

Schütte, Reinhard; Kari, Mohamed

Article — Published Version

Cloud Enterprise Systems – State of the Art und Herausforderungen für Unternehmen

HMD Praxis der Wirtschaftsinformatik

Provided in Cooperation with:

Springer Nature

Suggested Citation: Schütte, Reinhard; Kari, Mohamed (2025) : Cloud Enterprise Systems – State of the Art und Herausforderungen für Unternehmen, HMD Praxis der Wirtschaftsinformatik, ISSN 2198-2775, Springer Fachmedien Wiesbaden GmbH, Wiesbaden, Vol. 62, Iss. 1, pp. 5-24, <https://doi.org/10.1365/s40702-025-01141-3>

This Version is available at:

<https://hdl.handle.net/10419/319053>

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.



<http://creativecommons.org/licenses/by/4.0/deed.de>



Cloud Enterprise Systems – State of the Art und Herausforderungen für Unternehmen

Reinhard Schütte  · Mohamed Kari

Eingegangen: 14. September 2024 / Angenommen: 2. Januar 2025 / Online publiziert: 14. Januar 2025
© The Author(s) 2025

Zusammenfassung Die Softwarehersteller propagieren für Enterprise Systems das Cloud Computing. So sollen sämtliche Komponenten von Enterprise Systems langfristig nur noch als Software as a Service genutzt werden. Aufgrund der Abhängigkeit vieler Anwendungsunternehmen von den wenigen Anbietern (Microsoft, SAP, Oracle, Infor, Sage, Salesforce etc.) stellt sich die Frage, welchen Entwicklungen die Applikationsarchitekturen und IT-Prozesse unterworfen sein werden. Ausgehend von einer Entfaltung dessen, was unter Enterprise Systems verstanden wird und welche Präsuppositionen den nachfolgenden Abschnitten unterliegen, wird der State of the Art heutiger Enterprise Systems kurz skizziert. Es wird eine Referenzarchitektur entworfen, in der die Services der drei großen Hyperscaler verortet werden. Zuletzt werden die strategischen, prozessualen und applikationsmäßigen sowie systemtechnischen Implikationen des Cloud Computing für die Anwendungsunternehmen aufgezeigt. Der Beitrag verdeutlicht, dass die technologischen Paradigmen und Systems-Engineering-Vorgehensweisen im Cloud Computing die zukünftige Einführung von Enterprise Systems und auch deren Betrieb nachhaltig verändern werden. Der Beitrag schließt mit einem Ausblick auf zukünftige Herausforderungen für die Forschung und die betriebliche Praxis.

Schlüsselwörter Enterprise Systems · Cloud Computing · SaaS · PaaS · Applikationsarchitektur · ERP-Einführung

✉ Reinhard Schütte · Mohamed Kari
Fakultät für Informatik, Universität Duisburg-Essen, Essen, Deutschland
E-Mail: reinhard.schuette@icb.uni-due.de

Cloud Enterprise Systems – State of the Art and Challenge for Businesses

Abstract Software manufacturers are propagating cloud computing for enterprise systems; in the long term, it can be assumed that all components of enterprise systems will only be provided as software as a service. Due to the dependence of many application companies on the few oligopolists (Microsoft, SAP, Oracle, Infor, Sage, Salesforce, etc.), the question arises as to how application architectures and IT processes will develop accordingly. Starting with an explanation of what is meant by enterprise systems and which presuppositions underlie the following sections, the state of the art of today's enterprise systems is briefly outlined. A reference architecture for cloud computing is outlined, in which the different services of the three major hyperscalers are situated. Building on this, the strategic, process-related, application-related and system-related implications of cloud computing for domain companies are highlighted. The article makes it clear that the technological paradigms and systems engineering approaches in cloud computing will permanently change the future implementation of enterprise systems and their operation. The article concludes with an outlook on future challenges for research and operational practice.

Keywords Enterprise Systems · Cloud Computing · SaaS · PaaS · Application Architecture · ERP Implementation

1 Problemstellung und Entfaltung der Präsuppositionen

Enterprise Systems werden gemeinhin nicht mehr als ein monolithisches System verstanden. Stattdessen wird in der Literatur die Einschätzung geteilt, dass derartige Systeme sich aus – je nach Autor – fünf (Gronau 2021) bis sechs (Schütte 2024) unterschiedlichen Systemen zusammensetzen, die abweichende funktionale und technische Aufgaben abdecken: Enterprise Resource Planning (ERP), Supplier Relationship Management (SRM), Supply Chain Management (SCM), Customer Relationship Management (CRM), Business Intelligence (BI) und nach dem vorliegenden Verständnis optional auch ein Shop-System (Schütte 2023, 2024).

Dabei sind Enterprise Systems dadurch geprägt, dass ihre Komponenten Standard-Anwendungssysteme sind, die einen hohen Integrationsgrad anstreben (Leyh und Wendt 2018; Gronau 2021; Kurbel 2021). Die mit dem Aufkommen des Cloud Computing induzierten technologischen Herausforderungen für Enterprise Systems sind erheblich, wie der vorliegende Beitrag darstellen wird. Die erforderlichen Änderungen in bestehenden Systemen haben ihre Ursache möglicherweise auch in den – bezüglich der Cloud-Technologien – geringen technologischen Fortschritten, die die Standardsoftwarehersteller bisher realisiert haben. Dabei wird in diesem Beitrag unter Cloud Computing (Hentschel et al. 2018; Hentschel und Leyh 2016) nicht nur der Verkauf einer Lösung als Software as a Service (SaaS) verstanden. Stattdessen unterstellt dieser Beitrag, dass mit dem Begriff eines „Cloud-basierten Systems“ die Realisierung von Eigenschaften des Cloud-Native oder Serverless Computing

gemeint ist, welche jedoch durch Enterprise Systems von heute nicht oder oft nur sehr begrenzt erfüllt werden.

Daraus ergeben sich die beiden zentralen Problembereiche, die dieser Beitrag adressiert:

- Welche Veränderungen in der Architektur von Enterprise Systems sind zu erwarten und wie wirkt sich dies auf die Evolution der Architekturen in den Anwendungsunternehmen aus?
- Welche Herausforderungen und Risiken ergeben sich aus den veränderten architektonischen Gegebenheiten für die im Einsatz befindlichen Applikationsarchitekturen?

Der Diskussion der Problembereiche seien dabei folgende Präsuppositionen vorangestellt, um dem geneigten Leser die Basis der Bewertung nachvollziehbar zu entfalten:

- Ein Enterprise System ist nicht nur ein monolithisches System, sondern die Komposition von mehreren Systemen, die auch nicht nur von einem Softwarehersteller stammen muss (*Architektur- und Heterogenitätspräsupposition*).
- Die ökonomische und auch technische Bedeutung des Cloud-Native Computing für die Softwarehersteller führt dazu, dass diese die Anwendungsunternehmen technologisch in die Cloud „treiben“ und alternative Technologieangebote geringer und vor allem weniger leistungsfähig werden (*Cloud-Dominanzpräsupposition*).
- Ein Enterprise-System-Hersteller dominiert die Anwendungssystemlandschaft von Unternehmen im inhaltlichen und auch technologischen Sinne (*betriebswirtschaftliche und technologische Herstellerdominanz-Präsupposition*).

Da Enterprise Systems von den Softwareherstellern betriebswirtschaftlich vorge-dacht werden, ist der Lösungsraum für das einführende Unternehmen für wesentliche wertschöpfende Prozesse betriebswirtschaftlich mit ganz wenigen Ausnahmen zwingend unterspezifiziert (*Abbildungsunmöglichkeitspräsupposition*). Es gibt lediglich zwei Fälle, in denen dieser wichtige Aspekt geringer ausgeprägt ist: Entweder Unternehmen sind an der Entwicklung der Lösungen beim Hersteller beteiligt oder die Anforderungen der Unternehmen sind tatsächlich im Standard vollumfänglich abbildbar. Die Erkenntnisse aus der Einführung von Standardsoftware belegen dabei aber, dass die Individualisierungsanforderungen erheblich sind und die letztgenannte Annahme unrealistisch ist. Auch die empirischen Ergebnisse belegen, dass in den Wertschöpfungsbereichen von Unternehmen eine solche Situation nicht vorliegt (Buxmann und König 1997; Brehm et al. 2002; Hansen et al. 2023).

