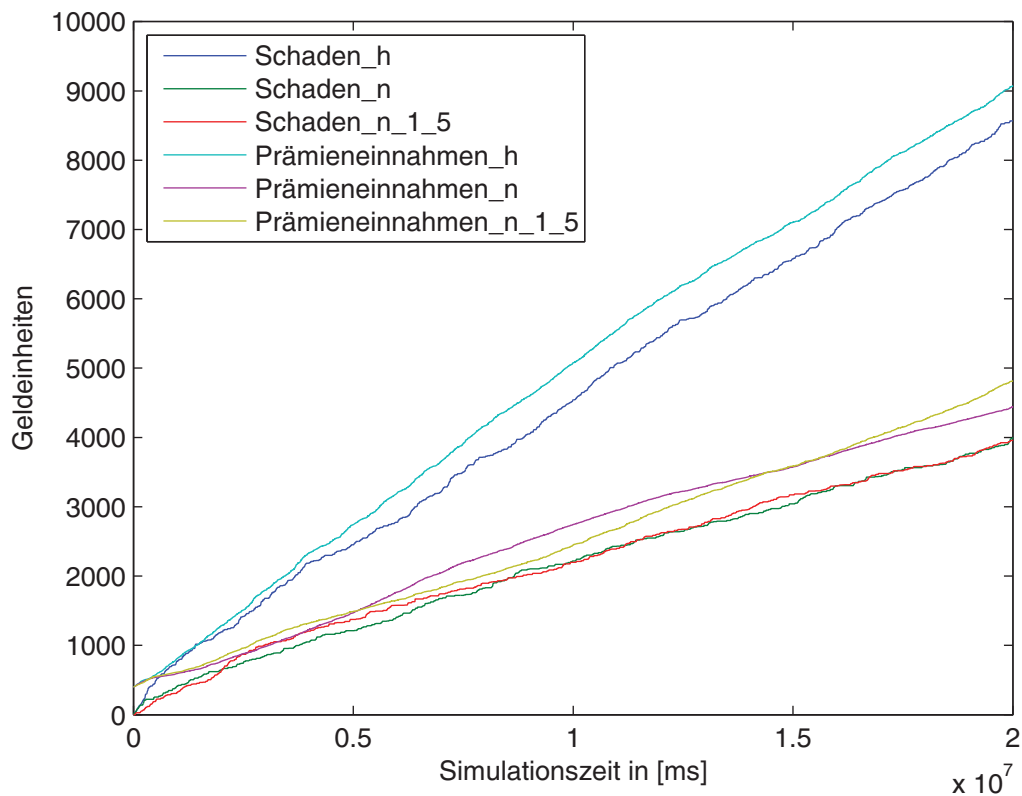


Abbildung 7.9: Prämieneinnahme- und Schadensprozess der Grid-Versicherung mit statischem (Szenario `_n_1_5`, wobei `_1_5` für einen statischen Sicherheitszuschlag von 1,5 steht) und dynamischen (Szenarien `_n` und `_h`) Sicherheitszuschlägen; das Subskript `_n` entspricht einem Simulationslauf mit niedrigem Risiko und das Subskript `_h` steht für einen Simulationslauf mit hohem Risiko.

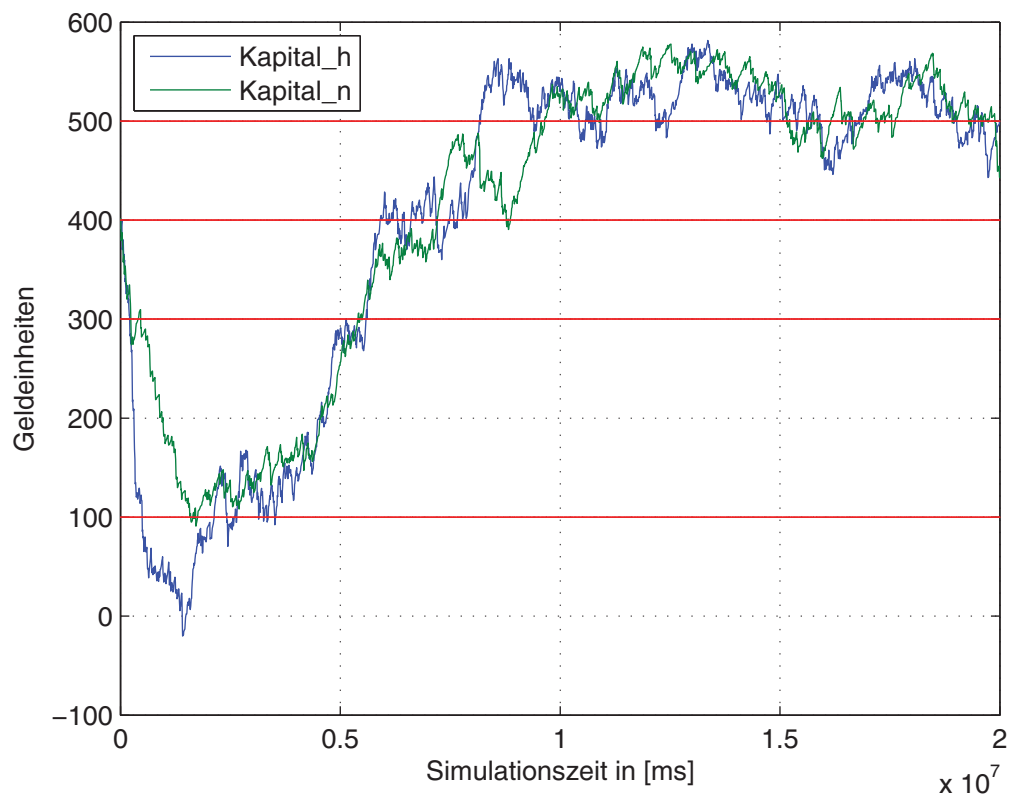


Abbildung 7.10: Entwicklung des Kapitals der Versicherung mit dynamischen Sicherheitszuschlägen; die waagrechten Linien kennzeichnen die Schwellenwerte der Sicherheitszuschläge.

Sicherheitszuschlags auf 2,0 der kurzzeitige Ruin der Versicherung nicht vermieden werden. Ein höheres Anfangskapital würde in diesem Szenario den Ruin der Versicherung vermeiden. Langfristig führt in beiden Risikoprofilen die Strategie der Versicherung zu einer schnellen Wiederherstellung des ursprünglichen Kapitals, das im Fortlauf der Simulationsläufe zwischen 450 und 580 Geldeinheiten schwankt. Die Versicherung erreicht ihr Ziel, langfristig ein Kapital von 400 zu halten. In beiden Szenarien würde ein Sicherheitszuschlag von 1,3 zu einer Zunahme des Kapitals der Versicherung führen, ein Sicherheitszuschlag von 1,0 zu einer Abnahme des Kapitals.

Der durchschnittliche Sicherheitszuschlag, der erhoben wird, liegt im Szenario mit hohem Risiko bei 1,07 und im Szenario mit niedrigem Risiko bei 1,08. Dabei gehen nur Werte in die Berechnung ein, die nach einer Simulationszeit von  $1,0 * 10^7$  gemessen wurden. Der durchschnittliche Sicherheitszuschlag kann als Indikator dafür interpretiert werden, wie gut die Versicherung das Risiko in den Simulationsläufen einschätzt bzw. wie groß der Fehler bei der Einschätzung der Risiken ist. Die Versicherung benötigt in beiden Szenarien den durchschnittlichen Sicherheitszuschlag von 7-8%, um ein Kapital von 500 zu halten. Dieser muss vom Versicherungsnehmer getragen werden.

In den Simulationsläufen der folgenden Abschnitte kommt die oben beschriebene dynamische Strategie der Versicherung zum Einsatz, die in den meisten Fällen einen Ruin der Versicherung verhindern kann und zu einer schnellen Wiederherstellung des Kapitals der Versicherung beiträgt. Auswirkungen der Strategie der Versicherung auf die Versicherungsnachfrage werden im folgenden Abschnitt betrachtet.

### 7.2.3 Versicherungsnachfrage

Die Analyse der Versicherungsnachfrage gliedert sich in eine Analyse auf Ebene aller Basic-Services und in eine Analyse auf Ebene der fünf verschiedenen Basic-Service-Typen. Ziel der Untersuchung ist es, das Verhalten der Basic-Services bei gegebenem Versicherungsangebot zu untersuchen. Dabei werden die Fragestellungen des Simulationsentwurfs aus Kapitel 6 aufgegriffen und näher betrachtet.

#### 7.2.3.1 Konfiguration des Lernalgorithmus

Grundlage der Versicherungsnachfrage stellt der Lernalgorithmus dar, der die Strategie des Agenten festlegt. Der in Kapitel 3 eingeführte co-evolutionäre Algorithmus wurde um zwei Gene

erweitert, die in Kapitel 5 näher erläutert wurden. In diesem Abschnitt steht die konkrete Konfiguration des Algorithmus im Vordergrund, so wie er in den folgenden Abschnitten für die Simulationsläufe verwendet wird. Der Algorithmus wurde so eingestellt, dass die Agenten das Ziel verfolgen, langfristig einen hohen Nutzen aus ihren Transaktionen zu ziehen.

Im Detail wird folgende Ausprägung der Konfigurationsparameter verwendet:

- `maturityThreshold = 3`: Der Agent wartet drei erfolgreiche Transaktionen ab, bevor er seine Strategie in Form des Genotyps an die benachbarten Basic-Service-Agenten verschickt. Nur wenn ein Agent über drei Transaktionen erfolgreich war, kann seine Strategie als Vorbild für andere Agenten dienen.
- `learning.hop.count = 3`: Die Lerninformationen werden bei jedem Lernschritt an alle Knoten verschickt, die in einem Radius von drei Knoten um den Sender herum im Netzwerk liegen. Aus Gründen der Performanz wird die Ausbreitung der Lerninformationen begrenzt, um die simulierte Netzwerkinfrastruktur nicht zu stark zu belasten.
- `courterThreshold = 10`: Ein Basic-Service-Agent sammelt Lerninformationen von mindestens 10 weiteren Basic-Service-Agenten, bevor er diese heranzieht, um seine eigene Strategie mit den empfangenen Informationen zu verbessern.
- `crossoverProbability = 0.5`: Hat ein Agent mindestens 10 Lerninformationen gesammelt und eine Strategie ausgewählt, so passt er seine Strategie durch die Anwendung eines Crossover an. Die Crossover-Wahrscheinlichkeit gibt an, mit welcher Wahrscheinlichkeit – hier 50% – ein Gen aus den ausgewählten Genotypen in die eigene Strategie aufgenommen wird. Ein Crossover-Wert von 0,5 stellt eine gute Wahl zwischen einer Lernstrategie, in der sich schnell erfolgreiche Verhaltensweisen durchsetzen sollen und einer Lernstrategie, mit der es den Agenten möglich sein soll, langfristig eine eigene, erfolgreiche Strategie aufzubauen, dar.
- `crossOverSelectionModel = 1`: Dieser Konfigurationsparameter beschreibt, welches Modell angewandt wird, um eine Strategie für einen anschließenden Crossover auszuwählen. Das in den folgenden Simulationsläufen verwendete Auswahlverfahren wählt immer die beste der 10 empfangenen Strategien aus. Die eigene Strategie spielt bei diesem Auswahlverfahren keine Rolle.
- `mutationProbability = 0.05`: Die Mutationswahrscheinlichkeit legt fest, wie häufig die eigene Strategie einer kleineren Veränderung unterzogen wird. Durch die Mutation entstehen neue Strategien, die im System bisher nicht vorhanden waren. Die Agenten können damit

ihre Strategie Stück für Stück verbessern. Im konkreten Szenario werden Gene mit einer Wahrscheinlichkeit von 5% einer Mutation unterzogen.

- `gaussWidth = 0.1`: Dieser Parameter legt fest, wie stark bei einer Mutation ein Gen verändert wird. In der verwendeten Einstellung beträgt die Änderung des Gens bei einer Mutation maximal 10% vom aktuellen Wert. Charakteristisch für die Mutationskonfiguration ist eine geringe Mutationshäufigkeit, kombiniert mit einem großen Anpassungsschritt. Dies führt zu wenig neuen Strategievarianten innerhalb der Basic-Service-Agenten.

Diese Konfiguration bestimmt das Lernen der beiden Gene *Risikoeinschätzung* und *Zahlungsbereitschaft* der Agenten, die die Versicherungsnachfrage bestimmen. Nach jeder erfolgreichen Transaktion werden die Gene einem Lernprozess unterzogen, der nach dem in Kapitel 3 beschriebenen Schema abläuft. Zu Beginn der Simulation werden die beiden Gene zufällig in einem Intervall zwischen 0 und 1 initialisiert. Die Initialisierung der Gene zwischen 0 und 1 führt zu einer breiten, initialen Startkonfiguration, aus der sich die Agenten durch Lernen an ihre individuell beste Strategie anpassen können. Erfolgreiche Strategien setzen sich durch, weniger erfolgreiche Strategien sterben über die Zeit aus.

### 7.2.3.2 Analyse der Versicherungsnachfrage

Die Analyse der Versicherungsnachfrage zielt darauf ab, das Verhalten der 200 Basic-Service-Agenten in ihrer Gesamtheit zu untersuchen. Im Detail werden die Fragestellungen aus Kapitel 6 näher betrachtet. Die Versicherungsnachfrage verwendet in den folgenden Simulationsläufen ausschließlich das Modell einer freiwilligen Versicherungsnachfrage. In jedem Simulationslauf werden 20.000 Benutzeranfragen für die Risikoprofile *hohes und niedriges Risiko* simuliert. Die Versicherungsnachfrage folgt der im vorangegangenen Abschnitt vorgestellten Konfiguration. Ziel der Analyse der Versicherungsnachfrage ist, Unterschiede und Gemeinsamkeiten des Verhaltens der Basic-Service-Agenten in den beiden unterschiedlichen Risikoprofilen aufzuzeigen. Die Analyse erfolgt unter Einbezug sowohl der beiden Gene *Risikoeinschätzung* und *Zahlungsbereitschaft* als auch des Ex-Post-Nutzens der Basic-Service-Agenten.

Abbildung 7.11 zeigt die Anzahl der versicherten und unversicherten Basic-Services für die beiden Risikoprofile mit hohem und niedrigem Risiko über die Zeit. Die Gesamtanzahl der erfolgreichen Basic-Service-Allokationen ist in beiden Simulationsläufen in etwa gleich, was zeigt, dass die Verhandlungsstrategie des Ressourcenmanagements in beiden Szenarien eine ähnliche hohe Allokationsrate erzielen kann. Zu Beginn der Simulation liegt der Anteil der Versicher-

ten bei 50%. Die Basic-Service-Agenten haben zu diesen Zeitpunkt noch keine Informationen über erfolgreiche Strategien anderer Agenten, sodass sie vom Erfolg ihrer eigenen Strategie abhängig sind. Mit der Zeit nimmt der Anteil an versicherten Basic-Services schnell zu, während gleichzeitig die Strategie der Risikoakzeptanz nur von vereinzelt Agenten angewandt wird. Ein vollständiges Verschwinden der unversicherten Basic-Service-Agenten konnte nicht beobachtet werden. Einzelne Agenten im System sind offensichtlich damit erfolgreich, keine Versicherung abzuschließen, während die überwiegende Mehrheit der Agenten langfristig die Strategie eines Risikotransfers auf die Grid-Versicherung vorzieht.

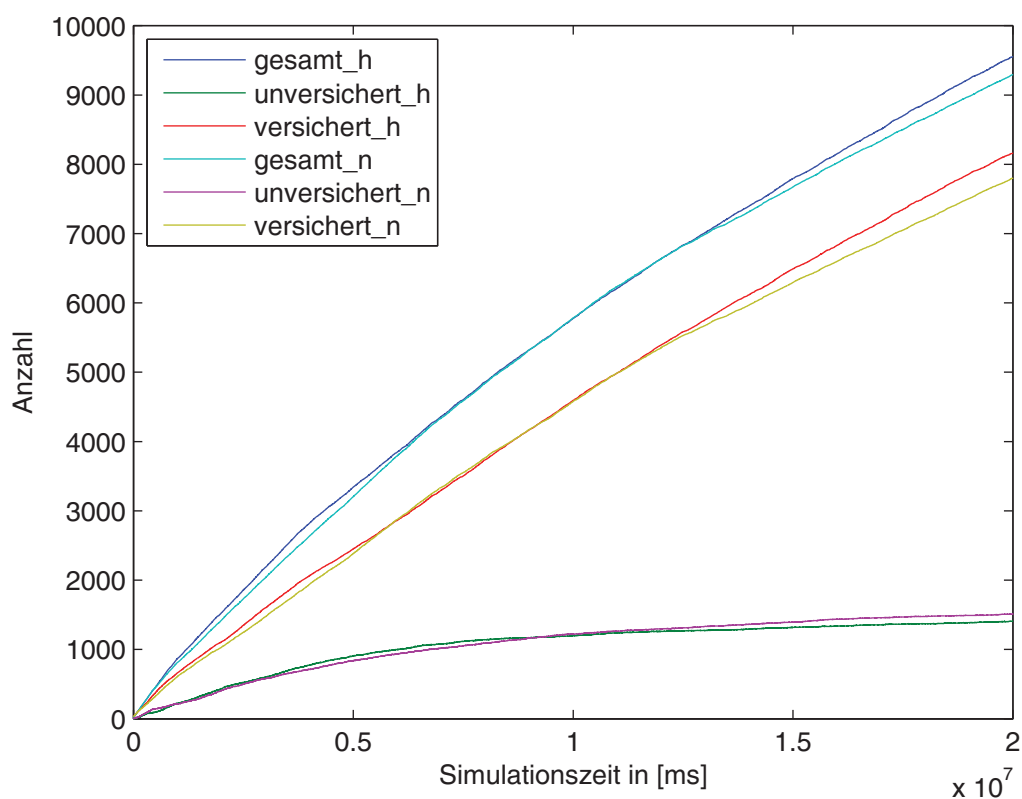


Abbildung 7.11: Die Entwicklung der Anzahl an versicherten und unversicherten Basic-Services über die Zeit für das Risikoprofil mit hohem (Subskript \_h) und niedrigem Risiko (Subskript \_n)

Ein Unterschied zwischen den Risikoprofilen hinsichtlich der Präferenzen für eine Risikobehandlungsstrategie lässt sich nicht erkennen. In den Simulationsläufen beider Risikoprofile dominiert die Strategie des Risikotransfers auf die Versicherung. Die Basic-Service-Agenten können durch Abschluss einer Versicherung in beiden Szenarien langfristig einen höheren Nutzen realisieren als ohne Versicherung.



Den realisierten Nutzen der Agenten zeigen die Abbildungen 7.12 für das niedrige Risiko und 7.13 für das hohe Risiko. Die Zeitserien der Abbildungen präsentieren den Verlauf des Ex-Post-Nutzens der Basic-Services während eines Simulationslaufs. Ein Punkt im Diagramm entspricht einem realisierten Nutzen eines Agenten. In der Farbe Grün sind die unversicherten Basic-Services markiert. Die versicherten Agenten sind blau gekennzeichnet. Die Ex-Post-Nutzen-Werte der unversicherten Agenten weisen zwei Extremwerte auf: Entweder können sie ihren Nutzen zu 100% realisieren oder – im Fall einer gescheiterten Ausführung des Basic-Services – keinen Nutzen realisieren. Wie schon in Abbildung 7.11 gezeigt, nimmt die Gruppe der Unversicherten deutlich ab.

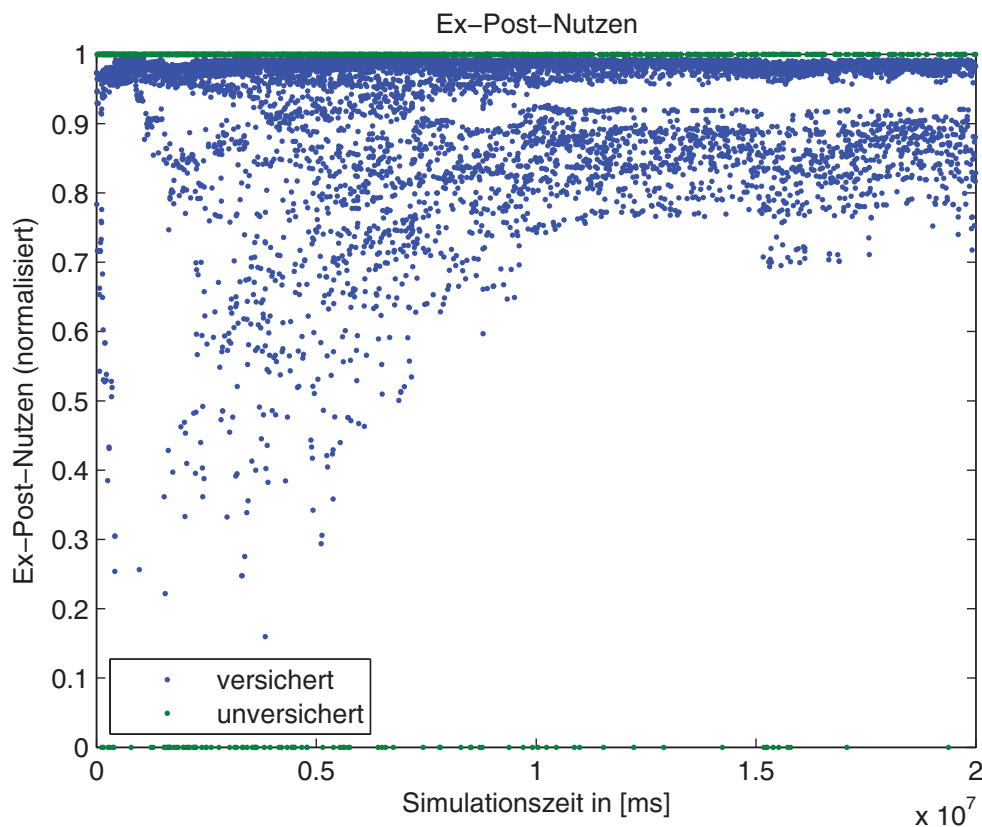


Abbildung 7.12: Zeitserie der Ex-Post-Nutzen der Basic-Services mit niedrigem Risiko

Die Gruppe der versicherten Basic-Services zeigt zu Beginn ein breites Band realisierter Nutzenwerte, wobei die Verteilung der Messwerte ist mit einem Median bei 0,94 stark linksschief ist. Zu Beginn der Simulation zeigen die Tariffaktoren der Versicherung starke Ausschläge. Zusätzlich wird der Preis der Versicherungspolice durch hohe Sicherheitszuschläge stark verteuert, sodass der Ex-Post-Nutzen des versicherten Basic-Service sinkt. In dieser Phase der Simulation

gibt es auch eine große Gruppe an unversicherten Basic-Services. Haben sich die Prämien stabilisiert und die Sicherheitszuschläge der Versicherung wieder abgebaut, gewinnt die Strategie eines Risikotransfers auf die Versicherung weitere Vorteile, die sich in einem deutlich gesteigerten Ex-Post-Nutzen der Versicherungsnehmer widerspiegelt. Es bilden sich zwei über die Zeit stabile Wertebereiche – ein Bereich zwischen 0,95 und 0,99 und ein zweiter Bereich zwischen 0,95 und 0,75 – des Ex-Post-Nutzens heraus, die auf die unterschiedlichen Prämienhöhen hindeuten, die die Basic-Services bezahlen müssen.

Ein ähnliches Verhalten manifestiert sich im Szenario mit hohem Risiko, das Abbildung 7.13 zeigt. Ebenfalls ist zu Beginn der Simulation durch die teuren Preise der Versicherungspolice der Ex-Post-Nutzen deutlich verringert. Der Effekt des Abnahme des Sicherheitszuschlags ist in diesem Szenario im Zeitbereich zwischen  $0,5 \cdot 10^7$  und  $0,75 \cdot 10^7$  erkennbar, führt jedoch zu einer weniger deutlichen Zunahme des Ex-Post-Nutzens.

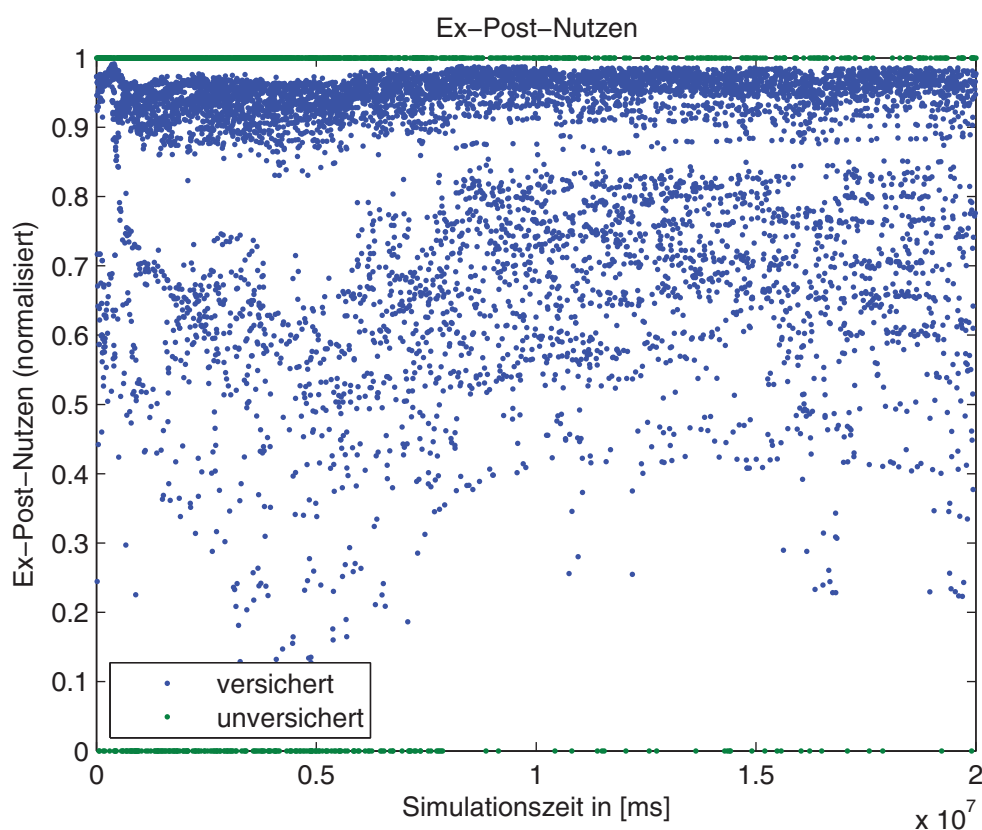


Abbildung 7.13: Zeitserie der Ex-Post-Nutzen der Basic-Services mit niedrigem Risiko

Es bilden sich wie im Szenario mit niedrigem Risiko zwei Wertebereiche heraus, die darauf hindeuten, dass bestimmte Basic-Services gehäuft Prämien ähnlicher Höhe bezahlen müssen. Ein

Wertebereich umfasst die Ex-Post-Nutzen zwischen 0,9 und 0,98, der zweite Wertebereich die Ex-Post-Nutzen zwischen 0,85 und 0,4. Die darunter liegende Messpunkte können als Ausreißer interpretiert werden. Allgemein gilt, dass sich das höhere Risiko auf einen geringeren Ex-Post-Nutzen niederschlägt und das Band der realisierten Nutzenwerte deutlich breiter ist.

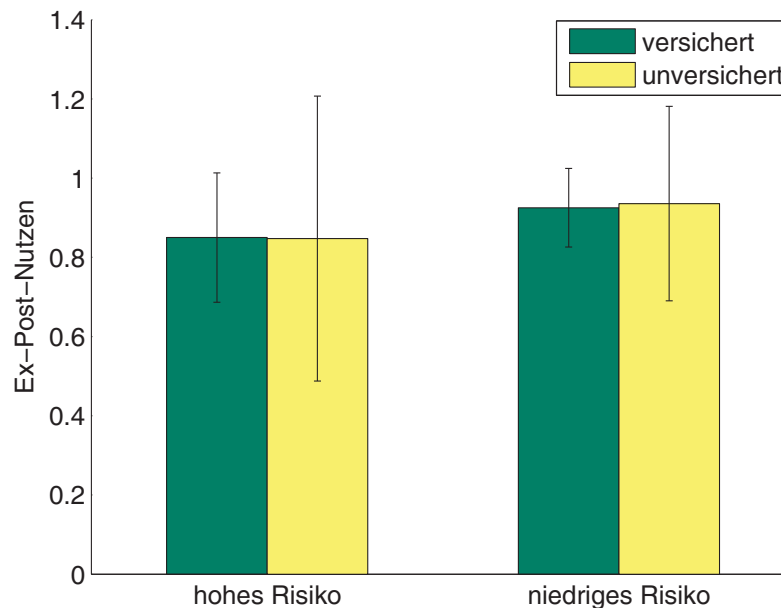


Abbildung 7.14: Vergleich der Mittelwerte und Standardabweichungen der Strategien Risikoakzeptanz (unversicherte Basic-Services) und Risikotransfer auf eine Versicherung (versicherte Basic-Services) für die Risikoprofile mit niedrigem und hohem Risiko

Abbildung 7.14 zeigt die Mittelwerte und Standardabweichungen der Ex-Post-Nutzen für die versicherten und unversicherten Basic-Service-Agenten in beiden Risikoprofilen. Im Mittel sind die Strategien Risikoakzeptanz und Risikotransfer gleich, jedoch weisen die Standardabweichungen der Ex-Post-Nutzen von versicherten Agenten deutliche Unterschiede auf. In einem Szenario mit niedrigem Risiko lässt sich im Mittel ein höherer Nutzen realisieren als in einem Szenario mit hohem Risiko.

Der Vorteil einer Versicherungspolice ergibt sich für die Basic-Services aus der deutlich verringerten Standardabweichung des Ex-Post-Nutzens. Sowohl im Szenario mit hohem Risiko als auch im Szenario mit geringem Risiko ist der versicherte Basic-Service einer geringeren Volatilität ausgesetzt, sein Ziel, langfristig einen möglichst hohen Nutzen zu realisieren, zu erreichen. Er wählt aus diesem Grund die Strategie eines Risikotransfers auf eine Versicherung.

**Analyse des Gens *Risikoeinschätzung*** Die Strategie der Versicherungsnachfrage wird durch zwei Gene festgelegt, die durch einen co-evolutionären Algorithmus im Zeitablauf verändert werden. Ziel der Betrachtung ist es, Verhaltensmuster der Agenten zu bestimmen, die das Systemverhalten beschreiben.

Als Grundlage der Analyse dient die Zeitserie des Gens, die mithilfe einer Dichtefunktion visualisiert wird <sup>7</sup>. Die Dichtegrafiken zeigen eine Abnahme der Dichte über Simulationsdauer. Dies ist bedingt durch den Rückgang der erfolgreichen Ressourcenzuweisungen durch den Einsatz des dezentralen Allokationsansatzes. Die Agenten erlernen die Strategie des jeweiligen Verhandlungspartners und werden zu härteren Verhandlungspartnern, die dazu tendieren, eher eine Verhandlung abzubrechen, in der sie ihre Ziele nicht erfüllen können, anstatt sie zu einem schlechteren Verhandlungsergebnis abzuschließen.

Die Analyse der Zeitserien erfolgt anhand der beiden Risikoprofile mit hohem und niedrigem Risiko. Abbildung 7.15 zeigt die Zeitserie des Gens *Risikoeinschätzung* für das Risikoprofil mit niedrigem Risiko. Zu Beginn der Simulation wird der Startwert des Gens mithilfe eines Zufalls-generators erzeugt, sodass Startwerte gleichmäßig im Intervall zwischen 0 und 1 verteilt sind. Ausgehend von der gleichmäßigen Verteilung der Startwerte wird der zeitliche Verlauf des Gens protokolliert und deren Dichte bestimmt. Zu Beginn des Simulationslaufs zeigt sich ein homogenes Dichtebild, das sich über den Verlauf der Zeit stark ändert. Eine Risikoeinschätzung nahe 1 verschwindet fast vollständig. Dies ist gleichbedeutend damit, dass die Einschätzung der Agenten abnimmt, einen Dienst mit einer Wahrscheinlichkeit von 100% zu bekommen.

Die Agenten lernen, dass die Ausführung eines Basic-Service mit einem Risiko verbunden ist und bewerten den Erwartungsnutzen mit einer Versicherung über dem Erwartungsnutzen ohne Versicherung. Die meisten Agenten bevorzugen einen Wert der Risikoeinschätzung nahe 0. In diesem Fall schätzen die Agenten das Risiko einer Basic-Service-Ausführung als sehr hoch ein. Weitere Wertebereiche bilden sich bei 0,7 und bei 0,35 heraus. An diesen Stellen sind ebenfalls Agenten anzutreffen, die mit dieser Wertekonfiguration einen hohen Ex-Post-Nutzen erzielen können. Eine detailliertere Analyse der einzelnen Basic-Service-Typen im folgenden Abschnitt wird diese Muster näher untersuchen.

Im Simulationslauf mit einem hohen Risiko, der in Abbildung 7.16 gezeigt wird, ist ein ähnli-

---

<sup>7</sup>Eine Dichtefunktion hat den Vorteil, Häufungen in den Messdaten grafisch darzustellen zu können und damit eine Analyse der Messdaten zu vereinfachen. Die Berechnung der Dichte erfolgt durch ein symmetrisches Intervall von 25 Datenpunkten, um den aktuellen Datenwert herum. Gegen Ende der Simulation entstehen Artefakte geringer Dichte, da fehlende Datenpunkte zu Blöcken von 25 Elementen mit 0 Werten aufgefüllt werden müssen. Dies ist bedingt durch den Algorithmus der Dichteberechnung und wird bei der Interpretation der Grafiken berücksichtigt.

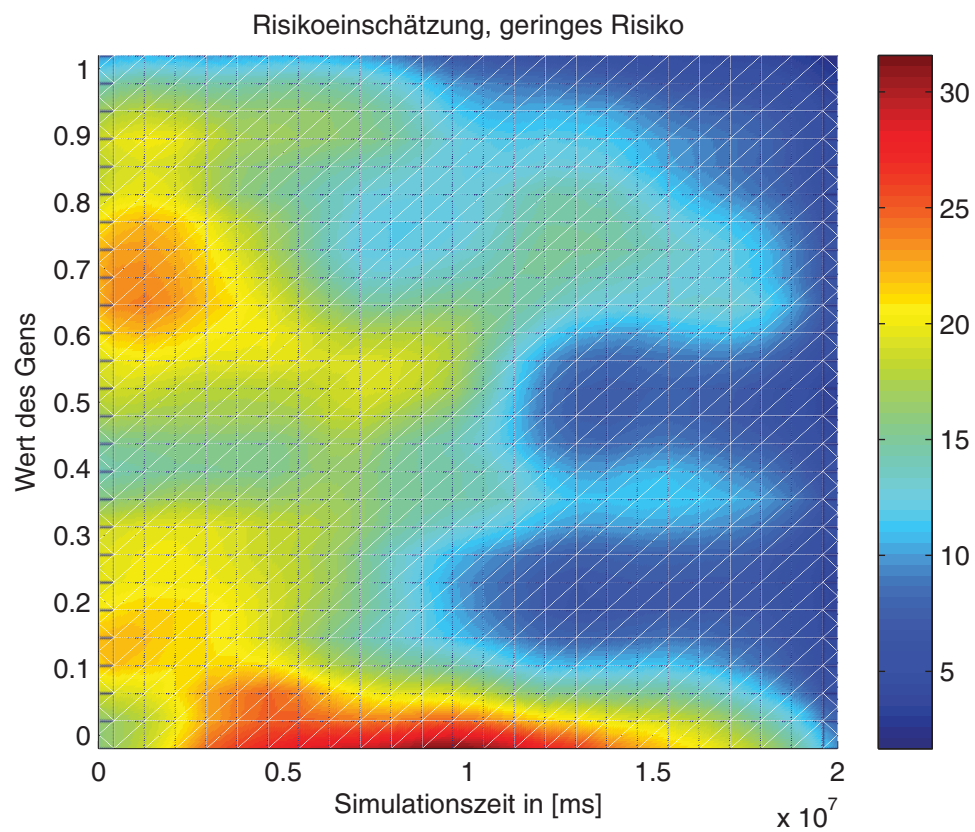


Abbildung 7.15: Zeitserie des Gens *Risikoeinschätzung* für das Risikoprofil mit niedrigem Risiko

ches Bild wie im Simulationslauf mit dem niedrigen Risiko zu beobachten. Die Basic-Service-Agenten reagieren auf das deutlich höhere Risiko, indem sie Strategien mit einer Risikoeinschätzung von 1 deutlich schneller im System aussterben lassen. Gegen Ende der Simulation findet sich eine Reihe von Agenten, die mit einer Risikoeinschätzung von 0,55 erfolgreich sind. Im Vergleich zum Szenario mit niedrigem Risiko bedeutet dies eine Senkung der maximalen Risikoeinschätzung um einen Wert von 0,15. In ihrer Gesamtheit schätzen die Basic-Services das Risiko als deutlich höher ein.

