Opriel, Sebastian; Strobel, Gero; Otto, Boris; Möller, Frederik

Article — Published Version

Data Sovereignty in Inter-organizational Information Systems

Business & Information Systems Engineering

Suggested Citation: Opriel, Sebastian; Strobel, Gero; Otto, Boris; Möller, Frederik (2024) : Data Sovereignty in Inter-organizational Information Systems, Business & Information Systems Engineering, ISSN 1867-0202, Springer Fachmedien Wiesbaden, Wiesbaden, Vol. 67, Iss. 6, pp. 833-853,
https://doi.org/10.1007/s12599-024-00893-4

This Version is available at:
https://hdl.handle.net/10419/333371

Bus Inf Syst Eng 67(6):833–853 (2025)
https://doi.org/10.1007/s12599-024-00893-4

RESEARCH PAPER

Data Sovereignty in Inter-organizational Information Systems

Findings from Demand and Capacity Management in the Automotive Industry

Sebastian Opriel · Frederik Möller · Gero Strobel · Boris Otto

Received: 22 December 2022 / Accepted: 9 July 2024 / Published online: 29 August 2024

Abstract Car manufacturers and suppliers in the Automotive industry increasingly face the issue of optimization of highly complex supply chains that need to accommodate each customer's precise demands, requiring a vast array of parts and information to be available at the right place and at the right time. This involves data sharing between organizations, which is hindered by various issues, such as fear of data misappropriation by the data receiver or the involuntary disclosure of business secrets. The paper proposes design principles for a novel type of Inter-Organizational Information System, which addresses these challenges through the technical implementation of data sovereignty. The study reports on an Action Design Research study in the Automotive industry between a car manufacturer and a 1st-tier supplier. It contributes (a) design requirements, (b) design features, (c) an instantiation, and (d) design principles for this type of data sovereign inter-organizational information system.

Accepted after two revisions by Óscar Pastor.

S. Opriel
TU Dortmund University/sovity GmbH, August-Schmidt-Straße 1, 44227 Dortmund, Germany
e-mail: sebastian.opriel@tu-dortmund.de

F. Möller (✉)
TU Braunschweig/Fraunhofer ISST, Mühlenpfordtstraße 23, 38106 Braunschweig, Germany
e-mail: frederik.moeller@tu-braunschweig.de

G. Strobel
University Duisburg-Essen, Universitätsstraße 9, 45141 Essen, Germany

B. Otto
TU Dortmund University/Fraunhofer ISST, August-Schmidt-Straße 1, 44227 Dortmund, Germany

Keywords Data sovereignty · Inter-organizational information systems · Usage control · Action design research · Design principles

1 Introduction

Automotive supply chain participants, such as car manufacturers, contract and coordinate thousands of suppliers on different supply chain levels (tiers) to obtain materials, parts, and services (e.g., Doran 2004; Kern and Wolff 2019). The production relies on strictly and accurately paced delivery concepts, such as Just-in-Time (JIT) or Just-in-Sequence (JIS), to manufacture highly customized automobiles for their customers. These supply chains are riddled with complex decisions, such as identifying correct parts and suppliers, that could be optimized through transparency by sharing data along the supply chain (Ahmed and Omar 2019; Aldoori 2019; Wiengarten et al. 2013). Sharing these data is hindered by a variety of barriers and tensions, including a lack of trust, fear of data misappropriation, or the unknown value of the data an organization is to share (e.g., Fassnacht et al. 2023; Jussen et al. 2023, 2024; Opriel et al. 2021). More specifically, organizations fear that sharing data leads to competitive disadvantages in price negotiations or worse positions in supply chain power struggles (Kumar et al. 2018; Opriel et al. 2021). As opposed to physical goods and products, digital data can be easily copied, shared, or reproduced with almost zero marginal cost (Shapiro et al. 1998; Veit et al. 2014). Subsequently, these data require different sovereignty standards that can be summarized under the umbrella of 'data sovereignty,' which we see as the "self-determination of individuals and organizations with regard to the use of their data" (Jarke et al. 2019, p. 550).

Data sovereignty is especially relevant when organizations share sensitive information,[1] in which partners in inter-organizational settings require a basic level of mutual trust. To further distinguish different trust levels, control is a significant factor, which is defined as the "willingness of a party to be vulnerable to the actions of another party (…) irrespective of the ability to monitor or control that other party" (Mayer et al. 1995, p. 712). Subsequently, less trust is needed between supply chain participants when the control an organization has over their data and how they are used is formalized through, e.g., legally binding contracts or usage control (UC) mechanisms. Our work starts precisely at this point as we report on the design and implementation of an inter-organizational information system (IOIS) between an Original Equipment Manufacturer (OEM) and a supplier in the German automotive industry. Traditionally, this information system class is a shared system between two parties through a software artifact (Cash and Konsynski 1985). Klein et al. (2005, p. 170) define IOISs as "embedded or deployed" systems that build "interfirm networks," such as Enterprise Resource Planning systems (ERP). These IOISs offer a variety of advantages for organizations since they enable the automated exchange of information between suppliers and customers, making processes more efficient and transparent through supply chain integration (Holland 1995; Johnston and Vitale 1988; Saeed et al. 2011). Supply chains benefit from deploying IOISs by generating continuous information flow and visibility across the chain of involved actors (e.g., Lee et al. 2014).

This study adopts a design-oriented approach (Hevner et al. 2004) and reports on designing an IOIS with features to implement situation-specific data sharing and protection against misuse by data receivers in their systems based on UC policies, which aims to motivate data providers to share sensitive data. We do this by pursuing the following research question: *How to design an IOIS enabling the sovereign exchange of sensitive information in automotive supply chains?*

In this paper, we report on an Action Design Research (ADR) study following the elaborated ADR process of Mullarkey and Hevner (2019) as part of a multi-paper project initiated with a conference paper (see Opriel et al. 2021) based on a dissertation (see Opriel 2021). The conference paper focused exclusively on the elicitation of the problem space and requirements. Within this publication, we significantly extended the context of the ADR-study, including generating design features, artifact building, evaluation, and formulating design principles. Subsequently, the value in this extension is the actual implementation of the artifact addressing the design requirements, as well as the extraction of design principles for data sovereign IOISs.

The ADR study spanned 17 months of operation and interaction between one OEM and one 1st tier German automotive supplier. ADR is explicitly suitable in this setting since we had the chance to actively design an artifact in the field with practitioners, enabling us to learn from and with practice and extract and formalize valuable design knowledge about data sovereign IOISs (Sein et al. 2011). We can infer the novelty of data sovereignty as an organizational problem, given the emergence of a variety of well-funded European initiatives that exclusively tackle this issue. Examples are Gaia-X (2024), which is domain agnostic, and Catena-X (2024), which explicitly focuses on the German automotive industry. Against this background of many large-scale European research initiatives focusing precisely on these issues, these results are highly relevant for research and practice. The constellation we investigate exists manifold in various relationships between OEMs and n-tier suppliers, not just in the automotive industry. Subsequently, designing a data sovereign IOIS is at the core of Information Systems (IS) research, at the interplay between digitalization and building novel IT artifacts to solve organizational problems in digital transformation (Hevner et al. 2004; Legner et al. 2017).

The remainder of this article is structured as follows: Sect. 2 introduces IOISs and data sovereignty as key concepts in our study design. Section 3 details our approach to the ADR study. Sections 4 to 6 illustrate our findings structured alongside the ADR process. In Sect. 7, we formalize the knowledge we gained as design principles. Section 8 discusses contributions to theory and practice, limitations, and closes the article with a conclusion.

2 Background

Below, we illustrate the concepts of *IOISs* and *data sovereignty*. Both concepts are required to understand and position the key artifact of our ADR study, a data sovereign IOIS. It aims to implement data sovereignty in inter-organizational data sharing in a shared software between a car manufacturer and an OEM. The overview below is not a systematic review but introduces relevant concepts necessary to understanding the study.

2.1 Inter-organizational Information Systems

IOISs are an established technical artifact to integrate stakeholders via shared software (e.g., suppliers or customers) to reduce the effort in information sharing and

[1] "[I]nformation that are exchangeable from a technical and legal point of view, but whose exchange is not desirable due to mistrust, unclear benefits, or other concerns" (Opriel et al. 2021).

positively affect competitive advantage by lowering transaction costs (Johnston and Vitale 1988). Especially in supply chains, IOISs enable instant, transparent information exchange and outperform non-IOIS-based supply chains based on asynchronous data sharing (Lee et al. 2014). IOISs are "defined as automated information systems shared by two or more companies, [that] will significantly contribute to enhanced productivity, flexibility, and competitiveness of many companies" (Cash and Konsynski 1985). Such systems can be configured in many ways tailored to organizations' specific use cases, prompting various potential outcomes (e.g., electronic markets) (Holland 1995). Based on a systems theory perspective, IOISs can be divided into bilateral (e.g., buyers and sellers) and multilateral systems according to their system topology (Choudhury 1997). In multilateral systems, IOISs are a technological intermediary between organizations (e.g., one or multiple buyers and sellers). Subsequently, these IOISs work as a boundary object between different organizations, which requires them to engage to some degree of formalized collaboration that exceeds their individual boundaries (Kumar and van Dissel 1996). IOISs provide electronic channels interfacing with two or more organizations, giving them instant access and visibility to information in the circle of organizations embedded in the IOIS ecosystem (Lee et al. 2014). These IOIS ecosystems can develop around a shared proprietary or open software artifact, which determines the specifics of data flow (e.g., data standards, semantics, processes) (Lyytinen and Damsgaard 2011). A state-of-the-art example of IOIS ecosystems is *data spaces*, which function as a boundary object between different stakeholders and offer them a 'space' to act digitally with their data (Möller et al. 2022a).