2 State of the Art heutiger Enterprise Systems und deren Einführung

Enterprise Systems werden technologisch und betriebswirtschaftlich von den jeweiligen Softwareherstellern dominiert. Auch wenn es in unterschiedlichen Domänen viele Anbieter von Standardanwendungssystemen gibt, so dominieren aus einer Marktanteilsperspektive wenige, große Anbieter den Markt: Oracle, SAP, Microsoft

und Salesforce; Auch Sage und Workday sind in diesem Zusammenhang noch zu erwähnen (Sarferaz 2022, 2023). Unabhängig von dem Standardsoftwarehersteller sind Applikationsarchitekturen entstanden, in deren Kern die Enterprise-Systemskomponenten dominieren, auch wenn weitere wichtige Applikationen existieren. Die Kommunikation zwischen und die Integration mit diesen Komponenten bildet dabei die Basis oder „das Backbone“ der Applikationsarchitekturen (Olson und Kesharwani 2010). Dabei gibt es mindestens zwei Fundamentalaspekte, die in der Literatur angedeutet werden, allerdings in ihrer Wirkmächtigkeit für die Applikationsarchitekturen nicht ernsthaft diskutiert werden.

Erstens hat sich in der Literatur die Einschätzung etabliert, dass die Rede von „generischen Standardsystemen“ (Klaus et al. 2000; Gronau 2021) ein zielführendes Leitbild für ERP-Systeme sein sollte. Die technische Realisierung dieser Anforderungen durch das Anwendungsunternehmen (oder deren Implementierungspartner) in der Software wird in der Literatur mehrheitlich als Customizing bezeichnet (Gronau 2021; Kurbel 2021; Klaus et al. 2000), wobei dies im engeren Sinne eher eine Parametrisierung darstellt (Schütte 2024). Der Umfang an Parametern ist dabei erheblich (vgl. Dittrich und Vaucouleur 2008) und diese Parametrisierung unterstellt, dass die vergrößerte Code-Basis mit einer darüberliegenden Parametrisierungsschicht Vorteile mit sich bringt. Damit werden implizite Annahmen getroffen, die im Widerspruch zur Abbildungsunmöglichkeitspräsupposition aus dem vorhergehenden Abschnitt dieses Beitrags stehen könnten. Ein solches Zielbild von Standardsoftware unterstellt die Möglichkeit hohe Standardisierungsgrade dadurch zu erzielen, dass unterschiedliche Anforderungen von Unternehmen in der Standardsoftware durch eine Erweiterung der Codebasis realisiert werden können oder es eine – im Wortsinn ernsthafte – generische Ableitungsmöglichkeit des Softwarecodes aus Anforderungen der Unternehmen gibt. Hier wird die Annahme nicht geteilt, denn die Anpassungsmaßnahmen in der Standardsoftware sind den zitierten empirischen Belegen zufolge immens und haben sich im Zeitablauf nicht verringert (Dittrich und Vaucouleur 2008). Hierzu sei auch exemplarisch der ehemaligen CIO von Aldi Nord zitiert: „Man ist schlussendlich hingegangen und hat sehr auf die bestehenden Bedarfe der jeweiligen Fachbereiche bzw. Länderspezifika gehört und hat eben nicht standardisiert, sondern in hohem Maße individualisiert. Wenn man das mal so vergleicht, sich das Coding mal ansieht, sind ungefähr 70 % der Codestrecken, die wir heute nutzen, individueller Natur. Das heißt, tatsächlich selber entwickelt bzw. modifiziert, im SAP-Sprachgebrauch, und nur ungefähr 30 % der Codestrecke sind von der Software vom Hersteller übernommen“ (Hoepfner 2021). Ungeachtet der enormen Individualisierung von Standardsystemen gilt es die Bedeutung von Standardkomponenten mit Ausnahme der BI-Komponente auf zwei Ebenen zu beachten. Zunächst ist aus einer Metaebene die konzeptionelle und terminologische Bedeutung derartiger Systeme erheblich. Es sind grundsätzliche betriebswirtschaftliche Konzeptionen, die die Art der Aufgabenkoordination festlegen. Beispielsweise determiniert die Kopplung des Rechnungswesens mit der Materialwirtschaft oder dem Vertrieb in einem SAP-System die „Sicht auf die ökonomische Wirklichkeit“. Auf einer Objektebene ist zu beachten, dass der strukturelle und prozessuale Rahmen des Unternehmens von Standardsoftwareherstellern vorgegeben wird und stets auch limitierend wirkt. Dabei sind die meisten aktuell im Einsatz befindlichen Systeme

auf Datenmodellen basierend, die den Wunsch nach einer integrierten Datenhaltung durch Inkaufnahme als starr zu bezeichnender Datenstrukturen realisiert haben. Beispielsweise ist das alte SAP-Datenmodell aus dem R/2-System, in dem die zentralen Organisationseinheiten des Systems repräsentiert sind, noch immer in großen Teilen aktuell. Mehr als 50 Jahre sind diese Datenstrukturen und die darauf basierenden Algorithmen stabil geblieben.

Zweitens wird auch die technologische Basis in der Diskussion sehr selten problematisiert. Dazu gehört die Art der Datenhaltung und des Datenbanksystems, der Programmiersprache und der Entwicklungsumgebung und nicht zuletzt die Softwarearchitektur und für den Betrieb besonders bedeutend das Deployment und die Softwarelogistik. Letztlich dürfte das Technologiealter der funktionalen Komponenten sehr hoch sein, was auch eine technologische Last für die Anwendungsunternehmen bedeutet. Aufgrund der Vielzahl von Stammkunden wirken diese technologischen Gegebenheiten wiederum ganz im Sinne des Innovator's Dilemma von Christensen (1997) als Problem für eine wirkliche Neuentwicklung von ERP, CRM, SCM oder SRM-Komponenten. Im Ergebnis sind die heutigen Applikationsarchitekturen damit einerseits durch die betriebswirtschaftliche Prägung der wesentlichen Komponenten eines Enterprise Systems limitiert und ihrer Weiterentwicklung eingeschränkt. Andererseits wird der Betrieb der Applikationen und deren Weiterentwicklung aus technischer Sicht durch die Standardsoftwarehersteller vorgegeben. Der letztgenannte Aspekt erscheint vordergründig nicht so bedeutend zu sein, ist aber für die Organisation der IT und auch die Einführung der Systeme von hoher Bedeutung. Die Kompetenzbasis der IT-Ressourcen lässt sich nicht so einfach verändern, ist sie doch in den handelnden Personen begründet. Deren Wissen ist an die unterschiedlichen Ecosysteme technisch und auch ökonomisch gebunden, übergreifende und tiefe Domänen- und Technologiekenntnisse sind überaus selten vorhanden.

Folgerichtig werden von den Standardsoftwareherstellern in Abhängigkeit von dem Gegenstand der Einführung (den Komponenten eines ES, also BI, SCM, SRM, ERP, CRM und Shop-System) und dem Deployment-Modell in der Cloud unterschiedliche Vorgehensmodelle mitsamt der die Einführung unterstützenden Tools für das Application Lifecycle Management offeriert (Aral et al. 2024), die auch noch anhand der verfolgten Zwecke des Vorgehensmodells variiert werden (z. B. Produktentwicklung eines Template-Systems, Rollout eines Template-Systems, Release-Wechsel). Zusätzlich bieten die im Ecosystem (Wulfert 2024) der Softwarehersteller befindlichen Implementierungspartner i. d. R. adaptierte, mitunter auch eigene Vorgehensmodelle, um sich von Wettbewerbern zu differenzieren. Die Partner im Ecosystem reichen von global agierenden Systemhäusern über Schulungsanbieter bis hin zu Freelancern. Die bei diesen Partnern im Fokus stehenden Vorgehensmodelle lehnen sich in der Regel an eher klassische sequenzielle oder iterative oder an agile Vorgehensmodelle an. Dabei ist die Art des Vorgehens bei einer Standardsoftware nicht so frei wählbar, wie dies bei einer Individualsoftware der Fall ist. Die Vorgabe der betriebswirtschaftlichen Inhalte in Form von Datenstrukturen und Prozessen führt ebenso wie die Art der Realisierung dergleichen in Softwarecode dazu, dass das Vorgehen inhaltlichen und technologischen Restriktionen unterworfen ist. Damit wird auch der Einsatz „echter“ agiler Vorgehensweisen in komplexen transak-