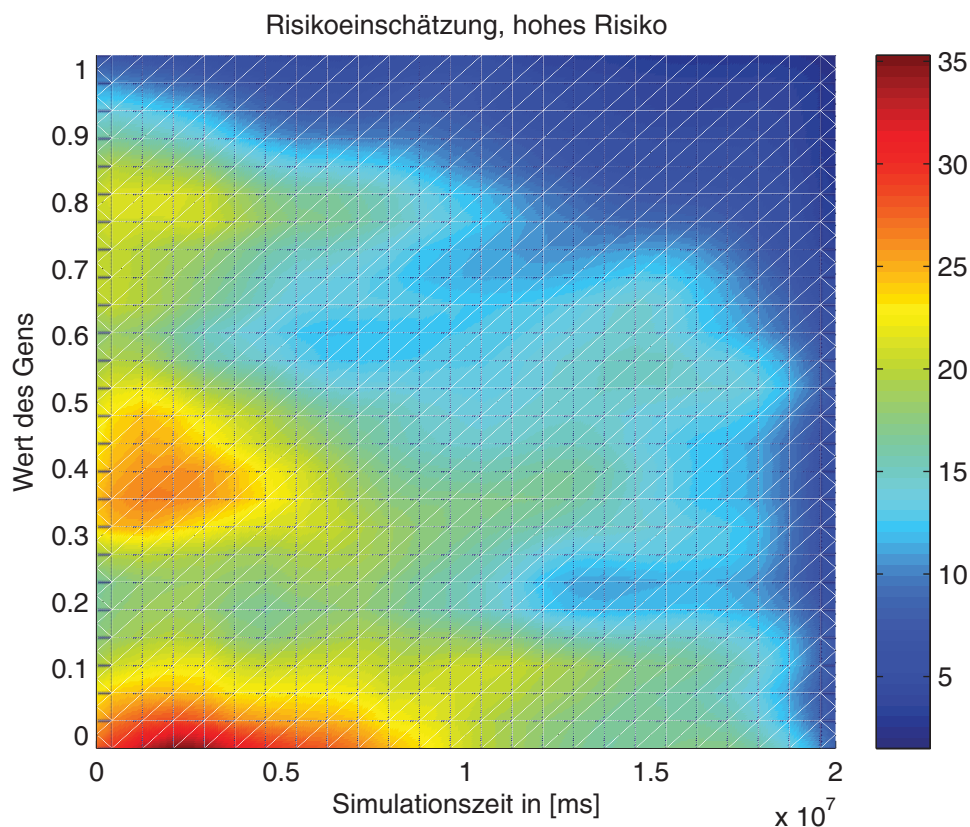


Abbildung 7.16: Zeitserie des Gens *Risikoeinschätzung* für das Risikoprofil mit hohem Risiko

Die größte Menge an Agenten findet sich wie im Szenario mit niedrigem Risiko zwischen 0 und 0,1. Hier wird das Risiko eines Basic-Service-Ausfalls als sehr hoch eingeschätzt und der Erwartungsnutzen einer Strategie mit Versicherung erfährt eine stärkere Gewichtung.

Ein Vergleich der beiden Szenarien führt zu folgenden Ergebnissen:

- Die Agenten reagieren auf das Risiko, indem sie die Risikoeinschätzung senken und den Erwartungsnutzen eines Risikotransfers stärker gewichten und mehr Versicherungspolice



nachfragen.

- Es bilden sich verschiedene Wertebereiche der Risikoeinschätzung aus und damit verschiedene, erfolgreiche Strategien, mit denen die Basic-Services auf das Risiko einer Basic-Service-Ausführung reagieren.

**Analyse des Gens *Zahlungsbereitschaft*** Das Gen *Zahlungsbereitschaft* steuert den Betrag, den die Basic-Service-Agenten bereit sind, für die Versicherungsprämie auszugeben. Bewertet ein Basic-Service den Erwartungsnutzen einer Basic-Service-Ausführung mit Versicherung höher als ohne Versicherung, wird anschließend mithilfe des Gens *Zahlungsbereitschaft* bestimmt, ob der Agent bereit ist, die geforderte Prämie zu bezahlen. Ein Wert des Gens von 1 bedeutet, dass der Basic-Service bereit ist, die Höhe des Transaktionspreises für die Versicherungsprämie auszugeben. Bei einem Wert von 0 ist der Agent nicht bereit, einen Preis für die Versicherungspolice zu bezahlen.

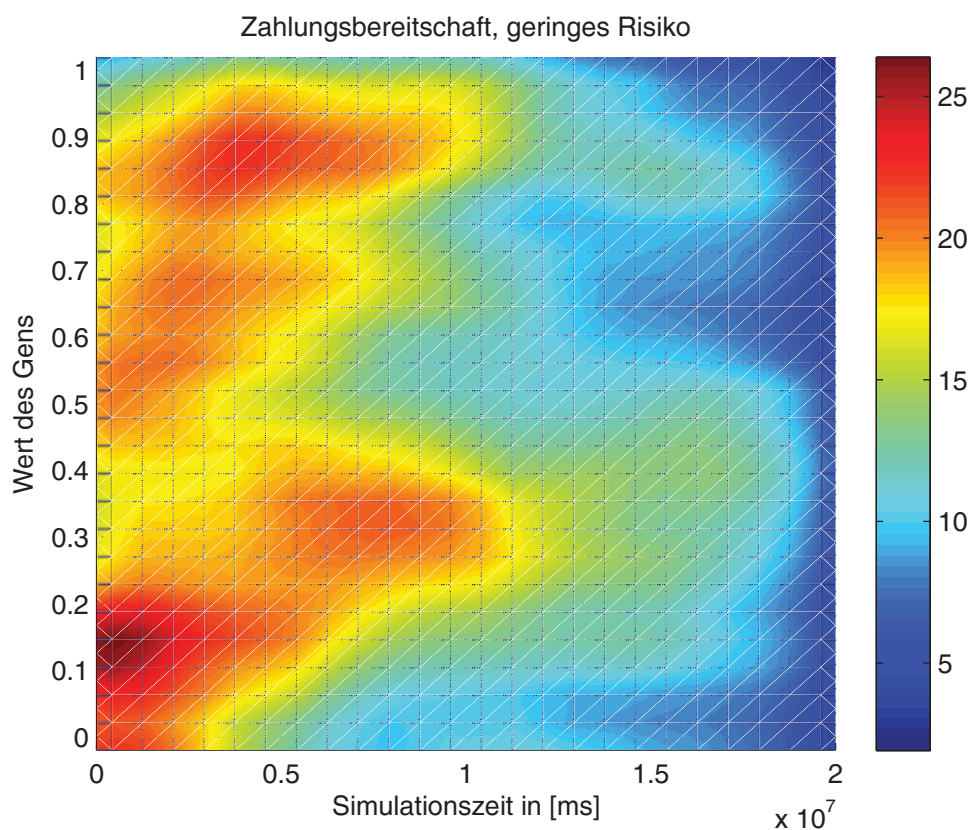


Abbildung 7.17: Zeitserie des Gens *Zahlungsbereitschaft* für das Risikoprofil mit niedrigem Risiko

Der Vergleich des Gens Zahlungsbereitschaft benutzt die beiden Risikoprofile mit niedrigem und hohem Risiko. Analog zur Betrachtung des Gens Risikobereitschaft erfolgt die Analyse der Zeitserie unter Verwendung der oben eingeführten Dichtegrafik. Abbildung 7.17 zeigt die Zeitserie des Gens Zahlungsbereitschaft für das Risikoprofil mit niedrigem Risiko. Es bilden sich zwei Wertebereiche für die Zahlungsbereitschaft der Basic-Service-Agenten heraus: ein weniger breites Band bei zwischen 0,8 und 0,9 und ein breiteres Band zwischen 0,1 und 0,5. Die erste Gruppe der Basic-Service-Agenten ist bereit, zwischen 80% und 90% des Transaktionspreises für die Versicherungsprämie auszugeben, während die zweite Gruppe bereit ist, für 10% bis 50% des Transaktionspreises in eine Versicherungspolice zu investieren.

Abbildung 7.18 zeigt die Zahlungsbereitschaft der Basic-Service-Agenten für das Szenario mit hohem Risiko. Deutlich stellt sich hier eine erhöhte Zahlungsbereitschaft der Basic-Service-Agenten gegenüber dem Szenario mit niedrigem Risiko dar. Sind im ersteren Szenario die Agenten überwiegend nur bereit, 10 bis 50% für die Versicherungsprämie zu bezahlen, ist die Bereitschaft der Agenten im Szenario mit hohem Risiko zwischen 50 und 100% angesiedelt. Das erhöhte Risiko einer Basic-Service-Ausführung schlägt sich auf eine erhöhte Zahlungsbereitschaft nieder. Vereinzelt weisen Agenten eine sehr geringe Zahlungsbereitschaft auf. Dies weist auf Agenten hin, die eine Risikoakzeptanzstrategie bevorzugen.

Ein Vergleich der beiden Szenarien führt zu folgenden Ergebnissen:

- Die Basic-Service-Agenten reagieren auf eine Risikoerhöhung, indem sie die Zahlungsbereitschaft für die Versicherungsprämie erhöhen.
- Es bilden sich Wertebereiche für die Gene *Zahlungsbereitschaft* und *Risikoeinschätzung* heraus, die eine größere Dichte als andere Wertebereiche haben. Dies lässt die Schlussfolgerung zu, dass erfolgreiche Strategien in dichteren Gebieten zu finden sind und weniger erfolgreiche Strategien langsam im Zeitablauf aussterben.

### 7.2.3.3 Analyse der Basic-Services

Im Mittelpunkt dieses Abschnitts steht die Analyse der 5 Basic-Service-Typen. Dabei werden die Erkenntnisse des vorangegangenen Abschnitts aufgegriffen und vertieft. Ziel ist es bei dieser Untersuchung, Muster, die auf Ebene der Gesamtpopulation beobachtet wurden, bestimmten Basic-Service-Typen zuzuordnen. Des Weiteren werden die Kosten der Versicherung aus Nachfragersicht detaillierter analysiert. Die folgenden Untersuchungen werden für das Szenario mit hohem Risiko durchgeführt.



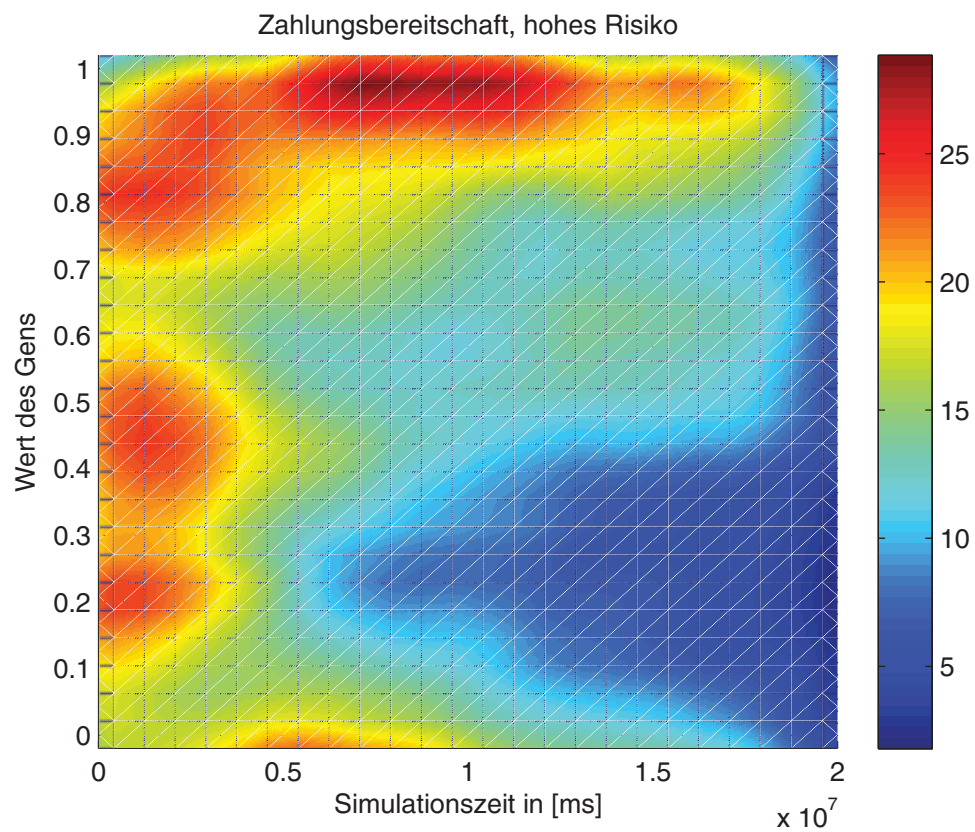


Abbildung 7.18: Zeitserie der Gens *Zahlungsbereitschaft* für das Risikoprofil mit hohem Risiko

Abbildung 7.19 zeigt die Dichtegrafiken des Gens *Risikoeinschätzung* für die 5 simulierten Basic-Service-Typen. In Klammern hinter dem Namen des Basic-Service sind zur besseren Orientierung die Anzahl der nachgefragten Recheneinheiten und die Länge der Ausführungszeit angegeben.

Die Zeitserien der Gene zeigen deutliche Unterschiede zwischen den einzelnen Basic-Service-Typen. Es lassen aus den Beobachtungen für die einzelnen Basic-Services folgende Erkenntnisse ableiten:

- Die Gruppen der Basic-Service-Typen reagieren gemeinschaftlich auf das Risiko, dem sie ausgesetzt sind mit dem Ziel, einen möglichst hohen, durchschnittlichen Ex-Post-Nutzen zu erzielen. Die Agentengruppe bs1 mit dem geringsten Risiko schätzt das Risiko auch am geringsten ein, während die Gruppe der Basic-Services mit dem größten Risiko (bs5) auch das Risiko am höchsten einschätzt. Alle weiteren Basic-Service-Gruppen (bs2, bs3, und bs4) ordnen sich dazwischen ein, wobei sich ein breites Band an Werten in diesen Gruppen herausbildet. Eine Anomalie zeigt die Gruppe der bs4, deren Wertebereich dem Wertebereich des Basic-Service bs2 bei gleichzeitig deutlich höherem Risiko entspricht. Dieses Phänomen hat seinen Ursprung in der Konstruktion des Versicherungstarifs und wird bei der Analyse der Kosten der Versicherung näher betrachtet.
- Die Dichte ist ein Indikator dafür, wie viele Transaktionen eines Basic-Service-Typs über die Zeit erfolgreich abgeschlossen werden. Dieser schwankt im Zeitablauf der Simulation, was auf eine unterschiedlich starke Nachfrage nach einem bestimmten Basic-Service-Typ zu einem bestimmten Zeitpunkt zurückzuführen ist.

Die Zeitserien des Gens *Zahlungsbereitschaft* für die 5 Basic-Service-Typen zeigt Abbildung 7.20. Die Abbildung belegt, dass die Zahlungsbereitschaft abnimmt, je geringer das Risiko ist, dem eine bestimmte Agentengruppe ausgesetzt ist. Die Agenten passen ihre Zahlungsbereitschaft der erwarteten Höhe der Versicherungsprämie an, um sicher zu stellen, dass sie genügend Budget für die Versicherungsprämie zur Verfügung haben.

Wie bereits im Gen *Risikoeinschätzung* zu sehen, besetzen die Agenten der Gruppe bs1 und bs5 die beiden Extrempunkte des Wertebereichs des Gens, während sich die anderen Basic-Services dazwischen ansiedeln. Ebenfalls deutlich zu erkennen ist die Anomalie der Gruppe bs4, die ein ähnliches Verhalten wie Gruppe bs2 zeigt, jedoch einem deutlich höheren Risiko ausgesetzt ist. Aufschluss darüber gibt die folgende Untersuchung der Kosten der Versicherungspolice.

Zusammenfassend richten sich die Agenten über die Zeit relativ zueinander aus und passen ihre

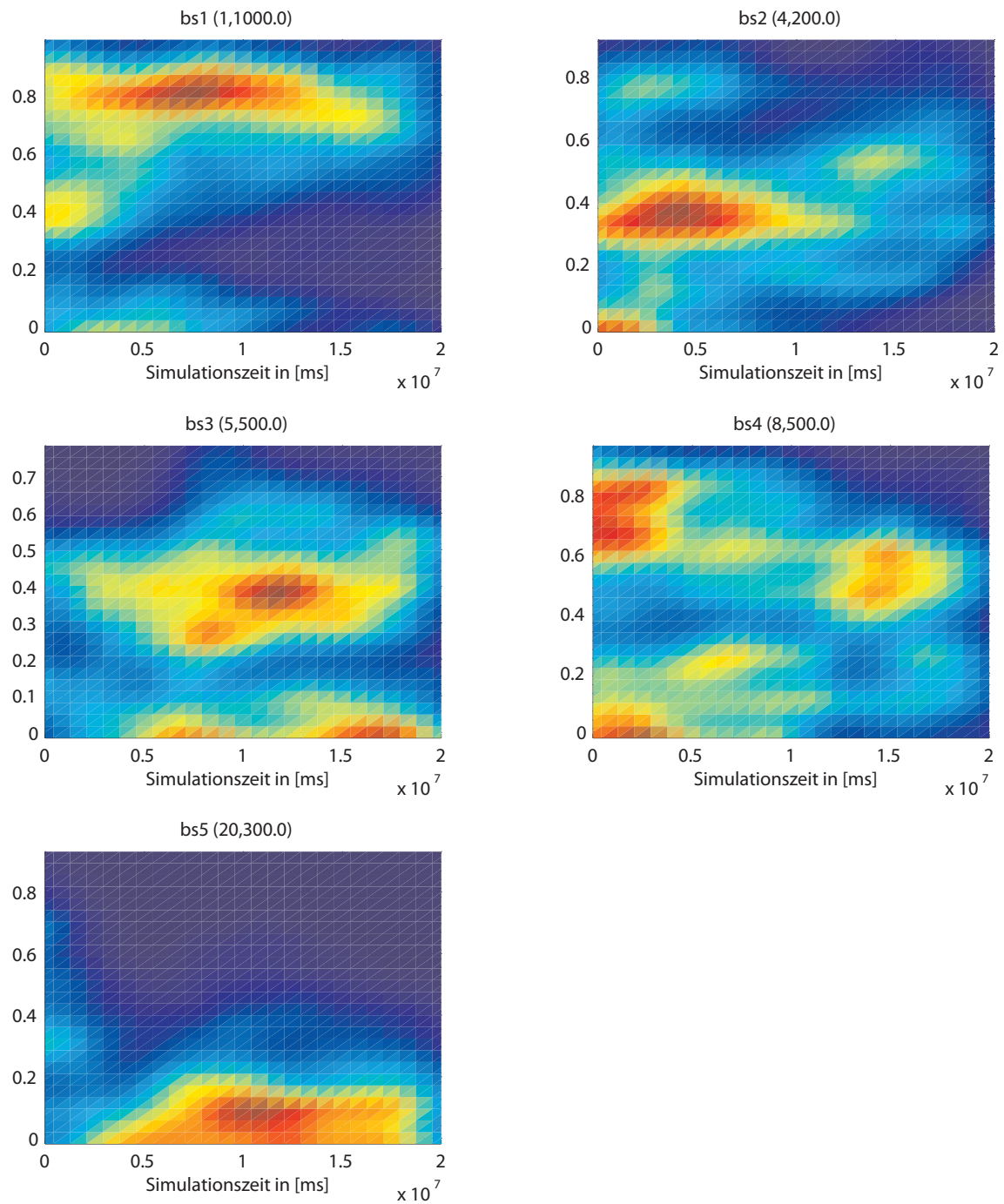


Abbildung 7.19: Zeitserien des Gens *Risikoeinschätzung*, aufgliedert nach den 5 Basic-Service-Typen für das Szenario mit hohem Risiko

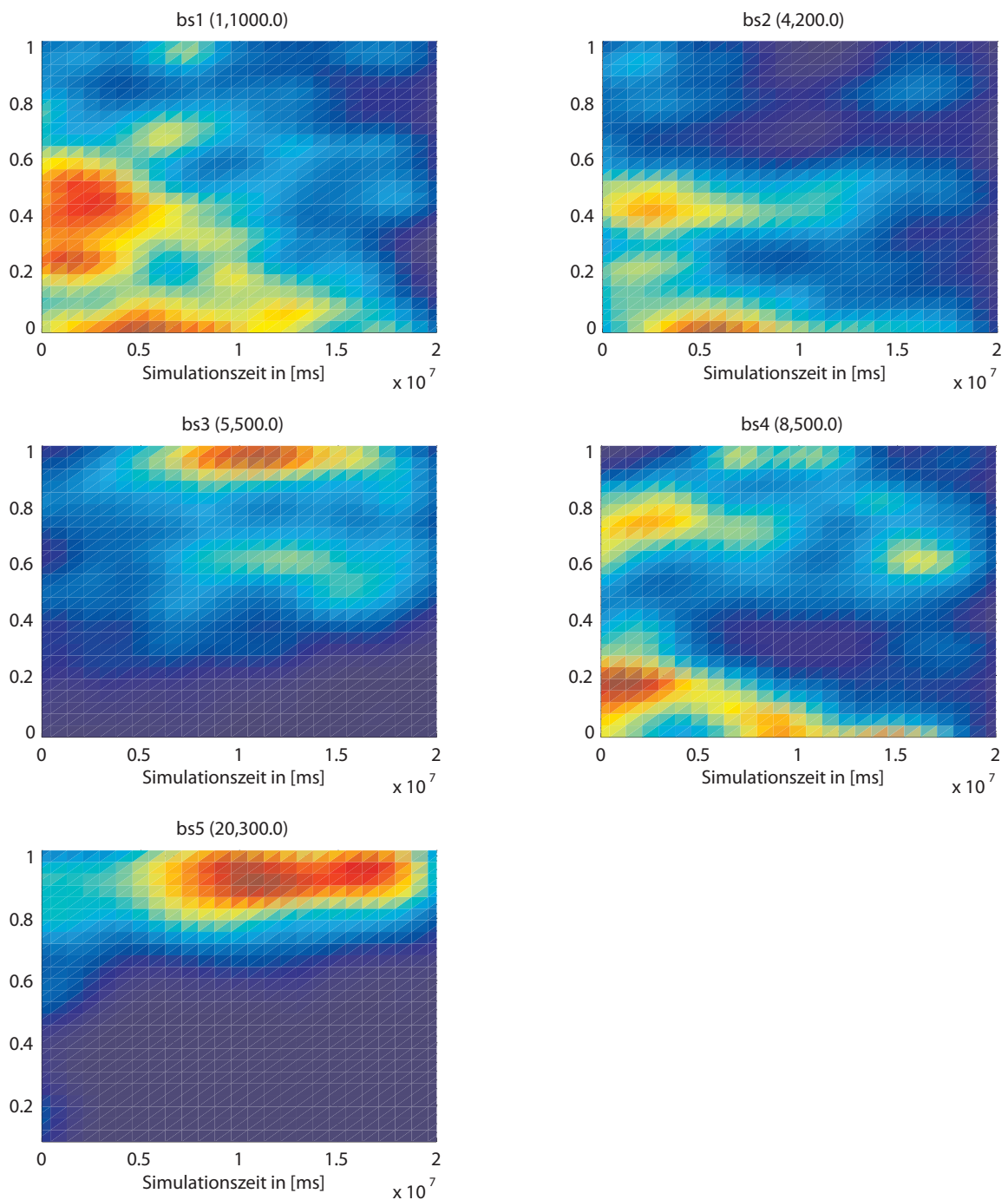


Abbildung 7.20: Zeitserien des Gens *Zahlungsbereitschaft*, aufgegliedert nach den 5 Basic-Service-Typen für das Szenario mit hohem Risiko

Strategie dem gegebenen Risikoprofil an. Das Ziel der Agenten, einen im Durchschnitt hohen Ex-Post-Nutzen zu erreichen, ist einem kontinuierlichen Prozess unterworfen, bei dem sich eine überwiegende Mehrheit der Agenten einen Risikotransfer auf eine Versicherung wählt. Jede Gruppe von Basic-Service-Agenten wählte ihre individuell am besten passende Strategie, um mit dem auftretenden Risiko umzugehen.

**Kosten der Versicherungspolice** Anhand der Kosten der Versicherung wird der Einfluss der Prämien auf die Strategie der Basic-Services als Nachfrager der Grid-Versicherung näher betrachtet. Zusätzlich werden die Mehrkosten analysiert, die die Gruppe der Basic-Services zu tragen hat.

Den Anteil der Kosten der Versicherungspolice an den Gesamtkosten einer Basic-Service-Ausführung zeigt Abbildung 7.21. Für jede erfolgreiche Transaktion mit einer Versicherung wurde der Anteil der Kosten der Versicherungspolice an den Gesamtkosten der Transaktion berechnet. Der Anteil an den Gesamtkosten der Transaktion, der für den Risikotransfer bezahlt werden muss, schwankt zwischen 5 und 40% in Abhängigkeit von der Ausprägung des Risikos. Es zeigt sich, dass Basic-Services mit geringem Risiko (bs1 und bs2) eine deutlich geringere Prämie bezahlen müssen als Basic-Services, die einem höheren Risiko ausgesetzt sind (bs3 und bs4). Lediglich die Gruppe der bs4-Agenten zeigt einen deutlich verringerten Anteil der Prämien an den Gesamtkosten trotz hohen Risikos.

Die Basic-Service-Agenten richten sich nicht nur nach ihrem tatsächlichen Risiko aus, sondern beziehen bei ihrer Anpassung die Versicherungsprämie mit ein. Sie richten ihre Strategie der Versicherungsnachfrage auf die Höhe der Versicherungsprämien aus wie die Gruppe der bs4 deutlich zeigt. Da der Anteil der Prämien, den die Basic-Services in dieser Gruppe bezahlen muss, in etwa dem Anteil der Prämien der bs2-Gruppe entspricht, weist die Strategie der bs4 die oben beobachteten Ähnlichkeiten mit der Strategie der bs2 auf.

Diese im Verhältnis zu bs3 und bs5 niedrigen Versicherungsprämien sind auf die Konstruktion des Versicherungstarifs zurückzuführen. Die Basic-Services vom Typ bs4 sind bezüglich des Tarifmerkmals Recheneinheit in einer Risikogruppe mit den bs5-Agenten und hinsichtlich des Tarifmerkmals Ausführungszeit in einer Risikogruppe mit den bs3-Agenten. Da die Tariffaktoren für die Risikogruppen bestimmt werden, kommt es zu dem Effekt, dass die Basic-Services vom Typ bs4 niedrige Prämien bezahlen müssen. Der Schaden, den die bs4-Agenten verursachen, werden auf die den Risikogruppen zugehörigen bs3-Agenten und bs5-Agenten verteilt. Durch die wenig besetzte Tarifmatrix fehlt dadurch eine größere Gruppe an Basic-Services, auf

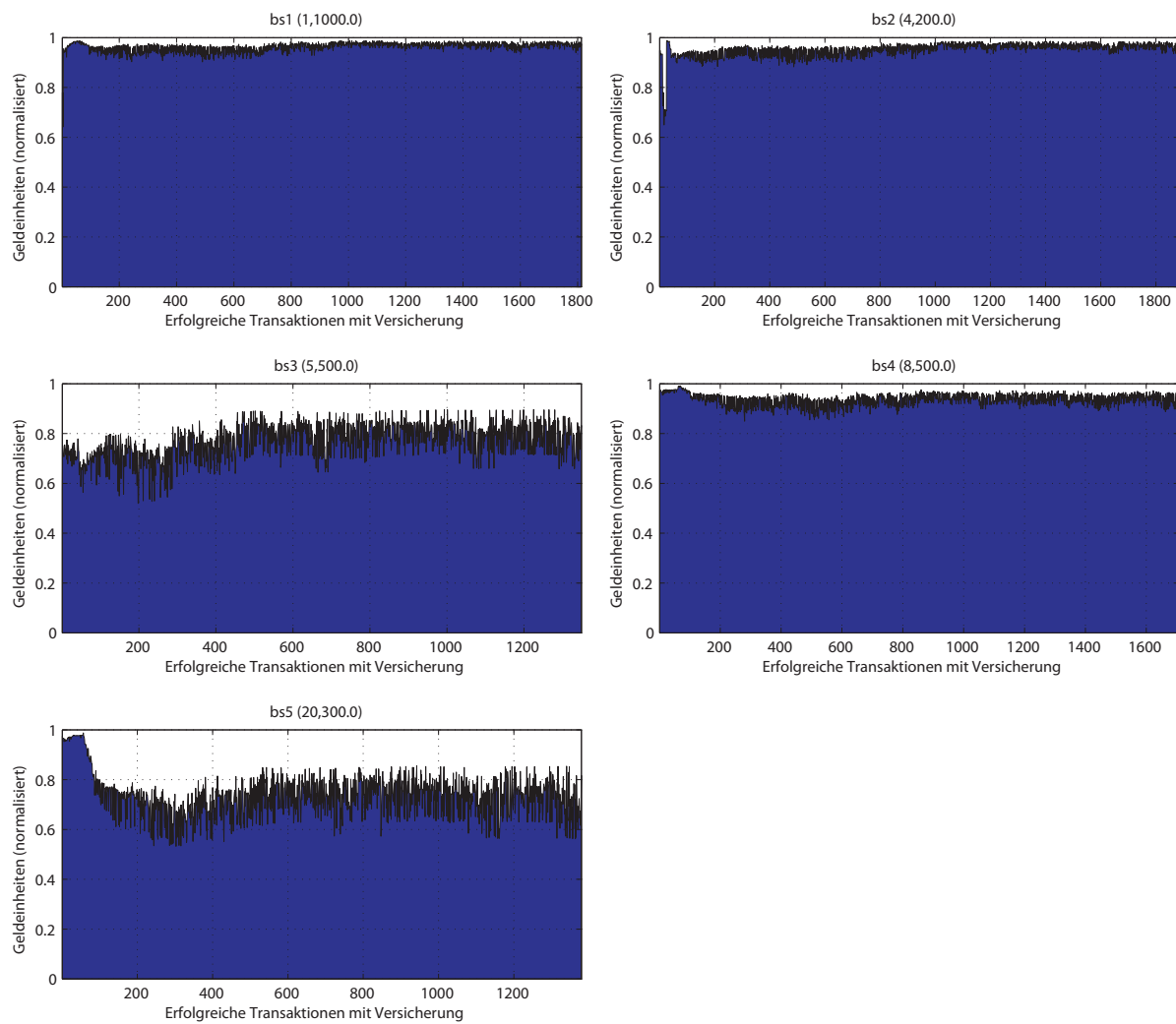


Abbildung 7.21: Anteil der Kosten der Versicherungspolice an den Gesamtkosten einer Basic-Service-Ausführung für die 5 verschiedenen Basic-Service-Typen; der blau gekennzeichnete Bereich entspricht den Kosten der Ressourcen, der weiße Bereich kennzeichnet die Kosten der Versicherungsprämie. Zur besseren Vergleichbarkeit sind die Werteskalen der Geldeinheiten auf das Intervall zwischen 0 und 1 normalisiert.

die die Schäden verteilt werden könnte. Es ist zu erwarten, dass mit Zunahme der Belegung der Tarifmatrix diese Effekte weniger stark ausgeprägt sind, da die Schäden einer Risikogruppe besser an die einzelnen Basic-Service-Typen verteilt werden können.

## 7.3 Simulationsergebnisse Teil 2: Analyse der Gesamtperformanz

Der zweite Teil der Simulationsstudie führt eine Analyse der Gesamtperformanz des Grid-Systems durch, in dem die in Kapitel 6 eingeführten Metriken gemessen und zum Social-Loss-Index aggregiert werden. Dieser Aggregationsprozess wird in mehreren Stufen durchgeführt, wobei die Informationsdichte zunimmt. Mithilfe der Indikatoren und Indizes der Metrikpyramide werden in den folgenden Abschnitten die Risikobehandlungsstrategien Risikoakzeptanz (RS1) und Risikodiversifikation durch Redundanz (RS2) mit der Strategie eines Risikotransfers auf eine Versicherung (RS3) analysiert. Der Vergleich findet unter der Verwendung von drei Risikoprofilen statt, wobei die beiden bekannten Risikoprofile mit niedrigem und hohem Risiko aus dem vorangegangenen Abschnitt durch ein Risikoprofil mit sehr hohem Risiko ergänzt werden.

In Abschnitt 7.3.1 werden der Aufbau und die Konfiguration der Szenarien erläutert. Die Analyse der Indikatoren beinhaltet Abschnitt 7.3.2, während die Analyse der Indizes in Abschnitt 7.3.3 erfolgt.

### 7.3.1 Aufbau und Konfiguration der Szenarien

Der Aufbau des Simulationsszenarios folgt dem in diesem Kapitel eingeführten Simulationsszenario mit 200 Basic-Services und 50 Ressourcen-Services, die auf ein Netzwerk mit 200 Knoten verteilt sind. In jedem Simulationslauf werden 2000 Benutzeranfragen simuliert. Dieses grundlegende Simulationsszenario wird nun abhängig von der jeweiligen Risikobehandlungsstrategie wie folgt variiert:

- RS1 (Risikoakzeptanz): Diese Strategie simuliert Ressourcenkonsumenten, die alle Risiken akzeptieren. Im Erfolgsfall einer Serviceausführung werden sie durch einen Nutzen von 1 belohnt. Bei Ausfall eines Basic-Service während seiner Ausführung erreichen sie einen Nutzen von 0. Ein Basic-Service nimmt bei einer fehlgeschlagenen Serviceausführung keinen weiteren Versuch wahr, die Ausführung des Basic-Service ein weiteres Mal zu starten. Dies entspricht dem Risikoakzeptanzmodell, wie es in Kapitel 4 eingeführt wurde.



- RS2 (Risikodiversifikation durch Redundanz): Die Strategie der Redundanz findet sich häufig in der Praxis, um die Ausführung von Anwendungen auf risikobehafteten Ressourcen durchzuführen. Aus diesem Grund wird sie neben der Risikoakzeptanzstrategie zur Bewertung der Gesamtperformanz der Versicherung herangezogen. Dabei gelten für die Redundanzstrategie folgende Annahmen:
  - Die redundante Ausführung eines Basic-Service erfolgt auf Ressourcen des gleichen Anbieters. Es wird dabei angenommen, dass die Serviceausführungen auf verschiedenen Ressourcenbündeln eines Ressourcenanbieters unabhängig voneinander sind.
  - Es wird unabhängig von der Risikoeinschätzung des Agenten immer die doppelte Menge an Ressourcen nachgefragt, der Basis-Service wird doppelt ausgeführt. Dabei fallen die doppelten Kosten einer Transaktion an.
- RS3 (Risikotransfer auf eine Versicherung): Für die Analyse der Gesamtperformanz wird die Versicherung von allen Basic-Services zwangsweise abgeschlossen. Dies schließt eine Wahlmöglichkeit aus und erhöht die Vergleichbarkeit der Strategien, da in den beiden alternativen Strategien ebenfalls keine Wahlmöglichkeiten existieren. Weiterhin gelten für die Versicherungsstrategie aus Gründen der Vereinfachung des Szenarios folgende Annahmen:
  - Die Versicherung stellt Ersatzressourcen zur Kompensation des Schadens bereit. Die Ersatzressourcen werden in vollem Umfang geliefert, wobei die Versicherung keinen Preisrisiken bei Wiederbeschaffung der Ressourcen ausgesetzt ist.
  - Die Ersatzressourcen sind zu 100% zuverlässig.
  - Die Schadensregulierung bzw. die Bereitstellung der Ersatzressourcen erfolgt ohne Verzögerung. Es wird keine Zeit für die Allokation der Ersatzressourcen benötigt.