2.2 Data Sovereignty and Usage Control Policies

Sovereignty originates in the Latin *superanus* and refers to exercising political power and authority in the organization of a state (Britannica 2023). This notion of sovereignty in the political sphere has since been shifted into the digital world and combined with a range of new concepts, such as *digital sovereignty*, *technological sovereignty, cyber sovereignty*, *virtual sovereignty,* or *data sovereignty* (Hellmeier and von Scherenberg 2023; Hummel et al. 2021, Jarke 2020). These concepts are utilized at different application levels, from the individual to the organizational to the (inter-)national. Hellmeier and von Scherenberg (2023) propose contextualizing and differentiating *digital*, *technological*, and *data sovereignty*. They find that technological sovereignty refers to a nation's sovereignty or authority about its technological infrastructure positioned in international politics. Digital sovereignty is one lower abstraction level of economic authority over IT infrastructure and a body of digital expertise required to act in the digital sphere, i.e., the "digital literacy of a population" (Hellmeier and von Scherenberg 2023, p. 10). Data sovereignty refers to the individual or organizational level and describes the power to determine what happens with the data of an individual or organization. Hummel et al. (2021) find that *cyber sovereignty* often occurs with issues of defense on a governmental level and *internet sovereignty* with legislation and international relationships. From these recent studies (Hellmeier and von Scherenberg 2023; Hummel et al. 2021), we can infer that *data sovereignty* is the most relevant concept for our research since we strive to implement mechanisms for trusted data sharing on an organizational level and not a (inter-)national one. Data sovereignty in itself is not clearly defined but has a bouquet of ways for interpretation. For example, *national data sovereignty* means data are not allowed to leave the jurisdiction of a particular country (e.g., Amoore 2016) or that data is subject to its specific laws (e.g., Polatin-Reuben and Wright 2014). A significant research stream explicitly focuses on data sovereignty to preserve and establish self-determination and rights for data of and about *indigenous people* (e.g., Kukutai and Taylor 2016).

Regardless of the definition or field of application, basic mechanisms such as the notion of control or restricting the flow and use of data are pivotal in establishing data sovereignty (Hellmeier and von Scherenberg 2023; Hummel et al. 2021; Kukutai and Taylor 2016; Otto 2022). In our case, we focus on *organizational data sovereignty*, granting organizations control over their data. Subsequently, we adopt the understanding of data sovereignty as "the self-determination of individuals and organizations with regard to the use of their data" (Jarke et al. 2019, p. 550) in the minimal bilateral relationship of sharing data between a data provider and a data receiver, which governs the control of data of both parties, i.e., what the providers can expect to happen to their data and what the receiver is allowed to do with it (Otto and Jarke 2019). This definitional approach is reflected in up-to-date definitions of *data sovereignty* in both the individual and the organizational context (see Table 1).

While there are a variety of concepts for operationalizing data sovereignty, such as Digital Rights Management (DRM) (Park and Sandhu 2004; Zhaofeng Ma 2017), we focus on UC policies, which are "rules for the usage of data between the provider of the data and the customer" (Zrenner et al. 2019, p. 480). We explicitly draw from UC policies since it is a synergy of the concepts explained above and is the chosen method in the European research landscape (e.g., Eitel et al. 2021). The context of the data-sharing environment somewhat outlines how UC policies need to be formulated. For example, these policies can be influenced by the interests or preferences of either the data

Table 1 Selected definitions of data sovereignty (based on Hellmeier et al. 2023; Hummel et al. 2021)

Citation	Definition
Alboaie and Cosovan (2017), p. 86	"we consider data sovereignty to be the ability of the user to have full control over his data and the entities to which it is shared or revoked"
Sarabia-Jácome et al. (2019), p. 101	"Consequently, the data sovereignty concept arises, which is defined as the ability of the data owner to decide itself how to share and use its data."
Otto (2022), p. 5	"Data sovereignty refers to the capability of a legal entity or natural person to determine and execute usage rights when it comes to their data"

provider, the data owner, or both. They also can be predetermined by legal requirements or prior UC policies defined by a previous data owner (Pretschner et al. 2006). Zrenner et al. (2019) report on two examples of UC policies in bottleneck management in supply chains from different data provider perspectives (OEM and supplier), structured alongside provisions (the content of the policy), fixed time interval, obligations, and which are eternally valid. In one example, for specific UC policies, the OEM gives the supplier a time frame of three days to use the data and then mandates that this data must be deleted. The OEM also defines some eternally valid obligations, such as a prohibition on forwarding the data to other companies or using the data for purposes other than bottleneck management. On the suppliers' side, the supplier gives the OEM a 14-day time frame to use the data and mandates that they be deleted afterward. Similar to policies defined by the OEM, the supplier prohibits forwarding data to other companies, using the data for other purposes, or using the data to the "detriment of the supplier" (p. 485). We draw on these fundamental UC mechanisms through the research process to develop a data-sovereign IOIS and create descriptive design knowledge for future IOISs.

3 Research Design

We report on an in-depth ADR study in the German automotive industry between an OEM and a 1st tier supplier. The study aimed to design an artifact that enables both parties to exchange sensitive information while upholding their data sovereignty. This data originating from production planning systems includes *sensitive information* about *stocks*, the *suppliers' production plans*, *key information from the OEM's demand*, and *capacity analysis*. Given the sensitivity of the data, the industrial partners insist on implementing mechanisms to foster and ensure that the data's recipient is non-opportunistic (Khurana et al. 2011). The underlying scenario is highly complex since it considers various dimensions, such as security, trust, and motivation of two parties to share information in a supply chain. We opted to conduct an ADR study because of its explicit requirement to oscillate between research and practice in designing novel artifacts. ADR addresses "a problem situation encountered in a specific organizational setting by intervening and evaluating; and (2) construct[s] and evaluat[es] an IT artefact that addresses the class of problems typified by the encountered situation" (Sein et al. 2011, p. 40). Referring to the previous quote from Sein et al. (2011), this is precisely the situation we faced: solving an organizational IT problem through the design and introduction of a novel IT artifact belonging to the class of information systems *data sovereign IOISs*. In particular, we follow the elaborated ADR process of Mullarkey and Hevner (2019) and formalize the design knowledge we produced in our design process as five design principles.

3.1 ADR Cycles

We adopted a subset of the elaborated ADR process in diagnosis (1) and design (2) and reduced the scope of the original implementation stage (3) to the development of the artifact. To clearly distinguish between the development process and the extensive evaluation, we adjusted the last stage (4) to be *pilot*. In this phase, we tested the IOIS in a real business context and collected feedback from the field. Each of our four stages consists of an iterative five-step process with activities: problem formulation **(P)**, artifact creation **(A)**, evaluation **(E)**, reflection **(R)**, and learning **(L)** (see Fig. 1).

Our entry point into the diagnosis stage was *problem-centered*, originating in the field with practitioners of the German automotive industry and their problems exchanging sensitive data in *demand and capacity management*. The overall ADR process followed a linear structure from one stage to the next, with one backward iteration to the design stage. However, several transitions lead to minor changes and improvements in the artifact. For example, following an agile software development approach, some implementation decisions had drawbacks from the initial concept (design stage) or even the requirements (diagnosis