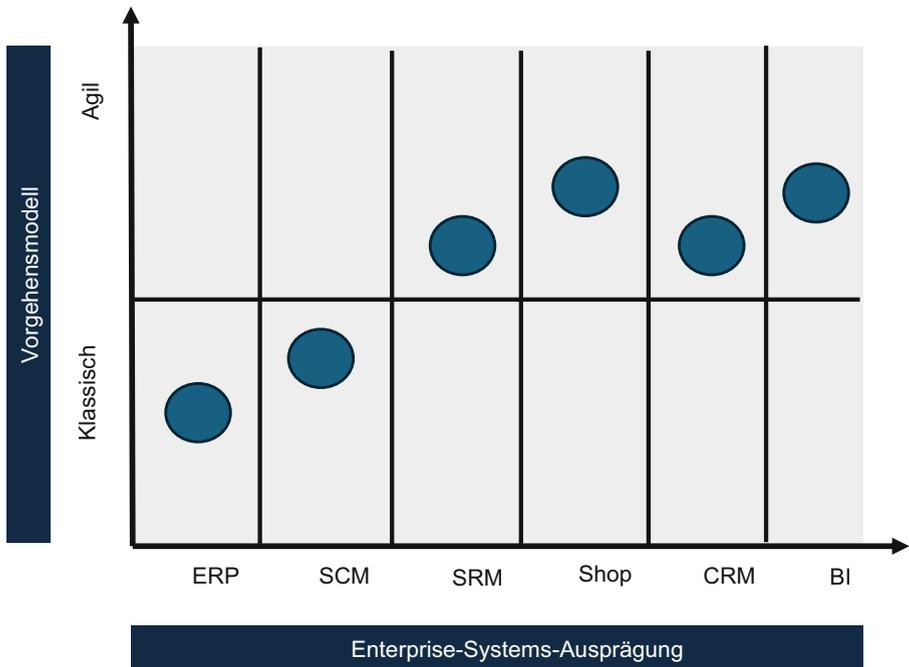


Abb. 1 Dominante Vorgehensmodelltypen bei Komponenten von ES-Systemen. Die Grafik trägt das dominante Vorgehensmodell gegen die Enterprise-Systems-Komponenten auf. ERP und SCM werden weiterhin weitgehend im klassischen Modell entwickelt, während SRM, Shop, CRM und BI-Systeme vorrangig dem agilen Vorgehensmodell folgend entwickelt werden. (Eigene Darstellung)

tionalen Systemen innerhalb eines Enterprise Systems selten praktiziert. Abb. 1 gibt einen Eindruck wieder, wie wenig verbreitet agile Vorgehensmodelle nach Einschätzung der Autoren entgegen gegenteiligen Ankündigungen der Hersteller in realen Projekten im Enterprise-Systems-Kontext sind.

Dabei sei vor allem darauf hingewiesen, dass ein SCM- oder ein ERP-System weitaus stärker transaktional komplexere betriebswirtschaftliche Prozesse realisiert, als dies in den anderen Systemen der Fall ist. Aufgrund der damit verbundenen Integrationsproblematik in diesen Systemen werden daher auch klassische Vorgehensmodelle erforderlich.

Unabhängig von dem Vorgehensmodelltyp werden „Referenz-Vorgehensmodellartig“ die fünf Phasen Projektvorbereitung, Anforderungsdefinition (oder in der Sprache der Praxis auch als Business Blueprint bezeichnet), Realisierung, Produktionsvorbereitung und Go Live unterschieden (Kurbel 2021). In diesen Phasen werden Aufgaben mit Problemlösungstechniken gelöst und mittels Darstellungstechniken die Lösungen dokumentiert (vgl. zum Verständnis von Phasen, Aufgaben, Darstellungs- und Problemlösungstechniken auch die Ausführungen bei Teubner 1999). Dabei sind die im Rahmen der Phasen wahrzunehmenden Aufgaben von der spezifizierten Domäne und der technologischen Basis abhängig. Die realen Einführungsfälle werden durch die Vorgehensmodelle des jeweiligen Standardsoftwareherstellers geprägt (sei es SAP, Oracle, Microsoft, Salesforce, o. a.), die i. d. R. durch hersteller-

spezifische Tools für das Application Lifecycle Management unterstützt werden. Die beiden zentralen Phasen, die Anforderungsdefinition und die Realisierung, weisen dabei standardsoftwarespezifische Aspekte auf:

- Die *Anforderungsdefinition* ist nicht frei, sondern folgt je nach Vorgehensmodell und Ziel der Systemeinführung den in der Software realisierten betriebswirtschaftlichen Vorgaben des Standardsoftwareherstellers (z. B. Greenfield und Brownfield der Systeme/Organisation und Ausrichtung am Standardsoftwaresystem als systemtechnischem Ziel oder betriebswirtschaftlich sinnvolle Ausrichtung an den Anforderungen des Unternehmens). Dabei herrscht in der Praxis zu Beginn des Projektes die Vorgabe vor, dass so viel Standard wie möglich und so wenig Individualisierung wie nötig realisiert werden soll. Es werden dabei unterschiedliche Arten von Vorlagen verwendet, wie auf konzeptioneller Ebene Prozessmodelle oder auch eingerichtete Systeme mit Stammdaten und konfigurierten Prozessmodellen (z. B. SAP Market Standard).
- Bei der *Realisierung* folgt diese den technischen Rahmenbedingungen der heutigen Systemkomponenten, die i. d. R. einen historischen Hintergrund haben und damit auch die eingesetzten Paradigmen und Services begrenzen.

Diese beiden Aspekte werden im nachfolgenden Kapitel aufgegriffen. Für die spätere Analyse der Einführungsveränderung durch neue Architekturprinzipien in einem Cloud Native Computing ist die Kenntnis beider Aspekte fundamental.

3 Architekturen im Cloud Computing

3.1 Cloud-Computing-Architekturen aus der Perspektive der Cloud Provider

3.1.1 Überblick

Cloud Computing verleiht Hardware diejenigen Skalierbarkeits- und Automatisierungseigenschaften, die davor nur Software vorbehalten waren. Das Bereitstellen eines Computers, zehntausender Computer, eines globalen Netzwerks von Satelliten-Bodenstationen oder sogar eines Quantencomputers erfordert nicht mehr als eine Internetverbindung und das Ausführen einiger Codezeilen (und einer Kreditkartennummer). Von der Bereitstellung einer Webseite für eine lokale Schülerzeitung bis hin zum Betrieb von Netflix können alle erforderlichen Systemressourcen (Computer, Netzwerk, Datenbanken, Benutzerauthentifizierungsdienste oder GPU (Graphics Processing Units)-Cluster, die für das Training im Rahmen von Machine-Learning-Workloads benötigt werden) im gewünschten Layout in Software definiert und vollautomatisch bereitgestellt werden (Armbrust et al. 2009; Barroso et al. 2003; Waibel 2020).

Das Cloud Computing soll für die Anwendungsunternehmen erheblich vereinfachte und beschleunigte Entwicklungs- und Betriebsprozesse ermöglichen. Dabei ist die Realisierung der Ziele nicht nur abhängig von der angebotenen Servicebreite und -tiefe der Cloud-Anbieter, sondern vor allem auch von den Fähigkeiten des einsetzenden Unternehmens. Das zentrale Wertversprechen der drei großen Cloud-

Hyperscale – Amazon, Alphabet und Microsoft – besteht darin, eine gut gestaltete Provisioning API (Application Programming Interface) anzubieten, die Kunden-Requests entgegennimmt und die Ressourcen entsprechend automatisch bereitstellt (Fehling und Leymann 2018). Im Kontext dieses Beitrags kommt der Produkt-Portfolio-Architektur der Cloud Provider eine besondere Rolle zu, um ihre Wirkung auf die Erstellung und den Betrieb von Enterprise Systems aus einer Architektorentwicklungsperspektive beurteilen zu können.

3.1.2 *Serviceniveau-orientierte Einordnung*

Die Architektur der Cloud-Provider wird kanonischerweise in Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), und Software-as-a-Service (SaaS) untergliedert, unter denen sich einzelne als Produkt angebotene Services subsumieren lassen (Anselmi et al. 2017).