Diese drei Risikobehandlungsstrategien werden neben den beiden bekannten Risikoprofilen mit niedrigem und hohem Risiko zusätzlich mit dem in Abbildung 7.22 gezeigten Risikoprofil mit sehr hohem Risiko simuliert. Ein Basic-Service bs5 weist in diesem Risikoprofil eine sehr geringe Zuverlässigkeit von zirka 30% aus, während ein Basic-Service bs2 weiterhin eine hohe Zuverlässigkeit hat. Die Bandbreite der Zuverlässigkeit ist in diesem Risikoprofil stärker ausgeprägt. Ziel ist es, durch eine weitere Erhöhung des Risikos die Grenzen der Risikobehandlungsstrategien zu untersuchen. In den beiden folgenden Abschnitten werden die drei Risiko-steuerungsverfahren in diesen drei verschiedenen Risikoprofilen untersucht, wobei im folgenden Abschnitt zuerst die Indikatoren betrachtet werden und im darauf folgenden Abschnitt die daraus



abgeleiteten Indizes.

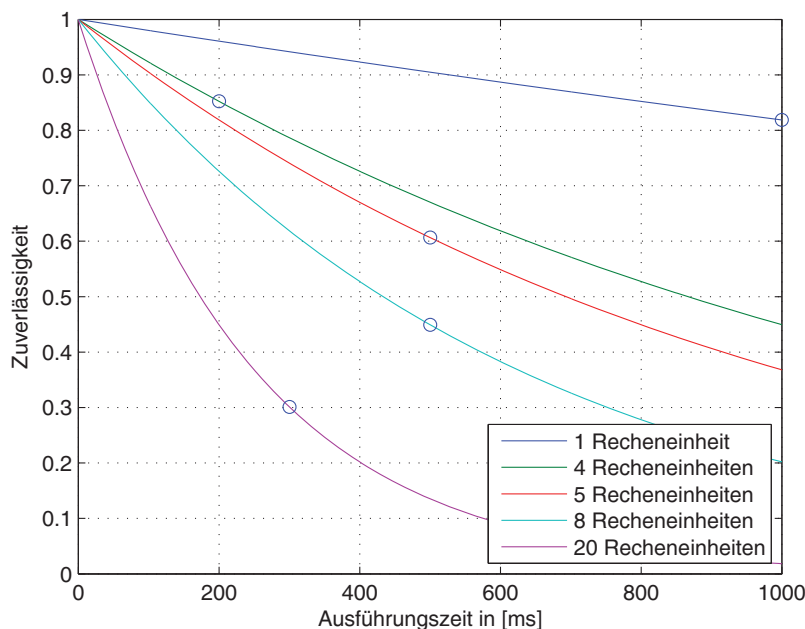


Abbildung 7.22: Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor  $k = 2,0 \cdot 10^{-4}$

### 7.3.2 Analyse der Indikatoren

Die Analyse der Indikatoren erfolgt über den Mittelwert und die Standardabweichung, wobei die Stichprobe alle gemessenen Ereignisse eines Simulationslaufs umfasst. Ziel der Analyse ist ein Vergleich der drei Risikostrategien in verschiedenen Risikoprofilen hinsichtlich der 8 ausgewählten Indikatoren. Die Annahme der Unabhängigkeit der Indikatoren wurde durch die Berechnung der Krosskorrelation zwischen je zwei Indikatoren überprüft. Dabei wurden Werte, die im Maximum zwischen  $+0,1$  und  $-0,1$  liegen, berechnet. Dies bedeutet, dass es keine linearen Abhängigkeiten zwischen den Indikatoren gibt, sodass die Annahme der Unabhängigkeit der Indikatoren als erfüllt gelten kann.

Abbildung 7.23 zeigt die Mittelwerte der Indikatoren und Abbildung 7.24 die dazugehörigen Werte der Standardabweichung für das Risikoprofil mit hohem Risiko. Die Abbildungen der Risikoprofile mit niedrigem und sehr hohem Risiko finden sich im Anhang C.2. Die genauen Werte der Indikatoren fasst Tabelle 7.8 für das Risikoprofil mit hohem Risiko zusammen.

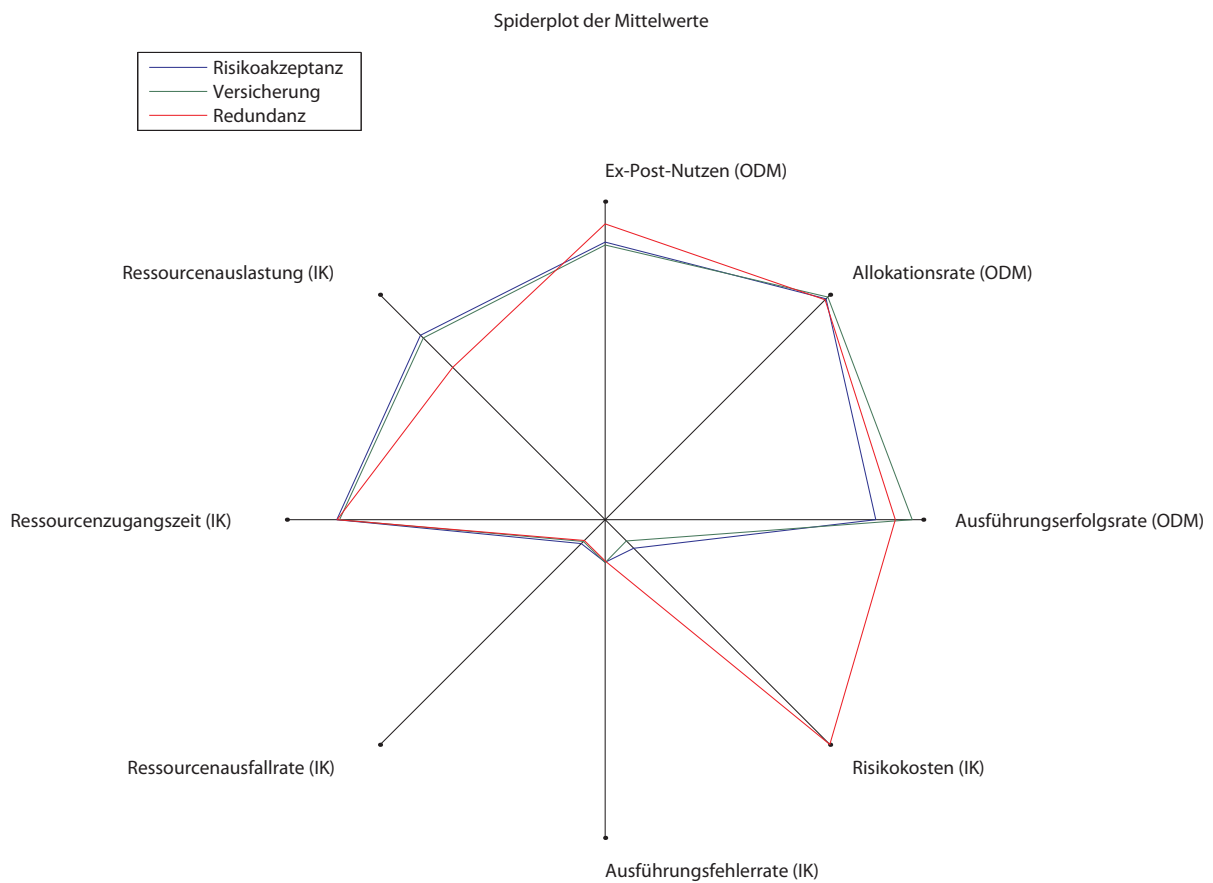


Abbildung 7.23: Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko; alle Werte sind normiert auf das Intervall zwischen 0 und 1.

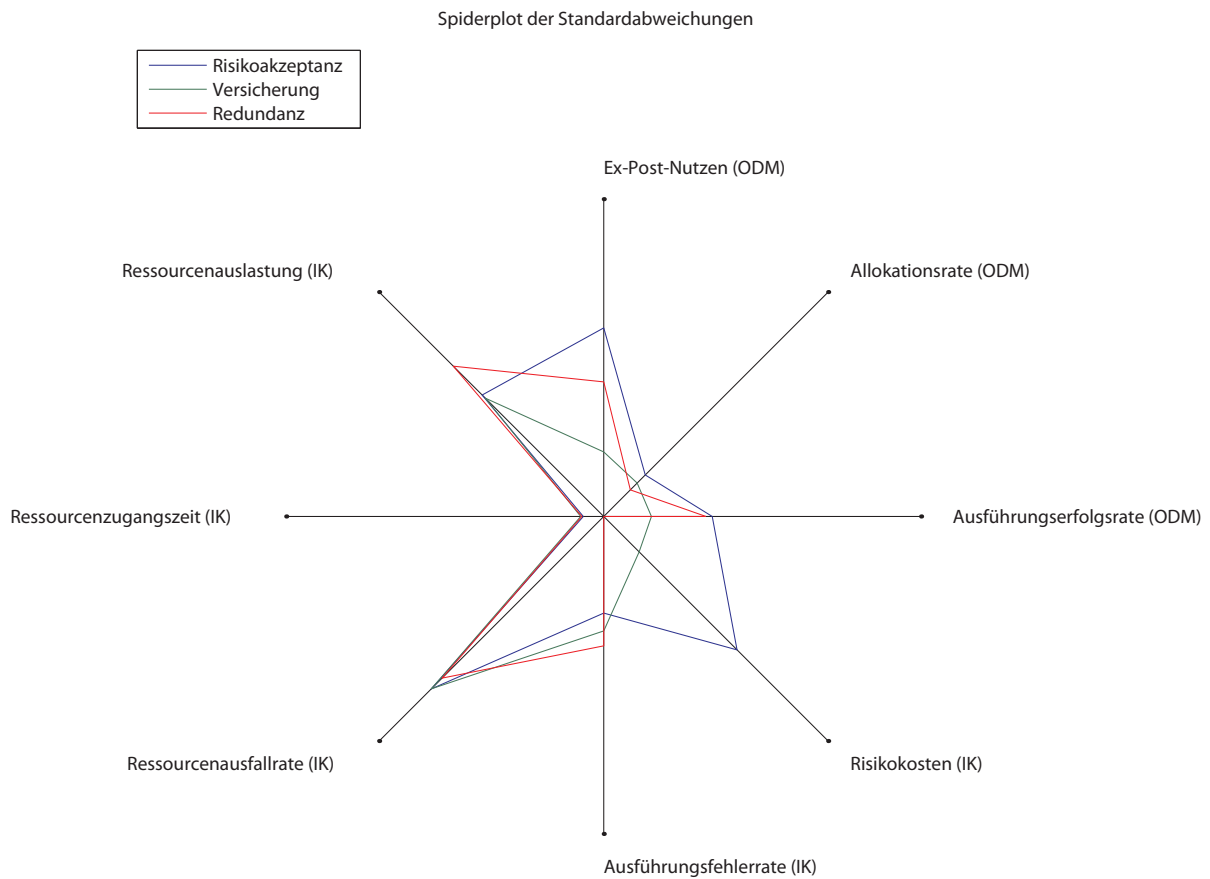


Abbildung 7.24: Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko; alle Werte sind normiert auf das Intervall zwischen 0 und 1.

Es lassen sich folgende Erkenntnisse für die einzelnen Indikatoren in den oben beschriebenen Szenarien ableiten:

- **Ex-Post-Nutzen:** In allen untersuchten Risikoprofilen zeigt die Redundanzstrategie den höchsten mittleren Ex-Post-Nutzen, wobei der mittlere Ex-Post-Nutzen eines Ressourcenkonsumenten einen ähnlichen hohen mittleren Ex-Post-Nutzen im Szenario mit sehr hohem Risiko erreicht. In den Szenarien mit niedrigem und hohem Risiko liegen die Strategien Risikoakzeptanz und Versicherung bezüglich des mittleren Ex-Post-Nutzens gleichauf, während der mittlere Ex-Post-Nutzen im Risikoprofil mit sehr hohem Risiko bei Risikoakzeptanz deutlich abnimmt. Dies weist auf hohe Prämien hin, die an die Versicherung zu bezahlen sind, um das Risiko zu transferieren.

Ein einheitlicheres Bild ergibt sich bei den Standardabweichungen des Ex-Post-Nutzens in den unterschiedlichen Risikoszenarien. Hier zeigt sich deutlich der Vorteil der Versicherung, deren Ex-Post-Nutzen eine deutlich geringere Standardabweichung aufweist. Wie bereits im Abschnitt 7.2 gezeigt, führt diese geringere Streuung des Ex-Post-Nutzens zu einer Bevorzugung der Versicherung durch die Basic-Services. Ein Vergleich der beiden Strategien Risikoakzeptanz und Redundanz macht deutlich, dass ein zunehmendes Risiko den mittleren Ex-Post-Nutzen eines Ressourcenkonsumenten bei Risikoakzeptanz stark vermindert.

- **Allokationsrate:** Dieser Indikator ist eine Kennzahl des Ressourcenmanagement, die den Erfolg der Allokationsmechanismus misst. Die gemessenen Werte aller drei Risikoprofile und Risikobehandlungsstrategien deuten auf ein sehr stabiles Verhalten des Allokationsmechanismus hin, der nur eine sehr geringe Anzahl an Fehlallokationen aufweist.
- **Ausführungserfolgsrate:** Der Indikator Ausführungserfolgsrate gibt Hinweise darauf, wie viele durch das Ressourcenmanagement zugewiesenen Ressourcen durch die Ausführung eines Basic-Service erfolgreich konsumiert und nicht verschwendet werden. Es werden alle erfolgreichen Ausführungen von Basic-Services zur Berechnung der Rate herangezogen. Im Fall einer Redundanzstrategie werden nur diejenigen Ausführungen nicht gezählt, bei denen in beiden Ressourcenbündeln bei der Ausführung eines Basic-Service einen Fehler auftritt. Sind die Ressourcen versichert, garantiert die Versicherung eine Ausführung des entsprechenden Basic-Service. Dies führt dazu, dass die Versicherung in allen untersuchten Szenarien die höchste Ausführungserfolgsrate bei gleichzeitig geringster Streuung besitzt. Die beiden Strategien Risikoakzeptanz und Redundanz folgen mit teilweise deutlichem Abstand. Je höher das Risiko wird, desto größer wird der Abstand zwischen den

Strategien hinsichtlich des Indikators Ausführungserfolgsrate. Ein Ressourcenkonsument sollte bei erhöhtem Risiko von einer Risikoakzeptanzstrategie hin zu einer Strategie mit Redundanz oder Versicherung schwenken.

- **Risikokosten:** Als Risikokosten werden die Kosten bezeichnet, die entstehen, einen Basic-Service auf den risikobehafteten Ressourcen auszuführen. Die Risikokosten der Risikoakzeptanzstrategie geben einen Hinweis darauf, wie hoch die Kosten sind, die der Ressourcenkonsument selbst tragen muss. Im Szenario mit einer Redundanzstrategie geben die Risikokosten an, wie hoch die Mehrkosten durch die Redundanz sind. Bei der Versicherungsstrategie sind die Risikokosten die Kosten der Versicherungsprämie.

Die Kosten der jeweiligen Risikostrategie unterscheiden sich dabei deutlich. Die im Mittel höchsten Kosten fallen bei einer Redundanzstrategie an, da hier unabhängig vom Risiko immer die doppelte Menge an Ressourcen verbraucht wird. Der Mehrkosten liegen hier bei 100%. Die durchschnittlich geringsten Risikokosten müssen bei einer Versicherung aufgewandt werden. Zusätzlich weist der Risikotransfer auf eine Versicherung eine sehr geringe Streuung auf, was einen Hinweis darauf gibt, dass die Versicherungsprämien um ihren Mittelwert wenig schwanken. Ein Vergleich der Risikokosten für die unterschiedlichen Risikoprofile zeigt, dass die Versicherung in allen untersuchten Szenarien im Mittel die geringsten Risikokosten aufweist. Je höher das Risiko ist, desto deutlicher werden die Vorteile der Versicherung gegenüber der Risikoakzeptanzstrategie. Die Risikokosten der Versicherung nähern sich dabei den Risikokosten der Redundanzstrategie an.

- **Ressourcenfehlerrate und Ausführungsfehlerrate:** Beide Indikatoren charakterisieren den Ausfall der Ressourcen, wobei die Ressourcenfehlerrate die grundlegende Fehlerrate der Ressourcen beschreibt und die Ausführungsfehlerrate den Ausfall eines Ressourcenbündels während einer Basic-Service-Ausführung abbildet. Beide Indikatoren weisen einen niedrigen Mittelwert und hohe Streuung der Werte auf. Dies entspricht den Messergebnissen der in Kapitel 2 vorgestellten Studien realer Grid-Systeme mit heterogenen Ressourcen. Diese Charakteristik wird durch die Ressourcenfehlerrate abgebildet. Die hohe Streuung der Ausführungsfehlerrate ist ein Indikator der Streuung der unterschiedlichen Anwendungsanforderungen in Bezug auf ihre Ausführungszeit und die verbrauchten Ressourcen.
- **Ressourcenzugangszeit:** Diese Zeit repräsentiert den zeitlichen Mehraufwand, der für die Allokation der Ressourcen und das Risikomanagement benötigt wird. Die Ausführungszeit des Dienstes selbst wird nicht in die Ressourcenzugangszeit mit einbezogen. Alle drei

Tabelle 7.8: Werte der Indizes der Metrikpyramide für das Szenario mit hohem Risiko; die Spalte M beinhaltet die Mittelwerte und die Spalte S die Standardabweichung der Werte

Indikator	Akzeptanz		Versicherung		Redundanz	
	M	S	M	S	M	S
Erfolgsrate (ODM)	0,8510	0,2078	0,9654	0,0913	0,9122	0,1575
Allokationsrate (ODM)	0,9820	0,0598	0,9900	0,0424	0,9781	0,0717
Ex-Post-Nutzen (ODM)	0,8731	0,3331	0,8635	0,2227	0,9303	0,2547
Ressourcenauslastung (IK)	0,8201	0,2077	0,8081	0,2100	0,6776	0,2742
Ressourcenzugangszeit (IK)	0,8430	0,0248	0,8346	0,0283	0,8422	0,0254
Ressourcenausfallrate (IK)	0,1059	0,3078	0,0962	0,2948	0,0923	0,2895
Ausführungsfehlerrate (IK)	0,1336	0,2022	0,1344	0,2015	0,1295	0,2064
Risikokosten (IK)	0,1269	0,3331	0,0943	0,0995	0,9990	0,0000

untersuchten Szenarien zeigen hier im Mittel die gleiche Wertausprägung bei kleiner Standardabweichung. Dies bedeutet, dass der Abschluss einer Versicherung im Verhältnis zur Ressourcenallokation nicht signifikant mehr Zeit benötigt.

- **Ressourcenauslastung:** Die Ressourcenauslastung der verfügbaren Ressourcen unterscheidet sich deutlich von der Risikoakzeptanz und Versicherung auf der einen Seite und der Redundanzstrategie auf der anderen Seite. Dies liegt darin begründet, dass bei einer Redundanzstrategie Ressourcenanbieter mit einer kleineren Kapazität Anfragen nach der doppelten Ressourcenmenge nicht erfüllen können und dadurch ihre Ressourcen nur schlecht ausgelastet sind, während Ressourcenanbieter mit hoher Kapazität ihre Auslastung steigern können. Dies senkt im Mittel die Auslastung der Ressourcen bei einer gleichzeitig hohen Standardabweichung. Bei Risikoakzeptanz und Versicherung verändert sich die Ressourcenauslastung nicht, da die Risikoakzeptanzstrategie bei fehlerhafter Basic-Service-Ausführung keinen weiteren Versuch mehr unternimmt und die Versicherung die Ressourcen selbst bereitstellt, sodass in diesen Szenarien keine weiteren Ressourcen auf dem Spotmarkt bei Bedarf nachgefragt werden.

In allen untersuchten Szenarien weist der Risikotransfer auf eine Versicherung sehr gute Ergebnisse im Vergleich zu Risikoakzeptanz und Redundanz für die untersuchten Indikatoren auf. Durch die dynamische Anpassung der Prämien hat eine Risikobehandlungsstrategie mit einer Versicherung deutliche Vorteile gegenüber den anderen Strategien und kann ohne Anpassung in verschiedenen Risikoprofilen eingesetzt werden, während die Strategie der Risikoakzeptanz eine

ähnliche Performanz bei geringem Risiko erreicht, jedoch bei hohem Risiko nicht zu empfehlen ist. Bei hohem Risiko weist die Redundanzstrategie eine bessere Performanz auf. Um eine gesamtheitliche Aussage treffen zu können, welche Strategie in welchem Szenario zu bevorzugen ist, werden im folgenden Abschnitt die Indikatoren zu Indizes aggregiert und miteinander verglichen.

### 7.3.3 Analyse der Indizes

Dieser Abschnitt analysiert die Indizes Infrastrukturkosten (IK), On-Demand-Verfügbarkeit (ODM) und den übergeordneten Gesamtindex Social-Loss, der in Kapitel 6 eingeführt wurde. Der Index Infrastrukturkosten aggregiert die Indikatoren Ressourcenauslastung, Ressourcenzugangszeit, Ressourcenausfallrate, Ausführungsfehlerrate und Risikokosten, während der Index On-Demand-Verfügbarkeit die Indikatoren Ex-post-Nutzen, Allokationsrate und Ausführungserfolgsrate zusammenfasst<sup>8</sup>. Beide Indizes, Infrastrukturkosten und On-Demand-Verfügbarkeit, werden mit gleichem Gewicht in die Berechnung des Social-Loss-Index einbezogen. Die Mittelwerte und die Standardabweichungen der Indizes sind in Abbildung 7.25 für das Risikoprofil mit hohem Risiko dargestellt. Die entsprechenden Grafiken für die Risikoprofile mit niedrigem und sehr hohem Risiko finden sich im Anhang C.2. Die Aussagen, die für das Risikoprofil mit hohem Risiko gemacht werden, gelten ähnlich auch für diese Risikoprofile.

Ein Vergleich der drei Strategien zeigt:

- **Redundanz:** Die Vorteile eines höheren ODM-Index werden durch die erhöhten Risikokosten und eine niedrigere mittlere Auslastung der Ressourcen, die zu einem deutlich erhöhten IK-Index führen, egalisiert. Eine Erhöhung des Risikos verschlechtert den Social-Loss um 4%.
- **Risikoakzeptanz:** Die Indizes der Risikoakzeptanzstrategie zeigen keine Ausschläge wie bei der Redundanz. Die Abweichung bezüglich des Infrastrukturindex fällt kleiner aus als die Abweichung bezüglich der des ODM-Index im Vergleich zur Versicherung. Eine Erhöhung des Risikos verschlechtert den Social-Loss um 8%, was doppelt so viel wie bei der Redundanzstrategie ist und 5% schlechter als mit der Risikotransferstrategie.
- **Versicherung:** Über alle Indizes hinweg zeigt die Versicherung die beste Performanz. Bei einer Erhöhung des Risikos verschlechtert sich der Social-Loss um 3%, was den besten Wert im Vergleich zu den beiden Alternativstrategien darstellt.

---

<sup>8</sup>Siehe Kapitel 6.5 für die Details der Berechnung.

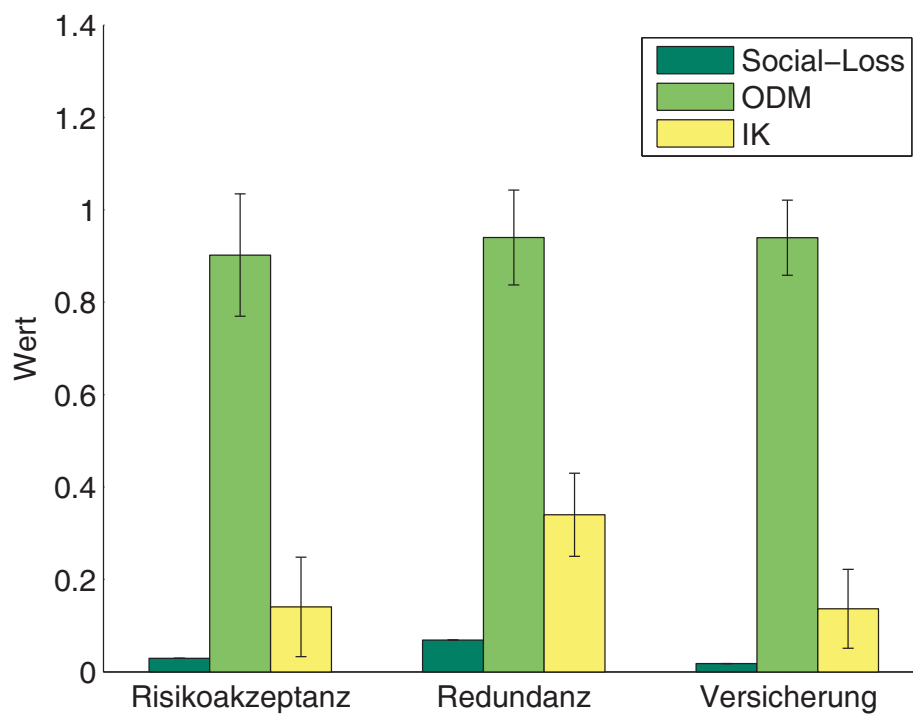


Abbildung 7.25: Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Loss-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko



Tabelle 7.9: Werte der Indizes der Metrikpyramide für das Szenario mit hohem Risiko

Index	Redundanz	Akzeptanz	Versicherung
Social-Loss	0,0690	0,0293	0,0181
s_ODM	0,1027	0,1324	0,0815
m_ODM	0,0598	0,0980	0,0604
s_IK	0,0899	0,1078	0,0854
m_IK	0,3402	0,1407	0,1364

Tabelle 7.10: Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wertes für das Szenario mit hohem Risiko

ODM-Gewichtung	IK-Gewichtung	Social-Loss		
		Redundanz	Akzeptanz	Versicherung
0,1	0,9	0,1128	0,0310	0,0243
0,3	0,7	0,0909	0,0301	0,0212
0,5	0,5	0,0690	0,0293	0,0181
0,7	0,3	0,0470	0,0284	0,0150
0,9	0,1	0,0251	0,0275	0,0118

In Tabelle 7.10 werden die Werte des Social-Loss-Index für verschiedene Gewichtungen der beiden Indizes On-Demand-Verfügbarkeit (ODM) und Infrastrukturkosten (IK) abgebildet. Ausgehend von einer identischen Gewichtung der beiden Indizes, werden die Gewichtung des ODM-Index erhöht und gesenkt und die berechneten Werte des Social-Loss-Index für die drei Risikobehandlungsstrategien angegeben. Hierbei zeigt sich, dass die Attraktivität der Redundanzstrategie mit erhöhter Gewichtung des ODM-Index zunimmt und für eine Gewichtung von 0,9 des ODM-Index eine höhere Gesamtperformanz als die Risikoakzeptanzstrategie zeigt. Der Verlust, der durch die Mehrkosten in der Redundanzstrategie entsteht, wird durch die Vorteile hinsichtlich einer erhöhten Verfügbarkeit ausgeglichen. Die beste Strategie in allen Kombinationen stellt jedoch die Versicherung dar, die ebenfalls von der Erhöhung des Gewichts des ODM-Index profitiert.

Bei der Auswahl der richtigen Risikobehandlungsstrategie kommt es darauf, wie stark die Indizes ODM und IK gewichtet werden. Steht eine hohe Verfügbarkeit im Vordergrund, so bieten sich eine Versicherungsstrategie oder eine Strategie durch Redundanz an, wobei eine Redundanzstrategie einen mehr als doppelt so hohen Social-Loss im Vergleich zur Versicherungsstrategie

erwarten lässt. Stehen die Infrastrukturkosten im Vordergrund bei der Auswahl der Risikobehandlungsstrategie, so ist der Einsatz einer Risikoakzeptanz- oder einer Versicherungsstrategie zu empfehlen. Hier fällt der Unterschied des Social-Loss-Index deutlich geringer aus; die Versicherung weist einen um 20% geringeren Social-Loss auf.

Mit der Zu- und Abnahme der Risikos eines Simulationsszenarios verschiebt sich auch der Zeitpunkt, ab wann eine Strategie einer anderen vorzuziehen ist, sodass eine Abwägung der Risikostrategie individuell auf das jeweilige Risikoprofil abgestimmt werden muss<sup>9</sup>. Nimmt das Risiko zu, so gewinnt die Strategie der Redundanz an Attraktivität, während die Strategie einer Risikoakzeptanz bei abnehmenden Risiko einen geringeren Social-Loss aufweist. Die Versicherungsstrategie ist hingegen weniger stark vom individuellen Risikoprofil des Szenarios abhängig. Sie weist sowohl bei zunehmendem Risiko wie auch bei abnehmendem Risiko einen geringen Social-Loss auf. Die Strategie des Risikotransfers auf eine Versicherung ist somit die effizienteste der untersuchten Risikomanagementstrategien in den drei Risikoprofilen unter Einbezug der oben beschriebenen Indikatoren und Indizes der Metrikpyramide.

## 7.4 Zusammenfassung

Ausgehend vom Aufbau realer Grid-Systeme wurde zu Beginn des Kapitels ein realistisches Grid-Szenario abgeleitet. Das Szenario umfasst 5 verschiedene Basic-Service-Konfigurationen und 5 verschiedene Ressourcenbündelkonfigurationen, die von den Amazon EC2-Cloud-Computing Services abgeleitet wurden. Für dieses Szenario wurde anschließend ein 2-fach klassifizierender Versicherungstarif mit je 4 Merkmalsausprägungen eines Tarifmerkmals für die Ausführung von Basic-Services auf standardisierten Ressourcenbündeln definiert. Dieser Versicherungstarif ordnet die Basic-Services anhand der Tarifmerkmale Ausführungszeit und Anzahl der Recheneinheiten in die entsprechende Tarifgruppe ein.

Im ersten Teil der Simulationsstudie steht das Verhalten der Grid-Versicherung und der Versicherungsnachfrage im Vordergrund. Die wichtigsten Ergebnisse, die dabei gewonnen wurden, sind:

- **Versicherungsperiode:** Eine Verkürzung der Versicherungsperiode reduziert die Schadenssumme, die in einer Periode anfällt. Dies führt zu weniger stark ausgeprägten Ausschlägen

---

<sup>9</sup>Die Vergleichstabelle des Social-Loss-Index für die Simulationsszenarios mit geringem Risiko und sehr hohem Risiko finden sich in Anhang C.2.

zwischen den Versicherungsperioden und damit zu weniger starken Schwankungen der Versicherungsprämien.

- **Entwicklung der Tariffaktoren:** Die Tariffaktoren zeigen ein oszillierendes Verhalten. Dies lässt sich auf die Verteilung der Schäden zurückführen, die im simulierten Grid-Szenario stark linksschief ist. Die teuersten Ressourcenbündel fallen am häufigsten aus, da sie die größte Menge an Recheneinheiten verbrauchen. Zusätzlich treten Spätschäden auf, für die Reserven durch das Marginalsummenverfahren gebildet werden müssen. Jedoch ist der Einfluss der Spätschäden gegenüber der linksschiefen Verteilung der Schäden gering.
- **Sicherheitszuschlag:** Der Sicherheitszuschlag legt die Strategie der Versicherung fest. Die Implementierung einer dynamischen Anpassung der Sicherheitszuschläge bei Unter- oder Überschreiten festgelegter Schwellenwerte gibt der Versicherung die Möglichkeit, auf verändertes Risiko zu reagieren und den Ruin der Versicherung zu vermeiden.
- **Versicherungsnachfrage aller Versicherungsnehmer:** Langfristig kann ein Ressourcenkonsument durch den Abschluss einer Versicherung einen höheren Ex-Post-Nutzen erzielen als mit der Strategie einer Risikoakzeptanz. Nur wenige Basic-Services, jene, die geringem Risiko ausgesetzt sind, entscheiden sich für eine Akzeptanz des Risikos.
- **Versicherungsnachfrage der einzelnen Basic-Service-Typen:** Die Einschätzung des Risikos und die Zahlungsbereitschaft des Versicherungsnehmers unterscheiden sich zwischen den 5 Basic-Service-Typen deutlich. Die einzelnen Gruppen von Basic-Service-Agenten passen ihre Strategie der von der Versicherung geforderten Prämie an, um langfristig einen hohen Ex-Post-Nutzen zu erzielen. Zusätzlich richten sich die einzelnen Basic-Service-Gruppen ihrem Risiko entsprechend zueinander aus. Die Gruppe der Basic-Services mit dem geringsten Risiko wählt eine entgegengesetzte Strategie zu der Gruppe der Basic-Services mit dem größten Risiko.

Im zweiten Teil der Simulationsstudie stand die Gesamtperformanz des Ressourcen- und Risikomanagements für drei Risikobehandlungsstrategien im Vordergrund. Neben der Strategie eines Risikotransfers auf die Grid-Versicherung wurden die Strategien der Risikoakzeptanz und der Risikodiversifikation durch Redundanz evaluiert. Die Analyse erfolgte durch die in Kapitel 6 eingeführte Metrikpyramide. Eine getrennte Analyse der Indikatoren und Indizes kommt zu dem Ergebnis, dass ein Risikotransfer auf eine Versicherung in verschiedenen Risikoprofilen den geringsten Social-Loss im Vergleich zu den beiden Alternativverfahren aufweist. Der Grund hierfür liegt in der garantierten Ausführung des Basic-Service durch die Versicherung und die dynamische Anpassung der Risikokosten an die auftretenden Schäden.

Zusammenfassend zeigt der in diesem Kapitel simulierte und analysierte Risikotransfer auf eine Versicherung einen effizienten Ansatz zum Umgang mit den Risiken bei der Ausführung von Services auf Grid-Ressourcen.



# Kapitel 8

## Zusammenfassung und Fazit

*„Computing may someday be organized as a public utility just as the telephone system is organized as a public utility.“ (John McCarthy) [Carr08, S. 59]*

Die zunehmende Standardisierung von Grid- und Cloud-Computing-Technologien ermöglicht es Unternehmen, bei Bedarf Software-Services und Infrastrukturressourcen als Utility zu beziehen und die konsumierte Leistung nach einem Pay-per-Use-Modell abzurechnen [Rapp04]. Beim Bezug der Ressourcen und Services treten jedoch technische Risiken auf, die die Zuverlässigkeit beeinträchtigen können [IJSE07] [DaPZ07]. In den vorangegangenen Kapiteln wurde gezeigt, dass die technischen Risiken, die bei der Ausführung von Grid-Services auf Grid-Ressourcen auftreten können, versicherbar sind. Die Simulationsergebnisse weisen darauf hin, dass Grid-Versicherungen ein effizienteres Verfahren zur Behandlung der technischen Risiken darstellen als Risikoakzeptanz und Redundanz hinsichtlich der untersuchten Metriken. Zusätzlich wurde gezeigt, dass ein ökonomisches, dezentrales Ressourcenallokationsverfahren von einem integrierten Risikomanagement profitiert und die Zuverlässigkeit des Grid- bzw. Cloud-Computing-Systems erhöhen kann.

Im diesem abschließenden Kapitel werden die wesentlichen Ergebnisse zusammengefasst, thematisch eingeordnet, Einschränkungen der Simulationsstudie diskutiert und zukünftige Anwendungsmöglichkeiten von Versicherungen in Grid- und Cloud-Computing-Systemen vorgestellt. Abschnitt 8.1 gibt eine zusammenfassende Übersicht über die einzelnen Kapitel der Arbeit. Implikationen für die Forschung thematisiert Abschnitt 8.2. Hier werden die Ergebnisse der Arbeit aufgegriffen und mögliche Erweiterungen diskutiert. Der letzte Abschnitt dieses Kapitels, Abschnitt 8.3, gibt einen Überblick über Anwendungsszenarien für Ressourcenkonsumenten, Ressourcenanbieter und Versicherungen und beschließt die Arbeit mit einem Fazit.

## 8.1 Zusammenfassung der Arbeit

Kapitel 1 skizzierte die Vorteile von Grid- und Cloud-Computing-Systemen und motivierte den Einsatz von Risikomanagement, um auftretende technische Risiken zu steuern. Das Ziel der Arbeit, eine Versicherung zum Transfer des technischen Risiko zu implementieren und zu evaluieren, wurde ebenfalls in diesem Kapitel vorgestellt. Für die Zielerreichung wurden 4 Forschungsfragen definiert, die im weiteren Verlauf der Arbeit betrachtet wurden.