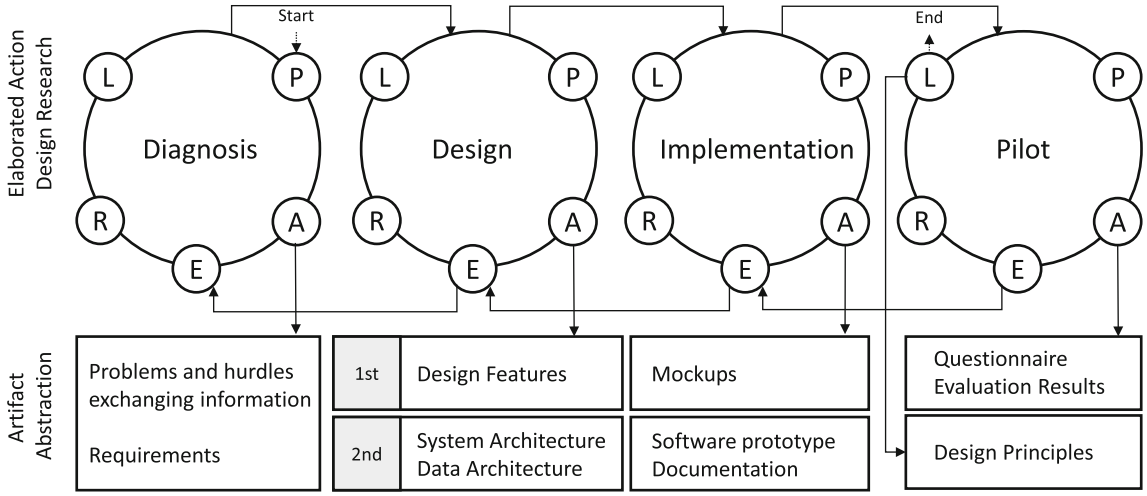

Fig. 1 Elaborated ADR adapted from Mullarkey and Hevner (2019)

stage). This is somewhat expected since designing is a search process and real design problems are seldom idealized but instead messy, making the problem and solution space dynamic (Hevner et al. 2004). The frequency of interactions between the research team and practice resulted in over 240 appointments in 305 business days, eight workshops (avg. seven participants), and nine steering committees (avg. 12 participants) (see Appendix A, available online via http://link.springer.com). The study involved 90 individuals from 17 companies, including the ADR team, which consists of 18 individuals. Initially, we identified problems and derived a set of requirements for an IOIS for this scenario. In the first iteration of the design cycle, we generated a set of 18 design features of a potential software artifact addressing design requirements. In this stage, we produced the first mock-ups in the implementation stage. On this, we gathered feedback from the field. We went back into the design stage to build the system and data architecture, which were implemented as a prototype in the subsequent implementation stage. Finally, we piloted the prototype in a real business setting.

4 Exploration of the Problem-Space (Diagnosis Cycle)

In the first stage of the ADR research project, we diagnosed, framed, and scoped the problem space (i.e., problems and design requirements) as well as the associated objective for the artifact (Goel and Pirolli 1992). The motivation of the study is the optimization of demand and capacity management in automotive supply chains. The OEM has *short-term*, *mid-term*, and *long-term planning*. The short-term planning spans several weeks before production, in which the OEM specifies its call-offs regularly. Since call-offs can vary plus or minus five percent due to differences in logistics systems or adjustments resulting from rescheduling production sequences for just-in-sequence deliveries, higher transparency of suppliers' production processes can support adjusting production programs more swiftly. In the case of bottlenecks, the alignment of suppliers and OEMs is even more time-critical for rescheduling production plans. Real-time access to OEM demands or suppliers' production capacities fosters process optimization on both sides. Subsequently, to leverage this potential and engage in a problem-motivated ADR process, the goal was to design an IOIS to exchange sensitive information about *stocks* (OEM and supplier), *production numbers* (supplier), and *demands* (OEM). Another mandatory condition from the field is the ability of the IOIS to be transferable to more scenarios in the supplier-customer relationship, to empower existing processes digitally, and to generate advantages on both sides (OEM and supplier) to enhance the applicability and acceptance. We pursue this by extracting design principles from the *artifact* and *design process* (see Möller et al. 2020; Sein et al. 2011), which we formulate reflectively for this case but abstract them so that they are valid in the boundaries of IOISs in OEM and supplier relationships, detached from the specific data that is exchanged. This procedure makes the findings instantiable in comparable scenarios of similar boundary conditions (Chandra Kruse and Seidel 2017). After setting these constraints, we established a common understanding of highly relevant concepts, constructs, and terminology in this domain. We consulted practice to get an in-depth understanding of how data exchange works in automotive supply chains (see Table 2).

First, we investigated the *problem space* (e.g., Maedche et al. 2019) by formulating problems in supply chain data sharing (see Appendix B). We elicited these problems from experts in the field, case documentation, and literature (see

Table 2 Overview of interviews to explore the problem space

ID	Duration (min)	Company	Department	Position	Experience
E1	60	OEM a	Innovation	Project Manager	3 years
E2	40	OEM a	Innovation	Life Cycle Manager	8 years
E4	60	OEM a	Corporate IT	IT Architect	5 years
E3	48	OEM b	Innovation	IT Manager	4,5 years
E6	26	OEM b	Logistics	Product Manager	5 years
E5	60	OEM c	Analytics	Senior Data Scientist	4 years
E7	62	Supplier d	Controlling	Head of CSCO	19 years
E8	55	Consulting e	Supply Chain	Capacity Manager	8 years
E9	58	Consulting f	Supply Chain	Controller	6 years

also Opriel et al. 2021). In the next step, we transferred them into a set of design requirements for an IOIS for sovereign data exchange. The result from the Diagnosis cycle is a set of 37 *design requirements* (see Table 3). We triangulated requirements from different data sources, such as field data from the case (e.g., calls, meetings, workshops, documentation, notes, slides, or concept documents), interviews with nine experts from seven different companies (see Table 2) (Opriel et al. 2021). The interview study was conducted by a single researcher in three phases, with three interviews each. Interviews were conducted both in-person and virtual and were recorded for analysis with the consent of the participants. Subsequently, these recordings were transcribed and analyzed using MAXQDA 2020. To ensure internal validity within the analysis, particularly regarding the coding process (Saldaña 2015), the analysis was primarily carried out operationally by one researcher from the ADR team. However, the results were validated and discussed by the other research team members. A theoretical saturation was reached within the last phase, and no new insights were gained. In the concluding phase, a total of 37 design requirements were thoughtfully extracted from the raw data. We analyzed 397 raw statements from field data, interviews, and literature and condensed them to 37 design requirements. The coding process followed a structured, iterative approach, commencing with an initial exploratory open coding phase (Corbin and Strauss 2014). This was further refined by aligning the coding framework with the IEEE software requirements specification (SRS) (ISO/IEC/IEEE 2018) for structure and comparability to categorizing the statements into business and system requirements. Each design requirement addresses at least one *design problem*, as indicated in the respective tables (see Table 3) and a mapping diagram (see Appendix D).

5 Extracting Design Features: 1st Design and Implementation Cycle

5.1 Design Cycle

Based on the initial exploration of the problem space and the formulation of design requirements, we conceptualized 18 design features (see Fig. 2). A detailed step-by-step illustration of the relationships between design challenges, requirements, and features can be found in Appendix D.

These design features are specific and artifact-near chunks of instantiated design knowledge (Schoormann et al. 2021). To uncover the "intertwined relations between requirements and design" (Hesse and Paech 2016, p. 189), we first conceptualized all design features through a structured analysis of supported processes, (user) interfaces, sketched systems, source code, and system architectures (e.g., runtime-environment or database schema). The resulting features were analyzed and rated for relevance and clustered inductively afterward. We did this iteratively until no more changes were made. In this process, we generated five categories within the 18 design features, which we labeled according to the summative content of design features (see Appendix C):

(1) Business Case Design Features: One of two ways to generate benefits from implementing a new software artifact is to either generate new revenue or reduce cost by optimizing existing processes (e.g., Boehm 2003). In our case, the software artifact should optimize existing inter-organizational processes through features fostering transparency (F01). In traditional supply chains, B2B data exchange occurs under contractual obligations, which leads to two issues: **First**, it is hard to identify business cases that produce revenue based on data sharing when data use is strictly demarcated and regulated. **Second**, sensitive information was not monetized in our case, making their valuation nearly impossible. To mitigate these issues, the design feature *quid pro quo* (F02) implements mechanisms to only share data once both parties share data of the same perceived value. The OEM shared planned production

Table 3 Design requirements for data sovereign IOISs and the corresponding problems they address (see Appendix B and Opriel et al. 2021)

ID	Design requirement	Addressed problems
R01	Positive ROI	P02, P09, P12, P14
R02	Appropriate KPIs	P05, P06, P14
R03	Transferable, adaptable, applicable	P02, P08, P13
R04	Low cost for implementation	P02, P08, P14
R05	Low know-how requirements	P02, P08, P13
R06	Mutual benefits	P08, P09, P13
R07	Accessible, exchangeable, combinable	P04, P12, P14
R08	Solve a specific problem	P11
R09	Collaborative approach	P02, P07, P08, P10, P11, P13, P15
R10	Foster trust	P01, P07, P09, P15
R11	No third-party data access	P15, P16
R12	Decentralized data storage	P03, P15, P16
R13	Runnable in arbitrary environments	P08, P13, P14
R14	Platform features	P05, P12, P13, P14
R15	Usage control mechanisms	P15, P16
R16	Link usage policies with contracts	P15, P16
R17	Traceability of data	P15, P16
R18	Log data access and usage	P15, P16
R19	Precise connection configuration	P03, P15
R20	Quick establishable connections	P01, P03, P12, P13
R21	Real-time data exchange	P03
R22	Agree mutually on connections	P05, P08, P15
R23	Up-to-date and correct data	P05, P06
R24	Accurate, consistent, complete data	P05, P06, P09, P10
R25	Define data types precisely	P05, P06
R26	Standardized and structured formats	P04, P06, P12
R27	Machine-readable usage policies	P01, P05, P15
R28	Mutually authenticate and authorize	P15, P16
R29	No secret changes in configuration	P07, P16
R30	Prevent malicious data extraction	P07, P15, P16
R31	Prevent data misuse	P07, P15, P16
R32	Data thriftiness	P07, P15
R33	Automatically exchange data	P03, P05, P06
R34	Retrieve data from host systems	P05, P06
R35	Standardized API	P03, P04, P06, P12
R36	Combine usage policies and payload	P04, P15, P16
R37	Provide a GUI with high usability	P13, P14