In diesem Beitrag werden die IaaS-Komponenten und PaaS-Komponenten anhand des Kriteriums der „Self-Hostability“ unterschieden. Es stellt sich die Frage, ob ein Service durch einen Cloud-Verwender selbst auf anderen Low-Level-Cloud-Ressourcen in vergleichbarer Weise gehostet werden könnte. Eine IaaS-Komponente kann nicht selbst gehostet werden, wobei ein PaaS-Komponente durch Inanspruchnahme anderer Cloud-Ressourcen selbst betrieben werden könnte. Ein NFS (Network File Service)-kompatibler File Server (etwa AWS EFS, Azure Files, oder Google Cloud Filestore) ebenso wie ein S3-kompatibler Object Store (etwa AWS S3, Azure Blob Storage, oder Google Cloud Storage) könnte etwa in Eigenregie durch Verwendung virtueller Maschinen, daran angehängter virtueller Festplatten und Installation eines NFS-Kernel-Server selbst betrieben werden. In der Diktion dieses Beitrags wäre es damit eine PaaS-Komponente. Im Gegensatz dazu ist eine virtuelle Festplatte als IaaS-Komponente einzuordnen, da sie nicht selbst durch Software-Installation betrieben werden kann, sondern – trotz aller dazwischenliegenden Virtualisierungsebenen – durch den Cloud Provider in Korrespondenz zu physischem Festplattenspeicher bereitgestellt werden muss.

Darüber hinaus werden in diesem Beitrag PaaS-Komponenten und SaaS-Komponenten danach unterschieden, ob die vornehmliche Nutzergruppe der Komponenten Softwareentwickler und systemtechnische Nutzer sind oder eine Endnutzer-Schnittstelle vorliegt. Dieses Kriterium dient als Indikator für die Positionierung im digitalen „Wertschöpfungsprozess“ eines Systems.

Aus softwaretechnischer und softwarearchitektonischer Sicht ist die Frage, ob die darunterliegenden Schichten als IaaS und PaaS bereitgestellt werden oft von geringer Bedeutung. Ob etwa eine Postgres-Datenbank auf einer virtuellen Linux-Maschine unmittelbar im Hoheitsgebiet des Cloud-Users installiert wurde, oder auf einer virtuellen Linux-Maschine beim Cloud Provider installiert ist, erscheint für den Anwendungsserver kaum unterscheidbar. In beiden Fällen genügen IP-Adresse und Port zur ODBC-Schnittstelle der Datenbank, um SQL-Abfragen auszuführen.

Wengleich es kaum Studien zur Wirkung des Cloud Computing auf die Softwarequalität gibt, wird angenommen, das auch jenseits der Bewertung der Skalierbarkeit als einer Qualitätsdimension von Software die Qualität weiterer Evaluationskriterien steigt. Ebenso gibt es kaum Daten dazu, welche ökonomischen oder qualitativen

Effekte die Entscheidung von PaaS vs IaaS prägen. Jedoch wird insbesondere bei Domänenunternehmen vielfach die Bereitstellungsgeschwindigkeit und die Qualität durch den Einsatz von PaaS-Produkte anstelle von IaaS-Produkten steigen. So setzt die Verwendung von PaaS-Komponenten in der Cloud voraus, dass diese mittels der API des Cloud Providers vollautomatisch provisioniert werden. Während selbstentworfenen IaaS-Systemarchitekturen durch Veränderungen, etwa mittels interaktiver Befehle über die Cloud Provider GUI (Graphical User Interface) oder CLI (Command Line Interface) in undokumentierte oder gar unreplizierbare Systemzustände führen, können, machen es jedoch die Infrastructure-as-Code-Schnittstellen der Cloud Provider einfach, Services selbstdokumentierend, versioniert, und deklarativ und damit auf Knopfdruck reproduzierbar bereitzustellen. Zudem dürfte das eigene Shell-Script dem Shell-Script des Cloud-Providers zur Konfiguration des Datenbank-Servers, aller IP-Adressen, etc. unterlegen sein. Weiterhin kann unmittelbar auf die Integrationsmöglichkeiten mit VPC (Virtual Private Cloud), Secrets Manager und API Gateways zurückgegriffen werden. Traditionell ist vielen Entwicklungsprojekten zu eigen, manuelle Veränderungen an diesen Architekturkomponenten vorzunehmen. Dies verhindert wiederum die automatisierte Bereitstellung des Gesamtsystemzustandes, ein Beispiel für die obige Forderung, die IT-Prozesse an den Cloud-Möglichkeiten auszurichten, um den potenziellen Wirtschaftlichkeitsvorteil nicht zu einem -nachteil werden zu lassen. Die PaaS-Bereitstellung forciert eine Automatisierbarkeit und Reproduzierbarkeit der Infrastruktur-Bereitstellung, die zu positiven Effekten in der Softwarequalität führt. Diese ökonomischen Mehrwerte dürften Zusatzaufwendungen, etwa bei der Erstinstallation, überkompensieren. Es wird deutlich, dass den PaaS-Komponenten dadurch eine besondere Bedeutung für die effiziente Implementierung von Applikationen durch Domänenunternehmen zukommt. Diesen Vorteilen stehen jedoch Abhängigkeiten vom PaaS-Anbieter und Quasi-Lizenzgebühren entgegen, die durch höhere Kosten für die PaaS-Komponente gegenüber dem reinen IaaS-„Materialeinsatz“ entstehen.

Die Services auf den Ebenen IaaS, PaaS und SaaS sind je nach betrachteter Ebene, aus Sicht der Betreiber von Cloud-Plattformen, als homogen bzw. heterogen bei den jeweiligen Anbietern zu bewerten. Die IaaS-Komponenten werden durch physische Hardware und einige weitgehend zeitlich persistente Kernkonzepte der Informatik repräsentiert. Demzufolge ist auch eine ausgeprägte Austauschbarkeit zwischen den unterschiedlichen Anbietern gegeben. Noch immer scheinen in diesem Bereich erhebliche Skalierungseffekte zu existieren, allerdings weniger qualitative Differenzen. Im Gegensatz dazu weisen die PaaS-Angebote durch die Veränderbarkeit der Softwaresysteme Differenzierungspotenziale auf. Das SaaS-Angebot der Hyperscaler ist angesichts der Domänenorientiertheit von Geschäftssoftware gering ausgeprägt und wird weitgehend über Marketplaces und Partner abgebildet. Diese stellen dann ihre Systeme auf den PaaS-Ressourcen des jeweiligen Cloud Providers bereit.

3.1.3 Referenzarchitektur für die Service(-klassen) von Cloud-Anbietern

Bei einer serviceorientierten (aus Sicht der Hersteller auch funktionsorientierten) Betrachtung tritt die diskutierte IaaS-/PaaS-Frage zugunsten der Betrachtung von Technologie-Komponenten und deren Schnittstellen zurück. Ob der Datenbankser-

ver vom gleichen Cloud-Provider, einem Second-Tier-Provider, im Self-Hosting auf den Ressourcen eines anderen Hyperscalers, oder auf einem Bare-Metal-Server im eigenen Rechenzentrum liegt, ist technisch nicht wesentlich, solange nur die Schnittstelle und die Leistungsfähigkeit den Anforderungen entsprechen.

Die serviceorientierte Perspektive orientiert sich an den funktionalen Architekturkomponenten der Cloud-Anbieter, die bei den Applikationsarchitekturen der Unternehmen zum Einsatz kommen. So unterscheiden sich die Cloud-Angebote von AWS, Microsoft Azure und Google in vielen Details aufgrund unterschiedlicher Strategien. Nach Einschätzung der Autoren verfolgt etwa AWS die Strategie, für jeden Kunden „das Richtige“ anbieten zu wollen, so dass eine Form des „Wildwuchses“ mitsamt von Angebotsüberschneidungen oder logischen Dopplungen von Services entsteht. Google hingegen scheint eine Strategie der Opinonated Structure zu verfolgen, die die Integration von Services durch das Vordenken des besten Wegs anstrebt. Dies mag aber auch nur die subjektive Bewertung der Autoren sein, da hier nicht der wissenschaftliche Raum für eine derartige Erörterung zur Verfügung steht. Ungeachtet der Unterschiede bei einzelnen Services lassen sich wiederkehrende Service- oder funktionale Komponenten generalisieren. Abb. 2 gibt eine Referenz-Architektur wieder, die ihre Struktur aus der Analyse der Produktlandschaft der Hyperscaler erhalten hat.