Die technischen Konzepte und daraus entstehende Risiken von Grid-Systemen wurden in Kapitel 2 betrachtet und analysiert. Das Kapitel führte die Open-Grid-Services-Architektur (OGSA) ein, die eine serviceorientierte Architektur zur Realisierung von Grid-Systemen darstellt und grundlegende Schnittstellen zum Bezug von Grid-Ressourcen und Grid-Services als Web-Service festlegt [FKSB<sup>+</sup>06]. Mithilfe der Open-Grid-Services-Architektur und ihren standardisierten Schnittstellen ist es möglich, Risiken, die durch fehlerhafte Spezifikation entstehenden Risiken zu verringern bzw. im Idealfall zu eliminieren. Des Weiteren werden Dienste zum Messen und Überwachen der Servicequalität bereitgestellt, die benötigt werden, um eine fehlerhafte Ausführung und den dadurch entstandenen Schaden feststellen zu können.

Die folgenden Abschnitte gaben eine Übersicht über verschiedene Ausprägungen und Anwendungsfelder von Grid-Systemen. Studien über die technischen Risiken in Grid-Systemen folgten der Darstellung der Grid-System-Ausprägungen. Im Fokus der Studien standen Risiken beim Einsatz von rechenintensiven Grid-Systemen, die je nach Konfiguration und Aufbau eine Zuverlässigkeit von zirka 70% aufwiesen [IJSE07]. Das Modell von Dai [DaPZ07] zur Modellierung der Zuverlässigkeit von Grid-Systemen wurde vorgestellt. Mit seiner Hilfe wurde der stochastische Prozess des Ausfalls eines Grid-Systems im Simulationsmodell modelliert.

In Kapitel 3 wurde das abstrakte Simulationsmodell der CATNETS Grid-Ökonomie eingeführt, das zwei mit einander verknüpfte Grid-Märkte beschreibt [ERSR<sup>+</sup>05]. Auf dem Ressourcenmarkt werden standardisierte Ressourcenbündel gehandelt, während auf dem Servicemarkt der Fokus auf dem Handel von homogenen Services liegt. Zusätzlich wurde das Netzwerkmodell der Simulation beschrieben, das einen wichtigen Einfluss auf die Grid- und Cloud-Systeme haben kann, da der Bezug und die Benutzung der Ressourcen und Services über ein meist öffentliches Netzwerk erfolgen.

Bei der Suche nach Ressourcen- bzw. Serviceanbietern kommt der Peer-To-Peer-Ansatz *Flooding* zum Einsatz [DiPi06]; die Verhandlung der Güter erfolgt auf beiden Märkten durch eine bilaterale Verhandlung unter Verwendung des monotonic-concession Protokolls [ZIRo96]. Die

Verhandlungsstrategie des dezentralen, ökonomischen Allokationsmechanismus passt sich durch einen co-evolutionären Algorithmus an die dynamische Grid-Umgebung an [SmTa98]. Dieser Algorithmus kommt im weiteren Verlauf der Arbeit ebenfalls bei der Strategie der Versicherungsnachfrage zum Einsatz. Abschließend wurden in diesem Kapitel die Anforderungen an das integrierte Risikomanagement definiert und die Erweiterungen des Simulationsmodells daraus abgeleitet.

Das Kapitel 4 gab einen Überblick über den aktuellen Stand des Risikomanagements in Grid-Systemen. Ein speziell auf Grid-Systeme bezogener Risikomanagementzyklus wurde eingeführt [DGPB<sup>+</sup>06] und 4 allgemeine Verfahren zur Behandlung von Risiken im Kontext von Grid-Systemen diskutiert. Von diesen 4 Verfahren stand der Risikotransfer auf eine Versicherung im Vordergrund. Die Verfahren Risikoakzeptanz und Risikodiversifikation durch Redundanz dienen als Referenzszenarien in der Simulationsstudie. Die Vermeidung von Risiken wurde nicht weiter verfolgt, da auf einem Grid-System immer ein potenzielles Risiko besteht, dass der angeforderte Grid-Service nicht zur Verfügung steht.

Das Modell der Grid-Versicherung stand im Mittelpunkt von Kapitel 5. Ausgehend von einer allgemeinen Beschreibung der Funktionen und Aufgaben von Versicherungen wurde die Versicherbarkeit von Grid-Risiken mithilfe des Modells von Karten [Kart72] näher untersucht, mit dem Ergebnis, dass technische Grid-Risiken versicherbar sind. Die Bestandteile und Ausprägungen von Versicherungen im Kontext von Grid-Systemen standen in den folgenden Abschnitten im Vordergrund. Eine Analyse der Bestandteile einer Versicherung identifizierte eine Reihe an Parameter, die in der Simulation näher auf ihren Einfluss und ihre Entwicklung über die Zeit hin untersucht wurden. Die Parameter sind die Versicherungsperiode, die Tarifmerkmale, der Sicherheitszuschlag und die Strategie der Versicherungsnachfrage.

Die Berechnung der Versicherungsprämie erfolgte durch einen multiplikativen Tarif, der 2 Tarifmerkmale verwendet und das Marginalsummenverfahren zur Schätzung der Tariffaktoren einsetzt. Die Strategie der Versicherungsnachfrage wurde durch zwei Parameter bestimmt: die persönliche Risikoeinschätzung des jeweiligen Software-Agenten und die individuelle Zahlungsbereitschaft für die Versicherungsprämie. Diese Parameter werden durch den in Kapitel 3 beschriebenen Lernalgorithmus verändert.

Abschließend wurden in diesem Kapitel verschiedene Möglichkeiten der Integration der Versicherung in die CATNETS Grid-Ökonomie diskutiert. Aus Gründen der Vereinfachung der Komplexität wurde dabei festgelegt, dass die Versicherung als Monopol implementiert wird und die Kompensation der Schäden monetär erfolgt.



Kapitel 6 beschrieb die Simulation der Grid-Versicherung und bereitete die Evaluierung der Simulationsergebnisse vor. Zu Beginn des Kapitels wurde die Forschungsmethode Simulation eingeführt und von einer rein analytischen Betrachtung und dem realen Funktionstest an einem Prototypen oder produktiv eingesetztem Grid-System abgegrenzt. Der Prozess zur Durchführung einer Simulationsstudie nach Law [Law07] wurde eingeführt, an dessen Ablaufschema sich die Struktur der Simulationsstudie in dieser Arbeit orientiert. Die folgenden Abschnitte des Kapitels beschrieben den Aufbau und die Konfiguration der Simulationsumgebung sowie die Implementierung der Grid-Versicherung als Software-Agenten.

Auf Grundlage der Versicherungsimplementierung wurden der Simulationsentwurf skizziert und detailliert der Untersuchungsgegenstand, die dazugehörige Fragen und die Metriken zur Evaluation des Verhaltens der Grid-Versicherung definiert, die den ersten Teil der Simulationsstudie umfassen. Für den zweiten Teil der Simulationsstudie, die einen Vergleich der Gesamtperformanz der Grid-Versicherung zu den beiden Referenzstrategien Risikoakzeptanz und Redundanz vornimmt, wurde eine Auswahl an Metriken vorgestellt, die über mehrere Aggregationsstufen einen Gesamtindex zur Messung der Gesamtperformanz bereitstellen [SREC<sup>+</sup>06] [SEVC<sup>+</sup>07].

Die Evaluation der Grid-Versicherung stand im Mittelpunkt von Kapitel 7. Ausgehend von einer detaillierten Besprechung des Simulationsszenarios wurden mithilfe von Zeitserien die Grid-Versicherung und die Versicherungsnachfrage untersucht. Im Vergleich zu den beiden Referenzmodellen Risikoakzeptanz und Risikodiversifikation durch Redundanz kann die Grid-Versicherung dynamisch auf die Risiken der untersuchten Grid-Szenarien reagieren und so zu einem erhöhten Nutzen der Ressourcenkonsumenten beitragen. Die Evaluation der Metrikpyramide zeigte, dass die Strategie der Risikoakzeptanz und die Strategie der Grid-Versicherung bei niedrigem Risiko ähnliche Ergebnisse erreichten, während bei hohem Risiko die Strategien Redundanz und Grid-Versicherung vergleichbar sind.

## 8.2 Implikationen für die Forschung

Die Ergebnisse der vorliegenden Arbeit unterliegen Restriktionen, die sich insbesondere aus dem methodischen Vorgehen der Computersimulation ergeben. Da das Simulationsmodell eine abstrakte Abbildung eines Grid-Systems darstellt, bedarf es weiterer Untersuchungen an einem Prototypen oder produktivem Grid-System, um die Ergebnisse zu validieren. Diese Vorgehensweise bietet Ansatzpunkte für die weitere wissenschaftliche Auseinandersetzung mit dem Risikomanagement und speziell dem Risikotransfer auf eine Versicherung im Kontext von Grid- und

Cloud-Computing-Systemen. In den folgenden Absätzen werden offene Fragen der Simulationsstudie aufgeworfen und mögliche Erweiterungen diskutiert.

**Analyse der Grid-Risiken** Die Analyse der Risiken von Grid- und Cloud-Computing-Systemen ist eine wichtige Voraussetzung für die Bewertung der Risiken durch eine Grid-Versicherung. Die technischen Risiken dieser Systeme sind bisher nur in vereinzelt Studien betrachtet worden, die vor allem Systeme aus der Wissenschaft analysiert haben. Weitergehende Analysen der Risiken sind notwendig, um eine detaillierte Erfassung der Risiken zu erreichen und Risiken kommerzieller Anbieter von Cloud-Computing-Systemen zu ermitteln. Hierfür müssen die bestehenden Monitoring-Systeme um die detaillierte Erfassung von qualitativen und quantitativen technischen Risiken erweitert werden und Kennzahlen zur Zuverlässigkeit der Systeme den Kunden bzw. der Versicherung zur Verfügung gestellt werden. Mit einer Verbreiterung der Datenbasis über die technischen Risiken lassen sich auch die Modelle zur Bewertung der technischen Risiken verbessern wie das in dieser Arbeit verwendete Modell von Dai [DaPZ07].

Die vorliegende Arbeit konzentrierte sich auf technische Risiken, die eine Ausführung eines Grid-Service scheitern lassen. Mit dem Auftreten von Märkten, auf denen homogene Ressourcenbündel gehandelt werden, können weitere Risiken wie Preisrisiken oder moralische Risiken auftreten, die in ein zukünftiges Risikomanagement mit einbezogen werden müssen. Beispiele für erste Marktplätze für Rechenressourcen sind das europäische Forschungsprojekt SORMA<sup>1</sup> und der kommerzielle Marktplatz Zimory<sup>2</sup>.

**Entwurf von Versicherungen für Grid- und Cloud-Computing-Systeme** Der untersuchte Versicherungstarif verwendete ein multiplikatives Modell mit den zwei Tariffaktoren *Ausführungszeit* und *Anzahl an Recheneinheiten*. Dieses Modell versicherte die Ausführung eines Service auf Grid-Ressourcen. Neben Grid-Systemen, die Rechenressourcen bereitstellen, finden sich auch Grid-Systeme, die auf die Verwaltung großer Datenmengen spezialisiert sind. Neben den Ressourcen Recheneinheiten und Datenspeicher ist als dritte wichtige Ressource Netzwerkbandbreite zu nennen, die ebenfalls einem technischen Risiko unterworfen ist. Die beiden Ressourcen Datenspeicher und Bandbreite blieben in der Untersuchung dieser Arbeit aus Gründen der Komplexität unberücksichtigt. Sie sind jedoch Kandidaten für weitere, speziell angepasste Versicherungstarife. Hierfür müssen weitere Tarifmerkmale identifiziert und die Anzahl der Tarifklassen festgelegt werden.

---

<sup>1</sup><http://www.sorma-project.org/> (Letzter Abruf: 12.08.2008)

<sup>2</sup><http://www.zimory.com/> (Letzter Abruf: 12.08.2008)

Das Marginalsummenverfahren zur Berechnung der Prämien ist auf 2 Tarifmerkmale in dieser Arbeit beschränkt. Bei Aufnahme weiterer Tarifmerkmale zur genaueren Bewertung des Risikos nimmt auch der Aufwand der Berechnung der Tariffaktoren zu. Die Komplexität des Marginalsummenverfahrens kann mit  $O(n^x)$  angegeben werden, wobei  $n$  die Anzahl an Tariffaktoren und  $x$  die Anzahl an Tarifmerkmalen darstellt. Dies bedeutet, dass die Komplexität exponentiell mit der Anzahl der Tarifmerkmale zunimmt, sodass die Anzahl der Tarifmerkmale möglichst klein gehalten werden soll, um die Berechnungszeiten für die Aktualisierung der Prämien nicht zu stark zu erhöhen.

**Simulation einer voll automatisierten Grid-Versicherung** Die Simulationsstudie in dieser Arbeit zeigte, dass sich Grid-Versicherungen voll automatisiert mit Multi-Agenten-Systemen realisieren lassen. Jedoch ergaben sich durch die Wahl der Simulationsplattform technische Beschränkungen, die verhinderten, dass eine größere Anzahl von Agenten simuliert wurde. Eine Limitierung stellte die sequenzielle Abarbeitung der an die Versicherung gesandten Nachrichten dar. Dies führte dazu, dass Schadensmeldungen erst in der folgenden Periode bearbeitet wurden und somit Spätschäden entstanden. Die Implementierung der Versicherung in einem produktiven System sollte in mehrere, nebenläufige Prozesse aufgeteilt werden, wobei der Prozess zur Schadensabwicklung parallel zum Prozess der Prämieinnahme realisiert werden sollte.

**Evaluierung einer Versicherung für Grid-Systeme** Die Evaluierung der Grid-Versicherung erfolgte im Vergleich zur Risikoakzeptanz und zur Risikodiversifikation durch Redundanz. In allen untersuchten Szenarien zeigte die Grid-Versicherung teilweise deutliche Vorteile hinsichtlich der betrachteten Metriken gegenüber den Alternativstrategien, die als Referenz dienten. Jedoch sind weitere Simulationsläufe notwendig, um die Grid-Versicherung an einer konkreten Konfiguration eines Grid-Systems zu evaluieren und zu erwartende Verbesserungen im Risikomanagement daraus abzuleiten. Im Umkehrschluss sind auch weitere Untersuchungen an produktiven Grid-Systemen notwendig, die dazu verwendet werden können, das Simulationsmodell zu verbessern. Dieser iterative Prozess kann dazu dienen, Stück für Stück das Risikomanagement für Grid- und Cloud-Computing-Systeme zu verbessern.

In der Simulationsstudie nicht betrachtet wurden Strategien, in denen ein Ressourcenkonsument zwischen Risikoakzeptanz, Risikotransfer, und Risikodiversifikation individuell entscheiden und diese Strategien mischen kann. Dies könnte ebenfalls einen weiteren Ansatzpunkt für zukünftige Forschung darstellen.

## 8.3 Implikationen für die Praxis

Die Implikationen für die Praxis werden für die 2 Rollen Ressourcenkonsument und Versicherung untersucht. Das Ziel besteht darin, die in dieser Arbeit gewonnenen Ergebnisse bei der Entwicklung zukünftiger Grid- und Cloud-Computing-Systeme zu berücksichtigen.

**Verbesserung des Risikomanagements für Ressourcenkonsumenten** Ressourcenkonsumenten haben bei der Festlegung ihrer Strategie zur Behandlung der technischen Risiken einen flexiblen Ansatz, den sie in ihrer Strategie berücksichtigen können. Der Risikotransfer auf eine Grid-Versicherung bietet den Ressourcenkonsumenten die Möglichkeit, ihre Risikokosten im Vergleich zu einer Redundanzstrategie zu senken.

Im kommerziellen Umfeld finden sich aktuell vor allem Strategien zur Behandlung der technischen Risiken, die Redundanz einsetzen, um ausgefallene Grid-Ressourcen zu kompensieren. Anbieter wie *Scalr*<sup>3</sup>, *RightScale*<sup>4</sup> oder *weoceo*<sup>5</sup> bieten gegen Zahlung einer Grundgebühr Ressourcenkonsumenten einen Service an, der ausgefallene Amazon EC2 Ressourcenbündel automatisiert durch neue Amazon EC2 Ressourcenbündel ersetzt mit dem Ziel, die damit verbundene Unterbrechung der Serviceausführung möglichst kurz zu halten. Die neuen Ressourcenbündel werden dabei dem Ressourcenkonsumenten im vollen Umfang in Rechnung gestellt.

Ein Vergleich der kommerziellen Anbieter mit dem Versicherungsansatz dieser Arbeit zeigt, dass bei den Angeboten der kommerziellen Anbieter kein Risikotransfer stattfindet. Die kommerziellen Anbieter von Risikomanagementlösungen übernehmen das Risiko nicht. Sie garantieren weder eine Lieferung von Ersatzressourcen noch eine monetäre Kompensation des Schadens. Hier bietet sich der Einsatz einer Versicherungslösung an, die dem Konsumenten die Möglichkeit gibt, die Risiken zu transferieren.

**Entwicklung von neuen Versicherungstarifen durch die Versicherungswirtschaft** Die Versicherungswirtschaft hat die Möglichkeit, neue Versicherungstarife im Bereich von Grid- und Cloud-Computing-Systemen anzubieten und damit ihr Angebot über klassische Elektronikversicherungen hinaus zu erweitern. Eine Voraussetzung ist dabei eine voll automatisierte Umsetzung der Schadensabwicklung, in Analogie zu der von IBM skizzierten Vision der Versicherungswirtschaft für das Jahr 2020 [Bisk06]. Die Entwicklung und Einführung von Versicherungstarifen für

---

<sup>3</sup><http://www.scalr.net> (Letzter Abruf: 12.08.2008)

<sup>4</sup><http://www.rightscale.com/> (Letzter Abruf: 12.08.2008)

<sup>5</sup><http://weoceo.weogeo.com/> (Letzter Abruf: 12.08.2008)

Grid- und Cloud-Computing-Systeme erfordert domänenspezifisches Wissen, sodass es sich für Versicherungen anbietet, in Zusammenarbeit mit Ressourcenanbietern neue Versicherungstarife zu entwickeln. In einem interdisziplinären Team erfolgen die Implementierung und der Betrieb durch den Ressourcenanbieter, während die Ausgestaltung des Tarifs das Versicherungsunternehmen übernimmt.

Die in dieser Arbeit vorgestellten Ergebnisse bieten sowohl für die Forschung als auch für die Praxis Ansatzpunkte zur weiteren Betrachtung bzw. für den Aufbau neuer, kostenpflichtiger Zusatzservices, die ein effizientes Risikomanagement der technischen Grid-Risiken unterstützen. Die Arbeit zeigt, dass Grid-Versicherungen einen Beitrag zur Verbesserung der Risikobehandlung in Grid- und Cloud-Computing-Systemen leisten können und schafft damit die Grundlage für zukünftige Anwendungsmöglichkeiten in der Praxis.

# Anhang



# Anhang A

## Implementierung der Grid-Versicherung

### A.1 Die Klasse *InsuranceAgent*

```
package org.catnets.optorsim.risk;

import java.util.HashMap;
import java.util.concurrent.ConcurrentLinkedQueue;

import org.apache.log4j.Logger;
import org.catnets.optorsim.infrastructure.AlnContainer;
import org.catnets.optorsim.infrastructure.AlnSite;
import org.catnets.optorsim.negotiations.Negotiator;
import org.catnets.optorsim.negotiations.P2P;
import org.catnets.optorsim.negotiations.messages.InsuranceClaimMessage;
import org.catnets.optorsim.negotiations.messages.InsuranceNotificationMessage;
import org.catnets.optorsim.negotiations.messages.InsurancePayMessage;
import org.catnets.optorsim.negotiations.messages.InsuranceRequestMessage;
import org.catnets.optorsim.negotiations.messages.Message;
import org.catnets.optorsim.time.GridTime;
import org.catnets.optorsim.time.GridTimeFactory;

public class InsuranceAgent extends Thread implements Negotiator {

    //initialise logger
```



```
private static Logger log =
Logger.getLogger(InsuranceAgent.class);

// the input box of messages waiting to be read
private ConcurrentLinkedQueue<Message> _messageQueue =
new ConcurrentLinkedQueue<Message>();

// made false when shutting down BSB
private boolean _iAmAlive;

// the local p2p mediator to communicate messages
private P2P _p2pMediator;

// Instance of GridTime
private GridTime _time = GridTimeFactory.getGridTime();

//simulator parameter
//private OptorSimParameters _optorSimParameters =
// OptorSimParameters.getInstance();

//insurance model
private InsuranceModelResource model = null;

//insurance period
private long period = 4000;

//counts the periods
private int periodCounter = 0;

public InsuranceAgent(int id, P2P p2p) {
super(GridTimeFactory.getThreadGroup(), "IA" + id + "@"
+ "CentralInsuranceSite");
_p2pMediator = p2p;
_iAmAlive = true;
}
```

```
//log.debug("Successful init of insurance");

//create InsuranceModel
model = new InsuranceModelResource();

//set insurance period
this.period = model.getPeriod();
}

/**
 * run method
 * periodically checks the input box queue and processes the messages
 */
public void run() {
while (!_iAmAlive) {

//wait for incoming messages
standBy();

//check if its time to adapt premium model
//call insurance model to update
if (_time.getRunningTimeMillis() > periodCounter*period) {
//log.debug("update tariff factors");
//update tariff factors
this.model.updateTariffFactors();
//update safety loading
this.model.updateSafetyLoading();
//update period counter
periodCounter++;
}

// process messages
Message msg = _messageQueue.poll();
```

```
if (msg instanceof InsuranceRequestMessage) {
    this.handleRequestMessage(msg);
} else if (msg instanceof InsurancePayMessage) {
    this.handlePayMessage(msg);
} else if (msg instanceof InsuranceClaimMessage) {
    this.handleClaimMessage(msg);
}

} //end while
} //end run

/**
 * handle insurance request
 * @param msg
 */
private void handleRequestMessage(Message msg) {
    HashMap<String,Double> iData =
        new HashMap<String,Double>();
    iData.put(InsuranceConstants.CONSUMED_RESOURCES,
        (Double)((HashMap)msg.getContent()).get(
            InsuranceConstants.CONSUMED_RESOURCES));
    iData.put(InsuranceConstants.EXECUTION_TIME,
        (Double)((HashMap)msg.getContent()).get(
            InsuranceConstants.EXECUTION_TIME));

    //call insurance model
    double premium = this.model.getPremium(
        msg.getSender(), iData);

    //send premium value to requester
    InsuranceNotificationMessage retMsg =
        new InsuranceNotificationMessage(
            msg.getMyNegotiation(), this,
            msg.getSender(), null, premium, null);
```

```
_p2pMediator.acceptMessage(retMsg);

} //end handleRequestMessage

/**
 * handle payment
 * @param msg
 */
private void handlePayMessage(Message msg) {
//forward premium payment to the model
this.model.setPremium(msg.getSender(),
(HashMap)msg.getContent(),
                msg.getPrice(), periodCounter);
} //end handlePayMessage

/**
 * handle insurance claim
 * @param msg
 */
private void handleClaimMessage(Message msg) {
//get compensation for claim
double compensation =
                this.model.getCompensationForClaim(
                msg.getSender(), periodCounter);

//send message with compensation back to insured agent
InsuranceNotificationMessage retMsg =
                new InsuranceNotificationMessage(
msg.getMyNegotiation(), this,
                msg.getSender(), null, compensation, null);
_p2pMediator.acceptMessage(retMsg);
} //end handleClaimMessage

/**
 * The InsuranceAgent waits until it receives a message
```

```
    */
private void standby() {
    // If there are messages to process, return immediately.
    synchronized(_messageQueue) {
        if (!_messageQueue.isEmpty()) {
            return;
        }
    }
    _time.gtWait(this);
}

/**
 * This method shuts down the agent. The method is called at the end of the
 * simulation by the simulation control.
 *
 * It can be used to write metrics at the end of the simulation.
 */
public void shutDownIA() {
    //System.out.println("Vers: Shutdown insurance agent");
    _iAmAlive = false;
    _time.gtNotify(this);
}

/**
 * add message to the input box queue
 * @param msg
 */
public synchronized void addMessage(Message msg) {
    // add message to message queue
    synchronized(_messageQueue) {
        _messageQueue.add(msg);
    }
    // notify agent thread
    _time.gtNotify(this);
}
```

```
/**
 * new toString method
 */
public String toString() {
return getName();
}

/**
 * method from the interface
 */
public AlnSite getSite() {
return AlnContainer.getInstance().getIASite();
} //end getSite
} //end class
```

## A.2 Die Klasse *InsuranceModelResource*

```
package org.catnets.optorsim.risk;

import java.io.FileInputStream;
import java.util.HashMap;
import java.util.Properties;
import java.util.Vector;

import org.apache.log4j.Logger;
import org.catnets.optorsim.OptorSimParameters;
import org.catnets.optorsim.negotiations.Negotiator;
import org.catnets.optorsim.time.GridTime;
import org.catnets.optorsim.time.GridTimeFactory;
import org.catnets.optorsim.utils.MetricsLogger;

/**
 * Insurance model of the resources with 2 tariff factors:
```

```
* - consumed resources
* - execution time
*/
public class InsuranceModelResource {

//initialise logger
private static Logger log =
Logger.getLogger(InsuranceModelResource.class);

//aggregated claims per risk classification
double[][] claim = new double[5][5];

//number of insured agents per risk classification
double[][] insured = new double[5][5];

//tariff factors
//the key is a concatenated String of tariff factor id and its shape
//the value is the list of indices for the factors array
HashMap<String,Vector<Integer>> tariffFactors =
new HashMap<String,Vector<Integer>>();

//the factors array stores the tariff factors
//an array of primitive data type double speeds up the computation step
double[][] factors = new double[2][4];

//index of tariff factors and their intervals
//which build a 2-dimensional index for access to the matrix
//this index is needed to access the tariff factor map for the
//matrix indices
HashMap<String,Vector<Double>> index =
new HashMap<String,Vector<Double>>();

//agent list
HashMap<Negotiator,HashMap<String,Object>> agents =
```

```
new HashMap<Negotiator,HashMap<String,Object>>();

// metrics logger
private static MetricsLogger logger = MetricsLogger.instance();

// Instance of GridTime
protected GridTime _time = GridTimeFactory.getGridTime();

//total premium
private double totalPremium = 400.0;

//safetyLoading of the premium [1.0,2.0]
private double safetyLoading = 1.8;

//insurance period in [milliseconds]
private long period = 4000;

private long totalRequests = 0;

public long getPeriod() {
return period;
}

/**
 * constructor of the insurance model
 */
public InsuranceModelResource() {
buildDataStructure();
initDataStructure();
} //end constructor

/**
 * build data structure for insurance
 * the current implementation realises 2 tariff factors
 */
```



```
private void buildDataStructure() {

//get configuration parameters
OptorSimParameters _param = OptorSimParameters.getInstance();

Properties prop = new Properties();
    try {
        prop.load(new FileInputStream(
            _param.getInsuranceInitFile()));
    } catch (Exception e) {
        e.printStackTrace();
        System.exit(1);
    }

//definition of the tariff classes
//only the upper limit of the interval is defined
Vector<Double> index1 = new Vector<Double>();
index1.add(Double.parseDouble(
    prop.getProperty(InsuranceConstants.
        CONSUMED_RESOURCES+".interval.1")));
index1.add(Double.parseDouble(
    prop.getProperty(InsuranceConstants.
        CONSUMED_RESOURCES+".interval.2")));
index1.add(Double.parseDouble(
    prop.getProperty(InsuranceConstants.
        CONSUMED_RESOURCES+".interval.3")));
index1.add(Double.parseDouble(
    prop.getProperty(InsuranceConstants.
        CONSUMED_RESOURCES+".interval.4")));

//put HashMap into index structure
index.put(InsuranceConstants.CONSUMED_RESOURCES, index1);

//three intervals for network failure
```

```
Vector<Double> index2 = new Vector<Double>();
index2.add(Double.parseDouble(
prop.getProperty(InsuranceConstants.
    EXECUTION_TIME+".interval.1")));
index2.add(Double.parseDouble(
prop.getProperty(InsuranceConstants.
    EXECUTION_TIME+".interval.2")));
index2.add(Double.parseDouble(
prop.getProperty(InsuranceConstants.
    EXECUTION_TIME+".interval.3")));
index2.add(Double.parseDouble(
prop.getProperty(InsuranceConstants.
    EXECUTION_TIME+".interval.4")));

//put HashMap into index structure
index.put(InsuranceConstants.EXECUTION_TIME, index2);

//initialise safety loading
safetyLoading = Double.parseDouble(
    prop.getProperty("safety.loading"));

//initialise budget
totalPremium = Double.parseDouble(
    prop.getProperty("budget"));

//read insurance period
period = Long.parseLong(prop.getProperty("period"));

} //end buildDataStructure

/**
 *
 */
private void initDataStructure() {
//map tariff factor names to array index
```

```
Vector<Integer> v = new Vector<Integer>();
v.add(new Integer(0));
v.add(new Integer(0));
tariffFactors.put(InsuranceConstants.
    CONSUMED_RESOURCES+
index.get(InsuranceConstants.
    CONSUMED_RESOURCES).get(0),v);
v = new Vector<Integer>();
v.add(new Integer(0));
v.add(new Integer(1));
tariffFactors.put(InsuranceConstants.CONSUMED_RESOURCES+
index.get(InsuranceConstants.CONSUMED_RESOURCES).get(1),v);
v = new Vector<Integer>();
v.add(new Integer(0));
v.add(new Integer(2));
tariffFactors.put(InsuranceConstants.CONSUMED_RESOURCES+
index.get(InsuranceConstants.CONSUMED_RESOURCES).get(2),v);
v = new Vector<Integer>();
v.add(new Integer(0));
v.add(new Integer(3));
tariffFactors.put(InsuranceConstants.CONSUMED_RESOURCES+
index.get(InsuranceConstants.CONSUMED_RESOURCES).get(3),v);
v = new Vector<Integer>();
v.add(new Integer(1));
v.add(new Integer(0));
tariffFactors.put(InsuranceConstants.EXECUTION_TIME+
index.get(InsuranceConstants.EXECUTION_TIME).get(0),v);
v = new Vector<Integer>();
v.add(new Integer(1));
v.add(new Integer(1));
tariffFactors.put(InsuranceConstants.EXECUTION_TIME+
index.get(InsuranceConstants.EXECUTION_TIME).get(1),v);
v = new Vector<Integer>();
v.add(new Integer(1));
v.add(new Integer(2));
```

```
tariffFactors.put(InsuranceConstants.EXECUTION_TIME+
index.get(InsuranceConstants.EXECUTION_TIME).get(2),v);
v = new Vector<Integer>();
v.add(new Integer(1));
v.add(new Integer(2));
tariffFactors.put(InsuranceConstants.EXECUTION_TIME+
index.get(InsuranceConstants.EXECUTION_TIME).get(3),v);

factors[0][0] = 0.5;
factors[0][1] = 0.5;
factors[0][2] = 0.5;
factors[0][3] = 0.5;
factors[1][0] = 0.5;
factors[1][1] = 0.5;
factors[1][2] = 0.5;
factors[1][3] = 0.5;

//start with an initial population of 10 insured agents
for (int i=0;i<insured.length;i++) {
for (int j=0;j<insured.length;j++){
insured[i][j]=1;
}
}

} //end initDataStructure

/**
 * compute premium and return premium value
 * for the given data
 * @return
 */
public double getPremium(Negotiator agent, HashMap<String,Double> data) {
Vector<String> cellIndex = new Vector<String>();
//get cell index data to access factor array
```

```
cellIndex = this.getCellIndexString(data);

//counts the total number of agents with AND
//without insurance
totalRequests++;

//access claim array for base premium
//for the current implementation of a 4x4 array the cell [3][3]
//double basePremium = claim[3][3]/insured[3][3];
//the current base premium is assumed to be 1.0
//this will simplify the tariff factor computation
double basePremium = 1.0;

//log.debug("getPremium: basePremium == "+basePremium);

//read the tariff factors and multiply them to the base premium
double premium = basePremium;
Vector<Integer> v = null;
for (String key:cellIndex){
v=tariffFactors.get(key);
premium = premium * factors[v.get(0)][v.get(1)];
}

//log.debug("getPremium: premium == "+premium);

String value = "" + this.totalRequests + " " +
(this.totalRequests - insured[4][4]) + " " +
premium + " " + (premium*safetyLoading)+ " " +
safetyLoading;

logger.logMetric(_time.getRunningTimeMillis(),
"InsuranceModel",
"insuranceTotalRequests",value);

//return premium
```

```
return premium * safetyLoading;
} //end getPremium

/**
 * receive premium payment and update data structures
 * @param agent the insured agent
 * @param data the data which describes the risk
 * @param premium the premium the agent pays and received for his request
 */
public void setPremium(Negotiator agent,
HashMap<String,Double> data, double claimValue, int period) {

Vector<Integer> cellIndex = new Vector<Integer>();

//log.debug("setPremium: agent == "+agent);

//store data for agent
HashMap<String,Object> agentData = new HashMap<String,Object>();
agentData.put("data", data);
agentData.put("premium", data.get("premium"));
agentData.put("period", period);

totalPremium = totalPremium + data.get("premium");
//*claim[3][3]/insured[3][3];
this.logTotalPremium();

HashMap<String,Double> iData = new HashMap<String,Double>();
iData.put(InsuranceConstants.CONSUMED_RESOURCES,
(Double)(data).get(InsuranceConstants.CONSUMED_RESOURCES));
iData.put(InsuranceConstants.EXECUTION_TIME,
(Double)(data).get(InsuranceConstants.EXECUTION_TIME));

cellIndex = this.getCellIndexInteger(iData);

//log.debug("iData == " + iData);
```

```
//System.out.println("cellIndex[i] == "+cellIndex.get(0));
//System.out.println("cellIndex[j] == "+cellIndex.get(1));

//access cell and increase insured agents value by one
int i = cellIndex.get(0);
int j = cellIndex.get(1);
insured[i][j]++;

agentData.put("cellIndex", cellIndex);
agentData.put("claimValue", claimValue);
this.agents.put(agent, agentData);

//log.debug("setPremium: current paying agents == "+agentData);
} //end setPremium

/**
 * return the compensation for claim
 * @param agent the insured agent
 * @return claim value
 */
public double getCompensationForClaim (Negotiator agent, int period) {

//log.debug("getCompensationForClaim: agent == "+agent);

//get data for agent
HashMap<String, Object> agentData = agents.get(agent);
if (agentData == null) {
//if agentData equals null we should return 0.0, because the agent
//didn't pay his premium
//log.debug("getCompensationForClaim: value(agent did not pay premium) == "+0.0);
return 0.0;
}
double claimValue = ((Double)agentData.get("claimValue")).doubleValue();
Vector<Integer> cellIndex = (Vector<Integer>)agentData.get("cellIndex");
```

```
//access cell and increase insured agents value by one
int i = cellIndex.get(0);
int j = cellIndex.get(1);
claim[i][j] = claim[i][j] + claimValue;

//log.debug("getCompensationForClaim: value(claimValue) == "+claimValue);

int premiumPeriod = ((Integer)agentData.get("period")).intValue();
int claimPeriod = period;
int difference = claimPeriod-premiumPeriod;
int isDifference = 0;
if (difference!=0) {
isDifference = 1;
}

//write metric
logger.logMetric(_time.getRunningTimeMillis(), "InsuranceModel",
"insuranceSpaetschaeden",""+premiumPeriod+" "+claimPeriod+" "+
difference+" "+isDifference);

//return claim value
return claimValue;
} //end getCompensationForClaim

/**
 * start computation of new tariff factors
 */
public void updateTariffFactors () {

//new factors
double[][] newFactors = new double[2][4];

//temporary array of insured*tariffFactor array
double[][] insuredTariffSum = new double[2][4];
```



```
//compute row sum -> x_i
for (int i=0;i<claim.length-1;i++) {
//reset sums
insured[i][4] = 0;
claim[i][4] = 0.0;
insuredTariffSum[0][i] = 0.0;
for (int j=0;j<claim.length-1;j++) {
//claim row sum
claim[i][4] = claim[i][4] + claim[i][j];
//insured row sum
insured[i][4] = insured[i][4] + insured[i][j];
//sum for computation of x_i
insuredTariffSum[0][i]=
insuredTariffSum[0][i]+insured[i][j]*factors[0][j];
}
}

//compute column sum
for (int j=0;j<claim.length-1;j++) {
//reset sums
claim[4][j] = 0.0;
insured[4][j] = 0;
insuredTariffSum[1][j]= 0.0;
for (int i=0;i<claim.length-1;i++) {
//claim column sum
claim[4][j] = claim[4][j] + claim[i][j];
//insured row sum
insured[4][j] = insured[4][j] + insured[i][j];
//sum for computation of y_j
insuredTariffSum[1][j]=
insuredTariffSum[1][j]+insured[i][j]*factors[1][i];
}
}
```

```
//reset sum
claim[4][4] = 0.0;
insured[4][4] = 0;
//sum row and column sum
double checkSumClaim = claim[4][4];
double checkSumInsured = insured[4][4];
for (int i=0;i<claim.length-1;i++){
claim[4][4] = claim[4][4]+claim[4][i];
insured[4][4] = insured[4][4] + insured[4][i];
//for checkSums
checkSumClaim = checkSumClaim + claim[i][4];
checkSumInsured = checkSumInsured + insured[i][4];
}

if (claim[4][4]==checkSumClaim && insured[4][4]==checkSumInsured){
log.debug("sum computation successful");
} else {
log.debug("sum computation failed");
}

for (int i=0;i<2;i++) {
for (int j=0;j<4;j++){
System.out.print(" "+insuredTariffSum[i][j]);
}
System.out.println("");
}

//failure factor
for (int i=0;i<4;i++){
if (insuredTariffSum[0][i]==0.0){
newFactors[0][i]= factors[1][i];
} else {
newFactors[0][i]=
```

```
claim[i][4]/insuredTariffSum[0][i];
}
}

//share factor
for (int i=0;i<4;i++){
if (insuredTariffSum[1][i]==0.0) {
newFactors[1][i]= factors[1][i];
} else {
newFactors[1][i]=
claim[4][i]/insuredTariffSum[1][i];
}
}

//set adapted factors
for (int i=0;i<2;i++) {
for (int j=0;j<4;j++){
if (newFactors[i][j]!=0) {
factors[i][j] = newFactors[i][j];
}
}
}

this.logFactorMatrix();
this.logClaimMatrix();
this.logInsuredMatrix();

this.printClaimMatrix();
this.printInsuredMatrix();
this.printFactorMatrix();
} //end updatePremium

/**
 *
```

```
*/
public void updateSafetyLoading(){

double capital = totalPremium - claim[4][4];

//write metric "capital"
logger.logMetric(_time.getRunningTimeMillis(), "InsuranceModel",
"insuranceCapital", ""+capital);

//write metric "safety loading"
logger.logMetric(_time.getRunningTimeMillis(), "InsuranceModel",
"insuranceSafetyLoading", ""+this.safetyLoading);

if (capital < 100) {
this.safetyLoading = 2.0;
} else if (capital < 300) {
this.safetyLoading = 1.9;
} else if (capital < 400) {
this.safetyLoading = 1.7;
} else if (capital < 500) {
this.safetyLoading = 1.3;
} else {
//capital >= 500
this.safetyLoading = 1.0;
}

}

/**
 * get cell index for received insurance data
 * @param data
 * @return
 */
private Vector<String> getCellIndexString(HashMap<String,Double> data){
Vector<Double> riskIntervals = null;
```

```
double value = 0.0;
double tmpVal = 0.0;
Vector<String> cellIndex = new Vector<String>();

//goal of the implementation should be the concatenation of the index for
//access of the tariff factors list

//select the tariff factor
for (String key:data.keySet()) {
log.debug("key == "+key);
riskIntervals = index.get(key);
value = data.get(key);
log.debug("value == "+value);
//select a suitable interval
boolean intervalFound = false;
for (Double key2: riskIntervals){
//convert key2 to double value for comparison
tmpVal = key2.doubleValue();
log.debug("tmpVal == "+tmpVal);
//check if current risk value fits to an
//predefined interval
if (value<=tmpVal && intervalFound==false){
//we have found an index which fits in the smallest
//interval
cellIndex.add(key+key2);
intervalFound=true;
}
}
}

log.debug("getCellIndexString: cellIndex == "+cellIndex);

return cellIndex;
}
```

```
/**
 * get cell index for received insurance data
 * @param data
 * @return
 */
private Vector<Integer> getCellIndexInteger(HashMap<String,Double> data){
Vector<Double> riskIntervals = null;
double value = 0.0;
double tmpVal = 0.0;
Vector<Integer> cellIndex = new Vector<Integer>();

//select the tariff factor
for (String key:data.keySet()) {
riskIntervals = index.get(key);
value = data.get(key);
//select a suitable interval
boolean intervalFound = false;
for (int i=0;i<riskIntervals.size();i++ ){
//convert key2 to double value for comparison
tmpVal = riskIntervals.get(i);
//check if current risk value fits to an
//predefined interval
if (value<=tmpVal && intervalFound==false){
//we have found an index which
//fits in the smallest interval
cellIndex.add(i);
intervalFound=true;
}
}
}

return cellIndex;
}

} //end class
```