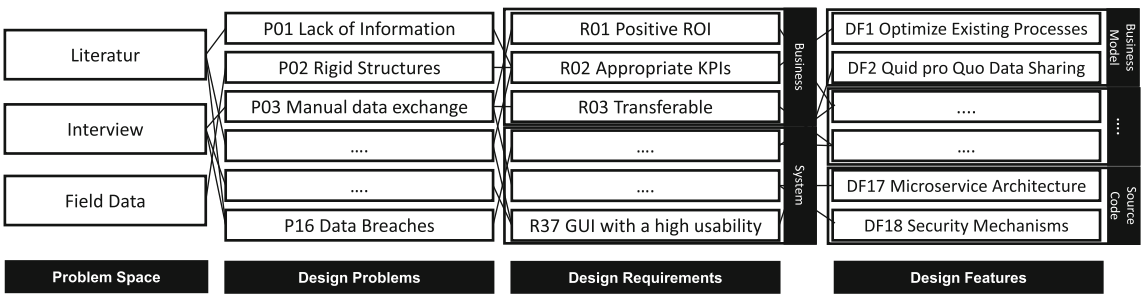

Fig. 2 Matching process of design problems to design features

demands in exchange for the suppliers' planned production outcomes. This results in data acting as a currency that can be exchanged for data of the same perceived non-monetary value. This approach avoids challenges arising from data valuation and prevents the generation of imbalances in data provider and data receiver relationships.

(2) Functionality Design Features: The company's guidelines or policies might prohibit third-party software or specific service providers. Subsequently, a necessary design feature is running a decentralized and self-administrated system in one's own IT authority (F03). The ADR team decided to make the software prototype more accessible with a front-end enabling users to visualize data (F04). A challenge for every connected system is semantic operability (Vernadat 2009). Due to differences in the information and definition of data types of both industry partners, the ADR team set up a glossary with precise definitions, general terms, and other relevant key vocabularies (F05), as well as to enable legitimate users of the system to establish new data connections quickly (F06). The software focuses on data exchange, management, and access (F07). Building the IOIS focused on these topics guarantees that it is complementary and adds value instead of generating redundancies to existing software solutions for demand and capacity bottleneck management. Leveraging existing standards for secure data exchange, the ADR team decided to implement the software based on the standards of the International Data Spaces (IDS) (F08).

(3) Data Access and Exchange Design Feature: Nine different data types were identified in two workshops and multiple telephone discussions. The team agreed on a subset of three specific types to be transmitted through the software: **(1) stocks**, **(2) OEM demands**, and **(3) supplier production information** (F09). The most critical issue rationalizing the implementation of P2P data sharing is that no third party is supposed to be involved in this process (apart from potential cloud service hosting providers) (F10). Generally, data sharing can be triggered by two mechanisms: push and pull. The data provider triggers a push, and a pull is triggered by the data consumer (F11). Applying a data pull assures a data provider that data are just transferred when necessary and demanded by the data consumer. The consumer can be assured that just essential information is received. Two different options were applied to pull data with the software: **First** is an on-demand pull, where a system or user requests the most up-to-date information. **The second** is a scheduled pull, which is triggered by the system. A daily number of allowed data refreshes limits the on-demand pull mechanism. On the one hand, it limits a malicious participant from pulling information at a high frequency (to get a live view), which would be an inefficient implementation of a data push strategy. On the other hand, it allows participants to pull information as flexibly as needed in daily business. The software should provide up-to-date information by avoiding manual inputs as much as possible. Subsequently, the software requires the end-to-end appliance of IT systems and APIs (F12). Fostering transparency within the supply network is the ultimate objective of the ADR study. Logging is used to provide transparency in the usage and access of information. As trust plays a significant role between humans and, therefore, operative personnel, logging in to IT systems like corporate firewalls is insufficient since employees usually do not have access to such systems. The software logs each data request under consideration of general data protection regulation (GDPR), which is visible to system users (F13).

(4) Usage-Control Design Features: As the software focuses on end-users, we argue that UC and its enforcement provide significant advantages against intentional and unintentional data extraction compared to current data handling. The software defines four UC policies – two defining rights to provide access to received data by other systems and two defining preconditions for data usage (F14):

- *Front-end access* Three levels of protection prevent data extraction from the front end. The first and most restrictive deny all data access via the front-end API. The second level allows access with front-end restrictions like disabled text selection or printing, while the third level allows unrestricted data access to the front end.
- *External access* A significant requirement of the single-case study's ADR team is to allow company systems to access received data for processing purposes. Conceptually, we distinguish between no access, access by authorized and certified IDS-compliant systems, and open access to all company-internal systems.
- *Data retention* It is a matter of defining how long data are accessible after receiving them. This leads to the self-deletion of data, which avoids unintentionally forgetting to delete information about a business partner. Additionally, the data provider can ensure that the receiver does not work with expired data when more recent information might be available, finally avoiding misconceptions and mistakes.
- *Data history* Restricts the number of data snapshots that can be stored, which is implemented as a limited FIFO (first in, first out) queue. A limited series of data snapshots already opens the potential for optimizations, like analyzing changes in planning or performing data-validity measures, while a comprehensive series opens opportunities to analyze planning changes or gain other possibly unintended insights.

To enforce described and technically specified usage policies, we use an XACML-based (OASIS 2013) architecture (F15) (see Appendix F).

(5) Source-Code Design Features: To gain trust and avoid unnecessary license costs, free and open-source software (OSS) without a copy-left policy can be used, like MIT or Apache 2.0 license (F16). Besides financial benefits, widely used OSS tends to be more secure because it relies on a broad community that checks and enhances the source code to make it more reliable and stable. Adopters of OSS also have the opportunity to check used libraries, which raises trust and confidence in the resulting software. From an architectural perspective, encapsulating atomic services as so-called microservices (F17) offers enormous advantages in maintenance, scaling, and control, especially in extensive and inter-organizational system landscapes (Jamshidi et al. 2018). In data sovereignty, security is a central aspect to be regarded (F18). Thus, a range of security standards is used, like TLS encryption for data transfers, encryption for the database, or basic authentication for users and APIs.

5.2 Implementation Cycle

In the 1st iteration of the implementation cycle, we implemented and visualized the design features as mock-ups for two reasons: **First**, mock-ups are an intuitive medium to foster discussion and agreement within the ADR team on what the software should look like and what it should be capable of. **Second**, they are a visual tool to help guide the developers in implementing the front end and are crucial in representing the problem space through an information system (Wand and Weber 1995). Figure 3 shows the mock-up of a production plan screen. The general layout was designed straightforwardly and consisted of four core elements (a navigation bar (left), a top header, a footer bar, and a central content field). The screenshot visualizes some design features. On the one hand, the design was aligned with an available open-source web framework (F04 and F16). On the other hand, it focuses strictly on processes (F01) and data exchange (F07). Filters enable selective visualization of information. With the final version of the mock-ups, 19 screens were designed. The results of this cycle were evaluated iteratively in a two-stage process. The starting point was several focus groups with the project managers of the stakeholders. Recurring discussions about the design were held within the focus groups, and the design decisions were made. Based on this, the steering committee approved the mockups and the associated requirements as the final decision-makers.

6 Building the Software Prototype: The 2nd Iteration of Design and Implementation Cycles

6.1 Design Cycle

In the 2nd design cycle, we used the design features from the 1st iteration and integrated them into the system architecture. It ensures that the data provider and receiver (OEM and 1st tier supplier) hold sovereignty over their data through policy enforcement and decentralized data storage. The back end follows the IDS Connector specification and enables the connection of in-house data sources such as ERP systems or warehouse management systems. The software prototype defines a standardized API that allows direct integration into other systems or connections via a bridge implementation. This enables transferring data to different back ends through IDS messaging protocols subject to UC policies (see Fig. 4).