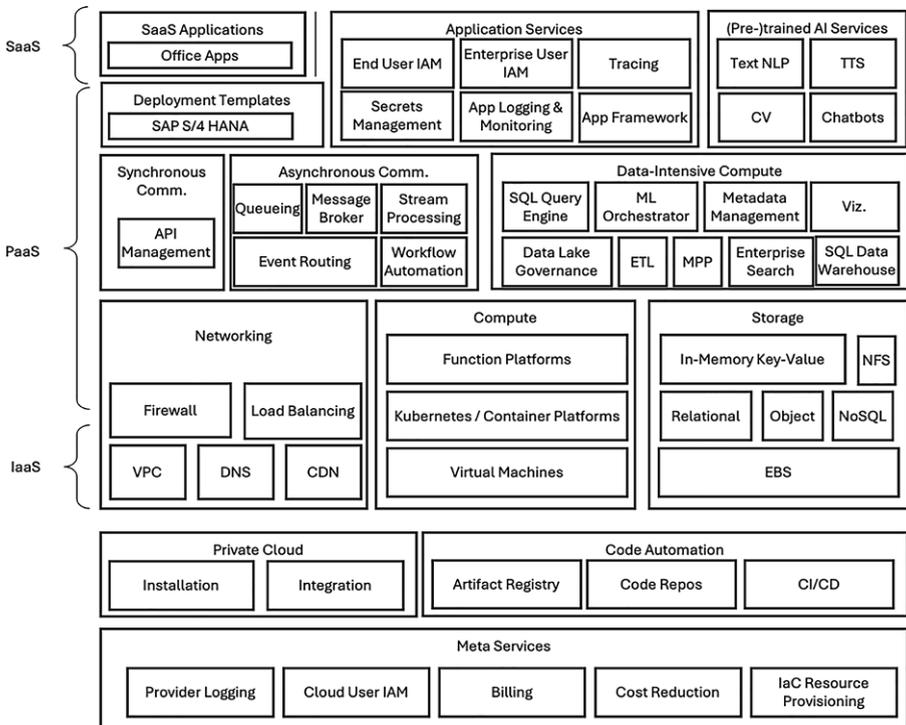


Abb. 2 Service-(klassen)orientierte Betrachtung von Cloud-Architekturen aus Perspektive der Cloud-Anbieter. (Eigene Darstellung)

Die Referenzarchitektur gestattet nun die Gegenüberstellung der Services der einzelnen Anbieter AWS, Azure und Google Cloud. Die entsprechende Tabelle ist aufgrund ihres Umfangs im Anhang als Tab. 1 dargestellt, um dem geeigneten Leser eine Transparenz über die Vergleichbarkeit und auch Unterschiedlichkeit der einzelnen Services zu ermöglichen. Zudem kann Tab. 1 auch als eine Bestätigung für die Konstruktionsadäquanz der Referenzarchitektur verstanden werden (Schütte 1998). Die Services der Referenzarchitektur und die Einordnung der Services der Hyperscaler in die Architektur werden hier nicht weiter entfaltet.

3.2 Cloud-Computing-Architekturen aus Perspektive der Enterprise-Systems-Hersteller

Die Angebote der großen, führenden Hyperscaler dominieren und repräsentieren die technische Architektursicht der Enterprise-Systems-Anbieter. Allerdings bestehen nicht unerhebliche Wechselwirkungen, auch hinsichtlich der ökonomischen Interessenlage zwischen den beiden Anbietergruppen der Hyperscaler und ES-Anbieter.

Erstens stellt sich die Frage, inwiefern die Enterprise-Systems-Hersteller von den bestehenden Infrastrukturservices Gebrauch machen sollten. Beispielweise scheint Oracle mit seiner Oracle Cloud Infrastructure (OCI) die Strategie zu verfolgen, zumindest partiell in Konkurrenz zu AWS, Azure, und Google zu treten. Die Enterprise Systems Suite von Oracle namens Oracle Cloud Fusion kann dementsprechend nur aus der OCI bezogen oder mithilfe der Oracle Private Cloud Appliance on-premise betrieben werden. Eine Installation bei den drei Hyperscale Providern ist nicht möglich. Im Gegensatz dazu wird bei SAP eine andere Strategie bezüglich der Infrastrukturservices verfolgt. Zum einen bietet auch SAP eigene SaaS-Angebote an. Zum anderen erlaubt aber SAP auch das Deployment auf die IaaS-Ressourcen der Hyperscaler, welche entsprechende Dokumentationen bereitstellen.

Zweitens haben die Enterprise-Systems-Hersteller strategisch festzulegen, ob auf der Plattform-Ebene Konkurrenzservices angeboten werden sollen. Dabei gibt es zumindest traditionell zwei Ansatzpunkte, bei denen die ES-Hersteller bestehende Leistungen aus der On-Premise-Welt auch in die Cloud transformieren. Erstens sind Enterprise Architecture Integration Services für ES-Applikationslandschaften elementar, wie beispielsweise bei der SAP die PI (Process Integration). Diese werden auch in der Cloud von den ES-Herstellern angeboten, beispielsweise SAP mit dem Cloud Integration Service. Zweitens ist die BI-Komponente eines Enterprise Systems traditionell mit der Datenhaltung und der Analysesoftware Gegenstand des On-Premise-Angebotes. Die Hyperscaler bieten mit AWS Red Shift, Google Cloud BigQuery, und Azure Synapse eigene Lösungen an und versuchen durch eine verbesserte Kompatibilität mit der Cloud-Provider-Architektur Kundenvorteile zu versprechen. Demgegenüber bieten u. a. SAP mit SAP BW/4-HANA und SAP Data Warehouse Cloud oder Oracle mit dem Oracle Autonomous Data Warehouse eigene Data-Warehouse-Produkte an, die um Präsentationstechniken und Analysesoftware ergänzt werden. Die Enterprise System Provider versprechen den Anwendungsunternehmen eine bessere Integration mit den funktionalen Komponenten des Enterprise Systems (ERP, SCM, SRM, CRM, Shop-System).

Die Verortung der Infrastrukturservices mag aus Sicht der Anbieter auch relevant sein, für die Anwendungsunternehmen erscheint es aber weniger relevant, ob im obigen Beispiel Oracle oder ein Hyperscaler die Systeme betreibt. Dabei wird angenommen, dass die Leistungsfähigkeit für den Betrieb eines wirtschaftlichen Rechenzentrums auch bei den großen ES-Herstellern gegeben wäre, was mitunter bei einem Eigenbetrieb eines Anwendungsunternehmens nicht selbstverständlich ist.

4 Transformation von Enterprise Systems in die Cloud aus Sicht der Anwendungsunternehmen

4.1 Strategische Herausforderungen

Anwendungsunternehmen, die nachfolgend fokussiert werden sollen, haben bereits Enterprise Systems im Einsatz und stehen angesichts der Entwicklung bei den Softwareherstellern und auch bei den Hyperscalern vor der strategischen Herausforderung, die Cloud-Transformation zu planen.

Die strategischen Handlungsalternativen für die Anwendungsunternehmen können danach differenziert werden, ob einzelne Komponenten oder die gesamte ES-Applikationslandschaft transformiert werden. Zudem ist zu unterscheiden, ob viele Hersteller die ES-Landschaft prägen oder sie von einem Hersteller komplett dominiert wird. Bei der in diesem Beitrag fokussierten Situation eines dominanten Softwareherstellers ist zunächst zu prüfen, inwieweit die technologische Basis des Softwareherstellers die skizzierte Servicestruktur aus der Referenzarchitektur und exemplarisch auch der Hyperscaler Rechnung trägt. Es ist aus Applikationsarchitektursicht und der zugrundeliegenden Systemarchitektur fundamental, welche Architekturansforderungen die Unternehmen haben. Sofern das „Wie“ der Servicebereitstellung zum Unterscheidungskriterium des Cloud- vom On-Premise-Computing erhoben wird, werden viele traditionelle Softwareanbieter aktuell nicht die geeignete Lösung bereitstellen können. Im Gegensatz dazu gibt es Hersteller, die die geforderte systemarchitektonische Basis bereitstellen, bei denen die geforderte betriebswirtschaftliche Funktionalität allerdings nicht ausreichend ist. Dieser Zielkonflikt bei der Erfüllung der technologischen und betriebswirtschaftlichen Ziele bildet aktuell ein wesentliches Problemfeld der Cloud-Transformation. Es wirft das Problem auf, ob aus Sicht der Potenzialeigenschaften der IT auf diejenigen Anbieter verzichtet werden kann, die technologisch auch zukünftig die im vorhergehenden Kapitel skizzierten Technologieeigenschaften nicht unterstützen. Ansonsten könnte die Gefahr bestehen, dass die von den Herstellern prognostizierten Vorteile von Cloud-ES nicht erfüllt werden und damit die Kosten steigen, die Agilität nicht zunimmt und bei einer bedeutenden Rolle der IT für das Geschäftsmodell die Wettbewerbsfähigkeit gefährdet sein könnte. Die heutigen Systeme können bei vielen Komponenten die geforderten technischen Eigenschaften nicht aufweisen. So ist einem SAP ERP immer noch die alte Technologiebasis zu eigen. Die Kunden haben den Wechsel von R/3 auf SAP S/4HANA auf sich genommen, zumeist durch Wechsel der Technologiebasis und ohne größere Veränderung der Prozesse. Dabei hat die Cloudisierung der IT-Welt eine geringe Rolle gespielt. Nun soll die Transformation der alten techno-