# Anhang B

## Konfiguration des Simulationsszenarios

### B.1 Simulatorkonfiguration

```
#
# This file contains all the parameters required for OptorSim
# using the Properties class to store this information.
#
# Aln configuration files

aln.topology.file = examples/_Cloud_200N_50R_200BS_200CS/topology.conf
aln.bs.file = examples/_Cloud_200N_50R_200BS_200CS/bs_EC2_simple.conf
aln.arb.file = examples/_Cloud_200N_50R_200BS_200CS/arb_EC2_simple.conf
cs.configuration.file = examples/_Cloud_200N_50R_200BS_200CS/cs.conf

# market initialisation
market.decentral.file = examples/_Cloud_200N_50R_200BS_200CS/
    market_decentral_EC2_simple.properties

# insurance config file
insurance.configuration.file = examples/_Cloud_200N_50R_200BS_200CS/
    insurance.conf

#Number of complex services to be submitted
number.complexservices = 20000
```



```
#
# The categories of users available are:
# (1) Simple - wait cs delay between submitting CSs
# (2) Random - wait uniform random time between 0 and 2 * cs delay
users = 1
#
# The choice of the policy for the ComplexServiceDispatcher is:
# (1) random
# (2) queue length
#
policy = 1
#
# The cs delay is the interval in ms between the ComplexServiceDispatcher
# submitting each CS.
#
cs.delay = 1000
#
# The choice of access pattern generators is:
# (1) SequentialAccessGenerator - BSs are accessed in order.
# (2) RandomAccessGenerator - BSs are accessed using a flat random
#     distribution.
# (3) RandomWalkUnitaryAccessGenerator - BSs are accessed using a
#     unitary random walk.
# (4) RandomWalkGaussianAccessGenerator - BSs are accessed using a
#     Gaussian random walk.
# (5) RandomZipfAccessGenerator - BSs are accessed using a
#     Zipf distribution
#
access.pattern.generator = 1
# Shape parameter for Zipf-like distribution > 0
shape = 0.85
#
# The random seed for deciding which cs are chosen can be random or
# fixed.
random.seed = no
```



```
# actually the number of sites contacted
cfp_ann.hop.count = 2

# actually the number of sites contacted
learning.hop.count = 3

# discovery timeout
discovery.timeout = 500

# negotiation timeout
#timeout = 10000
timeout = 20000
timeout.reduction.factor = 0.5

# size of messages (in Kbytes) - size = 0
# implies instantaneous message delivery
message.size = 2
#
# Outputs negotiation information to negotiation.log. Can slow the
# simulation down a little bit.
#
negotiation.log = yes
#
#####
# Time Model #
#####
#
# use advanced grid time (yes) or not (no)
#
time.advance = no
#
#####
# Output path for metrics log file #
#####
```

```
metrics.path = ./log/
```

## B.2 Konfiguration der Versicherung

```
#  
# Insurance properties  
#  
  
# tariff classes for tariff attribute "consumedResources"  
# note: intervals must include all possible values  
# [1,3]  
consumedResources.interval.1 = 3.9  
# ]3,4]  
consumedResources.interval.2 = 4.1  
# [5,7]  
consumedResources.interval.3 = 7.1  
# [8,20]  
consumedResources.interval.4 = 20.1  
  
# tariff classes for tariff attribute "executionTime"  
# note: intervals must include all possible values  
executionTime.interval.1 = 210  
executionTime.interval.2 = 310  
executionTime.interval.3 = 510  
executionTime.interval.4 = 1010  
  
# initial insurance budget  
budget = 400  
  
# safety loading  
safety.loading = 1.0  
  
# insurance period [milliseconds]
```

```
period = 8000
```

## B.3 Marktkonfiguration

```
#####
# Setup of the EC2 service and resource #
# market configuration                  #
#####

#time slot size in [ms]
#time.slot = 100

#Internet Data Transfer
#$.100 per GB - all data transfer in
#
#$.170 per GB - first 10 TB / month data transfer out
#$.130 per GB - next 40 TB / month data transfer out
#$.110 per GB - next 100 TB / month data transfer out
#$.100 per GB - data transfer out / month over 150 TB

price.data.transfer.in = 1
price.data.transfer.out = 1.7

#define maximum execution time
#important for correct risk computation
execution.time.max = 1000

#####
# Basic Service 1 (bs1) - Standard Web Application #
# Main characteristic:                               #
# low requirements for CPU and storage               #
# Requests: Standard EC2 instance                   #
#####
```

```
bs1.seller.minPrice = 20
bs1.seller.maxPrice = 40
bs1.buyer.minPrice = 20
bs1.buyer.maxPrice = 40
bs1.hard.lower.limit = 20
bs1.hard.upper.limit = 40
bs1.resource.itemids = r2_0

bs1.execution.time = 1000

#bs1.cloud.time.slots = 10
#bs1.cloud.parallel.instances = 1
bs1.data.transfer.in = 1
bs1.data.transfer.out = 10

#####
# Basic Service 2 (bs2) - Financial Application 1 #
# Main characteristic: #
# medium CPU usage #
# Requests: High-CPU medium instance #
#####

bs2.seller.minPrice = 20
bs2.seller.maxPrice = 40
bs2.buyer.minPrice = 20
bs2.buyer.maxPrice = 40
bs2.hard.lower.limit = 20
bs2.hard.upper.limit = 40
bs2.resource.itemids = r2_3

bs2.execution.time = 500

#bs2.cloud.time.slots = 5
#bs2.cloud.parallel.instances = 10
```

```
bs2.data.transfer.in = 100
bs2.data.transfer.out = 300

#####
# Basic Service 3 (bs3) - Video Conversion #
# Main characteristic:                    #
# medium memory and storage, high cpu    #
# Requests: Standard Instance Large     #
#####

bs3.seller.minPrice = 20
bs3.seller.maxPrice = 40
bs3.buyer.minPrice = 20
bs3.buyer.maxPrice = 40
bs3.hard.lower.limit = 20
bs3.hard.upper.limit = 40
bs3.resource.itemids = r2_1

bs3.execution.time = 200

#bs3.cloud.time.slots = 2
#bs3.cloud.parallel.instances = 10
bs3.data.transfer.in = 500
bs3.data.transfer.out = 700

#####
# Basic Service 4 (bs4) - Data Mining    #
# Main characteristic:                    #
# high memory and storage                 #
# Requests: Standard Instance Extra Large #
#####

bs4.seller.minPrice = 20
bs4.seller.maxPrice = 40
```

```
bs4.buyer.minPrice = 20
bs4.buyer.maxPrice = 40
bs4.hard.lower.limit = 20
bs4.hard.upper.limit = 40
bs4.resource.itemids = r2_2

bs4.execution.time = 500

#bs4.cloud.time.slots = 5
#bs4.cloud.parallel.instances = 3
bs4.data.transfer.in = 500
bs4.data.transfer.out = 100

#####
# Basic Service 5 (bs5) - Financial Application 2 #
# Main characteristic:                               #
# high CPU usage                                    #
# Requests: High-CPU large instance                 #
#####

bs5.seller.minPrice = 20
bs5.seller.maxPrice = 40
bs5.buyer.minPrice = 20
bs5.buyer.maxPrice = 40
bs5.hard.lower.limit = 20
bs5.hard.upper.limit = 40
bs5.resource.itemids = r2_4

bs5.execution.time = 300

#bs5.cloud.time.slots = 3
#bs5.cloud.parallel.instances = 20
bs5.data.transfer.in = 600
bs5.data.transfer.out = 1000
```



```
#####  
# Resource bundles #  
#####  
# r1 == GB memory  
# r2 == number of compute instances  
# r3 == GB storage  
# r4 == 32 bit platform  
# r5 == 64 bit platform
```

```
#####  
# Standard Instance Small #  
# 1,7 GB of memory #  
# 1 EC2 Compute Unit #  
# 160 GB of instance storage #  
# 32 bit platform #  
# $0,10 per instance-hour #  
#####
```

```
r2_0.seller.minPrice =1  
r2_0.seller.maxPrice =2  
r2_0.buyer.minPrice =1  
r2_0.buyer.maxPrice =2  
r2_0.hard.lower.limit =1  
r2_0.hard.upper.limit =2  
r2_0.baseunit.r2 = 1  
r2_0.resourceids = r2
```

```
#####  
# Standard Instance Large #  
# 7,5 GB of memory #  
# 4 EC2 Compute Unit #  
# 860 GB of instance storage #  
# 64 bit platform #  
# $0,40 per instance-hour #
```

```
#####
```

```
r2_1.seller.minPrice =4
r2_1.seller.maxPrice =8
r2_1.buyer.minPrice =4
r2_1.buyer.maxPrice =8
r2_1.hard.lower.limit =4
r2_1.hard.upper.limit =8
r2_1.baseunit.r2= 4
r2_1.resourceids = r2
```

```
#####
```

```
# Standard Instance Extra Large #
# 15 GB of memory          #
# 8 EC2 Compute Unit      #
# 1690 GB of instance storage #
# 64 bit platform         #
# $0,80 per instance-hour #
#####
```

```
r2_2.seller.minPrice =8
r2_2.seller.maxPrice =16
r2_2.buyer.minPrice =8
r2_2.buyer.maxPrice =16
r2_2.hard.lower.limit =8
r2_2.hard.upper.limit =16
r2_2.baseunit.r2= 8
r2_2.resourceids = r2
```

```
#####
```

```
# High-CPU Medium Instance #
# 1.7 GB of memory        #
# 5 EC2 compute unit      #
# 350 GB of instance storage #
# 32 bit platform         #
```

```
# $0,20 per instance-hour      #
#####

r2_3.seller.minPrice =2
r2_3.seller.maxPrice =4
r2_3.buyer.minPrice =2
r2_3.buyer.maxPrice =4
r2_3.hard.lower.limit =2
r2_3.hard.upper.limit =4
r2_3.baseunit.r2= 5
r2_3.resourceids = r2

#####
# High-CPU Large Instance      #
# 7 GB of memory                #
# 20 EC2 compute unit          #
# 1690 GB of instance storage  #
# 64 bit platform              #
# $0,80 per instance-hour      #
#####

r2_4.seller.minPrice = 8
r2_4.seller.maxPrice = 16
r2_4.buyer.minPrice = 8
r2_4.buyer.maxPrice = 16
r2_4.hard.lower.limit = 8
r2_4.hard.upper.limit = 16
r2_4.baseunit.r2 = 20
r2_4.resourceids = r2

#####
# ID's of the available resource bundles #
#####

arb.itemids = r2_0 r2_1 r2_2 r2_3 r2_4
```

## B.4 Konfiguration der Ressourcenbündel

```
#
# Available resource bundles
# r1 == GB memory
# r2 == number of compute instances
# r3 == GB storage
# r4 == 32 bit platform
# r5 == 64 bit platform
# cluster configurations
# maximum number of compute instances
# typical cluster configuration see grid 5000 or d-grid
arb1 r2 20
arb2 r2 40
arb3 r2 50
arb4 r2 80
arb5 r2 100
```

## B.5 Konfiguration der Basic-Services

```
#
# Basic services
#
bs1 bs1 platinum r2 1
bs2 bs2 platinum r2 5
bs3 bs3 platinum r2 4
bs4 bs4 platinum r2 8
bs5 bs5 platinum r2 20
```

## B.6 Konfiguration der Complex-Services

```
#
```

```
# CS Table
#
# A CS name and a list of basic service needed.
#
\begin{cstable}
cs1 bs5
cs2 bs3
cs3 bs4
cs4 bs2
cs5 bs1
\end{cstable}
#
# CPS Schedule Table
# CPS site id, CS it will run
#
\begin{cspscheduletable}
0
1 cs1 cs2 cs3 cs4 cs5
2 cs1 cs2 cs3 cs4 cs5
3 cs1 cs2 cs3 cs4 cs5
4 cs1 cs2 cs3 cs4 cs5
5 cs1 cs2 cs3 cs4 cs5
6 cs1 cs2 cs3 cs4 cs5
7 cs1 cs2 cs3 cs4 cs5
8 cs1 cs2 cs3 cs4 cs5
9 cs1 cs2 cs3 cs4 cs5
10 cs1 cs2 cs3 cs4 cs5
11
12 cs1 cs2 cs3 cs4 cs5
13 cs1 cs2 cs3 cs4 cs5
14
15 cs1 cs2 cs3 cs4 cs5
16 cs1 cs2 cs3 cs4 cs5
17 cs1 cs2 cs3 cs4 cs5
18 cs1 cs2 cs3 cs4 cs5
```

19 cs1 cs2 cs3 cs4 cs5  
20 cs1 cs2 cs3 cs4 cs5  
21 cs1 cs2 cs3 cs4 cs5  
22  
23  
24 cs1 cs2 cs3 cs4 cs5  
25 cs1 cs2 cs3 cs4 cs5  
26 cs1 cs2 cs3 cs4 cs5  
27 cs1 cs2 cs3 cs4 cs5  
28 cs1 cs2 cs3 cs4 cs5  
29 cs1 cs2 cs3 cs4 cs5  
30  
31 cs1 cs2 cs3 cs4 cs5  
32 cs1 cs2 cs3 cs4 cs5  
33 cs1 cs2 cs3 cs4 cs5  
34 cs1 cs2 cs3 cs4 cs5  
35  
36 cs1 cs2 cs3 cs4 cs5  
37 cs1 cs2 cs3 cs4 cs5  
38  
39 cs1 cs2 cs3 cs4 cs5  
40 cs1 cs2 cs3 cs4 cs5  
41  
42 cs1 cs2 cs3 cs4 cs5  
43  
44  
45  
46 cs1 cs2 cs3 cs4 cs5  
47  
48  
49 cs1 cs2 cs3 cs4 cs5  
50 cs1 cs2 cs3 cs4 cs5  
51  
52  
53 cs1 cs2 cs3 cs4 cs5

54 cs1 cs2 cs3 cs4 cs5

55 cs1 cs2 cs3 cs4 cs5

56 cs1 cs2 cs3 cs4 cs5

57 cs1 cs2 cs3 cs4 cs5

58 cs1 cs2 cs3 cs4 cs5

59

60

61 cs1 cs2 cs3 cs4 cs5

62

63 cs1 cs2 cs3 cs4 cs5

64

65 cs1 cs2 cs3 cs4 cs5

66

67 cs1 cs2 cs3 cs4 cs5

68 cs1 cs2 cs3 cs4 cs5

69 cs1 cs2 cs3 cs4 cs5

70

71 cs1 cs2 cs3 cs4 cs5

72 cs1 cs2 cs3 cs4 cs5

73 cs1 cs2 cs3 cs4 cs5

74 cs1 cs2 cs3 cs4 cs5

75 cs1 cs2 cs3 cs4 cs5

76 cs1 cs2 cs3 cs4 cs5

77 cs1 cs2 cs3 cs4 cs5

78

79 cs1 cs2 cs3 cs4 cs5

80

81

82 cs1 cs2 cs3 cs4 cs5

83 cs1 cs2 cs3 cs4 cs5

84

85

86 cs1 cs2 cs3 cs4 cs5

87 cs1 cs2 cs3 cs4 cs5

88 cs1 cs2 cs3 cs4 cs5

89 cs1 cs2 cs3 cs4 cs5  
90 cs1 cs2 cs3 cs4 cs5  
91  
92 cs1 cs2 cs3 cs4 cs5  
93  
94  
95  
96 cs1 cs2 cs3 cs4 cs5  
97 cs1 cs2 cs3 cs4 cs5  
98 cs1 cs2 cs3 cs4 cs5  
99 cs1 cs2 cs3 cs4 cs5  
100 cs1 cs2 cs3 cs4 cs5  
101 cs1 cs2 cs3 cs4 cs5  
102  
103 cs1 cs2 cs3 cs4 cs5  
104 cs1 cs2 cs3 cs4 cs5  
105 cs1 cs2 cs3 cs4 cs5  
106 cs1 cs2 cs3 cs4 cs5  
107 cs1 cs2 cs3 cs4 cs5  
108  
109 cs1 cs2 cs3 cs4 cs5  
110  
111 cs1 cs2 cs3 cs4 cs5  
112 cs1 cs2 cs3 cs4 cs5  
113 cs1 cs2 cs3 cs4 cs5  
114 cs1 cs2 cs3 cs4 cs5  
115 cs1 cs2 cs3 cs4 cs5  
116  
117 cs1 cs2 cs3 cs4 cs5  
118 cs1 cs2 cs3 cs4 cs5  
119 cs1 cs2 cs3 cs4 cs5  
120 cs1 cs2 cs3 cs4 cs5  
121  
122 cs1 cs2 cs3 cs4 cs5  
123 cs1 cs2 cs3 cs4 cs5



124

125 cs1 cs2 cs3 cs4 cs5

126

127

128 cs1 cs2 cs3 cs4 cs5

129 cs1 cs2 cs3 cs4 cs5

130

131 cs1 cs2 cs3 cs4 cs5

132

133 cs1 cs2 cs3 cs4 cs5

134 cs1 cs2 cs3 cs4 cs5

135

136 cs1 cs2 cs3 cs4 cs5

137

138 cs1 cs2 cs3 cs4 cs5

139 cs1 cs2 cs3 cs4 cs5

140

141 cs1 cs2 cs3 cs4 cs5

142

143 cs1 cs2 cs3 cs4 cs5

144

145 cs1 cs2 cs3 cs4 cs5

146 cs1 cs2 cs3 cs4 cs5

147

148 cs1 cs2 cs3 cs4 cs5

149 cs1 cs2 cs3 cs4 cs5

150 cs1 cs2 cs3 cs4 cs5

151

152 cs1 cs2 cs3 cs4 cs5

153 cs1 cs2 cs3 cs4 cs5

154 cs1 cs2 cs3 cs4 cs5

155 cs1 cs2 cs3 cs4 cs5

156 cs1 cs2 cs3 cs4 cs5

157 cs1 cs2 cs3 cs4 cs5

158

159 cs1 cs2 cs3 cs4 cs5  
160 cs1 cs2 cs3 cs4 cs5  
161 cs1 cs2 cs3 cs4 cs5  
162 cs1 cs2 cs3 cs4 cs5  
163  
164  
165 cs1 cs2 cs3 cs4 cs5  
166 cs1 cs2 cs3 cs4 cs5  
167  
168 cs1 cs2 cs3 cs4 cs5  
169  
170 cs1 cs2 cs3 cs4 cs5  
171 cs1 cs2 cs3 cs4 cs5  
172  
173 cs1 cs2 cs3 cs4 cs5  
174  
175 cs1 cs2 cs3 cs4 cs5  
176  
177 cs1 cs2 cs3 cs4 cs5  
178 cs1 cs2 cs3 cs4 cs5  
179 cs1 cs2 cs3 cs4 cs5  
180 cs1 cs2 cs3 cs4 cs5  
181 cs1 cs2 cs3 cs4 cs5  
182  
183 cs1 cs2 cs3 cs4 cs5  
184  
185 cs1 cs2 cs3 cs4 cs5  
186  
187  
188 cs1 cs2 cs3 cs4 cs5  
189  
190 cs1 cs2 cs3 cs4 cs5  
191 cs1 cs2 cs3 cs4 cs5  
192 cs1 cs2 cs3 cs4 cs5  
193 cs1 cs2 cs3 cs4 cs5

```
194
195 cs1 cs2 cs3 cs4 cs5
196 cs1 cs2 cs3 cs4 cs5
197
198 cs1 cs2 cs3 cs4 cs5
199 cs1 cs2 cs3 cs4 cs5
\end{cspscheduletable}
#
# The probability each cs runs
#
\begin{csselectionprobability}
cs1 0.2
cs2 0.2
cs3 0.2
cs4 0.2
cs5 0.2
\end{csselectionprobability}
```

## B.7 Konfiguration des Lernalgorithmus

```
# send plumages
maturityThreshold = 3

# receive plumages
courterThreshold = 10

# crossover probability
crossoverProbability = 0.5

# mutation probability
mutationProbability = 0.05

# ring size
ringSize = 10
```

```
# crossOverSelectionModel
# (1) select best received plumage
crossOverSelectionModel = 1

# init float gene
gaussWidth = 0.1
min = 0.001
max = 0.999

# randomize genotype
genotype.randomize = yes
```



# Anhang C

## Evaluationsergebnisse

### C.1 Versicherungsperiode

### C.2 Metrikpyramide

Tabelle C.1: Werte der Indizes der Metrikpyramide für das Szenario mit niedrigem Risiko

Index	Redundanz	Akzeptanz	Versicherung
Social-Loss	0,0601	0,0178	0,0116
s_ODM	0,0728	0,0947	0,0392
m_ODM	0,0309	0,0531	0,0341
s_IK	0,0854	0,0924	0,0811
m_IK	0,3267	0,1236	0,1176

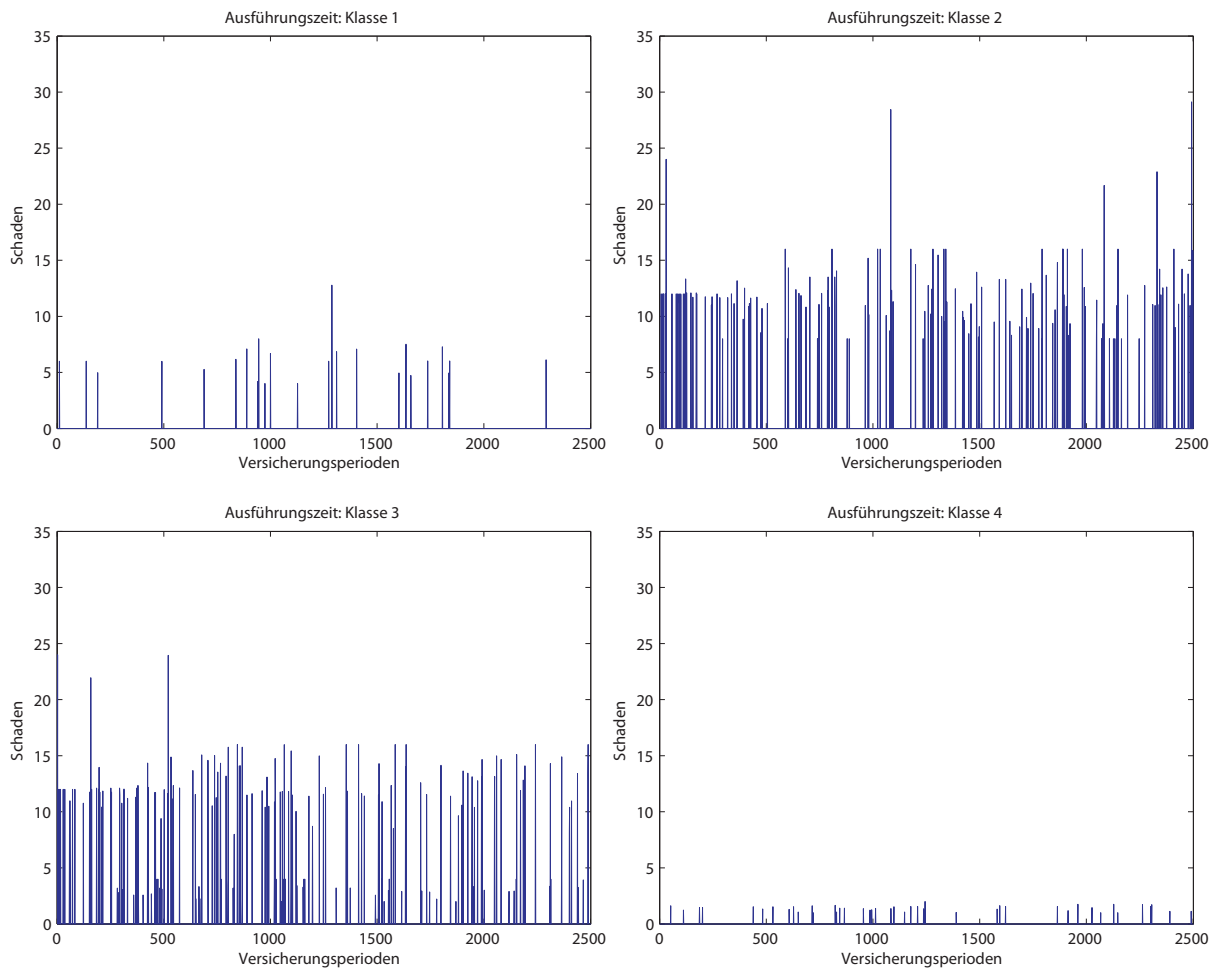


Abbildung C.1: Zeitserie der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals *Ausführungszeit* mit Risikoprofil *niedriges Risiko* und 8.000 ms Versicherungsperiode

Tabelle C.2: Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wert für das Szenario mit niedrigem Risiko

ODM-Gewichtung	IK-Gewichtung	Social-Loss		
		Redundanz	Akzeptanz	Versicherung
0,1	0,9	0,1033	0,0226	0,0186
0,3	0,7	0,0817	0,0202	0,0151
0,5	0,5	0,0601	0,0178	0,0116
0,7	0,3	0,0386	0,0154	0,0080
0,9	0,1	0,0170	0,0130	0,0045

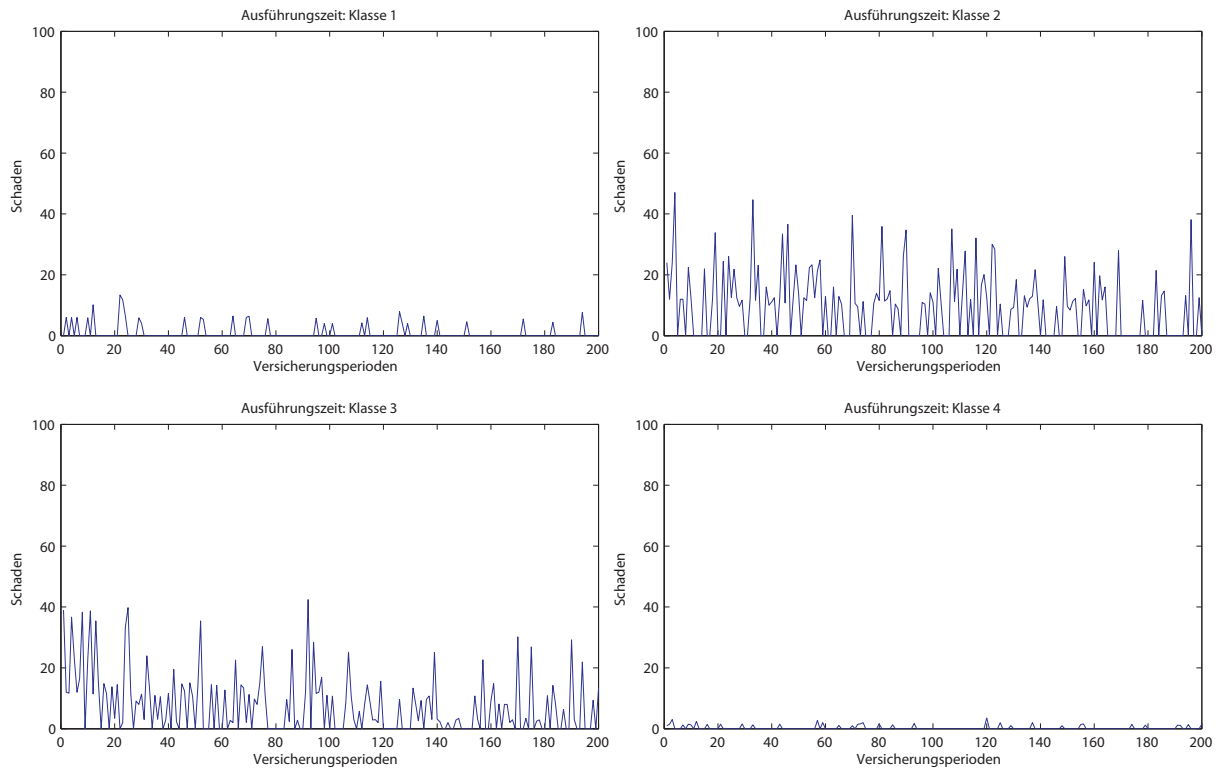


Abbildung C.2: Zeitserie der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals *Ausführungszeit* mit Risikoprofil *niedriges Risiko* und 100.000 ms Versicherungsperiode

Tabelle C.3: Werte der Indizes der Metrikpyramide für das Szenario mit sehr hohem Risiko

Index	Redundanz	Akzeptanz	Versicherung
Social-Loss	0,1095	0,0955	0,0395
s_ODM	0,1653	0,1977	0,1415
m_ODM	0,1570	0,2684	0,1272
s_IK	0,1026	0,1392	0,0995
m_IK	0,3956	0,2457	0,1815



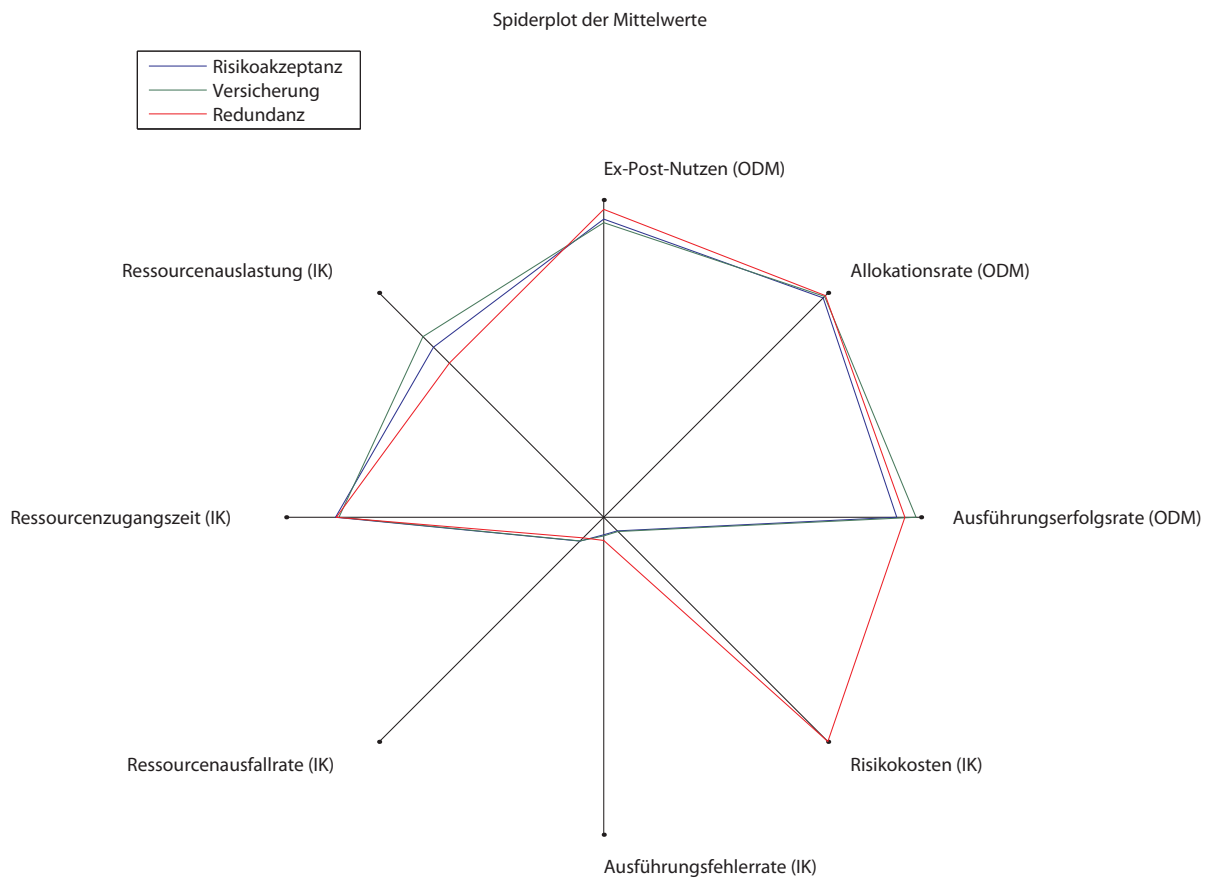


Abbildung C.3: Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko

Tabelle C.4: Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wert für das Szenario mit sehr hohem Risiko