6.2 Implementation Cycle

The design features outlined above cumulated in the implementation of the systems architecture. The front end was implemented and deployed on a web server. The back end was implemented in Java with the Spring Boot framework. OpenAPI specifications were developed to define the APIs used on the suppliers' side to implement the API within the SAP ERP system as a REST interface. On the side of the OEM, we used an existing API and integrated it. Using Java interfaces enables flexible integration of APIs. Based on the initial definition of the API for obtaining data from internal systems, we developed an information model holding all necessary data (see Appendix E). Particularly significant was the integration of UC design features into the information model. The emphasis was placed on data retention and data history. Whenever a data connection is established, basic connection information (e.g., date of connection establishment, connection partner, etc.) is stored by assigning a unique hash value to the connection. The information flow can be controlled and verified based on this unique connection key and the corresponding meta information. For example, it is possible to reduce the amount of information based on previously defined volumes, remove historical data, or restrict the total access to the data in terms of time.

6.3 Pilot Cycle and Evaluation

Given that the ADR project is embedded in an industry context, the resulting artifact was needed to produce value in an operational setting. The evaluation strategy follows a **proof-of-value** approach to assessing the usefulness of actual artifact deployment in the field (Nunamaker and

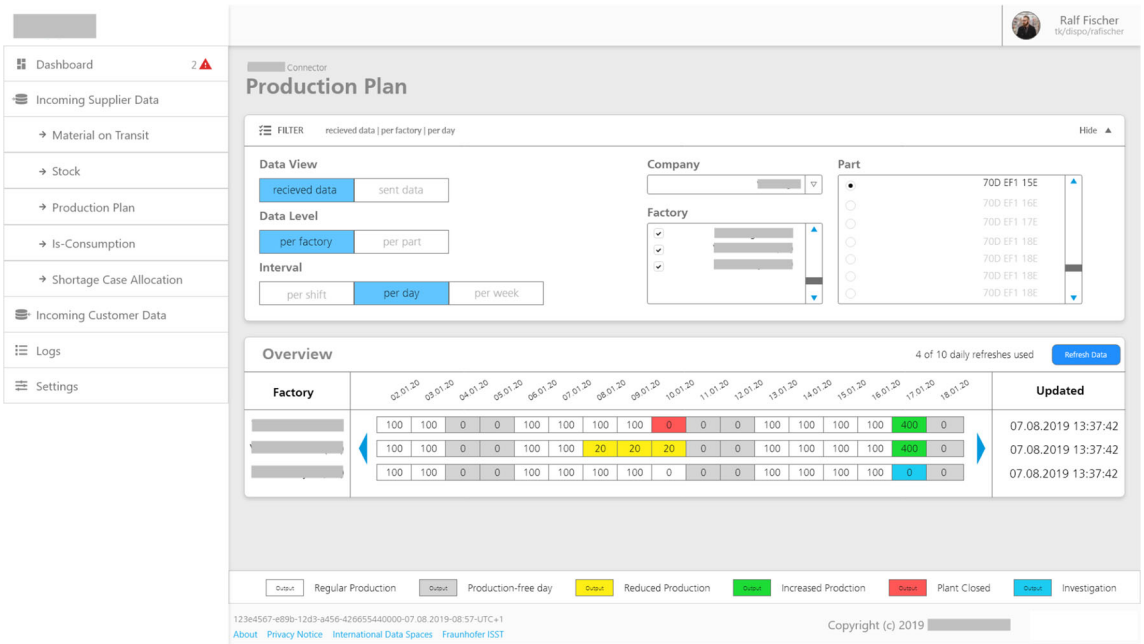

Fig. 3 Mock-up of the software's production plan page (Opriel 2021, p. 277)

Briggs 2012). For this, the artifact was deployed in a closed test phase to a narrow field of industry practitioners from both sides (OEM and 1st tier supplier) who would ultimately use it in an actual business environment. This was ensured since the project leads on the OEM and supplier side named individuals required for the piloting phase. The close test gave them access to the software and allowed them to exchange inventory and demand data as well as production information on four parts manufactured by the supplier and shipped to the OEM. The test was conducted with 13 user accounts (6 on each side and one additional API account on the OEM side). Table 4 illustrates the complete configuration protocol of the piloting phase.

The functionalities were gradually made available, and critical functions (e.g., usage control) were tested. Since there were no measurable KPIs to evaluate the artifact, we conducted a qualitative evaluation after **six months** of piloting (Prat et al. 2015; Venable et al. 2012). The group of pilot users included all users within the pilot system who were directly involved in the business process. From the piloting phase, we could extract **key observations** (see below). From a technical view, no significant errors occurred in the piloting phase. The prototype was used operatively to assess the value it can generate.

The evaluation process was based on a questionnaire and semi-structured interview with the opportunity to discuss statements and ratings. The questionnaire consisted of 32 questions divided into five blocks (spontaneous assessment, design and handling, functionality, relevance of displayed data, and open questions). The questionnaire was given to five individuals, who represented the sum of all users on both sides (OEM and supplier) that were involved with the four-part numbers during the pilot phase in procurement and production. It closed with a free text field

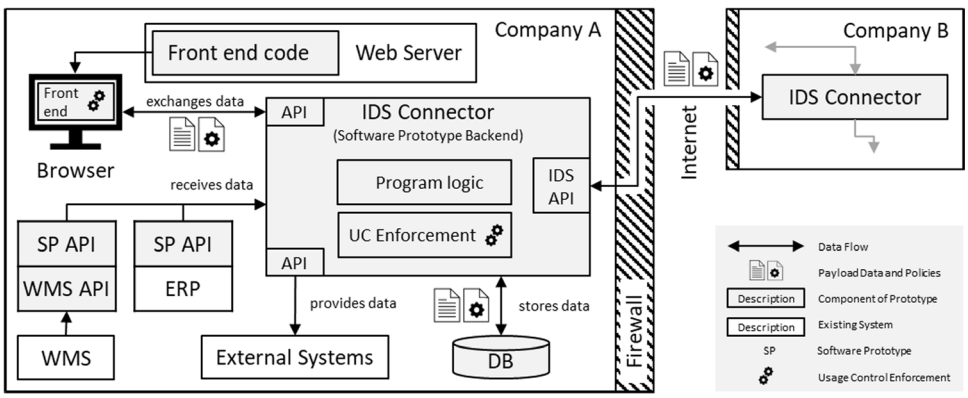

Fig. 4 The system architecture of the data sovereign IOIS

prompting participants to disclose open questions or suggestions. The quantitative evaluation was complemented by interviews with the five users of the pilot (see Table 5) that were held a few days after they had filled out the questionnaire. The interviews were held after the participants had filled out the questionnaire to go into in-depth discussions and collect feedback about the pilot. During the evaluation process, theoretical saturation was evident, and the project managers approved the pilot implementation requirements.

Based on this feedback, we condensed the evaluation results as follows. The participants were initially skeptical in the first cluster (**general assessment**). However, over time, the usefulness of the prototype materialized. Especially the shared data was something novel, i.e., "to see the data is quite a different thing than what we had so far." Yet, there is doubt whether the prototype will be used in the long run. Another point of critique was that the prototype did not cover all data but only a subset. This was to be expected since the prototype was not designed for the entire database at the time of testing. In terms of **design and handling**, users critiqued the color scheme ("a little bit grey in grey"). Generally, the prototype was perceived to be comprehensive and understandable, with the caveat that this point should be revisited once large data sets are visualizable. Handling of the prototype was intuitive for all users. However, they also wanted to implement multiple languages (German and Spanish). Some concerns permeated runtime when more data was available and specific downtimes when the connections had to be renewed.

In the **functionalities** cluster, users highlighted that the prototype "helps to get an overview" and saves the users "a lot of e-mails and phone calls." One of the major goals is fostering transparency. The accessible information on both sides (OEM and supplier) enhanced the users' sense of security since "If the range is enough until next week Wednesday, I am already a little more deeply relaxed." Another significant benefit of data availability is reducing "mentally tiring asynchronous e-mail traffic." The respondents highlighted areas for improvement, such as accessibility of information over a longer time span (weekly and monthly) or deploying the prototype in other fields (e.g., logistics). In terms of the **relevance of displayed data**, neither participants from management nor operative personnel recognized UC policies in the software. This is a good sign since they were implemented but did not impact operative work activities. Another interpretation of this result is that these UC policies were expected to be present since no one complained or highlighted any impact on their work. Another topic was the correctness of the displayed data. In this, we observe a distinction between those responsible for the case study that wanted the data to be displayed correctly. However, operative personnel assumed this data was correct, so this question of correctness was no topic for them. The users

Table 4 Key information about the piloting phase

Field	Description	
Duration	6 Months, 13 User Accounts (including one API account), 5 Users	
Functionality	Share information of four part numbers	
Transmitted Data	Inventory data (states: available, blocked, in-stock)	
Scope	Connection was limited to two plants (OEM and Supplier)	
Time limit	Unrestricted access to data	
Data update	Ten updates per day	
Usage Control	**Front-End Access:**	Level 1: Only for registered users
		Level 2: **Only for registered users; enforcement of the condition of use: no printing, no selecting/copying of text**
		Level 3: No access
	External Access:	Level 1: **Only for authorized systems**
		Level 2: Only for authorized systems that are at least certified with the IDS Base Security Profile; handover of the applicable Terms of Use
		Level 3: No access
	Storage Duration:	Store data for n minutes with $n \in N$ and $0 < n \leq \infty$. **(24 h)**
	Data history:	Store a maximum of n data points with $n \in N$ and $0 < n \leq \infty$ **(3 data points)**

The specific implementation in the piloting phase is highlighted in **bold**

Table 5 Participants in artifact evaluation

ID	Duration (min)	Company	Department	Position	Experience
E10	90	OEM a	DC-Management	DCM Manager	3 years
E11	56	OEM a	Logistics	Production Planer	3 years
E12	80	OEM a	Logistics	Head of Supply Chain	15 years
E13	68	OEM b	Logistics	Product Owner	25 years
E14	70	OEM b	Production	Capacity Manager	7 years

identified six additional areas positively affected by the implemented data sovereign IOIS (see Table 6).