logischen Basis in die Cloud (im Sinne von IaaS) die einer neuen Technologiebasis zugeschriebenen Eigenschaften realisieren können? Dies ist ein Widerspruch, der logisch nicht aufzulösen ist. Er wird mitunter kaum transparent, da die technologische Expertise der Entscheidungsträger gering und die Netzeffekte des jeweiligen Ecosystems groß sind. Allerdings wäre die strategisch-ökonomische Fragestellung zu beantworten, wie groß der technologische Vorteil echter Cloud-Systeme ökonomisch ist und wann dieser die Netzeffekte überkompensiert. Sobald dies der Fall ist, wäre der Zeitpunkt einer rationalen Entscheidung gegen den bestehenden Anbieter erreicht. Vorher erscheint eine solche Entscheidung wegen der Implikationen des Unternehmens in struktureller, terminologischer und prozessualer Hinsicht eher unwahrscheinlich zu sein.

Ein anderer wichtiger strategischer Aspekt besteht darin, die ökonomische Seite der Systemnutzung nicht zu vernachlässigen. Im Zuge der Cloud-Transformation werden die Aufwendungen für die Nutzungsrechte nicht günstiger, seien es die Nutzungsrechte der großen Softwarehersteller selbst oder auch die Leistungsangebote der Partner in dem Ecosystem (z. B. Adobe für Formulare). Sofern die geforderte Standardisierung in der Cloud dazu führt, dass immer mehr Nutzungsrechte erworben werden müssen, denen kein entsprechender Wirtschaftlichkeitsvorteil gegenübersteht, wären andere individualisierte Lösungen, auch On-Premise-Ansätze, potenziell vorzuziehen.

Einhergehend mit den beiden zuvor skizzierten Überlegungen sind auch die strategischen Implikationen der technologischen Herausforderungen für die langfristige Flexibilität des Unternehmens und des Betriebs und der Weiterentwicklung der Applikationslandschaft zu betrachten. In Abhängigkeit von den Flexibilitätsanforderungen des Unternehmens kann es erforderlich sein, die technologischen Anforderungen deutlich stärker auszuprägen, denn mit zunehmender Digitalisierung wird die technologische Basis von Enterprise Systems zunehmend bedeutender (Leyh und Wendt 2018).

4.2 Prozessuale Zwänge und Applikationsprobleme

Die für die Erfüllung der prozessualen Anforderungen des Unternehmens grundsätzliche Frage besteht darin, wie die Prozesse mittels der Standardsoftwarekomponenten unterstützt werden können. Daher sind Prozesse und Applikation in dem hier zu betrachtenden Systemumfeld untrennbar miteinander verwoben. Für die Realisierung von Anforderungen hat sich in den Standardapplikationen eine eigene Schicht, die Adaptionsschicht etabliert (Gronau 2021), die allerdings aus Sicht der Softwarearchitektur keine bedeutende Rolle spielt, denn der Softwarecode ist „vollständig“ in den Systemen verfügbar. Er wird anhand der Entscheidungen in der Adaptionsschicht nur unterschiedlich ausgeführt. Diese Situation hat dazu geführt, dass der Softwarecode zu umfangreich geworden ist, viel zu hohe Interdependenzen aufweist und damit jede Änderung und auch die anschließenden Tests enorm aufwendig geworden sind. Die Standardsoftwarehersteller sind daher aus zwei Gründen zu einer Veränderung ihrer Umsetzungsstrategie gekommen. Erstens zwingt der Umfang des Softwarecodes aus Gründen der Weiterentwicklung zu einer veränderten Softwarearchitektur, um die Modularisierung der Software zu ermöglichen. Zweitens wirkt

die technologische Entwicklung in der Cloud in die gleiche Richtung, denn es wird immer wichtiger, dass Entwickler ihre Artefakte schnell den Unternehmen und Nutzern zur Verfügung stellen können. Letztlich wird dies im Begriffspaar Continuous Integration und Continuous Delivery (CI/CD) besonderes deutlich. Es geht immer weniger darum, dass aus einer zentralistischen Sicht alles vorgedacht und überprüft werden sollte. Stattdessen ist auch für die laufende Weiterentwicklung mittlerweile akzeptiert, dass eine autonomere Entwicklung wirtschaftlich sinnvoll ist. Die durch CI/CD ermöglichte Automatisierung der Build-, Test- und Deployment-Prozesse ist daher insbesondere bei ES, die steten Veränderungsnotwendigkeiten im Tempo betriebswirtschaftlicher Anforderungen unterworfen sind, unerlässlich. Diese technologisch und betriebswirtschaftlich motivierte Flexibilitätsentwicklung hat ihre Grenzen in dem Integrationsanspruch der Systeme, die insbesondere für die Abbildung durchgängiger Prozesse erforderlich ist.

Zunächst wird in diesem Beitrag angenommen, dass die Softwarehersteller die technologisch gewünschte Entwicklungsrichtung nur einschlagen können, sofern sie ihre vielfältig vorgedachten Funktions- und Prozessalternativen oder -varianten erheblich reduzieren. Dies führt zu einer Reduzierung der durch das Enterprise System bzw. seiner Komponenten abbildbaren Anforderungen der Unternehmen (sofern diese nicht betriebswirtschaftlich irrelevant gewesen sind). Da nun die Anforderungen dann zukünftig nicht durch einen Software-as-a-Service bereitgestellt werden, bedarf es einer Umsetzung mit Hilfe von Services der PaaS-Ebene. Damit wird auch der Einsatz der in der Referenzarchitektur aufgeführten alternativen Services außerhalb der Ecosysteme von ES-Herstellern möglich. Zugleich deutet sich ein verstärkter Wettbewerb zwischen ES-Herstellern und Hyperscalern an, der aus Sicht der Anwendungsunternehmen wünschenswert sein kann. Diese können ihre Abhängigkeit von den ES-Herstellern dadurch erheblich reduzieren, denn sie verlagern ihre ES SaaS-Lösungen in PaaS-Lösungen, die dann wiederum von den ES-Herstellern unabhängig sein können. Das wiederum versuchen die ES-Hersteller zu verhindern, indem sie eigene PaaS-Lösungen anbieten und durch die Ausgestaltung von Nutzungsgebühren ökonomisch attraktiv zu gestalten.

In jedem Fall werden die SaaS-Lösungen damit nur in den Unternehmen eingesetzt werden können, wenn zusätzliche PaaS-Komponenten genutzt werden, um das Zusammenspiel einer schlanken SaaS-Lösung mit einer PaaS-Erweiterung zu ermöglichen. Aus Sicht der Applikationsarchitektur wird dies allerdings Herausforderungen mit sich bringen, die dann von den Unternehmen gelöst werden müssen. Die ehemals zentral motivierten Datenmodelle, die auch aus der Wissenschaft stets gefordert wurden und in unternehmensweiten Datenmodellen mitunter die Basis für den Erfolg heutiger ERP-Systeme und deren Nachfolger sind, werden damit obsolet. Die redundante Datenhaltung in modernen Microservice-Architekturen, die die Basis heutiger Entkopplungs- und Modularisierungsversuche darstellen, wird damit zu einer Herausforderung bei dem Entwurf der Systeme. In jedem Fall wird eine Verlagerung der konzeptionellen Entscheidungen von der Ebene der SaaS-Lösung auf die PaaS-Ebene erfolgen und damit von dem Softwarehersteller zu den Anwendungsunternehmen oder deren Implementierungspartnern. Diese Verlagerung und auch die Wahl der Plattform, deren Services genutzt werden, ist zudem ökonomisch von hoher Relevanz. Wie die obigen Ausführungen andeuten, muss der Standardsoft-

warehersteller versuchen, diese Abbildung der Anforderungen in seinem Ecosystem und seiner Plattform zu halten. So weist die SAP Business Technology Platform genau diese Entwicklung auf. Der Umsatz nimmt zu und die Implementierungspartner sind angehalten, die ehemals im Projekt erfolgte Entwicklung über die Plattform als weitere Lösungskomponenten zu entwickeln und anzubieten. Damit wird dann zukünftig der Standardsoftwarehersteller am Umsatz der Partner bei Wartungsgebühren profitieren, der Implementierungspartner auch und die Anwendungsunternehmen neben den Nutzungsgebühren über die SaaS-Services auch noch laufende Nutzungsgebühren für die über die PaaS genutzten Komponenten zu bezahlen haben. Dieses Spannungsfeld birgt enorme Gefahren, wenn die Systeme in der Zukunft so eingeführt würden wie dies in der Vergangenheit der Fall gewesen ist. Grundsätzlich dürfte dabei die Transformation der Applikationsarchitekturen in die Cloud auch ein Anlass sein, bei dem die Prozesse grundsätzlich überdacht werden sollten.