ODM-Gewichtung	IK-Gewichtung	Social-Loss		
		Redundanz	Akzeptanz	Versicherung
0,1	0,9	0,1555	0,0829	0,0422
0,3	0,7	0,1325	0,0892	0,0409
0,5	0,5	0,1095	0,0955	0,0395
0,7	0,3	0,0865	0,1017	0,0382
0,9	0,1	0,0635	0,1080	0,0369

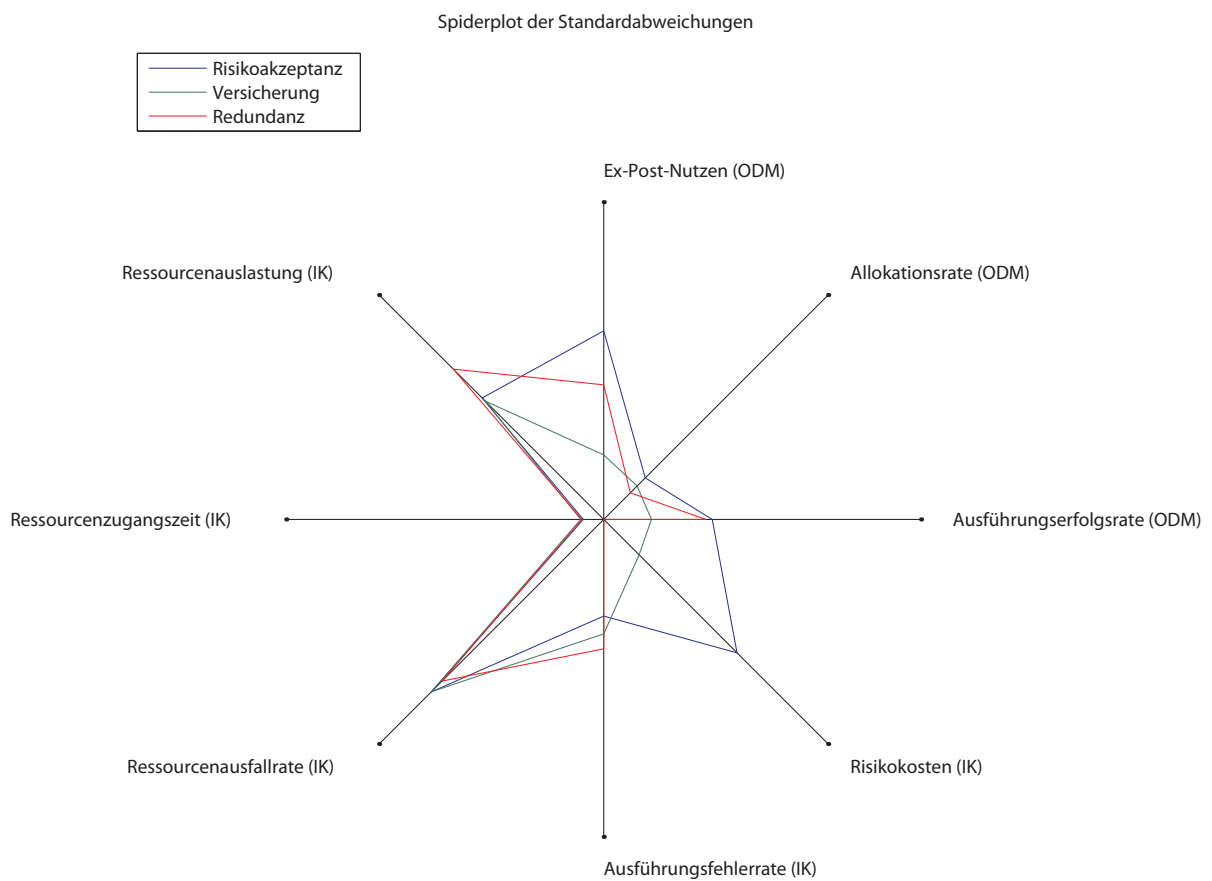


Abbildung C.4: Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko

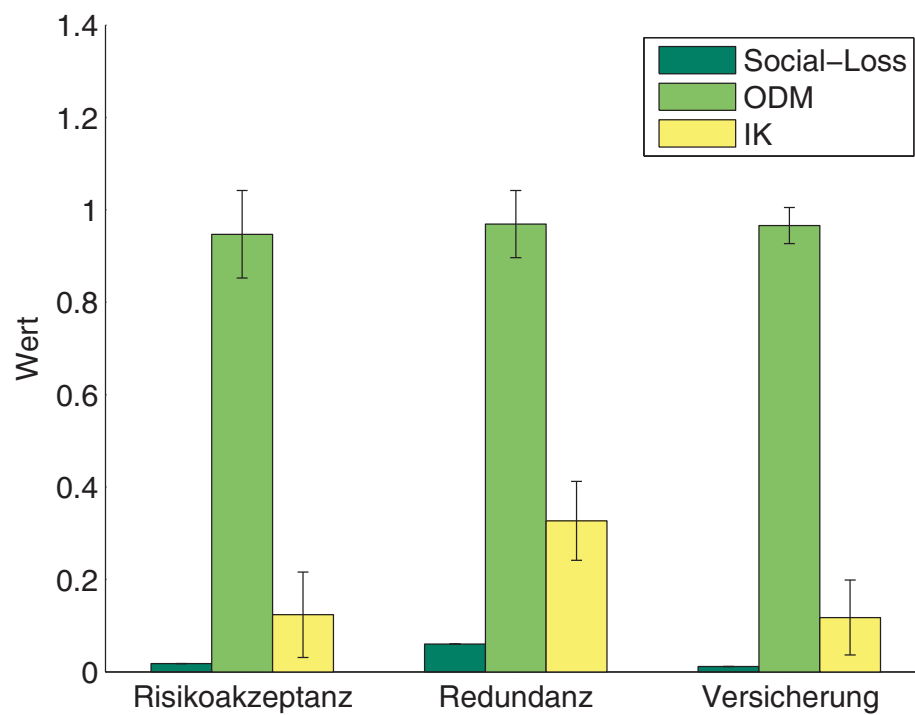


Abbildung C.5: Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Utility-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko

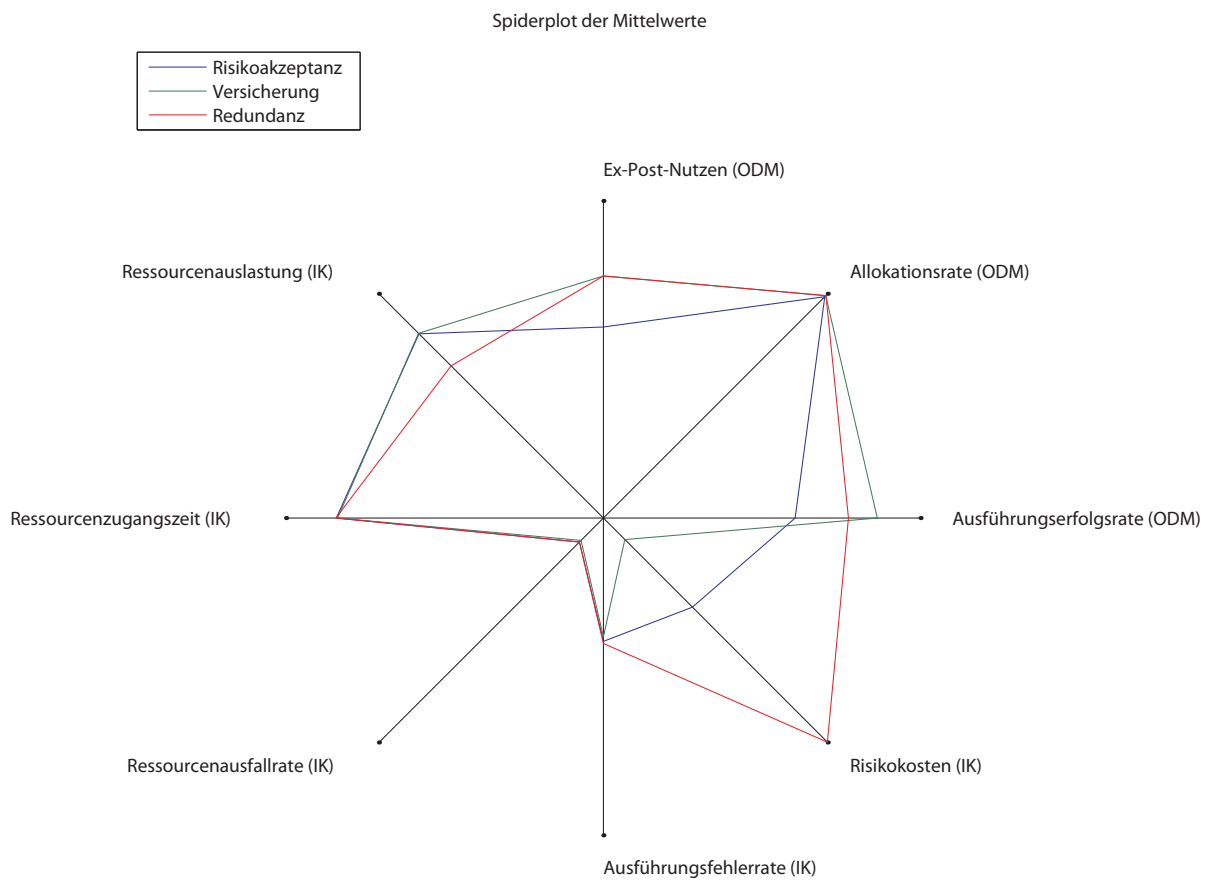


Abbildung C.6: Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko

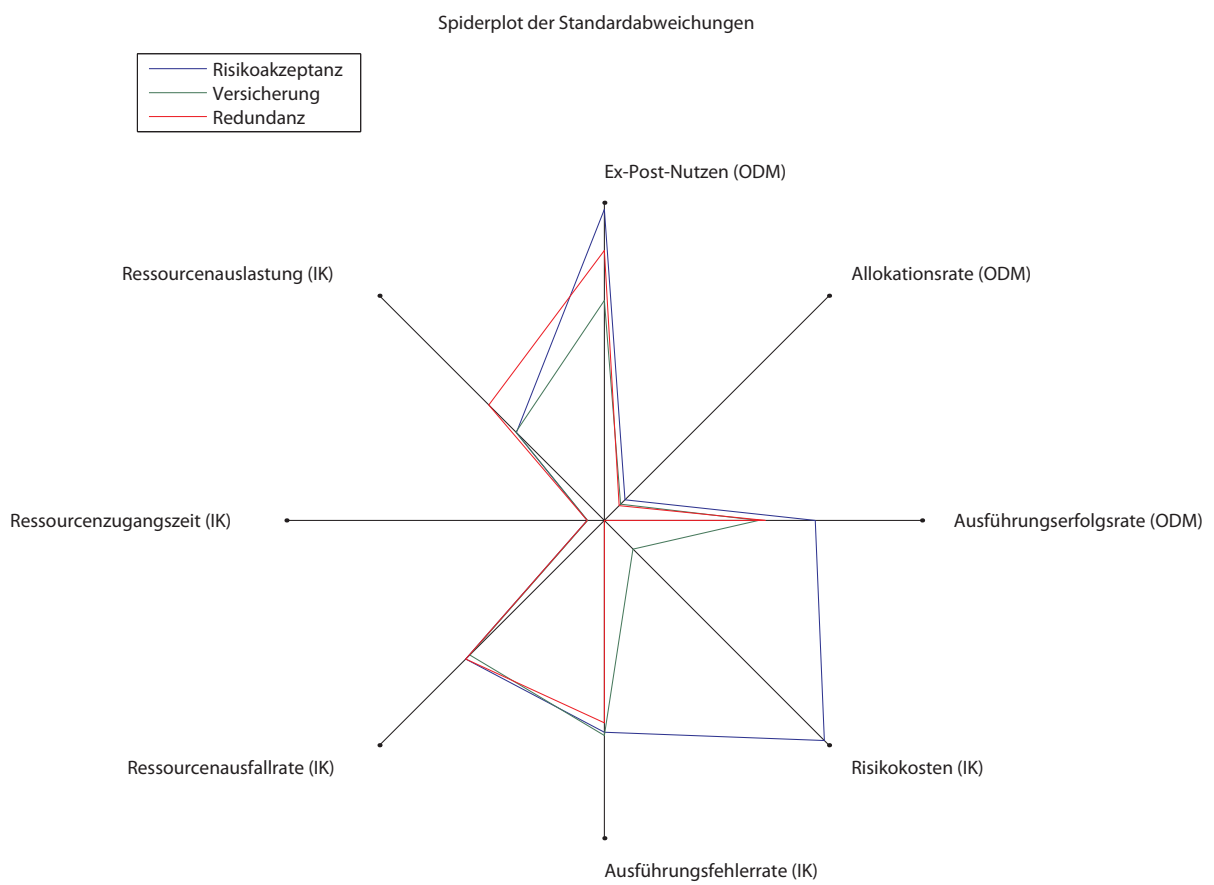


Abbildung C.7: Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko

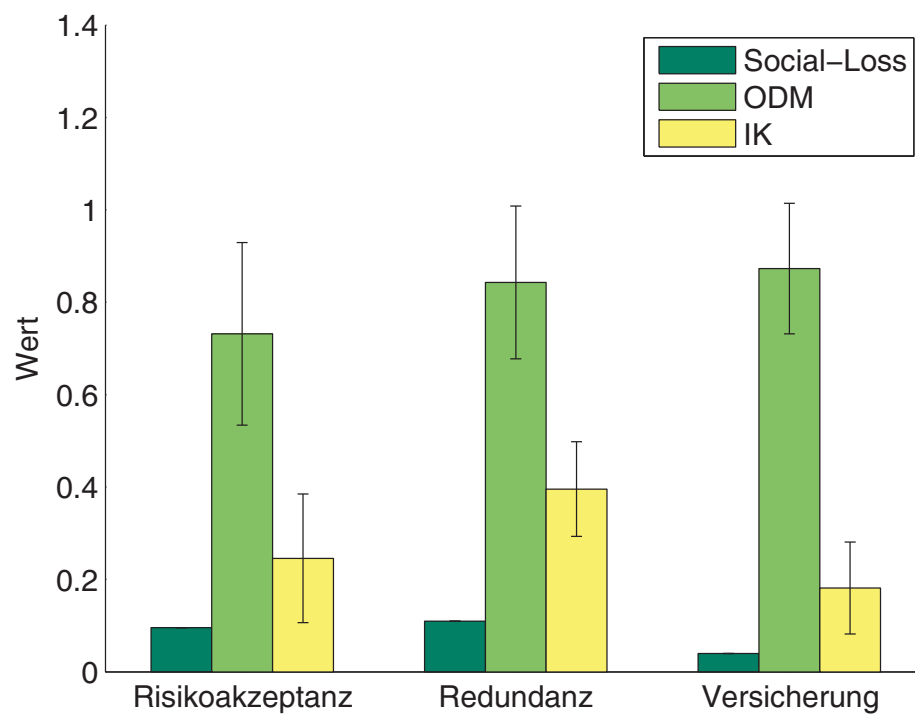


Abbildung C.8: Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Utility-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko



# Abbildungsverzeichnis

1.1	Technologien und Geschäftsmodelle zum Bezug von IT-Diensten und IT-Ressourcen	3
1.2	Gang der Untersuchung	14
1.3	Einordnung des technischen Risikos in den Kontext der Unternehmensrisiken [Seib06] [Wolk07] [Prok08]	16
2.1	Konzeptionelles Modell eines Application-Service-Provider-Geschäftsmodells unter Verwendung eines Grid-Systems [OISS07]	20
2.2	Schichtenmodelle der Grid- und Internet-Protokollarchitektur [JoEF04]	26
2.3	Fehlerontologie für Grid-Services und Grid-Ressourcen [HoFa08]	28
2.4	Aufbau eines Service-Level-Agreements nach dem WS-Agreement-Standard [ACDK <sup>+</sup> 06]	36
2.5	Abstraktes Modell eines Grid-Systems [FoGa01]	40
2.6	Risikobehaftete Komponenten von Grid-Systemen und ihre Zuweisung zum Schichtenmodell der Grid-Protokollarchitektur in Anlehnung an [DaPZ07]	56
3.1	Das Utility-Grid-Modell der CATNETS Grid-Ökonomie: Aufteilung der Grid-Dienste in einen Servicemarkt, auf dem Anwendungsdienste (Basic-Services) gehandelt werden und einen Ressourcenmarkt, auf dem elementare Dienste (Ressourcen-Services) gehandelt werden. [ERSR <sup>+</sup> 05]	63
3.2	Netzwerkmodell zur Modellierung der Risiken, denen die Marktteilnehmer ausgesetzt sind [SHES <sup>+</sup> 08]. Die Zahlenangabe innerhalb der Standorte legt die Ausfallwahrscheinlichkeit des Knotens fest.	67
3.3	Die Transaktionsphasen einer elektronischen Markttransaktion [MüEK03, S. 297ff.]	69
3.4	Aufbau der bilateralen Verhandlung: Nachfrager und Anbieter haben eine subjektive Vorstellung davon, in welchem Preisintervall ein Abschluss möglich ist. Die Grenze stellt bei beiden Verhandlungspartnern ihr individueller Reservationspreis dar.	73



4.1	An Grid-Systeme angepasster Risikomanagementzyklus der FERMA [Fede02]. . . . .	83
4.2	Entscheidungsbaum mit Auszahlungsmatrix (Nutzen) für Servicekonsument und Serviceanbieter bei Akzeptanz des Risikos . . . . .	84
4.3	Einordnung der Verfahren zur Risikobehandlung in ihren charakteristischen Einsatzbereich [Bank04] . . . . .	86
4.4	Einsatzszenarien verschiedener Versicherungsarten zum Risikotransfer [GrSh05]	91
4.5	Entscheidungsbaum bei Verwendung einer Versicherung zum Risikotransfer aus Sicht eines Ressourcenkonsumenten . . . . .	93
4.6	Risikoaverse Einstellung der Ressourcenkonsumenten in Anlehnung an [Bank04]	93
5.1	Bestandteile einer Versicherung [ZwEi07] . . . . .	98
5.2	Beispielhafte Zelle der Schadensstatistik . . . . .	104
5.3	Zeitbezogene Versicherungsperiode . . . . .	105
5.4	Transaktionsbezogene Versicherungsperiode . . . . .	106
5.5	Beispielhafte Schadensstatistik der Tarifmerkmale Ausführungszeit und Ressourcenmenge . . . . .	109
5.6	Klassifikation der versicherungstechnischen Risiken [Farn06, S. 85ff.] [Albr92, S. 8] [Wagn00, S. 139f.] . . . . .	113
6.1	Allgemeiner Simulationsprozess zur Durchführung einer Simulationsstudie nach Law [Law07, S. 67]. . . . .	123
6.2	Die drei wichtigsten Komponenten der CATNETS Simulationsumgebung [CSSE <sup>+</sup> 06]: der Szenariogenerator zur Erzeugung von Szenarien, der Simulator zur Simulation der Grid-Versicherung und die Evaluationskomponente zur Analyse der Messdaten. . . . .	128
6.3	Überblick über die Architektur des CATNETS Grid-Simulators mit einem Grid-Benutzer und einer Grid-Versicherung [CSSE <sup>+</sup> 06]. . . . .	131
6.4	Implementierung des Grid-Versicherungsagenten als UML-Klassendiagramm . . . . .	136
6.5	Verfügbare Nachrichten zur Kommunikation mit der Grid-Versicherung . . . . .	137
6.6	Die Klasse InsuranceNegotiation zur Kommunikation des Versicherungsnehmers (Basic-Service) mit der Grid-Versicherung . . . . .	137
6.7	Sequenzdiagramm eines beispielhaften Ablaufprozesses einer Grid-Versicherung in der CATNETS Grid-Ökonomie . . . . .	140
6.8	Allgemeine Methodik zum Erstellen eines zusammengesetzten Index [PiZC00, S. 42] [Boss99] . . . . .	145

6.9	Metrikenpyramide zur Evaluierung der Gesamtperformanz des Ressourcenmanagements unter Einbezug der Grid-Versicherung in Anlehnung an [SEVC <sup>+</sup> 07]	147
7.1	Netzwerk des Simulationsszenarios mit 200 Knoten und deren Netzwerkverbindung; das Netzwerk wurde mithilfe des CATNETS Szenariogenerators automatisiert erzeugt.	165
7.2	Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor $k = 5,0 * 10^{-5}$ (niedriges Risikoprofil)	170
7.3	Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor $k = 2,0 * 10^{-5}$ (hohes Risikoprofil)	171
7.4	Zweifach klassifizierender Versicherungstarif zur Versicherung der Basic-Service-Ausführung	172
7.5	Zeitreihe der Tariffaktoren für eine Versicherungsperiode mit 8.000 Millisekunden und die Risikoprofile <i>hohes und niedriges Risiko</i>	176
7.6	Zeitreihe der Tariffaktoren für eine Versicherungsperiode von 100.000 Millisekunden und die Risikoprofile <i>hohes und niedriges Risiko</i>	177
7.7	Zeitreihe der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals <i>Ausführungszeit</i> mit Risikoprofil <i>hohes Risiko</i> und 100.000 ms Versicherungsperiode	179
7.8	Zeitreihe der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals <i>Ausführungszeit</i> mit Risikoprofil <i>hohes Risiko</i> und 8.000 ms Versicherungsperiode	180
7.9	Prämieneinnahme- und Schadensprozess der Grid-Versicherung mit statischem (Szenario <i>_n_1_5</i> , wobei <i>_1_5</i> für einen statischen Sicherheitszuschlag von 1,5 steht) und dynamischen (Szenarien <i>_n</i> und <i>_h</i> ) Sicherheitszuschlägen; das Subskript <i>_n</i> entspricht einem Simulationslauf mit niedrigem Risiko und das Subskript <i>_h</i> steht für einen Simulationslauf mit hohem Risiko.	184
7.10	Entwicklung des Kapitals der Versicherung mit dynamischen Sicherheitszuschlägen; die waagrechten Linien kennzeichnen die Schwellenwerte der Sicherheitszuschläge.	185
7.11	Die Entwicklung der Anzahl an versicherten und unversicherten Basic-Services über die Zeit für das Risikoprofil mit hohem (Subskript <i>_h</i> ) und niedrigem Risiko (Subskript <i>_n</i> )	189

7.12	Zeitserie der Ex-Post-Nutzen der Basic-Services mit niedrigem Risiko . . . . .	190
7.13	Zeitserie der Ex-Post-Nutzen der Basic-Services mit niedrigem Risiko . . . . .	191
7.14	Vergleich der Mittelwerte und Standardabweichungen der Strategien Risikoakzeptanz (unversicherte Basic-Services) und Risikotransfer auf eine Versicherung (versicherte Basic-Services) für die Risikoprofile mit niedrigem und hohem Risiko	192
7.15	Zeitserie des Gens <i>Risikoeinschätzung</i> für das Risikoprofil mit niedrigem Risiko .	194
7.16	Zeitserie des Gens <i>Risikoeinschätzung</i> für das Risikoprofil mit hohem Risiko . .	195
7.17	Zeitserie des Gens <i>Zahlungsbereitschaft</i> für das Risikoprofil mit niedrigem Risiko	196
7.18	Zeitserie der Gens <i>Zahlungsbereitschaft</i> für das Risikoprofil mit hohem Risiko . .	198
7.19	Zeitserien des Gens <i>Risikoeinschätzung</i> , aufgegliedert nach den 5 Basic-Service-Typen für das Szenario mit hohem Risiko . . . . .	200
7.20	Zeitserien des Gens <i>Zahlungsbereitschaft</i> , aufgegliedert nach den 5 Basic-Service-Typen für das Szenario mit hohem Risiko . . . . .	201
7.21	Anteil der Kosten der Versicherungspolice an den Gesamtkosten einer Basic-Service-Ausführung für die 5 verschiedenen Basic-Service-Typen; der blau gekennzeichnete Bereich entspricht den Kosten der Ressourcen, der weiße Bereich kennzeichnet die Kosten der Versicherungsprämie. Zur besseren Vergleichbarkeit sind die Werteskalen der Geldeinheiten auf das Intervall zwischen 0 und 1 normalisiert. . . . .	203
7.22	Verteilung der Zuverlässigkeit der Basic-Services in Abhängigkeit der Ausführungszeit und der Recheneinheiten für den Skalierungsfaktor $k = 2,0 * 10^{-4}$ . . .	206
7.23	Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko; alle Werte sind normiert auf das Intervall zwischen 0 und 1. . . . .	207
7.24	Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko; alle Werte sind normiert auf das Intervall zwischen 0 und 1. . . . .	208
7.25	Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Loss-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil hohes Risiko . . . . .	213
C.1	Zeitserie der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals <i>Ausführungszeit</i> mit Risikoprofil <i>niedriges Risiko</i> und 8.000 ms Versicherungsperiode . . . . .	276

---

C.2	Zeitreihe der Schadenssummen für die Merkmalsausprägungen des Tarifmerkmals <i>Ausführungszeit</i> mit Risikoprofil <i>niedriges Risiko</i> und 100.000 ms Versicherungsperiode . . . . .	277
C.3	Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko . . . . .	278
C.4	Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko . . . . .	279
C.5	Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Utility-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil niedriges Risiko . . . . .	280
C.6	Mittelwerte der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko . . . . .	281
C.7	Standardabweichungen der Indikatoren für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko . . . . .	282
C.8	Vergleich der Infrastrukturkosten (IK), der On-Demand-Verfügbarkeit (ODM) und des Social-Utility-Index für die drei Risikobehandlungsstrategien Risikoakzeptanz, Versicherung und Redundanz im Risikoprofil sehr hohes Risiko . . . . .	283



# Tabellenverzeichnis

2.1	Messergebnisse zur Zuverlässigkeit der Job-Ausführung im LHC-Grid für drei Messperioden aus dem Jahr 2006 [LGWT06] . . . . .	53
2.2	Messergebnisse in Sekunden bei der Ausführung eines Test-Jobs in zwei Computational-Grid-Systemen, TeraGrid und GEON-Grid. Dieser Job benötigt 100 MB Daten als Eingabe, die im Rechenschritt aggregiert werden und abschließend in einer 500MB Datei an den Aufrufer zurückgeschickt werden. Über einen Zeitraum von drei Monaten wurde stündlich dieser Test-Job an beide Systeme geschickt. [KHOS <sup>+</sup> 06] . . . . .	54
3.1	Gegenüberstellung des Ausführungsmodells von CATNETS und das erweiterte Modell für diese Arbeit . . . . .	78
5.1	Beispielhafte Klassifizierung des Tarifmerkmals <i>Ausführungszeit</i> . . . . .	110
5.2	Beispielhafte Klassifizierung des Tarifmerkmals <i>Ressourcenmenge</i> . . . . .	111
7.1	Netzwerkaufbau und Ressourcenverteilung des Simulationsszenarios im Vergleich zu den Grid-Systemen Grid'5000, D-Grid und LHC-Computing-Grid. . . . .	163
7.2	Konfiguration der Amazon Elastic-Cloud-2 (EC2) Services und die daraus abgeleitete Konfiguration des simulierten Ressourcenmarkts . . . . .	166
7.3	Konfiguration der Nachfrage nach Ressourcen-Services durch Basic-Services und deren Reservationspreise . . . . .	168
7.4	Grundlegende Simulationsparameter des Szenarios . . . . .	174
7.5	Simulationsparameter zur Untersuchung der Versicherungsperiode unter verschiedenen Risikoprofilen . . . . .	174
7.6	Anteil Spätschäden für die untersuchten Simulationskonfigurationen . . . . .	181
7.7	Definition von Schwellenwerten für die Anpassung des Sicherheitszuschlags. . .	183

---

7.8	Werte der Indizes der Metrikpyramide für das Szenario mit hohem Risiko; die Spalte M beinhaltet die Mittelwerte und die Spalte S die Standardabweichung der Werte . . . . .	211
7.9	Werte der Indizes der Metrikpyramide für das Szenario mit hohem Risiko . . . .	214
7.10	Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wertes für das Szenario mit hohem Risiko . . . . .	214
C.1	Werte der Indizes der Metrikpyramide für das Szenario mit niedrigem Risiko . .	275
C.2	Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wert für das Szenario mit niedrigem Risiko . . . . .	276
C.3	Werte der Indizes der Metrikpyramide für das Szenario mit sehr hohem Risiko . .	277
C.4	Vergleich verschiedener ODM- und IK-Gewichtungen und des daraus resultierenden Social-Loss-Index-Wert für das Szenario mit sehr hohem Risiko . . . . .	278

# Abkürzungsverzeichnis

<b>IT</b>	Informationstechnologie
<b>IaaS</b>	Infrastructure-as-a-Service
<b>SaaS</b>	Software-as-a-Service
<b>ASP</b>	Application-Service-Provider
<b>SLA</b>	Service-Level-Agreement
<b>ERP</b>	Enterprise-Resource-Planning
<b>OGSA</b>	Open-Grid-Services-Architektur
<b>URIs</b>	Uniform-Resource-Identifiers
<b>WS</b>	Web-Services
<b>WS-Agreement</b>	Web-Service-Agreement
<b>XML</b>	eXtended Markup Language
<b>SOAP</b>	Simple-Object-Access-Protocol
<b>WSDL</b>	Web-Service-Description-Language
<b>UDDI</b>	Universal Description, Discovery, and Integration
<b>ECMWF</b>	European Centre for Medium-Range Weather Forecasts
<b>EGEE</b>	Enabling Grids for E-science
<b>BOINC</b>	Berkeley Open Infrastructure for Network Computing
<b>SETI@Home</b>	Search for Extraterrestrial Intelligence at Home
<b>CDNs</b>	Content-Delivery-Networks



<b>LHC</b>	Large-Hadron-Collider
<b>EC2</b>	Elastic-Compute-Cloud
<b>S3</b>	Simple-Storage-Service
<b>CERN</b>	Conseil Européen pour la Recherche Nucléaire
<b>RMS</b>	Ressourcenmanagementsystem
<b>CATNETS</b>	Catallaxy paradigm for decentralized operation of dynamic application networks
<b>CS</b>	Complex-Service
<b>RS</b>	Resource-Service
<b>FERMA</b>	Federation of European Risk Management Associations
<b>UML</b>	Unified Modeling Language
<b>ODM</b>	On-Demand Verfügbarkeit
<b>IK</b>	Infrastrukturkosten
<b>P2P</b>	Peer-To-Peer

# Literaturverzeichnis

- [AAEC96] Axtell, R.; Axelrod, R.; Epstein, J. M.; Cohen, M. D.: “Aligning simulation models: A case study and results”. In: *Computational & Mathematical Organization Theory*, Vol. 1(2), S. 123–141, doi:10.1007/BF01299065, ISSN 1381-298X (Print) 1572-9346 (Online), February 1996. 126
- [ABKM01] Andersen, D.; Balakrishnan, H.; Kaashoek, F.; Morris, R.: “Resilient overlay networks”. In: *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*, S. 131–145, ACM, New York, NY, USA, ISBN 1-58113-389-8, doi:http://doi.acm.org/10.1145/502034.502048, 2001. 71
- [ABKN<sup>+</sup>06] Asadzadeh, P.; Buyya, R.; Kei, C. L.; Nayar, D.; Venugopal, S.: *Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies*, Ch. 22, S. 431–458. *Wiley Series on Parallel and Distributed Computing*, John Wiley & Sons, Inc., doi:10.1002/0471732710, 2006. 23
- [ACDK<sup>+</sup>06] Andrieux, A.; Czajkowski, K.; Dan, A.; Keahey, K.; Ludwig, H.; Nakata, T.; Pruyne, J.; Rofrano, J.; Tuecke, S.; Xu, M.: “Web Service Agreement Specification (WS-Agreement)”. *Forschungsbericht, Open Grid Forum - Grid Resource Allocation Agreement Protocol (GRAAP) WG*, 2006. 35, 36, 285
- [ACKL<sup>+</sup>02] Anderson, D. P.; Cobb, J.; Korpela, E.; Lebofsky, M.; Werthimer, D.: “SETI@home: an experiment in public-resource computing”. In: *Communications of the ACM*, Vol. 45(11), S. 56–61, ISSN 0001-0782, doi:10.1145/581571.581573, URL <http://dx.doi.org/10.1145/581571.581573>, November 2002. 41
- [AdHu00] Adar, E.; Huberman, B.: “Free Riding on Gnutella”. In: *First Monday*, Vol. 5(10), 2000. 70

- [Albr92] Albrecht, P.: Zur Risikotransformationstheorie der Versicherung: Grundlagen und Ökonomische Konsequenzen. Verlag Versicherungswirtschaft, Karlsruhe, 1992. 112, 113, 286
- [AlIM07] Altmann, J.; Ion, M.; Mohammed, A. A. B.: “Taxonomy of Grid Business Models”. In: Proceedings of the 4th International Workshop on Grid Economics and Business Models (GECON 2007), Rennes, France, August 28, 2007, S. 29–43, Rennes, France, 2007. 4
- [Ande04] Anderson, D. P.: “BOINC: a system for public-resource computing and storage”. S. 4–10, URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1382809](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1382809), 2004. 41
- [AnMc07] Anderson, D. P.; McLeod, J.: “Local Scheduling for Volunteer Computing.” In: IPDPS, S. 1–8, IEEE, 2007. 42
- [APST05] Anderson, T.; Peterson, L.; Shenker, S.; Turner, J.: “Overcoming the Internet Impasse through Virtualization”. In: Computer, Vol. 38(4), S. 34–41, ISSN 0018-9162, doi:<http://dx.doi.org/10.1109/MC.2005.136>, April 2005. 19
- [Arro51] Arrow, K. J.: “Alternative Approaches to the Theory of Choice in Risk-Taking Situations”. In: Econometrica, Vol. 19, S. 404–437, 1951. 101
- [AsBu06] de Assuncao, M. D.; Buyya, R.: “InterGrid: A Case for Internetworking of Islands of Grids”. Technical Report GRIDS-TR-2006-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, February 2006. 49
- [ASEB07] de Assunção, M. D.; Streitberger, W.; Eymann, T.; Buyya, R.: “Enabling the Simulation of Service-Oriented Computing and Provisioning Policies for Autonomic Utility Grids”. In: J. Altmann; D. J. Veit (Hrsg.), Grid Economics and Business Models - 4th International Workshop, GECON 2007, *Lecture Notes in Computer Science*, Vol. Volume 4685/2007, S. 136–149, Springer Berlin / Heidelberg, Rennes, France, doi:10.1007/978-3-540-74430-6, 28 August 2007. 126
- [Axel97] Axelrod, R.: “Advancing the art of simulation in the social sciences”. In: Complexity, Vol. 3(2), S. 16–22, ISSN 1076-2787, doi:<http://dx.doi.org/10.1002/>

- (SICI)1099-0526(199711/12)3:2<16::AID-CPLX4>3.0.CO;2-K, ISSN:1076-2787, November/December 1997. 10, 11
- [BaBL02] Baker, M.; Buyya, R.; Laforenza, D.: “Grids and grid technologies for wide-area distributed computing”. In: *Softw. Pract. Exper.*, Vol. 32(15), S. 1437–1466, ISSN 0038-0644, doi:<http://dx.doi.org/10.1002/spe.488>, 2002. 2, 4, 5, 16, 19, 22, 24
- [BaGo83] Barro, R. J.; Gordon, D. B.: “Rules, discretion and reputation in a model of monetary policy”. In: *Journal of Monetary Economics*, Vol. 12(1), S. 101–121, URL <http://ideas.repec.org/a/eee/moneco/v12y1983i1p101-121.html> (LetzterAbruf: 20.05.2008), 1983. 158
- [Bake95] Baker, F.: “Requirements for IP Version 4 Routers”. Rfc 1812, Internet Engineering Task Force (IETF) - Network Working Group, <http://www.ietf.org/rfc/rfc1812.txt>, 1995. 23
- [Balt04] Baltaxe, D.: “The Adaptive Organization: An Examination of On Demand Computing”. Multiclient study, META Group, 2004. 3
- [Bank98] Banks, J.: *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, Ch. Principles of Simulation, S. 3–30. John Wiley & Sons, New York, ISBN: 978-0-471-13403-9, 1998. 10
- [Bank04] Banks, E.: *Alternative Risk Transfer: Integrated Risk Management through Insurance, Reinsurance and the Capital Markets*. Wiley Finance, Wiley & Sons, ISBN-13: 978-0470857458, January 2004. 85, 86, 93, 286
- [BBCD<sup>+</sup>03] Bertero, M.; Bonetto, P.; Carracciuolo, L.; D’Amore, L.; Formiconi, A.; Guaracino, M. R.; Laccetti, G.; Murli, A.; Oliva, G.: “MedIGrid: A Medical Imaging Application for Computational Grids”. In: *IPDPS ’03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, S. 252.2, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-1926-1, 2003. 39, 50
- [BBCF<sup>+</sup>04] Bartolini, C.; Boulmakou, A.; Christodoulou, A.; Farrell, A.; Salle, M.; Trastour, D.: “Management by Contract: IT Management driven by Business Objectives”. *Forschungsbericht HPL-2004-184*, Trusted Systems Laboratory, HP Laboratories Palo Alto, October 2004. 37