7 Reflection and Formalization of Learning

The section presents design principles for data sovereign IOISs crafted by reflecting on the instantiated artifact and the ADR study's design process (Eisenhardt 1989; Sein et al. 2011). Reflection is one of the critical paths to formalize design knowledge in ADR studies (Möller et al. 2020; Sein et al. 2011). The purpose of design principles is to formalize the *essential* design knowledge collected in a design process and make it available to others (Chandra Kruse et al. 2015), by "reflecting upon what has been done" (Gregor 2009, p. 7) and pushing the findings beyond "a single success story" (Chandra Kruse and Seidel 2017, p. 180). For this, they require a degree of abstraction, which we achieve by decoupling the instantiated artifact from the application scenario and characterizing a *generalized problem* and a *generalized solution* (Lee et al. 2011). This (de-)abstraction process is pivotal since design principles should address a class of artifacts that solves a class of problems (Sein et al. 2011). These design principles need to incorporate the relevant knowledge that leads to the successful artifact, which ultimately means that we formulated the design principles reflectively on what we identified as critical to the artifact's success (see *provenance* in the description of each design principle below). Additionally, we connected the resulting design principles with the design features they address in a mapping diagram to visualize the logical connection (see Appendix D). Table 7 lists the specific problem, class of problem, specific solution, and class of solution for our ADR study. In light of the initial research question, we derived the design principles focusing on the underlying key issue: to design the data sovereign IOIS so that users on the OEM and supplier sides are willing to share sensitive data.

We follow the codification mechanism of Gregor et al. (2020), as it aggregates prior knowledge on design principle formulation and has well-structured components. However, we did not use the tabular visualization but formulated the design principles as plain text drawing from the concepts of *aim*, *context*, *mechanisms*, and *rationale*. Our work builds on *justificatory knowledge*, from which we draw argumentation on "why the design works" (Gregor and Jones 2007, p. 322). In particular, we use justificatory knowledge to warrant the design principles we have extracted reflectively from the design process and design product (Goldkuhl 2004; Möller et al. 2022b): **First**, since our research deals with data sovereignty and the inter-organizational exchange of sensitive information, we draw from existing research on *UC policies* (e.g., Park and Sandhu 2004). **Second**, we draw from theories such as *Principal-Agent Theory*, *Justice Theory*, and *Deterrence Theory* to explain and warrant why the design principles we extract and formulate work (Goldkuhl 2004; Gregor and Jones 2007; Möller et al. 2022b). Below, we explain each design principle with a rationale, provenance, prescriptive statement, and justificatory knowledge.

7.1 Design Principle 1: Quid Pro Quo to Enable Equal Data Sharing and Circumvent Monetary Valuation of Data

Rationale Traditionally, supplier-customer relationships are imbalanced, as one party dominates the other and may dictate the use of specific software. Given the requirement to foster collaboration (R09), the first design principle calls for mutual benefits in data sharing. It incentivizes companies to share data by illustrating unambiguous benefits for both parties, such as reciprocally receiving sensitive and potentially valuable data. That notion is shared by Lempinen et al. (2012), who propose a design principle for balanced IOIS. Their argumentation illustrates the willingness to adopt the IOIS based on the users' balance of interests and power.

Provenance The design principle originates from the design process, in which strategies were sought to make both parties share data. We observed the dilemma of finding monetary value for data sharing between the OEM and the supplier, which, in our case, was not feasible or realistic to solve. This resulted in the need to implement data trading (data for data) with the boundary of it having perceived equal value and excluding monetary compensation.

The resulting design principle is: *To provide companies with an incentive to share data, in scenarios where at least*

Table 6 Six areas for application proposed by pilot users

Area	Description
Avoiding E-Mail Communication	The data sovereign IOIS reduced the e-mail 'ping-pong' required to collect information about supplier depth. This resulted in a reduction of necessary time (about 10–15 min)
Optimizing Transport Capacities	Vehicles often have leftover capacity, which was used ad hoc. Transparency about the depth of each part enabled optimized usage of transport capacities
Detection of arrived deliveries	The supplier can detect if and when the delivery has been successfully delivered to the OEM
Identification of Product delays	Unexpected problems in production can be assessed quickly. This results in fewer and shorter delivery delays, which can be communicated swiftly
Detection of Quality Problems	Both parties can identify parts that have been shifted to blocked stock. As a result, reworking and special transports can be planned more efficiently
Improvement of Personnel Planning	Information related to personnel can be used to assess (e.g., by checking short notification vacation) whether shifts can be taken out

Table 7 Phases of abstraction and de-abstraction of problems and solutions in DSR (Lee et al. 2011; Opriel 2021, p. 206)

Phase	Application to our study
Specific Problem	Exchange and use of sensitive inventory, production, and demand information in automotive supply networks between Tier-1 and OEM in the Automotive industry
Class of Problem	Exchange and use of information worthy of protection between two parties, each of which possesses equivalent information that is of benefit to the other party
Specific Solution	Data Sovereign IOIS for the exchange of sensitive inventory, demand, and production information in automotive supply networks between OEM and tier-1 under consideration of data sovereignty using usage control mechanisms
Class of Solution	Data Sovereign IOIS for the exchange of structured, sensitive data

one party has reservations about exchanging data, equivalent data should be offered, with offsetting variable criteria such as scope or quality criteria (e.g., accuracy or timeliness), in return for the release of data, to overcome valuation problems of data, to reduce positions of power, and to make unknown benefits of data tangible for others.

Justificatory Knowledge Since we faced the situation that this value cannot be attached monetarily, the solution was to identify data of equal value in use (e.g., planning data against planning data) and embed that data can only be shared if both parties share data of the same perceived value. This can be warranted by *justice theory* (e.g., Blau 1964; Markovsky 1985) and *social exchange theory* (e.g., Blau 1964). The suppliers and the OEM must get data of perceived equal value. The exchange of necessary information for the successful delivery of a service may be justified, whereas additional requested information may be seen as unnecessary and, therefore, require further compensation. The design principle aligns with *justice and social exchange theories* by providing a mechanism that overcomes complex discussions and negotiations regarding data value. Justice is established if the receivers exchange equivalent information with an expected and perceived equal value and equal cost. With regards to the developed software, *quid pro quo* is applied as a central data-sharing mechanism to overcome problems of assessing data valuation and to foster mutual benefits and, thus, collaboration. Establishing a data-sharing connection between two participants provides equivalent information, i.e., stock versus stock and demand versus output, and is a mandatory component (see Fig. 5, point A: "Select Data").

7.2 Design Principle 2: Open-Source Software and Components to Generate Information Transparency

Rationale The second design principle faces the challenge of data providers mistrusting the data receiver and handling its data according to the agreed-upon terms. Required trust is lowered significantly through enforcing UC policies. Nevertheless, the resulting IOIS is provided by a company or community, meaning that the user must ultimately trust the software provider. For example, the user must be confident that no backdoor exists to bypass UC policies

maliciously. The open-source paradigm drastically enhances transparency and enables the provider and the receiver to inspect the underlying source code of the IOIS. Nevertheless, open source does not necessarily refer to free software or that the source code may be used at will (Ven et al. 2008). The latter can be achieved through negotiating and enforcing non-disclosure agreements to enable source-code audits. Additionally, remote attestation can ensure that the recipient did not change the source code or the configuration (Gu et al. 2008).

Provenance The design principle was derived to give all parties maximum transparency in the technical development and execution of the software. Even though both parties had a trusting relationship, providing technical transparency proved critical. This was a crucial demand from the supplier to obtain the technical means to ensure that they can assess what happens to their data.