4.3 Systemtechnische Revolution des Cloud Computing und deren Auswirkungen auf das IT-Management

Die Aufgaben des IT-Managements und auch der operativen Betriebsprozesse werden in den Unternehmen durch das Cloud Computing auf der technischen Ebene durch die Andersartigkeit nachhaltig das „Wie“ der IT-Prozesse verändern. Nachfolgend sollen anhand der skizzierten Enterprise-Systems-spezifischen Kombination von SaaS- und PaaS-Lösungen die neuen Integrations- und Koordinationsanforderungen angedeutet werden.

Die *Integrationsanforderungen*, die aus der Verbindung mit SaaS-Lösungen entstehen (z.B. die Integration eines CRM von Salesforce mit dem ERP von SAP) bilden eine tradierte Aufgabe, die aufgrund der erwarteten Modularisierung von ES-Landschaften (Gartner 2020) weiter zunehmen werden. Dabei besteht aufgrund der zum Teil „erzwungenen“ Updates durch Cloud-Native-Anwendungen eine Aufgabe in der Unterstützung des Versionsmanagements durch wesentlich agilere IT-Prozesse. Eine bestehende Aufgabe, die dieses Problem ebenso betrifft, ist das Testmanagement. Das Testen wird zu einem laufenden Standardprozess, denn es sieht aktuell nach drei Versionierungen pro Jahr aus, so wie es Salesforce als erster großer Proponent und Anbieter des Cloud Computing für ES seit längerem praktiziert. Auch die Architektur- und Integrationsmanagementaufgaben werden wichtiger und können von dem IT-Management nicht mehr missachtet werden.

Die *Koordinationsherausforderungen* ergeben sich aus der institutionellen Komplexität, denn die ehemals dominant im eigenen Unternehmen wahrgenommenen Aufgaben sind zukünftig von einem Netz von Dienstleistern, Cloud-Anbietern und dem Unternehmen selbst wahrzunehmen. In diesem Zuge werden intern die Aufgaben zwischen der IT-Abteilung und den Fachbereichen neu verteilt und ehemals interne Aufgaben auch von Dienstleistern wahrgenommen. Damit setzt sich ein Trend fort, in dem sich die IT-Abteilung seit Jahren befindet und die IT-Wertschöpfungsketten verändern sich dramatisch. Damit wird auch die Diskussion um die Rollen von Chief Digital Officer und Chief Information Officer neu belebt und die IT-Systeme und deren Organisation wird viel stärker in dem Operations Management verankert werden, so die Prognose der Autoren. Für die Forschung ergeben

sich damit neue Herausforderungen, die effektive und effiziente Organisation der IT und ihrer Prozesse zu untersuchen.

Die Integrations- und Koordinationsproblematik wird durch einen mitunter zu gering beachteten Aspekt zusammengebracht: die Kompetenzen der Mitarbeiter. Die durch die Mitarbeiter repräsentierten Kompetenzen bilden die zentrale Knappheitsressource, an der sich die Entscheidungen auszurichten haben. Diese Überlegung, die dem Ausgleichsgesetz der Planung nach Gutenberg (1951, S. 126) folgt, erfordert aber zunächst eine Operationalisierung, die allerdings sowohl für Unternehmen als auch für die Wissenschaft sehr spannende, gestaltungsorientierte Einsichten bietet.

5 Fazit

Enterprise Systems stehen vor enormen Herausforderungen in Zeiten des Cloud Computing, denn die meisten der bestehenden Systeme weisen noch nicht die Modul-Charakteristika der in Abschn. 3 skizzierten Referenzarchitektur auf. Damit werden auch die Strategie und die zukünftige Einführungsweise sich erst in den nächsten Jahren sichtbar in den Unternehmen verändern. Es sind aufgrund der technologischen Bestrebungen bereits jetzt einige Trends offensichtlich. Zunächst wird die von den Herstellern zwingend verfolgte Reduktion der Systemkomplexität dazu führen, dass es mehr Standard im Kern der Systeme geben wird. Außerdem wird dadurch die Verbindung von SaaS- und PaaS-Lösungen zwingend, so dass mehr Individualentwicklungen möglich werden – bei einer Separierung der spezifischen Individualisierungswünsche in Side-by-Side-Ansätzen. Diese Differenzierung zwischen den auch gerne als „lean core“ bezeichneten ES-Komponenten und den spezifischen Anforderungen des Unternehmens eröffnet auch die Möglichkeit, die strategisch wichtigen Anforderungen für Flexibilität und Agilität des Unternehmens außerhalb des vom Hersteller dominierten Softwarekontext zu realisieren. Damit ist es möglich, auch unter Betrachtung der AI Capabilities, dass wir ein neues Zeitalter von Individualsoftware erwarten dürfen, welches die ES-Komponenten als Commodity betrachtet. In jedem Fall wird deutlich, dass das Cloud Computing die Spielregeln zwischen Herstellern und Anwendungsunternehmen verändert und letztlich fundamental für die gesamte unternehmerische Entwicklung ist.

6 Anhang

Tab. 1 Service-(Produkt-)Angebot der Hyperscaler auf den unterschiedlichen Serviceebenen. (Eigene Darstellung)

	AWS	Azure	Google Cloud
META-SERVICES			
<i>Logging</i>	CloudTrail	Monitor Logs	Cloud Audit Logs
<i>Cloud User IAM</i>	IAM	Identity Management	IAM
<i>Billing</i>	Billing and Cost Management	Billing	Cloud Billing
<i>Cost Reduction</i>	Recommender	Cost Optimization	Cost Management
<i>Infrastructure as Code Resource Provisioning</i>	CloudFormation	Azure Resource Manager	Terraform on Google Cloud
NETWORKING			
<i>VPC</i>	VPC	Virtual Network	VPC
<i>DNS</i>	Route 53	DNS	Cloud DNS
<i>CDN</i>	CloudFront	Front Door	Media CDN
<i>Firewall</i>	Web Application Firewall	Web Application Firewall	Cloud Armor
<i>Load Balancing</i>	Elastic Load Balancing	Load Balancer	Cloud Load Balancing
PRIVATE CLOUD			
<i>Integration at dedicated peering locations</i>	Direct Connect	Express Route	Cloud Interconnect
<i>Installation</i>	Outposts	Stack	Distributed Cloud
COMPUTE: OS VIRTUALIZATION			
<i>VMs</i>	EC2	Virtual Machines	Compute Engine
<i>Remote Desktop</i>	WorkSpaces	Virtual Desktop	Compute Engine
<i>Transient Batch Jobs (also compatible containers)</i>	Batch	Batch	Batch
COMPUTE: CONTAINER VIRTUALIZATION			
<i>Kubernetes</i>	EKS	AKS	Kubernetes Engine
<i>High-Level Container Platform</i>	Fargate	Container App	Cloud Run
<i>Low-Level Container Platform</i>	ECS	Container Instances	Cloud Run
COMPUTE: SERVERLESS			
<i>Functions</i>	Lambda	Functions	Cloud Functions

Tab. 1 (Fortsetzung)