- [BCCC<sup>+</sup>07] Buss, G.; Catalano, M.; Chacin, P.; Chao, I.; Eymann, T.; Freitag, F.; Joita, L.; Navarro, L.; Parasie, N.; Rana, O. F.; Schnizler, B.; Streitberger, W.; Veit, D.: "CATNETS - Deliverable D4.3 - Performance Evaluation". Forschungsbericht, Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik, 2007. 71, 122, 134, 147
- [BCCD<sup>+</sup>06] Bolze, R.; Cappello, F.; Caron, E.; Daydé, M.; Desprez, F.; Jeannot, E.; Jégou, Y.; Lanteri, S.; Leduc, J.; Melab, N.; Mornet, G.; Namyst, R.; Primet, P.; Quetier, B.; Richard, O.; Talbi, E.-G.; Touche, I.: "Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed". In: International Journal of High Performance Computing Applications, Vol. 20(4), S. 481–494, ISSN 1094-3420, doi:<http://dx.doi.org/10.1177/1094342006070078>, ISSN:1094-3420, November 2006. 11, 102
- [BCLW<sup>+</sup>04] Bucu, M. J.; Chang, R. N.; Luan, L. Z.; Ward, C.; Wolf, J. L.; Yu, P. S.: "Utility computing SLA management based upon business objectives". In: IBM Systems Journal, Vol. 43(1), S. 159–178, ISSN 0018-8670, 2004. 35, 38, 46, 101
- [BCMC<sup>+</sup>03] Bell, W. H.; Cameron, D. G.; Millar, A. P.; Capozza, L.; Stockinger, K.; Zini, F.: "Optorsim: A Grid Simulator for Studying Dynamic Data Replication Strategies". In: International Journal of High Performance Computing Applications, Vol. 17(4), S. 403–416, doi:[10.1177/10943420030174005](http://dx.doi.org/10.1177/10943420030174005), 2003. 127
- [BDFH<sup>+</sup>03] Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris, T.; Ho, A.; Neugebauer, R.; Pratt, I.; Warfield, A.: "Xen and the art of virtualization". In: SIGOPS Oper. Syst. Rev., Vol. 37(5), S. 164–177, ISSN 0163-5980, doi:<http://doi.acm.org/10.1145/1165389.945462>, 2003. 19
- [BeGa00] Beck, K.; Gamma, E.: More Java gems, Ch. Test-infected: programmers love writing tests, S. 357–376. Cambridge University Press, New York, NY, USA, ISBN:0-521-77477-2, 2000. 125
- [Berg07] van den Bergh, J. C.: "Evolutionary Thinking in Environmental Economics". Tinbergen Institute Discussion Papers 07-018/3, Faculty of Economics and Business Administration & Institute for Environmental Studies, Free University, De Boelelaan 1105, Amsterdam, Netherlands, URL <http://ideas.repec.org/p/dgr/uvatin/20070018.html>, February 2007. 72

- [Berl82] Berliner, B.: Die Grenzen der Versicherbarkeit von Risiken. Schweizer Rückversicherungsgesellschaft, Zürich, 1982. 100, 102
- [Bern03] Bernhard, M. G.: Strategisches IT-Management, Vol. 1, Ch. Service-Level-Management – Die IT als Supply-Chain organisieren, S. 137–161. Symposium Publishing, Düsseldorf, 2003. 34
- [Bégi08] Bégin, M.-E.: “An EGEE Comparative Study: Grids and Clouds - evolution or revolution?” Forschungsbericht, EGEE, May 2008. 3
- [BGIP05] Badica, C.; Ganzha, M.; Iordache, A.; Paprzycki, M.: “Rule-Based Framework for Automated Negotiation: Initial Implementation”. In: Proceedings of RuleML 2005, Galway, Ireland, 10-12 November 2005, Springer Verlag, 2005. 74
- [BHHK08] Buhl, H. U.; Heinrich, B.; Henneberger, M.; Krammer, A.: “Service Science”. In: Wirtschaftsinformatik, Vol. 50(1), S. 60–65, Januar 2008. 8, 30
- [BHMN<sup>+</sup>04] Booth, D.; Haas, H.; McCabe, F.; Newcomer, E.; Champion, M.; Ferris, C.; Orchard, D.: “Web Services Architecture”. Forschungsbericht, W3C Working Group, <http://www.w3.org/TR/ws-arch/>, February 2004. 30, 33
- [BhSC01] Bhoj, P.; Singhal, S.; Chutani, S.: “SLA management in federated environments”. In: Computer Networks, Vol. 35(1), S. 5–24, January 2001. 35
- [Birm93] Birman, K. P.: “The process group approach to reliable distributed computing”. In: Communications of the ACM, Vol. 36(12), S. 37–53, ISSN 0001-0782, doi: <http://doi.acm.org/10.1145/163298.163303>, December 1993. 2
- [Bisk06] Bisker, J.: “Insurance 2020 - Innovating beyond old models”. Forschungsbericht, IBM Institute for Business Value, Route 100, Somers, NY 10589 U.S.A., May 2006. 95, 225
- [BiVu99] Binmore, K.; Vulkan, N.: “Applying game theory to automated negotiation”. In: NETNOMICS, Vol. 1(1), S. 1–9, doi:10.1023/A:1011489402739, ISSN 1385-9587 (Print) 1573-7071 (Online), October 1999. 74
- [BLRS98] Buoudreau, M.-C.; Loch, K. D.; Robey, D.; Straud, D.: “Going global: Using information technology to advance the competitiveness of the virtual transnational organization”. In: Academy of Management Executive, Vol. 12(4), S. 120–128, 1998. 29

- [BMQL<sup>+</sup>07] Boss, G.; Malladi, P.; Quan, D.; Legregni, L.; Hall, H.: "Cloud Computing". Forschungsbericht, IBM Corporation, October 2007. 2
- [BoSc03] Boles, D.; Schmees, M.: "Kostenpflichtige Web-Services". In: W. Uhr; W. Esswein; E. Schoop (Hrsg.), *Wirtschaftsinformatik 2003/ Band I*, 2003. 4
- [Boss99] Bossel, H.: *Indicators for Sustainable Development: Theory, Method, Applications - A Report to the Balaton Group*. International Institute For Sustainable Development, Winnipeg, Manitoba, Canada, ISBN 1-895536-13-8, 1999. 144, 145, 286
- [BPSMM<sup>+</sup>06] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; Maler, E.; Yergeau, F.; Cowan, J.: "Extensible Markup Language (XML) 1.1 (Second Edition)". Forschungsbericht, W3C, August 2006. 33
- [Brae60] Braess, P.: *Versicherung und Risiko*. Gabler Verlag, Wiesbaden, 1960. 112
- [BrFr03] Bruns, C.; Freider: *Professionelles Portfoliomanagement*. Schäffer-Poeschel Verlag, Stuttgart, 3 Edition., ISBN 3-7910-2173-7, 2003. 84
- [BSEV08] Buss, G.; Streitberger, W.; Eymann, T.; Veit, D.: "Markets and Negotiations in the Grid Economy: How decentralized and centralized economic approaches perform". In: M. Bichler; T. Hess; H. Krmar; U. Lechner; F. Matthes; A. Picot; B. Speitkamp; P. Wolf (Hrsg.), *Multikonferenz Wirtschaftsinformatik (MKWI) 2008*, GITO-Verlag, Berlin, München, ISBN 978-3-940019-34-9, 26-28 Februar 2008. 69, 147
- [BSRB<sup>+</sup>08] Breikreutz, B.-J.; Stark, C.; Reguly, T.; Boucher, L.; Breikreutz, A.; Livstone, M.; Oughtred, R.; Lackner, D.; Bähler, J.; Wood, V.; Dolinski, K.; PDF, M. T. . . J.-.: "The BioGRID Interaction Database: 2008 Update". In: *Nucleic Acids Research*, Vol. 36(Database issue), S. D637–D640, doi:doi:10.1093/nar/gkm1001, 2008. 43
- [BuHW04] Burnecki, K.; Härdle, W.; Weron, R.: *Encyclopedia of Actuarial Science*, Ch. Simulation of risk processes. Wiley, Chichester, 2004. 142
- [BuKB02] den Burger, M.; Kielmann, T.; Bal, H. E.: "TOPOMON: A Monitoring Tool for Grid Network Topology". In: *ICCS '02: Proceedings of the International Conference on Computational Science-Part II*, S. 558–567, Springer-Verlag, London, UK, ISBN 3-540-43593-X, 2002. 101

- [BuPV08] Buyya, R.; Pathan, M.; Vakali, A. (Hrsg.): Content Delivery Networks, *Lecture Notes Electrical Engineering*, Vol. 9. Springer, ISBN: 978-3-540-77886-8, March 2008. 43
- [BuYV08] Buyya, R.; Yeo, C. S.; Venugopal, S.: “Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities”. In: Proceedings of The 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08), September 25-27,2008, DaLian,China, 2008. 2
- [Carr05] Carr, N. G.: “The End of Corporate Computing”. In: MITSloan Management Review, Vol. 46(3), S. 66–73, Spring 2005. 4, 5
- [Carr08] Carr, N. G.: The big switch: rewiring the world, from Edison to Google. W. W. Norton & Company, Inc., 500 Fifth Avenue, New York, NY 10110, 1 Edition., ISBN 978-0-393-06228-1, 2008. 1, 219
- [Casa07] Casanova, H.: “Benefits and Drawbacks of Redundant Batch Requests”. In: Journal of Grid Computing, Vol. 5(2), S. 235–250, doi:doi:10.1007/s10723-007-9068-6, URL <http://www.ingentaconnect.com/content/klu/grid/2007/00000005/00000002/00009068>, June 2007. 88
- [CBFN<sup>+</sup>08] Chao, I.; Brunner, R.; Freitag, F.; Navarro, L.; Chacin, P.; Ardaiz, O.; Joita, L.; Rana, O. F.: “A Decentralized Grid Market Infrastructure for Service Oriented Grids”. In: Wirtschaftsinformatik, Vol. 50(1), S. 25–30, Februar 2008. 125
- [CCSMN<sup>+</sup>04] Cameron, D. G.; Carvajal-Schiaffino, R.; Millar, A. P.; Nicholson, C.; Stockinger, K.; Zini, F.: “Analysis of Scheduling and Replica Optimisation Strategies for Data Grids using OptorSim”. In: Journal of Grid Computing, Vol. 2(1), S. 57–69, March 2004. 127
- [CDKN<sup>+</sup>02] Curbera, F.; Duftler, M.; Khalaf, R.; Nagy, W.; Mukhi, N.; Weerawarana, S.: “Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI”. In: IEEE Internet Computing, Vol. 06(2), S. 86–93, ISSN 1089-7801, doi:<http://doi.ieeecomputersociety.org/10.1109/4236.991449>, 2002. 34
- [CFKS<sup>+</sup>00] Chervenak, A.; Foster, I.; Kesselman, C.; Salisbury, C.; Tuecke, S.: “The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets”. In: Journal of Network and Computer Ap-



- plications, Vol. 23(3), S. 187–200, doi:10.1006/jnca.2000.0110, URL <http://citeseer.ist.psu.edu/chervenak99data.html>, July 2000. 43, 44, 45
- [CGSR<sup>+</sup>05] Catalano, M.; Giulioni, G.; Streitberger, W.; Reinicke, M.; Eymann, T.: “Evaluation and Metrics Framework”. Bayreuther Arbeitspapiere zur Wirtschaftsinformatik 6, IST-FP6-003769 CATNETS, URL <http://opus.ub.uni-bayreuth.de/volltexte/2007/362/> (LetzterAbruf: 20.05.2008), 2005. 158
- [ChKB05] Cheliotis, G.; Kenyon, C.; Buyya, R.: Peer-to-Peer Computing: The Evolution of a Disruptive Technology, Ch. Ten Lessons from Finance for Commercial Sharing of IT Resources, S. 244–264. Idea Group Inc., ISBN-13: 978-1591404293, 2005. 7
- [ChKe03] Cheliotis, G.; Kenyon, C.: “Autonomic Economics: Why Self-Managed e-Business Systems Will Talk Money”. In: Proceedings of IEEE Conference on E-Commerce '03 (CEC'03), June 2003. 8
- [CJSN03] Cao, J.; Jarvis, S. A.; Saini, S.; Nudd, G. R.: “GridFlow: Workflow Management for Grid Computing”. In: CCGRID '03: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid, S. 198, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-1919-9, 2003. 88
- [CKMT<sup>+</sup>03] Curbera, F.; Khalaf, R.; Mukhi, N.; Tai, S.; Weerawarana, S.: “The next step in Web services”. In: Commun. ACM, Vol. 46(10), S. 29–34, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/944217.944234>, 2003. 34
- [CSSE<sup>+</sup>06] Calabrese, G.; Schnizler, B.; Streitberger, W.; Eymann, T.; Zini, F.: “Simulator Development - Annual Report Year 2”. Bayreuther Arbeitspapiere zur Wirtschaftsinformatik 10, Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik, Bayreuth, URL <http://opus.ub.uni-bayreuth.de/volltexte/2007/364/> (LetzterAbruf: 15.05.2008), 2006. 128, 129, 130, 131, 286
- [Cusu08] Cusumano, M. A.: “The Changing Software Business: Moving from Products to Services”. In: Computer, Vol. 41(1), S. 20–27, ISSN 0018-9162, doi:<http://doi.ieeecomputersociety.org/10.1109/MC.2008.29>, January 2008. 2
- [CzFK99] Czajkowski, K.; Foster, I.; Kesselman, C.: “Resource Co-Allocation in Computational Grids”. In: HPDC '99: Proceedings of the 8th IEEE International

- Symposium on High Performance Distributed Computing, S. 37, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-0287-3, 1999. 89
- [Dabr08] Dabrowski, C.: “Reliability in Grid Computing Systems”. Forschungsbericht, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, URL [http://www.ogf.org/OGF\\_Special\\_Issue/GridReliabilityDabrowski.pdf](http://www.ogf.org/OGF_Special_Issue/GridReliabilityDabrowski.pdf) (LetzterAbruf:05.08.2008), 2008. 19, 32, 42, 43, 45, 55
- [DaPZ07] Dai, Y.-S.; Pan, Y.; Zou, X.: “A Hierarchical Modeling and Analysis for Grid Service Reliability”. In: IEEE Transactions on Computers, Vol. 56(5), S. 681–691, ISSN 0018-9340, doi:<http://doi.ieeecomputersociety.org/10.1109/TC.2007.1034>, 2007. 55, 56, 57, 58, 68, 82, 88, 102, 108, 139, 150, 169, 219, 220, 223, 285
- [Davi04] Davies, A.: “Computational intermediation and the evolution of computation as a commodity”. In: Applied Economics, Vol. 36(11), S. 1131–1142, available at <http://ideas.repec.org/a/taf/applec/v36y2004i11p1131-1142.html>, June 2004. 45
- [DaXP02] Dai, Y. S.; Xie, M.; Poh, K. L.: “Reliability Analysis of Grid Computing Systems”. In: PRDC '02: Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing, S. 97, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-1852-4, 2002. 55
- [DDKK<sup>+</sup>04] Dan, A.; Davis, D.; Kearney, R.; Keller, A.; King, R.; Kuebler, D.; Ludwig, H.; Polan, M.; Spreitzer, M.; Youssef, A.: “Web services on demand: WSLA-driven automated management”. In: IBM Syst. J., Vol. 43(1), S. 136–158, ISSN 0018-8670, 2004. 38, 89, 101
- [DGPB<sup>+</sup>06] Djemame, K.; Gourlay, I.; Padgett, J.; Birkenheuer, G.; Hovestadt, M.; Kao, O.; Voss, K.: “Introducing Risk Management into the Grid”. In: E-SCIENCE '06: Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, S. 28–35, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-2734-5, doi:<http://dx.doi.org/10.1109/E-SCIENCE.2006.92>, 2006. 81, 221
- [DiPi06] Dimakopoulos, M.-V. V.; Pitoura, M.-E.: “On the Performance of Flooding-Based Resource Discovery”. In: IEEE Trans. Parallel Distrib. Syst.,

- Vol. 17(11), S. 1242–1252, ISSN 1045-9219, doi:<http://dx.doi.org/10.1109/TPDS.2006.161>, 2006. 70, 220
- [DiSe68] Diamond, D. S.; Selwyn, L. L.: “Considerations for computer utility pricing policies”. In: Proceedings of the 1968 23rd ACM national conference, S. 189–200, ACM, New York, NY, USA, doi:<http://doi.acm.org/10.1145/800186.810580>, 1968. 4
- [Dool02] Dooley, K.: The Blackwell Companion to Organizations, Ch. Simulation Research Methods, S. 829–848. Blackwell Publishers Inc., London, 2002. 11, 120
- [Dunn06] Dunn, D.: “Many Data Centers Still Have No Risk Management Plan”. InformationWeek, URL <http://www.informationweek.com/shared/printableArticle.jhtml?articleID=183701425>, Letzter Abruf: 20.02.2008, March 2006. 7
- [DuRF05] Dumitrescu, C.; Raicu, I.; Foster, I.: “DI-GRUBER: A Distributed Approach to Grid Resource Brokering”. In: SC ’05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing, S. 38, IEEE Computer Society, Washington, DC, USA, ISBN 1-59593-061-2, doi:<http://dx.doi.org/10.1109/SC.2005.23>, 2005. 35
- [EdHa03] Edmonds, B.; Hales, D.: “Replication, Replication and Replication: Some Hard Lessons from Model Alignment”. In: Journal of Artificial Societies and Social Simulation, Vol. 6(4), S. online, URL <http://jasss.soc.surrey.ac.uk/6/4/11.html> (LetzterAbruf: 21.05.2008), 2003. 126
- [Eich88] Eichhorn, W. (Hrsg.): Measurement in Economics: Theory and Applications of Economic Indices. Physica, Heidelberg, 1988. 146
- [Elfa07] Elfatary, A.: “Dealing with change: components versus services”. In: Commun. ACM, Vol. 50(8), S. 35–39, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/1278201.1278203>, 2007. 29, 30
- [ENRS<sup>+</sup>06] Eymann, T.; Neumann, D.; Reinicke, M.; Schnizler, B.; Streitberger, W.; D., V.: “On the Design of a Two-Tiered Grid Market Structure”. In: Business Applications of P2P and Grid Computing, Multikonferenz Wirtschaftsinformatik (MKWI), Passau, Germany, Februar 2006. 62

- [Erne06] Ernest-Jones, T.: “Managing IT Business Risk - Safeguarding the organisation from IT failure”. Forschungsbericht, Mercury Interactive Corporation & Economist Intelligence Unit, 2006. 6
- [ERSR<sup>+</sup>05] Eymann, T.; Reinicke, M.; Streitberger, W.; Rana, O.; Joita, L.; Neumann, D.; Schnizler, B.; Veit, D.; Ardaiz, O.; Chacin, P.; Chao, I.; Freitag, F.; Navarro, L.; Catalano, M.; Gallegati, M.; Giulioni, G.; Schiaffino, R. C.; Zini, F.: “Catallaxy-based Grid markets”. In: Multiagent and Grid Systems, Vol. 1(4), S. 297–307, ISSN 1574-1702, ISSN:1574-1702, December 2005. 22, 61, 62, 63, 220, 285
- [ERSZ<sup>+</sup>05] Eymann, T.; Reinicke, M.; Streitberger, W.; Zini, F.; Giulioni, G.: “Analysis of Simulation Environment”. Bayreuther Arbeitspapiere zur Wirtschaftsinformatik 8, Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik, Bayreuth, URL <http://opus.ub.uni-bayreuth.de/volltexte/2007/359/> (LetzterAbruf:15.05.2008), 2005. 124
- [Eyma00] Eymann, T.: AVALANCHE - Ein agentenbasierter dezentraler Koordinationsmechanismus für elektronische Märkte. Dissertation, Albert-Ludwigs-Universität Freiburg, Wirtschaftswissenschaftliche Fakultät, 2000. 72, 73
- [EySH07] Eymann, T.; Streitberger, W.; Hudert, S.: “Global Grids - Making a Case for Self-organization in Large-Scale Overlay Networks”. In: U. Montanari; D. Sannella; R. Bruni (Hrsg.), Proceedings of Trusted and Global Computing (TGC) 2006, Vol. LNCS 4661, S. 51–68, Springer-Verlag, Berlin, Heidelberg, 2007. 75
- [EySM03] Eymann, T.; Sackmann, S.; Müller, G.: “Hayeks Katallaxie - Ein zukunftsweisendes Konzept für die Wirtschaftsinformatik?” In: Wirtschaftsinformatik, Vol. 45(5), S. 491–496, 2003. 72
- [Farn06] Farny, D.: Versicherungsbetriebslehre. Verlag Versicherungswirtschaft, 4 Edition., ISBN-13: 978-3899522051, 2006. 112, 113, 286
- [Fede02] Federation of European Risk Management Associations (FERMA): “A Risk Management Standard”. Forschungsbericht, Rue de la presse 4, Brussels, Belgium, URL <http://www.ferma.eu/Portals/2/documents/RMS/RMS-German.pdf> (LetzterAbruf:12.04.2008), 2002. 82, 83, 286

- [FKNT02a] Foster, I.; Kesselman, C.; Nick, J.; Tuecke, S.: “The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration”. Open Grid Service Infrastructure WG, Global Grid Forum, URL <http://citeseer.ist.psu.edu/foster02physiology.html>, 2002. 30
- [FKNT02b] Foster, I.; Kesselman, C.; Nick, J. M.; Tuecke, S.: “Grid Services for Distributed System Integration”. In: *Computer*, Vol. 35(6), S. 37–46, ISSN 0018-9162, doi:<http://dx.doi.org/10.1109/MC.2002.1009167>, 2002. 30
- [FKSB<sup>+</sup>06] Foster, I.; Kishimoto, H.; Savva, A.; Berry, D.; Djaoui, A.; Grimshaw, A.; Horn, B.; Maciel, F.; Siebenlist, F.; Subramaniam, R.; Treadwell, J.; Reich, J. V.: “The Open Grid Services Architecture, Version 1.5”. Forschungsbericht, Open Grid Forum, 2006. 31, 32, 220
- [FoGa01] Fox, G.; Gannon, D.: “Computational Grids”. In: *Computing in Science and Engg.*, Vol. 3(4), S. 74–77, ISSN 1521-9615, doi:<http://dx.doi.org/10.1109/5992.931906>, 2001. 31, 40, 285
- [FoJK04] Foster, I.; Jennings, N. R.; Kesselman, C.: “Brain Meets Brawn: Why Grid and Agents Need Each Other”. In: *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, S. 8–15, IEEE Computer Society, Washington, DC, USA, ISBN 1-58113-864-4, doi:<http://dx.doi.org/10.1109/AAMAS.2004.78>, URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1373456](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1373456), ISBN: 1-58113-864-4, 19-23 July 2004. 122
- [FoKe99] Foster, I.; Kesselman, C.: “Computational grids”. In: , S. 15–51, 1999. 22, 39
- [FoKe03] Foster, I.; Kesselman, C.: *The Grid 2: Blueprint for a New Computing Infrastructure (The Morgan Kaufmann Series in Computer Architecture and Design)*. Morgan Kaufmann, ISBN 1558609334, November 2003. 2, 5, 21, 30
- [FoKT01] Foster, I.; Kesselman, C.; Tuecke, S.: “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”. In: *Lecture Notes in Computer Science*, Vol. 2150, URL <http://citeseer.ist.psu.edu/foster01anatomy.html>, 2001. 19, 23, 29, 47
- [FoTu05] Foster, I.; Tuecke, S.: “Describing the elephant: the different faces of IT as service”. In: *Queue*, Vol. 3(6), S. 26–29, ISSN 1542-7730, doi:<http://doi.acm>.

- org/10.1145/1080862.1080874, July/August 2005. 19
- [FrWK02] Fremantle, P.; Weerawarana, S.; Khalaf, R.: “Enterprise services”. In: Communications of the ACM, Vol. 45(10), S. 77–82, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/570907.570935>, 2002. 33
- [Garf07] Garfinkel, S. L.: “Technical Report TR-08-07: An Evaluation of Amazon’s Grid Computing Services: EC2, S3 and SQS”. Forschungsbericht, Center for Research on Computation and Society, School for Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, August 2007. 4
- [GGKS02] Gottschalk, K. D.; Graham, S.; Kreger, H.; Snell, J.: “Introduction to Web Services Architecture”. In: IBM Systems Journal, Vol. 41(2), S. 170–177, URL <http://dblp.uni-trier.de/db/journals/ibmsj/ibmsj41.html#GottschalkGKS02>, 2002. 34
- [Gilb08] Gilbert, N.: Agent-based Models. Nr. 153 in Quantitative Applications in the Social Sciences, SAGE Publications, London, ISBN: 978-1-4129-4964-4, 2008. 121
- [GiNu05] Gissler, N.; Nunn, S.: “Grid Computing - Harnessing the power of many to deliver high performance”. Forschungsbericht, Accenture, 2005. 4, 5
- [GiTr05] Gilbert, N.; Troitzsch, K. G.: Simulation for the Social Scientist. Open University Press, McGraw-Hill Education, Maidenhead, Berkshire, England, 2 Edition., ISBN-13: 978-0335-21600-0, 2005. 10, 11, 121, 122
- [GJGB<sup>+</sup>05] Gagliardi, F.; Jones, B.; Grey, F.; Bégin, M.-E.; Heikkurinen, M.: “Building an infrastructure for scientific Grid computing: status and goals of the EGEE project”. In: Philosophical Transactions of the Royal Society, Vol. 363(1833), S. 1729–1742, doi:10.1098/rsta.2005.1603, August 2005. 41
- [GIRV04] Glass, R. L.; Ramesh, V.; Vessey, I.: “An analysis of research in computing disciplines”. In: Communications of the ACM, Vol. 47(6), S. 89–94, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/990680.990686>, June 2004. 10
- [GoLS03] Gordon, L. A.; Loeb, M. P.; Sohail, T.: “A framework for using insurance for cyber-risk management”. In: Communications of the ACM, Vol. 46(3), S. 81–85, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/636772.636774>, March 2003. 82, 90

- [Greg06] Gregor, S.: "The Nature of Theory in Information Systems". In: MIS Quarterly, Vol. 30(3), S. 611–642, September 2006. 10, 11
- [GrSh05] Grey, W.; Shi, D.: Handbook of Integrated Risk Management for E-Business: Measuring, Modeling, and Managing Risk, Ch. Enterprise Risk Management: A Value Chain Perspective, S. 1–32. J. Ross Publishing, 2005. 82, 91, 286
- [GrSi91] Gray, J.; Siewiorek, D.: "High-availability computer systems". In: Computer, Vol. 24(9), S. 39–48, ISSN 0018-9162, doi:10.1109/2.84898, September 1991. 156
- [GrSz02] Gray, J.; Szalay, A.: "The world-wide telescope". In: Communications of the ACM, Vol. 45(11), S. 50–55, ISSN 0001-0782, doi:http://doi.acm.org/10.1145/581571.581572, 2002. 43
- [Grze02a] Grzebiela, T.: "Insurability of Electronic Commerce Risks". In: HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7, S. 185–193, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-1435-9, 2002. 90, 92
- [Grze02b] Grzebiela, T.: Internetrisiken - Versicherbarkeit und Alternativer Risikotransfer. Gabler Edition Wissenschaft, Deutscher Universitäts-Verlag, Wiesbaden, ISBN: 3-8244-7754-8, 2002. 100, 102, 112
- [GFW<sup>+</sup>94] Grimshaw, A. S.; Wulf, W. A.; French, J. C.; Weaver, A. C.; Reynolds, P. F.: "Legion: The Next Logical Step Toward a Nationwide Virtual Computer". Technical Report CS-94-21, Department of Computer Science, University of Virginia, June 1994. 156
- [HaRE03] Hales, D.; Rouchier, J.; Edmonds, B.: "Model-to-Model Analysis". In: Journal of Artificial Societies and Social Simulation, Vol. 6(4), S. online, URL <http://jasss.soc.surrey.ac.uk/6/4/5.html> (LetzterAbruf: 21.05.2008), 2003. 126
- [Haye45] Hayek, F.: "The use of knowledge in Society". In: American Economist, Vol. 4, S. 519–530, 1945. 72
- [Haye08] Hayes, B.: "Cloud computing". In: Communications of the ACM, Vol. 51(7), S. 9–11, ISSN 0001-0782, doi:http://doi.acm.org/10.1145/1364782.1364786, ISSN:0001-0782, July 2008. 2



- [Herb92] Herbrich, M.: Kumulkontrolle. Gabler, Wiesbaden, 1992. 102
- [Hewi08] Hewitt, C.: “ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing”. In: IEEE Internet Computing, Vol. 12(5), S. 96–99, ISSN 1089-7801, doi:<http://doi.ieeecomputersociety.org/10.1109/MIC.2008.107>, September/October 2008. 2
- [HoFa08] Hofer, J.; Fahringer, T.: “A Multi-Perspective Taxonomy for Systematic Classification of Grid Faults”. In: PDP '08: Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), S. 126–130, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-3089-5, doi:<http://dx.doi.org/10.1109/PDP.2008.15>, 2008. 27, 28, 100, 285
- [HoKV06] Hovestadt, M.; Kao, O.; Voß, K.: “The First Step of Introducing Risk Management for Preprocessing SLAs”. In: Proceedings of IEEE International Conference on Services and Computing 2006 (SCC'06), S. 36–43, IEEE Computer Society, September 2006. 81, 82
- [Horn93] Horn, R. V.: Statistical Indicators For the Economic & Social Sciences. Cambridge University Press, Cambridge, UK, ISBN-13: 978-0521413336, 1993. 146
- [HuHo95] Hubermann, B. A.; Hogg, T.: “Distributed computation as an economic system.” In: Journal of Economic Perspectives, Vol. 9(1), S. 141–152, 1995. 61
- [HuSi05] Huhns, M. N.; Singh, M. P.: “Service-Oriented Computing: Key Concepts and Principles”. In: IEEE Internet Computing, Vol. 9(1), S. 75–81, ISSN 1089-7801, doi:<http://dx.doi.org/10.1109/MIC.2005.21>, 2005. 33
- [HwPa07] Hwang, J.; Park, J.: “Decision Factors of Enterprises for Adopting Grid Computing”. In: J. Altmann; D. J. Veit (Hrsg.), Proceedings of 4th International Workshop on Grid Economics and Business Models, GECON 2007, Rennes, France, August 2007, *Lecture Notes in Computer Science*, Vol. 4685/2007, S. 16–28, Springer Berlin / Heidelberg, doi:10.1007/978-3-540-74430-6, 2007. 49
- [IaFo04] Iamnitchi, A.; Foster, I.: Grid resource management: state of the art and future trends, Ch. A peer-to-peer approach to resource location in Grid environments,



- S. 413–429. Kluwer Academic Publishers, Norwell, MA, USA, ISBN 1-4020-7575-8, 2004. 70
- [IJSE07] Iosup, A.; Jan, M.; Sonmez, O. O.; Epema, D. H. J.: “On the dynamic resource availability in grids”. In: 8th IEEE/ACM International Conference on Grid Computing (Grid2007), S. 26–33, doi:10.1109/GRID.2007.4354112, September 2007. 27, 55, 101, 102, 169, 219, 220
- [ILJA<sup>+</sup>07] Iosup, A.; Li, H.; Jan, M.; Anoep, S.; Dumitrescu, C.; Wolters, L.; Epema, D.: “The Grid Workloads Archive”. Forschungsbericht, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands, 2007. 178
- [JAKP05] Juhas, Z.; Andics, A.; Kuntner, K.; Pota, S.: “Towards a Robust and Fault-Tolerant Multicast Discovery Architecture for Global Computing Grids”. In: Scalable Computing: Practice and Experience, Vol. 6(2), S. 23–33, June 2005. 32
- [JaLG08] Jaeger, P. T.; Lin, J.; Grimes, J. M.: “Cloud Computing and Information Policy: Computing in a Policy Cloud?” In: Journal of Information Technology and Politics, Vol. 5(3), S. forthcoming, 2008. 2, 3, 4
- [Jenn00] Jennings, N. R.: “On agent-based software engineering”. In: Artificial Intelligence, Vol. 117(2), S. 277–296, ISSN 0004-3702, doi:http://dx.doi.org/10.1016/S0004-3702(99)00107-1, ISSN:0004-3702, March 2000. 61
- [JhMF08] Jha, S.; Merzky, A.; Fox, G.: “Using Clouds to Provide Grids Higher-Levels of Abstraction and Explicit Support for Usage Modes”. Technical report, Pervasive Technology Labs, Indiana University, Bloomington, Indiana, USA, April 2008. 3
- [JiPMK07] Jiménez-Peris, R.; Patiño-Martínez, M.; Kemme, B.: “Enterprise Grids: Challenges Ahead”. In: Journal of Grid Computing, Vol. 5(3), S. 283–294, doi: 10.1007/s10723-007-9071-y, September 2007. 48
- [JoEF04] Joseph, J.; Ernest, M.; Fellenstein, C.: “Evolution of grid computing architecture and grid adoption models”. In: IBM Syst. J., Vol. 43(4), S. 624–645, ISSN 0018-8670, 2004. 19, 23, 26, 32, 47, 48, 49, 285

- [Kart72] Karten, W.: “Zum Problem der Versicherbarkeit und zur Risikopolitik des Versicherungsnehmers - betriebswirtschaftliche Aspekte”. In: *Zeitschrift für die gesamte Versicherungswirtschaft*, Vol. 2(3), S. 279–299, 1972. 96, 100, 101, 221
- [Kart93] Karten, W.: *Das Einzelrisiko und seine Kalkulation, Versicherungswirtschaftliches Studienwerk*, Vol. 2. Gabler Verlag, Wiesbaden, 4 Edition., 1993. 114
- [Kart00] Karten, W.: *Versicherungsbetriebslehre - Kernfragen aus entscheidungsorientierter Sicht*. Verlag Versicherungswirtschaft, Karlsruhe, 2000. 103
- [Kaye03] Kaye, D.: *Loosely Coupled: The Missing Pieces of Web Services*. RDS Press, ISBN 1881378241, 2003. 33
- [KeCh02] Kenyon, C.; Cheliotis, G.: “Architecture Requirements for Commercializing Grid Resources”. In: *HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, S. 215, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-1686-6, 2002. 5, 19
- [KeCh03] Kephart, J. O.; Chess, D. M.: “The Vision of Autonomic Computing”. In: *Computer*, Vol. 36(1), S. 41–50, ISSN 0018-9162, doi:<http://dx.doi.org/10.1109/MC.2003.1160055>, 2003. 46
- [KeCh04] Kenyon, C.; Cheliotis, G.: *Grid Resource Management: State of the Art and Future Trends*, Ch. *Grid Resource Commercialization: Economic Engineering and Delivery Scenarios*, S. 465–478. Kluwer Academic Publishers, Dordrecht, Netherlands, ISBN-13: 978-1402075759, 2004. 5, 8
- [KeCh05] Kenyon, C.; Cheliotis, G.: *Handbook of Integrated Risk Management for E-Business: Measuring, Modeling, and Managing Risk*, Ch. *Elements of Financial Risk Management for Grid and Utility Computing*, S. 169–191. J. Ross Publishing, 2005. 47, 90
- [KeLu03] Keller, A.; Ludwig, H.: “The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services”. In: *Journal of Network and Systems Management*, Vol. 11(1), S. 57–81, ISSN 1064-7570, doi:<http://dx.doi.org/10.1023/A:1022445108617>, March 2003. 4, 85
- [KHOS<sup>+</sup>06] Khalili, O.; He, J.; Olschanowsky, C.; Snaveley, A.; Casanova, H.: “Measuring the Performance and Reliability of Production Computational Grids”. In:

- Proceedings of the 7th IEEE/ACM International Conference on Grid Computing (GRID2006), S. 293–300, Barcelona, doi:10.1109/ICGRID.2006.311028, ISBN: 1-4244-0343-X, 2006. 27, 54, 101, 291
- [Knig21] Knight, F. H.: Risk, Uncertainty and Profit. The University of Chicago Press, Chicago, 1921. 15, 101
- [KoAV07] Kondo, D.; Anderson, D. P.; Vii, J. M.: “Performance Evaluation of Scheduling Policies for Volunteer Computing”. In: E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007), S. 415–422, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-3064-8, doi:http://dx.doi.org/10.1109/E-SCIENCE.2007.57, 2007. 42
- [KoLe04] Kossmann, D.; Leymann, F.: “Web Services”. In: Informatik-Spektrum, Vol. 27(2), Springer Berlin / Heidelberg, doi:10.1007/s00287-004-0378-9, ISSN 0170-6012 (Print) 1432-122X (Online), April 2004. 30, 33
- [KrBM02] Krauter, K.; Buyya, R.; Maheswaran, M.: “A taxonomy and survey of grid resource management systems for distributed computing”. In: Software - Practice & Experience, Vol. 32(2), S. 135–164, ISSN 0038-0644, doi:http://dx.doi.org/10.1002/spe.432, 2002. 22
- [Kreg03] Kreger, H.: “Fulfilling the Web services promise”. In: Commun. ACM, Vol. 46(6), S. 29–34, ISSN 0001-0782, doi:http://doi.acm.org/10.1145/777313.777334, 2003. 34
- [Law07] Law, A. M.: Simulation Modeling and Analysis. Industrial Engineering and Management Science, McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, 4 Edition., 2007. 10, 121, 123, 125, 222, 286
- [Lawt08] Lawton, G.: “Developing Software Online With Platform-as-a-Service Technology”. In: Computer, Vol. 41(6), S. 13–15, ISSN 0018-9162, doi:http://doi.ieeecomputersociety.org/10.1109/MC.2008.185, June 2008. 2
- [LaZV08] Laplante, P. A.; Zhang, J.; Voas, J.: “What’s in a Name? Distinguishing between SaaS and SOA”. In: IT Professional, Vol. 10(3), S. 46–50, ISSN 1520-9202, doi:http://doi.ieeecomputersociety.org/10.1109/MITP.2008.60, May/June 2008. 2

- [Lebr00] Lebrun, P.: “The Large Hadron Collider, A Megascience Project”. In: Proceedings of the 38th INFN Eloisatron Project Workshop on Superconducting Materials for High Energy Colliders, URL <http://documents.cern.ch/cgi-bin/setlink?base=preprint&categ=cern&id=lhc-project-report-374>, 2000. 43
- [LeDa07] Levitin, G.; Dai, Y.-S.: “Service reliability and performance in grid system with star topology”. In: Reliability Engineering & System Safety, Vol. 92(1), S. 40–46, doi:10.1016/j.ress.2005.11.005, January 2007. 55
- [LeRD03] Leff, A.; Rayfield, J. T.; Dias, D. M.: “Service-Level Agreements and Commercial Grids”. In: IEEE Internet Computing, Vol. 7(4), S. 44–50, ISSN 1089-7801, doi:<http://dx.doi.org/10.1109/MIC.2003.1215659>, 2003. 35
- [LGWT06] Li, H.; Groep, D.; Wolters, L.; Templon, J.: “Job Failure Analysis and Its Implications in a Large-Scale Production Grid”. In: e-science, Vol. 0, S. 27, doi:<http://doi.ieeeecomputersociety.org/10.1109/E-SCIENCE.2006.93>, 2006. 27, 53, 101, 291
- [LKAC<sup>+</sup>08] Litke, A.; Konstanteli, K.; Andronikou, V.; Chatzis, S.; Varvarigou, T.: “Managing service level agreement contracts in OGSA-based Grids”. In: Future Gener. Comput. Syst., Vol. 24(4), S. 245–258, ISSN 0167-739X, doi:<http://dx.doi.org/10.1016/j.future.2007.06.004>, April 2008. 35, 38, 101
- [LéMo07] Lévy-Garboua, L.; Montmarquette, C.: “A Theory of Satisfaction and Utility with Empirical and Experimental Evidences”. In: Conference of the French Economic Association, Behavioral and Experimental Economics, May 23-25, 2007, Lyon, France, Groupe d’Analyse et de Théorie Economique, Lyon, France, URL <http://www.gate.cnrs.fr/afse-jee/Papiers/37.pdf> (LetzterAbruf: 20.05.2008), May 2007. 143
- [LSWK<sup>+</sup>99] Lee, C. A.; Stepanek, J.; Wolski, R.; Kesselman, C.; Foster, I.: “A Network Performance Tool for Grid Environments”. In: ACM/IEEE SC 1999 Conference (SC ’99), IEEE Computer Society, Los Alamitos, CA, USA, ISBN 1-58113-091-0, doi:<http://doi.ieeeecomputersociety.org/10.1109/SC.1999.10020>, 1999. 101
- [Mack97] Mack, T.: Schadenversicherungsmathematik, *Schriftenreihe Angewandte Versicherungsmathematik*, Vol. Heft 28. Verlag Versicherungswirtschaft, Karlsruhe,

- ISBN-13: 978-3884879573, 1997. 111
- [Maes94] Maes, P.: “Agents that reduce work and information overload”. In: Communications of the ACM, Vol. 37(7), S. 30–40, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/176789.176792>, July 1994. 61
- [MaMP06] Maclaren, J.; Mc; Pickles, S.: “Co-Allocation, Fault Tolerance and Grid Computing”. In: Proceedings of the UK e-Science All Hands Meeting, S. 155–162, Engineering and Physical Sciences Research Council, Swindon, United Kingdom, September 2006. 32
- [Mark52] Markowitz, H. M.: “Portfolio Selection”. In: The Journal of Finance, Vol. VII(1), March 1952. 88
- [MaSm95] March, S. T.; Smith, G. F.: “Design and natural science research on information technology”. In: Decision Support Systems, Vol. 15(4), S. 251–266, ISSN 0167-9236, doi:[http://dx.doi.org/10.1016/0167-9236\(94\)00041-2](http://dx.doi.org/10.1016/0167-9236(94)00041-2), ISSN:0167-9236, December 1995. 10
- [McKe05] McKee, P.: “Grid—the ‘white knight’ for business?” In: BT Technology Journal, Vol. 23(3), S. 45–51, ISSN 1358-3948, doi:<http://dx.doi.org/10.1007/s10550-005-0029-y>, July 2005. 5
- [MCTH<sup>+</sup>06] Majlender, P.; Carlsson, C.; Tétard, F.; Heikkilä, M.; Gourlay, I.; Voß, K.; Birkenheuer, G.: “AssessGrid D1.2: Risk Management Evaluation”. Forschungsbericht, ASSESSGRID, 2006. 85, 88, 89
- [McCa04] Menasce, D. A.; Casalicchio, E.: “QoS in Grid Computing”. In: IEEE Internet Computing, Vol. 8(4), S. 85–87, Jul/Aug 2004. 48
- [Mein97] Meinecke, H.: Integriertes Risiko-Management für Unternehmens-Eigentümer. Dissertation, Universität St. Gallen, 1997. 15
- [Mein08] Meinel, T.: “Advance Reservation of Grid Resources via Real Options”. In: IEEE Joint Conference on E-Commerce Technology (CEC’08) and Enterprise Computing, E-Commerce and E-Services (EEE ’08), July 21-24, 2008, Crystal City, Washington, D.C., USA, 2008. 90
- [MüEK03] Müller, G.; Eymann, T.; Kreutzer, M.: Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft. Lehrbücher Wirtschaftsinformatik, Oldenbourg, München, 2003. 37, 69, 92, 285

- [Meye05] Meyer, U.: Paneuropäische Tarifstruktur in der Kfz-Haftpflichtversicherung – Abschlussbericht eines EU-Projekts, *Versicherungswissenschaftliche Studien*, Vol. 30, Ch. Kfz-Haftpflichtversicherung in Europa. Vergleichende Untersuchung der ökonomisch-statistischen Situation, S. 35–188. Nomos, Baden-Baden, ISBN: 978-3-8329-1499-8, 2005. 9, 108, 110, 111
- [MiFo04] Michalewicz, Z.; Fogel, D. B.: *How to Solve It: Modern Heuristics*. Springer, second edition Edition., ISBN 3540224947, URL <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20%&path=ASIN/3540224947>, December 2004. 72, 75
- [MiLe08] Mietzner, R.; Leymann, F.: “Towards Provisioning the Cloud: On the Usage of Multi-Granularity Flows and Services to Realize a Unified Provisioning Infrastructure for SaaS Applications”. In: *IEEE Congress on Services - Part I*, S. 3–10, doi:<http://doi.ieeecomputersociety.org/10.1109/SERVICES-1.2008.36>, 2008. 2
- [MiRo92] Milgrom, P.; Roberts, J.: *Economics, organizations, and management*. Prentice-Hall, 1992. 7
- [MKLN<sup>+</sup>02] Milojicic, D. S.; Kalogeraki, V.; Lukose, R.; Nagaraja, K.; Pruyne, J.; Richard, B.; Rollins, S.; Xu, Z.: “Peer-to-Peer Computing”. Forschungsbericht HPL-2002-57, HP Laboratories Palo Alto, URL <http://citeseer.ist.psu.edu/milojicic02peertopeer.html>, March 2002. 70
- [MLMB01a] Medina, A.; Lakhina, A.; Matta, I.; Byers, J.: “BRITE: An Approach to Universal Topology Generation”. In: *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems - MASCOTS’01*, Cincinnati, Ohio, August 2001. 129
- [MLMB01b] Medina, A.; Lakhina, A.; Matta, I.; Byers, J.: “BRITE: Universal Topology Generation from a User’s Perspective”. Forschungsbericht BUCS-TR-2001-003, Computer Science Department, Boston University, URL <http://www.cs.bu.edu/brite/publications/usermanual.pdf> (LetzterAbruf: 16.05.2008), April 2001. 129
- [MoBa03] Montresor, A.; Baboglu, O.: “Biology-Inspired Approaches to Peer-to-Peer Computing in BISON”. In: *Proceedings of the 3rd International Conference*

- on Intelligent System Design and Applications (ISDA'03), Advances in Soft Computing, S. 515–522, Springer-Verlag, Oklahoma, August 2003. 6
- [MoRW05] Moore, R.; Rajasekar, A.; Wan, M.: “Data Grids, Digital Libraries, and Persistent Archives: An Integrated Approach to Sharing, Publishing, and Archiving Data”. In: Proceedings of the IEEE, Vol. 93(3), S. 578–588, doi:doi:10.1109/JPROC.2004.842761, ISSN: 0018-9219, March 2005. 43
- [Mugl80] Mugler, J.: “Risikopolitische Strategien im Grenzbereich des Versicherbaren”. In: Zeitschrift für die gesamte Versicherungswirtschaft, Vol. 69(1), S. 71–87, 1980. 101
- [Murc04] Murch, R.: Autonomic Computing. On Demand Series, IBM Press, ISBN-13: 978-0131440258, April 2004. 46
- [MVNVAC<sup>+</sup>07] Moreno-Vozmedianob, R.; Nadimintia, K.; Venugopala, S.; Alonso-Condec, A. B.; Gibbinsa, H.; Buyya, R.: “Portfolio and investment risk analysis on global grids”. In: Journal of Computer and System Sciences, Vol. 73(8), S. 1154–1175, doi:doi:10.1016/j.jcss.2007.02.005, December 2007. 39
- [Nath05] Nathan, K.: Handbook of Integrated Risk Management for E-Business, Ch. Foreword, S. v–vi. J. Ross Publishing, 2005. 1
- [NeHO06] Neumann, D.; Holtmann, C.; Orwat, C.: “Grid-Economics”. In: Wirtschaftsinformatik, Vol. 48(3), S. 206–209, 2006. 7
- [Neum86] Neumann, P. G.: “On hierarchical design of computer systems for critical applications”. In: IEEE Trans. Softw. Eng., Vol. 12(9), S. 905–920, ISSN 0098-5589, 1986. 81
- [Nguy07] Nguyen, T.: Grenzen der Versicherbarkeit von Katastrophenrisiken. DUV, doi: 10.1007/978-3-8350-9612-7, 2007. 99, 101, 102
- [NoMa07] North, M. J.; Macal, C. M.: Managing Business Complexity - Discovering Strategic Solutions with Agent-Based Modeling and Simulation. Oxford University Press, Madison Avenue, New York, New York 10016, ISBN-13: 978-0-19-517211-9, 2007. 119, 121, 125, 126, 161
- [NWGO<sup>+</sup>08] Nurmi, D.; Wolski, R.; Grzegorzczak, C.; Obertelli, G.; Soman, S.; Youseff, L.; Zagorodnov, D.: “Eucalyptus : A Technical Report on an Elastic Utility Computing Architecture - Linking Your Programs to Useful Systems”. UCSB Com-



- puter Science Technical Report 2008-10, Computer Science Department, University of California, Santa Barbara, Santa Barbara, California 93106, 2008. 2
- [Obj07] Object Management Group: “OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2”. Forschungsbericht, Needham, MA 02494, U.S.A., URL <http://www.uml.org/> (LetzterAbruf: 21.05.2008), November 2007. 124, 134
- [OISS07] Ogawa, H.; Itoh, S.; Sonoda, T.; Sekiguchi, S.: “GridASP: an ASP framework for Grid utility computing: Research Articles”. In: *Concurrency and Computation: Practice & Experience*, Vol. 19(6), S. 885–891, ISSN 1532-0626, doi: <http://dx.doi.org/10.1002/cpe.v19:6>, 2007. 19, 20, 285
- [Orlo06] de Orlov, L. L.: “Grid Computing: Addressing Today’s Business Issues”. Forschungsbericht, Deloitte, 2006. 1
- [PaBu07] Pathan, M.; Buyya, R.: “A Taxonomy and Survey of Content Delivery Networks”. Technical Report GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Melbourne, Australia, <http://www.cs.mu.oz.au/apathan/Papers/CDN-Taxonomy.pdf>, February 2007. 45
- [PaDD05] Padgett, J.; Djemame, K.; Dew, P.: “Grid-Based SLA Management”. In: P. M. A. Sloot; A. G. Hoekstra; T. Priol; A. Reinefeld; M. Bubak (Hrsg.), *Advances in Grid Computing - EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Revised Selected Papers, Lecture Notes in Computer Science*, Vol. 3470, Springer, ISBN 3-540-26918-5, 2005. 89
- [PaGe03] Papazoglou, M. P.; Georgakopoulos, D.: “Service-Oriented computing: Introduction”. In: *Commun. ACM*, Vol. 46(10), S. 24–28, ISSN 0001-0782, doi:10.1145/944217.944233, URL <http://portal.acm.org/citation.cfm?id=944233>, October 2003. 33
- [Pale04] Paleologo, G. A.: “Price-at-Risk: A methodology for pricing utility computing services”. In: *IBM Systems Journal*, Vol. 43(1), S. 20–31, ISSN 0018-8670, 2004. 47



- [Papa03] Papazoglou, M. P.: "Service -Oriented Computing: Concepts, Characteristics and Directions". In: Proceedings of Fourth International Conference on Web Information Systems Engineering (WISE'03), S. 3, IEEE Computer Society, Los Alamitos, CA, USA, ISBN 0-7695-1999-7, doi:<http://doi.ieeeecomputersociety.org/10.1109/WISE.2003.1254461>, 2003. 6
- [Papa07] Papazoglou, M. P.: Web Services: Principles and Technology. Prentice-Hall, 2007. 33
- [PaVa06] Pallis, G.; Vakali, A.: "Insight and perspectives for content delivery networks". In: Commun. ACM, Vol. 49(1), S. 101–106, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/1107458.1107462>, 2006. 45
- [PeDa00] Peterson, L. L.; Davie, B. S.: Computernetze. dpunkt-Verlag, 2000. 23
- [PiRW03] Picot, A.; Reichwald, R.; Wigand, R.: Die grenzenlose Unternehmung. Information, Organisation und Management. Gabler, Wiesbaden, 2003. 1
- [Pitk75] Pitkänen, P.: "Tariff Theory". In: Journal of Actuarial studies in non-life insurance (ASTIN), Vol. 8(2), S. 204–228, URL [http://www.casact.org/library/astin/vol8no2/\(LetzterAbruf:10.07.2008\)](http://www.casact.org/library/astin/vol8no2/(LetzterAbruf:10.07.2008)), September 1975. 108
- [PiZC00] Pintér, L.; Zahedi, K.; Cressman, D. R.: Capacity Building for Integrated Environmental Assessment and Reporting: Training Manual. International Institute for Sustainable Development (IISD), Winnipeg, Manitoba, Canada, 2nd Edition., URL [http://www.iisd.org/PUBLICATIONS/pub.aspx?id=310\(LetzterAbruf:20.05.2008\)](http://www.iisd.org/PUBLICATIONS/pub.aspx?id=310(LetzterAbruf:20.05.2008)), 2000. 145, 286
- [PIJr06] Plaszczyk, P.; Jr., R. W.: Grid Computing: The Savvy Manager's Guide. Morgan Kaufmann Publishers, 2006. 49
- [PoFl00] Poess, M.; Floyd, C.: "New TPC benchmarks for decision support and web commerce". In: SIGMOD Rec., Vol. 29(4), S. 64–71, ISSN 0163-5808, doi:<http://doi.acm.org/10.1145/369275.369291>, ISSN:0163-5808, December 2000. 146
- [Pool70] Poole, W.: "Optimal Choice of Monetary Policy Instruments in a Simple Stochastic Macro Model". In: The Quarterly Journal of Economics,

- Vol. 84(2), S. 197–216, URL <http://ideas.repec.org/a/tpr/qjecon/v84y1970i2p197-216.html>, May 1970. 158
- [Prei98] Preist, C.: “Economic Agents for Automated Trading”. Forschungsbericht HPL-98-77, HP Laboratories Bristol, April 1998. 74
- [Prok08] Prokein, O.: IT-Risikomanagement - Identifikation, Quantifizierung und wirtschaftliche Steuerung. Gabler, 2008. 15, 16, 285
- [Prui81] Pruitt, D.: Negotiation behavior. Academic Press Inc., New York, 1981. 73
- [Rapp04] Rappa, M. A.: “The utility business model and the future of computing services”. In: IBM Systems Journal, Vol. 43(1), S. 32–42, ISSN 0018-8670, 2004. 219
- [RBJS03] de Roure, D.; Baker, M. A.; Jennings, N. R.; Shadbolt, N. R.: Grid Computing: Making the Global Infrastructure a Reality, Ch. 3 The evolution of the Grid, S. 65–100. Series in Communications Networking and Distributed Systems, John Wiley & Sons, 2003. 39
- [Rein07] Reinicke, M.: Dienstausswahlverfahren im Grid. Sierke Verlag, Göttingen, ISBN 978-3-933893-86-4, Januar 2007. 33, 69, 70
- [RoWe03] Ross, J. W.; Westerman, G.: “Architecting New Outsourcing Solutions: The Promise of Utility Computing”. Research Article CISR WP Nr. 337 and Sloan WP No. 4458-03, Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, October 2003. 2, 16, 17
- [RoWe04] Ross, J. W.; Westerman, G.: “Preparing for utility computing: The role of IT architecture and relationship management”. In: IBM Systems Journal, Vol. 43(1), S. 5–19, ISSN 0018-8670, 2004. 1, 19, 45, 46, 89
- [Schi01] Schierenbeck, H.: Ertragsorientiertes Bankmanagement - Band 2: Risiko und integrierte Rendite-/Risikosteuerung. Gabler, Wiesbaden, 2001. 16
- [Schm05] Schmidt, K. D.: Versicherungsmathematik. Springer, Berlin, 2. Edition., ISBN-13: 978-3540290971, 2005. 107, 110, 111

- [Schn07] Schnizler, B.: Resource allocation in the Grid - A Market Engineering Approach. Ph.d. thesis, Universität Karlsruhe, Institut für Informationswirtschaft und -management (IISM), 2007. 11
- [ScTa70] Schmidt, J. W.; Taylor, R. E.: Simulation and Analysis of Industrial Systems. Richard D. Irwin, Homewood, Illinois, 1970. 10
- [Seib06] Seibold, H.: IT-Risikomanagement. Oldenbourg, 2006. 16, 285
- [SEVC<sup>+</sup>07] Streitberger, W.; Eymann, T.; Veit, D.; Catalano, M.; Giulioni, G.; Joita, L.; Rana, O. F.: "Evaluation of Economic Resource Allocation in Application Layer Networks - A Metrics Framework". In: A. Oberweis; C. Weinhardt; H. Gimpel; A. Koschmider; V. Pankratius; B. Schnizler (Hrsg.), eOrganisation: Service-, Prozess-, Market-Engineering: 8. Internationale Tagung Wirtschaftsinformatik - Band 1, WI 2007, Karlsruhe, Germany, February 28 - March 2, 2007, S. 477–494, Universitätsverlag Karlsruhe, ISBN 978-3-86644-094-4, 2007. 126, 147, 153, 158, 222, 287
- [SEZS<sup>+</sup>07] Streitberger, W.; Eymann, T.; Zini, F.; Schnizler, B.; Vo, H. T. K.: "Simulator Development - Annual Report Year 3". Bayreuther Arbeitspapiere zur Wirtschaftsinformatik 26, Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik, Bayreuth, URL <http://opus.ub.uni-bayreuth.de/volltexte/2007/370/> (Letzter Abruf: 15.05.2008), 2007. 129, 130
- [SGMM03] Sahai, A.; Graupner, S.; Machiraju, V.; van Moorsel, A.: "Specifying and monitoring guarantees in commercial grids through SLA". In: Proceedings of International Symposium on 3rd IEEE/ACM Cluster Computing and the Grid, S. 292–299, doi:10.1109/CCGRID.2003.1199380, 2003. 35
- [Shah05] Shah, S.: Handbook of Integrated Risk Management for E-Business: Measuring, Modeling, and Managing Risk, Ch. Integrated Risk Management, S. 33–62. J. Ross Publishing, 2005. 84
- [SHES<sup>+</sup>08] Streitberger, W.; Hudert, S.; Eymann, T.; Schnizler, B.; Zini, F.; Catalano, M.: "On the Simulation of Grid Market Coordination Approaches". In: Journal of Grid Computing, Vol. 6(3), Springer Netherlands, doi:10.1007/s10723-007-9092-6, ISSN 1570-7873 (Print) 1572-9814 (Online), September 2008. 67, 285

- [Simo90] Simon, H. A.: "Prediction and prescription in systems modeling". In: *Operations Research*, Vol. 38(1), S. 7–14, 1990. 119
- [SiMS03] Silvestrov, D.; Malyarenko, A.; Silvestrova, E.: "Stochastic Modeling of Insurance Business with Dynamical Control of Investments". In: *Theory of Stochastic Processes*, Vol. 9(1-2), S. 184–205, 2003. 142
- [Sinz08] Sinz, E. J.: "SOA und die bewährten methodischen Grundlagen der Entwicklung betrieblicher IT-Systeme". In: *Wirtschaftsinformatik*, Vol. 50(1), S. 70–72, Januar 2008. 31
- [Ski02] Skillicorn, D.: "Motivating Computational Grids". In: *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and Grid (CC-GRID'02)*, 2002. 40
- [SMSM<sup>+</sup>02] Sahai, A.; Machiraju, V.; Sayal, M.; van Moorsel, A. P. A.; Casati, F.: "Automated SLA Monitoring for Web Services". In: *DSOM '02: Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, S. 28–41, Springer-Verlag, London, UK, ISBN 3-540-00080-1, 2002. 85
- [SmTa98] Smith, R. E.; Taylor, N.: "A Framework for Evolutionary Computation in Agent-Based Systems". In: *Proceedings of the 1998 International Conference on Intelligent Systems*, ISCA Press, 1998. 75, 221
- [SREC<sup>+</sup>06] Streitberger, W.; Reinicke, M.; Eymann, T.; Catalano, M.; Giulioni, G.: "Economic Evaluation Framework of Resource Allocation Methods in Service-Oriented Architectures". In: *CEC-EEE '06: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, S. 50, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-2511-3, doi:<http://dx.doi.org/10.1109/CEC-EEE.2006.39>, 2006. 222
- [SSPS04] Sivasubramanian, S.; Szymaniak, M.; Pierre, G.; van Steen, M.: "Replication for Web Hosting Systems". In: *ACM Computing Surveys*, Vol. 36(3), S. 291–334, [http://www.globule.org/publi/RWHS\\_cs.html](http://www.globule.org/publi/RWHS_cs.html), 2004. 43
- [StGF02] Stoneburner, G.; Goguen, A.; Feringa, A.: "Risk Management Guide for Information Technology Systems". *Forschungsbericht NIST Special Publication*

- 800-30, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2002. 82
- [Stie07] Stiehler, A.: "Systematisierung und Handelbarkeit von IT-Dienstleistungen". Forschungsbericht, Berlecon Research, Oranienburger Str. 32, 10117 Berlin, Februar 2007. 2, 3
- [Stoc07] Stockinger, H.: "Defining the grid: a snapshot on the current view". In: J. Supercomput., Vol. 42(1), S. 3–17, ISSN 0920-8542, doi:<http://dx.doi.org/10.1007/s11227-006-0037-9>, 2007. 23
- [Stro05] Strong, P.: "Enterprise grid computing". In: Queue, Vol. 3(6), S. 50–59, ISSN 1542-7730, doi:<http://doi.acm.org/10.1145/1080862.1080877>, 2005. 2, 4, 48
- [Suth68] Sutherland, I. E.: "A futures market in computer time". In: Commun. ACM, Vol. 11(6), S. 449–451, ISSN 0001-0782, doi:<http://doi.acm.org/10.1145/363347.363396>, June 1968. 90
- [TaGü05] Tamm, G.; Günther, O.: Webbasierte Dienste - Technologien, Märkte und Geschäftsmodelle. Physica-Verlag, doi:10.1007/b138788, 2005. 35
- [Tali02] Talia, D.: "The Open Grid Services Architecture: Where the Grid Meets the Web". In: IEEE Internet Computing, Vol. 06(6), S. 67–71, ISSN 1089-7801, doi:<http://doi.ieeecomputersociety.org/10.1109/MIC.2002.1067739>, 2002. 32
- [TaTr03] Talia, D.; Trunfio, P.: "Toward a Synergy Between P2P and Grids". In: IEEE Internet Computing, Vol. 7(4), S. 96–95, ISSN 1089-7801, doi:<http://dx.doi.org/10.1109/MIC.2003.1215667>, 2003. 21, 70
- [TeSh03] Teitelbaum, B.; Shalunov, S.: "What QoS research hasn't understood about risk". In: RIPQoS '03: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS, S. 148–150, ACM, New York, NY, USA, ISBN 1-58113-748-6, doi:<http://doi.acm.org/10.1145/944592.944603>, 2003. 82
- [ThLi04] Thain, D.; Livny, M.: The Grid: Blueprint for a New Computing Infrastructure, Ch. Building Reliable Clients and Services, S. 285–318. Morgan Kaufmann Publishers, 2004. 2
- [TrHS04] Trieschmann, J. S.; Hoyt, R.; Sommer, D.: Risk Management and Insurance. South-Western College Pub, 12 Edition., 2004. 100

- [TuBB03] Turner, M.; Budgen, D.; Brereton, P.: “Turning Software into a Service”. In: *Computer*, Vol. 36(10), S. 38–44, ISSN 0018-9162, doi:<http://doi.ieeecomputersociety.org/10.1109/MC.2003.1236470>, October 2003. 2
- [Vari08] Varia, J.: “Building GrepTheWeb in the Cloud, Part 1: Cloud Architectures”. Forschungsbericht, Amazon Web Services, URL [http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1632\(LetzterAbruf:02.08.2008\)](http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1632(LetzterAbruf:02.08.2008)), July 2008. 2
- [VeBR06] Venugopal, S.; Buyya, R.; Ramamohanarao, K.: “A taxonomy of Data Grids for distributed data sharing, management, and processing”. In: *ACM Computing Surveys*, Vol. 38(1), S. 3, ISSN 0360-0300, doi:<http://doi.acm.org/http://doi.acm.org/10.1145/1132952.1132955>, 2006. 43
- [VHML07] Vázquez, T.; Huedo, E.; Montero, R. S.; Llorente, I. M.: “Evaluation of a Utility Computing Model Based on the Federation of Grid Infrastructures”. In: *Proceedings of 13th International Euro-Par Conference, Rennes, France, August 2007, Lecture Notes in Computer Science*, Vol. 4641/2007, S. 372–381, Springer, doi:[10.1007/978-3-540-74466-5\\_41](https://doi.org/10.1007/978-3-540-74466-5_41), 2007. 49
- [Voss07] Voss, K.: “Enhance Self-managing Grids by Risk Management”. In: *ICNS '07: Proceedings of the Third International Conference on Networking and Services*, S. 27, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-2858-9, doi:<http://dx.doi.org/10.1109/ICNS.2007.61>, 2007. 81
- [Wagn00] Wagner, F.: *Risk Management im Erstversicherungsunternehmen: Modelle, Strategien, Ziele, Mittel*. Verlag Versicherungswirtschaft, Karlsruhe, ISBN-13: 978-3884878309, 2000. 112, 113, 286
- [Watk07] Watkins, K.: *Human Development Report 2007/2008*. Palgrave Macmillan, New York, New York, USA, URL [http://hdr.undp.org/en/media/hdr\\_20072008\\_en\\_complete.pdf](http://hdr.undp.org/en/media/hdr_20072008_en_complete.pdf) (LetzterAbruf:20.05.2008), ISBN 978-0-230-54704-9, 2007. 146
- [Waxm88] Waxman, B. M.: “Routing of multipoint connections”. In: *IEEE Journal on Selected Areas in Communications*, Vol. 6(9), S. 1617–1622, doi:[10.1109/49.12889](https://doi.org/10.1109/49.12889), ISSN: 0733-8716, 1988. 129

- [Weil92] Weill, P.: "The role and value of information technology infrastructure : some empirical observations". Working papers 240, Massachusetts Institute of Technology (MIT), Sloan School of Management, URL <http://ideas.repec.org/p/mit/sloanp/2414.html>, May 1992. 1
- [Weis07] Weiss, A.: "Computing in the clouds". In: netWorker, Vol. 11(4), S. 16–25, ISSN 1091-3556, doi:<http://doi.acm.org/10.1145/1327512.1327513>, ISSN:1091-3556, December 2007. 2, 19
- [Wied03] Wiedemann, A.: Erfolgsfaktor Risiko-Management, Chance für Industrie und Handel, Methoden, Beispiele, Checklisten, Ch. Identifikation, Messung und Steuerung finanzwirtschaftlicher Risiken in Unternehmen, S. 1–19. Gabler, 2003. 15
- [WiHe06] Wilde, T.; Hess, T.: "Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung". Forschungsbericht 2, Institut für Wirtschaftsinformatik und Neue Medien, Ludwig-Maximilians-Universität München, Ludwigstr. 28, München, 2006. 10
- [WiMS04] Wilkes, J.; Mogul, J.; Suermondt, J.: "Utilification". In: Proceedings of the 11th ACM SIGOPS European Workshop, Leuven, Belgium, 19-22 September 2004. 45
- [Wlad04] Wladawsky-Berger, I.: The Grid: Blueprint for a New Computing Infrastructure, Ch. The Industrial Imperative, S. 25–34. Morgan Kaufmann Publishers, 2004. 2
- [Wolk07] Wolke, T.: Risikomanagement. Oldenbourg, 2007. 16, 85, 86, 88, 89, 285
- [WoRu03] Wolf, K.; Runzheimer, B.: Risikomanagement und KonTraG. Konzeption und Implementierung. Gabler, 2003. 82
- [WoWe02] Wojciechowski, R.; Weinhardt, C.: "Web Services und Peer-to-Peer-Netzwerke". In: Peer-to-Peer: Ökonomische, technische und juristische Perspektiven, S. 99–118, Springer Verlag, Heidelberg, Berlin, 2002. 33
- [XiPD04] Xie, M.; Poh, K.-L.; Dai, Y.-S.: Computing System Reliability: Models and Analysis. Springer, URL <http://www.springer.com/west/home/math?SGWID=4-10042-22-33305879-0>, 2004. 55



- [YaAm01] Yacoub, S. M.; Ammar, H. H.: "A Methodology for Architectural-Level Reliability Risk Analysis". Forschungsbericht HPL-2001-132, Publishing Systems and Solutions Laboratory, HP Laboratories, Palo Alto, May 2001. 17
- [YBAY<sup>+</sup>07] Yeo, C. S.; Buyya, R.; de Assunção, M. D.; Yu, J.; Sulistio, A.; Venugopal, S.; Placek, M.: The Handbook of Computer Networks, Vol. III, Ch. 143 - Utility Computing on Global Grids. John Wiley & Sons, ISBN: 978-0-471-78461-6, December 2007. 46
- [YuBu05] Yu, J.; Buyya, R.: "A taxonomy of scientific workflow systems for grid computing". In: SIGMOD Rec., Vol. 34(3), S. 44–49, ISSN 0163-5808, doi:<http://doi.acm.org/10.1145/1084805.1084814>, 2005. 88
- [ZaSa05] Zanikolas, S.; Sakellariou, R.: "A taxonomy of grid monitoring systems". In: Future Generation Computer Systems, Vol. 21(1), S. 163–188, ISSN 0167-739X, doi:<http://dx.doi.org/10.1016/j.future.2004.07.002>, 2005. 101
- [ZhTZ07] Zhao, J. L.; Tanniru, M.; Zhang, L.-J.: "Services computing as the foundation of enterprise agility: Overview of recent advances and introduction to the special issue". In: Information Systems Frontiers, Vol. 9(1), S. 1–8, ISSN 1387-3326, doi:<http://dx.doi.org/10.1007/s10796-007-9023-x>, March 2007. 1
- [ZlRo96] Zlotkin, G.; Rosenschein, J. S.: "Mechanisms for Automated Negotiation in State Oriented Domains". In: Journal of Artificial Intelligence Research, Vol. 5, S. 163–238, URL <http://dblp.uni-trier.de/db/journals/corr/corr9610.html#cs-AI-9610101>, October 1996. 72, 220
- [ZwEi07] Zweifel, P.; Eisen, R.: Versicherungsökonomie. Springer, Berlin, ISBN-13: 978-3540441069, Februar 2007. 6, 7, 15, 96, 97, 98, 99, 113, 114, 122, 286





# Lebenslauf

## Werner Streitberger

### Persönlich Daten

Geburtsdatum/-ort: 2. Dezember 1977 in Donauwörth

Familienstand: ledig

### Werdegang

02/2009 - heute Projektmanager für Cloud Computing Sicherheit am Fraunhofer Institut für sichere Informationstechnologie (SIT), Garching bei München

09/2004 - 12/2008 Wissenschaftlicher Mitarbeiter am Lehrstuhl für Wirtschaftsinformatik der Universität Bayreuth

09/2004 - 12/2008 Dozent für Wirtschaftsinformatik an der Verwaltungsakademie Nürnberg

04/2004 - 08/2004 Informatiker bei der Kratzer Automation AG, Unterschleißheim

08/1988 - 03/2004 Studium der Informatik an der Technischen Universität München

09/1988 - 06/1997 Abitur am Gymnasium Donauwörth

Das hinter dem Begriff Grid-Computing stehende Konzept beschreibt verschiedene Lösungsansätze zur Umsetzung eines dynamischen Bezugs von IT-Ressourcen und Diensten innerhalb eines Unternehmens und über Unternehmensgrenzen hinweg. Jedoch gehen mit dem Einsatz von Grid-Computing-Systemen technische Risiken einher, deren Ursachen meist auf fehlerhafte Kommunikation und/oder auf den Ausfall von Ressourcen eines Standorts zurückzuführen sind. Dies schränkt die Zuverlässigkeit und Verfügbarkeit der Systeme ein und erfordert den Einsatz von Verfahren zur Behandlung dieser Risiken, da die Benutzer zuverlässige Ressourcen von einem Grid-Computing-System erwarten.

Der Arbeitsbericht identifiziert technische und ökonomische Risiken in Grid-Computing-Systemen und untersucht sie hinsichtlich ihrer Versicherbarkeit. Die in dieser Arbeit durchgeführten Simulationen zeigen, dass die Einführung einer Versicherung einen nachweislichen Einfluss auf die analysierten technischen und ökonomischen Kennzahlen haben und eine Verbesserung der Verfügbarkeit und Zuverlässigkeit der untersuchten Utility-Grid-Systeme erwarten lassen.