Thus, the design principle prescribes fostering transparency on a technical level to mitigate potential trust issues in proprietary or closed software: *To gain the trust of companies in the software that is developed and used on both – data sender and recipient – sides, open source technologies should be used, the core application should be made available as open source, and remote attestation mechanisms should be provided, to make it possible for users to verify the correct functioning of the program logic (proof of trust), so that the software provider and ultimately the software itself can be trusted.*

Justificatory Knowledge Principal-agent theory explains how organizations (actors) interact under knowledge asymmetries and how they can be overcome by the principal commissioning an agent to provide services (Braun and Guston 2003). In supply networks, this relationship manifests as customer–supplier relationships. In most cases, customers are unable or unwilling to provide the service themselves due to a lack of resources (e.g., business networks, machinery, equipment, or processes). In the context of developing the software, the agent (the developer of the software) has the knowledge and skills to build the software. If the software remained proprietary (i.e., closed source), the information asymmetry would be fostered due to the principal's inability to catch up with the knowledge of how the software was implemented. As the software aims to reduce power positions, the principal agent theory is also addressed by giving the principal transparency over the code. The same applies to the relationship between the software users, who are themselves in a principal-agent situation (i.e., supplier and customer). In case the customer (e.g., OEM) is the initiator who rolls out the software to all its suppliers, the open-source approach

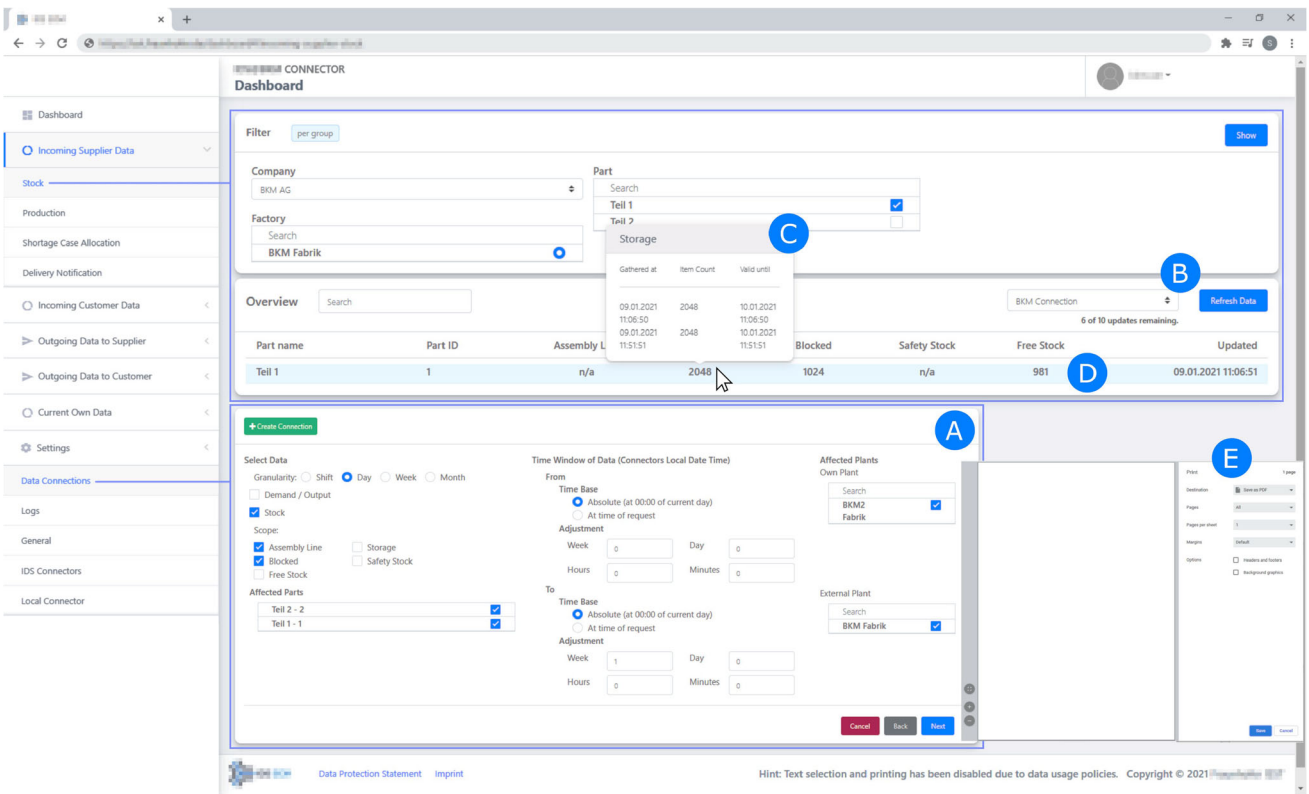

Fig. 5 Five design principles are instantiated in the artifact. A = fine-grained access and quid pro quo (DP 1 & 3), B = pull requests (DP 4), C, D, and E = instantiations of UC policies (DP 5)

also addresses the bilateral relationship between the customer (principal) and supplier (agent). Within the developed IOIS, an open-source approach is, on the one hand, pursued by using open-source libraries. On the other hand, the ADR team's researchers provided the source code of the developed software to the industry partner.

7.3 Design Principle 3: Fine-Grained Data Access to Avoid Unnecessary Data Exchange

Rationale Due to the information's sensitivity, the data must be transferred as precisely and condensed as possible. Transferring unnecessary, potentially sensitive data out of the initial purpose's scope is a pitfall for additional sender and receiver issues. Subsequently, the IOIS requires fine-grained, highly detailed configurations that can strip away all non-relevant information. For example, if a supplier wants to increase reaction time in quality-related cases, exchanging information about how many parts are in quality checking might be sufficient. Other types of information, e.g., free stock, are irrelevant and should not be transferred. An advantage is that fewer data must be stored, as non-relevant data are excluded. Given these data's frequency and volume, they can amount to a significant size due to several thousand OEM suppliers.

Provenance The design principle was developed to avoid sharing unnecessary data with potential sovereignty implications. This proved to be critical since acquiring additional data that was of no use to the data receiver needed to be prevented urgently. Not adhering to fine granular access would lead to additional barriers in data sharing, which is why the shared data needs to be meticulously demarcated. Subsequently, the design principle resulted from the key finding that data must be explicitly shared for a context, a demand, and highly restricted.

The resulting design principle reads as follows: *To enable users to provide other companies with a precisely definable data view with adequate data granularity (i.e., data scope), in special cases like temporary bottlenecks, data connections should be precisely configurable to avoid the misuse of unnecessarily additional available data.*

Justificatory Knowledge A fine granular adjustability of exchanged information helps to achieve a relationship that is perceived to be justified or 'fair' between the data provider and receiver (justice theory, social exchange theory) since data of equal value can be configured on a detailed level. Due to omitting unnecessary information, the value of the information provided but not needed by the receiver does not need to be rewarded. In the context of the developed IOIS, fine-grained access is integrated into the definition of data connections, allowing for precise definition, e.g., to transfer information types, parts, time windows, or affected own and partner's plants (see Fig. 5, point A).

7.4 Design Principle 4: P2P-Based Pull Strategy to Reduce Target Surface for Cyber-Attacks and Data Leaks

Rationale Recent reports on data leaks and warnings of cyber-attacks (e.g., Campbell 2019; Cowley 2018) demonstrate that centralized service platforms are a target due to the enormous amount of information they store. Next to customers and their suppliers' data, these platforms store information about the extensive supply chain network, making them an ideal target. Sensibly exchanging sensitive information is precious for attackers due to its high relevance and potential meaning for companies. Subsequently, the fourth design principle, P2P-based pull (DP4), recommends following a peer-to-peer strategy to reduce the probability of successful cyberattacks. Instead of data being pushed, they should be requested and pulled to reduce unnecessary information dissemination. Consequently, the data provider gets notified when the data receiver demands transaction authorization, and the recipient pulls the data.

Provenance The design principle was developed to avoid disseminating more information than is necessary to a data receiver. Additionally, the case partners did not want to implement a platform but required the data to be shared only bilaterally on-demand through pull queries. In alignment with DP3, this proved critical to avoid generating unnecessary data sovereignty complications.

To summarize, the design principle is as follows: *To give companies maximum control over their data, to avoid involving third parties, and to release as little data as possible within the general framework of data exchange, data storage should be decentralized, and data transfer should be triggered by the data owner or systems controlled by the data owner, in order to prevent power positions of possible service providers, to reduce data misuse and to avoid potential data breaches on the one hand for service providers and to mitigate them on the other hand for data users.*

Justificatory Knowledge A P2P-based pull allows the data provider to monitor when information is fetched and most likely to be processed by the data consumers. This mechanism has a corresponding deterrent effect in the context of data misuse on the part of the data user and reduces the risk at the organizational level (deterrence theory, e.g., Straub and Welke 1998). In the context of the IOIS, peer-to-peer communication is integrated with the data exchange process by two mechanisms to request data: Scheduled pulls and on-demand updates, which were

utilized as a part of the IOIS piloting process (Fig. 5 – point B).

7.5 Design Principle 5: UC Policies to Implement Data Sovereignty Technically

Rationale The core of the developed IOIS is enforcing usage control policies, which is the fifth design principle. While there are many different types of policies, as Eitel et al. (2021) show, a good starting point is obligations on when to delete data based on the usage and whether their processing purpose is fulfilled or based on a pre-defined validity window. These policies are essential, as they are the primary argumentative for sharing sensitive information if enforced strictly and followed precisely.