	AWS	Azure	Google Cloud
STORAGE			
<i>Block</i>	EBS	Disk Storage	Persistent Disk
<i>Object</i>	S3	Blob Storage	Cloud Storage
<i>Relational proprietary</i>	Aurora	SQL	AlloyDB, Cloud Spanner
<i>Relational 3rd party</i>	RDS	Database	SQL
<i>NoSQL</i>	DynamoDB	Table Storage, Cosmos DB	Datastore, Bigtable
<i>In-Memory Key-Value Store (Redis/Memcached)</i>	ElastiCache	Cache	Memorystore
<i>NFS/SMB</i>	EFS	Files	Filestore
DATA-INTENSIVE COMPUTE/WORKLOADS & ML WORKLOADS			
<i>SQL Query Engines (~ Presto)</i>	Athena	Synapse Polybase	Presto on Dataproc
<i>ML Orchestrator</i>	SageMaker	Machine Learning Studio	Vertex AI
<i>SQL Data Warehouse</i>	Redshift	Synapse Analytics	BigQuery
<i>Data Lake Governance</i>	Lake Formation	Data Share	Dataplex
<i>ETL</i>	Glue	Data Factory	Cloud Data Fusion
<i>MPP</i>	EMR	HDInsight	Dataproc
<i>Enterprise Search</i>	Kendra	AI Search	Cloud Search
<i>Visualization</i>	QuickSight	Power BI	Looker
<i>Data Metadata Management</i>	Glue Data Catalog	Data Explorer, Purview	Dataplex
SYNCHRONOUS COMMUNICATION			
<i>API Management</i>	API Gateway	API Management	API Gateway
<i>GraphQL</i>	AppSync	API Management (as a feature)	Apigee (as a feature)
ASYNCHRONOUS (EVENT/MESSAGE-BASED) COMMUNICATION			
<i>Message Broker (~ RabbitMQ)</i>	MQ, SQS + SNS	Service Bus	Pub/Sub
<i>Stream Processing (~ Kafka)</i>	Kinesis	Event Hubs	Partner Confluent Cloud
<i>Queuing</i>	SQS	Queue Storage	Pub/Sub
<i>Event Bus/Event Routing</i>	EventBridge	Event Grid	Eventarc
<i>Workflow automation</i>	Step Functions	Logic Apps	Workflows
GENERIC APP SERVICES			
<i>End User IAM</i>	Cognito	App Service Auth	Identity Platform, Firebase Auth
<i>Secret Management</i>	Secrets Manager	Key Vault	Secret Manager
<i>Web Hosting</i>	Amplify Hosting	Firebase Hosting	Static Web Pages
<i>Logging and Monitoring (~ OpenTelemetry, Splunk, Sentry)</i>	CloudWatch	Monitor Logs	Cloud Logging
<i>Enterprise User IAM</i>	IAM Identity Center	Entra ID	Cloud Identity

Tab. 1 (Fortsetzung)

	AWS	Azure	Google Cloud
<i>App Framework</i>	Amplify	App Service	Firebase
<i>Tracing</i>	X-Ray	Application Insights	Cloud Trace
(PRE-)TRAINED AI APIS			
<i>Text-to-Speech</i>	Polly	AI Speech	Text-to-Speech AI
<i>NLP</i>	Comprehend	AI Language	Natural Language
<i>Conversational AI (Chat-bots)</i>	Lex	AI Bot	Diagflow, Vertex AI Agent
<i>Translation</i>	Translate	AI Translator	Translate
<i>Computer Vision</i>	Rekognition	AI Vision	Vision AI
CODE AUTOMATION			
<i>CI/CD</i>	CodePipeline, CodeBuild, CodeDeploy, CodeCommit	GitHub, Azure DevOps, Repos	Cloud Build, Cloud Deploy, Source Repositories
<i>Registry</i>	ECR	Container Registry, Artifacts	Artifact Registry
FULL-STACK SAAS APPLICATIONS			
<i>SAP Deployment</i>	SAP on AWS (EC2, EBS, VPC, EFS)	SAP on Azure (VMs, Disk Storage, Virtual Network, Files)	SAP on Google Cloud (Compute Engine)

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen. Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

- Anselmi J, Ardagna D, Lui JC, Wierman A, Xu Y, Yang Z (2017) The economics of the cloud. TOMPECS, Bd. 2(4). ACM Transactions on Modeling and Performance Evaluation of Computing Systems
- Aral S, Brynjolfsson E, Gu C, Wang H, Wu DJ (2024) Understanding the returns from integrated enterprise systems: the impact of agile and phased implementation strategies. MISQ 48(2):749–774
- Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a Berkeley view of cloud computing. Technical report, Bd. UCB/EECS-2009-28. University of California, Berkeley
- Barroso L, Dean J, Hölzle U (2003) Web search for a planet: the Google cluster architecture. IEEE Micro 23(2003):22–28
- Brehm L, Heinzl A, Markus ML (2002) Tailoring ERP systems: a spectrum of choices and their implications. Proceedings if the 34th Annual Hawaii International Conference on Systems Sciences. Track:

- Organizational Systems and Technology, Minitrack: ERP System Issues and Answers, Island of Maui (Hawaii, USA), January 3–6, 2001
- Buxmann P, König W (1997) Empirische Ergebnisse zum Einsatz der betrieblichen Standardsoftware SAP R/3. *Wirtsch Inform* 39(4):331–338
- Christensen C (1997) The innovator's dilemma: when new technologies cause great firms to fail, 1. Aufl. Harvard Business Review Press
- Dittrich Y, Vaucouleur S (2008) Practices around customization of standard systems. In: Cheng L, Sillito J, Storey MD, Tessem B, Venolia G, de Souza CRB, Dittrich Y, John M, Hazzan O, Maurer F, Sharp H, Singer J, Sim SE (Hrsg) Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2008, Leipzig, Germany, Tuesday, May 13, 2008. ACM, S 37–40 <https://doi.org/10.1145/1370114.1370124>
- Fehling C, Leymann F (2018) Provisionierung. *Gabler Wirtschaftslexikon*. Springer Gabler (<https://wirtschaftslexikon.gabler.de/definition/provisionierung-53362>)
- Gartner (2020) The future of ERP is composable. <https://www.gartner.com/en/documents/3991664>. Zugegriffen: 10. Dez. 2022
- Gronau N (2021) ERP-Systeme. Architektur, Management und Funktionen des Enterprise Resource Planning, 4. Aufl. Walter de Gruyter, Berlin, Boston
- Gutenberg E (1951) Die Produktion, 1. Aufl. Bd. 1. Springer, Berlin, Heidelberg <https://doi.org/10.1007/978-3-662-21965-2>
- Hansen HF, Haddara M, Langseth M (2023) Investigating ERP system customization: a focus on cloud-ERP. *Procedia Comput Sci* 219:915–923
- Hentschel R, Leyh C (2016) Cloud Computing: Gestern, heute, morgen. *HMD* 53(2016):563–579
- Hentschel R, Leyh C, Petznick A (2018) Current cloud challenges in Germany: the perspective of cloud service providers. *J Cloud Comput* 7:5
- Hoepfner G (2021) Interview von Guido Hoepfner zur SAP-Einführung bei ALDI Nord durch Reinhard Schütte
- Klaus H, Rosemann M, Gable G (2000) What is ERP? *Inf Syst Front* 2(2):141–162
- Kurbel KE (2021) Enterprise resource planning und supply chain management in der Industrie, 9. Aufl. Springer, Berlin, Heidelberg
- Leyh C, Wendt T (2018) Enterprise Systems als Basis der Unternehmens-Digitalisierung. *HMD* 55:9–24
- Olson DL, Kesharwani S (2010) Enterprise information systems. Contemporary trends and issues. World Scientific, New Jersey et al.
- Sarferaz S (2022) Compendium on enterprise resource planning. Market, functional and conceptual view based on SAP S/4HANA. Springer, Berlin
- Sarferaz S (2023) ERP-Software: Funktionalität und Konzepte. Basierend auf SAP S/4HANA. Vieweg, Wiesbaden
- Schütte R (1998) Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsorientierte Modelle. Gabler, Wiesbaden
- Schütte R (2023) ERP-Systeme in der Ära des Cloud Computings. *ERP information*, Bd. 1/2023. DPI Verlag, S 15–19 https://doi.org/10.58678/erp-information_23-1_15-19
- Schütte R (2024) The next generation of ERP systems: problems of traditional ERP-Systems and the next wave of really standardized ERP-Systems. In: Strecker S, Jung J (Hrsg) Informing possible future worlds—Essays in honour of Ulrich Frank. Logos, Berlin, S 427–452
- Teubner A (1999) Organisations- und Informationssystemgestaltung : theoretische Grundlagen und integrierte Methoden. Gabler, Wiesbaden
- Waibel P (2020) Cloud-based elasticity for business processes and data storage. TU Wien (Dissertation)
- Wulfert T (2024) Selected perspectives on platforms in E-commerce ecosystems. Recommendations for the design and management of boundary resources and guidance on the orchestration of ecosystem participants. Springer, Berlin

Hinweis des Verlags Der Verlag bleibt in Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutsadressen neutral.