Provenance Implementing UC policies proved to be critical since they enabled the configuration of data usage, which resonated positively with the OEM and supplier. Ultimately, the parties agreed that the added value of technical UC policies was so great that they had to be incorporated in the artifact.

The resulting design principle can be defined as follows: "*To assure data issuing companies that data will only be used by their interests, in the systems of the data receiver, usage control should be exercised and enforced to reduce the probability respectively prevent misuse of data.*"

Justificatory Knowledge UC mechanisms on the data receiver's side can have different characteristics. In addition to those mechanisms used in the case study, policies like "log access and processing of information at a prescribed logging service" can foster deterrence of potential misuse. Even the simple ability of such mechanisms (e.g., dummy cameras) effectively influences users' behavior. According to deterrence theory, this perception of being punished or sanctioned is already enough to influence rule compliance positively (Wall et al. 2016). The developed software itself is equipped with four UC policies. Three are visible via the front end: Data retention (see Fig. 5 point C), where currently fetched information is valid for 24 h and will be deleted by policy enforcement afterward. As data history is configured to allow two snapshots, the other two fetched updates were overwritten and are no longer available. The front-end access policy restricts selection, and thus, copying text like at point D and printing via the browser's dialog is disabled by showing a blank page (see Fig. 5, point E). Subsequently, the justificatory knowledge is UC (Park and Sandhu 2004; Pretschner et al. 2006), and practitioner knowledge (Gregor and Hevner 2013) is engraved in the IDS's architectural guidelines (Otto et al. 2019).

8 Discussion

8.1 Reflecting on the Study

Generalizability of the results is a cornerstone of ADR projects (Sein et al. 2011). Our study proposes an artifact and design principles that address a highly typical scenario in different supply chains, i.e., that of an OEM and a supplier. Given this, the interplay of our results provides a concrete instantiation (e.g., March and Smith 1995) and generalized design knowledge that exceeds the confinements of the specific application scenario (e.g., Sein et al. 2011). Subsequently, we argue that the solution and design principles hold value on three levels. **First (1)**, while our study was done in the Automotive industry, the scenario of an OEM and a supplier sharing sensitive information for benefits, such as efficiency, is necessary and typical for every supply chain (e.g., fashion, food, or manufacturing) (Mentzer et al. 2001). Contributing to this is that the data that were actually exchanged (e.g., production data) are needed in every supply chain. While the complexity scale might be higher in the Automotive industry, our key findings can generally provide knowledge for the design of data sovereign IOIS for any OEM and supplier relationship. **Second (2)**, the same relationships that required optimization of capacity management exist in more supply chain constellations (e.g., in tier two or three supplier relationships). Subsequently, we see additional value in exploring and adapting data sovereign IOIS based on our findings alongside data sharing transactions upstream and downstream in the supply chain. **Third (3)**, while focusing on capacity management as a narrow piece of supply chain activities, we see ample opportunities to deploy such an artifact in different scenarios. Risk management in supply chains is a scenario with high face validity, which would benefit from efficient inter-organizational data sharing. The ADR's industry practitioners in the study were not selected randomly and may form a set specific to the Automotive industry, which fostered the adoption of the developed IOIS. We faced this issue by regularly presenting the current state of the IOIS to different audiences and requesting their opinion and thoughts. None of the feedback led to doubt that the solution might not be transferable to other business relationships and scenarios (see also Table 6). Even though the ADR study is bound to a single case, the solution can be adopted in other scenarios of sharing sensitive information with partners and different scenarios (as also propositioned by users in the evaluation) (Seddon and Scheepers 2012). Aside from shifting aims, e.g., towards ecological footprint, the concept of the software can be transferred to other domains like retail, where business partners need to collaborate. Also, the exchanged information could be extended towards equally or less sensitive

information, which in the latter case still reduces the risks of "losing" them.

The study showed that our results hold **validity** for practice. Significantly, these technical means protect a company's data sovereignty, which can help them sustain a global competitive advantage through protection against copying information. According to the involved industry partners, the developed IOIS led to a reduction in supply bottleneck situations. In addition, control over material flows could be improved, resulting in less effort due to avoiding e-mail checkbacks. The involved supplier reported that the system ensured the arrival of sent goods and earlier adjustments of internal processes (e.g., re-planning of shifts or preparation of measures) if anomalies were detected (e.g., booking received goods into quality check stocks at customers' sites). Also, a positive impact on the ecological footprint was reported due to optimizations of the remaining load capacities of transports, which were used to ship parts with the shortest stock ranges at the customer's site. The piloting phase and evaluation show that the artifact (and the corresponding design principles) fulfilled its goal: promote inter-organizational data sharing between an OEM and a supplier, which was achieved in concept but also in practice in a real business context.

8.2 Contributions to Practice and Research

From a **practice perspective**, the ADR study shows how to design an IOIS to protect corporate knowledge while sharing it with business partners. This is an example of a typical industry relationship between a car manufacturer and a supplier. In light of the growing relevancy of data sovereignty and sharing (e.g., see the European Strategy for Data European Commission 2020), it describes how integrating UC policies in IOISs can work and complement trust in data-sharing parties. Extending the instantiation, we propose design principles to enrich the knowledge base on IOISs through codified prescriptive design knowledge (Seidel et al. 2017). Equipping companies with technical means to exchange sensitive information sovereignly is a promising source to tap into leveraging data's completely new – yet undiscovered – potential. Using this newly available information to a prescribed extent could build the required trust for new business cases or optimizations of supply-chain processes (e.g., Richter and Slowinski 2019).

The main **research contribution** of this paper is a set of five design principles that make design knowledge available to assist researchers and practitioners in developing IOISs for sharing sensitive data demanding protection (generalized class of solution) (Cross 2001). Subsequently, we contribute to accumulating design knowledge on that particular class of artifact (vom Brocke et al. 2020). As these design principles were incorporated in the designed software artifact with positive feedback from practitioners, they manifest valuable design knowledge that can be used by others in similar scenarios to design data sovereign IOIS (Chandra Kruse and Seidel 2017). This is supported even more since users in the evaluation identified benefits that were not intended initially. For research, we contribute one of the first real-world case studies on inter-organizational data sharing adhering to technically implemented UC policies. Given that the European research landscape currently pursues precisely this type of research, our work is a starting point for others to build on. Still, it also provides key findings relevant to many other cases. One example worth highlighting is the difficulty of establishing a monetary equivalent compensation in the data-sharing process (at least in this case). From this, we derive a design principle that embodies a basic logic, namely that data sharing may depend not only on monetary values but also on the perceived equivalence of data. Educators in IS research could use our case as an illustrative example for ADR studies in larger-scope design projects. This is relevant since data sovereignty – and the data economy as a whole – become more relevant every day. Tangible cases that explicitly show how to create value in practice with IS researchers' toolset are desperately needed.

Lastly, our findings can motivate **further research**. Researchers can make use of our findings and apply them in a multi-case study to validate our observations and conclusions. One aspect is certainly the analysis of the acceptance of the created design knowledge in other instantiations based on the unified theory of acceptance and use of technology framework (Venkatesh et al. 2012). Also, the embedded UC technology may be enlarged and combined with other approaches, e.g., Digital Rights Management, Data Loss Prevention, or Blockchain. With a mid-level of abstract knowledge (Gregor and Hevner 2013), the design principles can evolve with further reflections and refinements to ultimately theories (Sein et al. 2011).

8.3 Limitations

Naturally, our study is subject to **limitations**. The IOIS includes system-invasive UC measures, such as disabling printing or text selection. Still, these measures are not a complete guarantee of data protection, as they can be bypassed using screenshots or disabling them in browsers' development consoles. Nevertheless, they significantly reduce the efficiency and benefit of possible malicious behavior. In the case of screenshots, they would be much more challenging and work-intensive to process compared to a received spreadsheet. Evading such safeguards also requires malicious acts beyond deniability due to ignorance. Another legal constraint is that some business-

relevant data might need to be stored for at least two years. A potential solution would be encrypting information with the data provider's public key in the data receiver's system. While the study was in-depth, its broadness is severely limited to the bilateral relationship of one OEM and one 1st tier supplier. While 1st tier suppliers usually produce parts highly specific to large OEMs, 2nd, 3rd, or n-tier suppliers typically have a more generic portfolio targeted to multiple customers. Potentially, that could be the source for data sovereignty pitfalls, as getting insights into generic production portfolios of n-th tier suppliers might reveal information about other customers. Yet, if customers can observe information, they gain critical transparency that potentially aggravates bullwhip effects (Cachon and Fisher 2000; Lee et al. 1997, 2000).

It is questionable if all design principles can be applied to their full extent. Prior described aspects regarding business cases, which may not allow publishing the software as open source or limit usage-policy enforcement due to the involvement of necessary legacy systems, are valid hurdles. Furthermore, quid pro quo assumes that both data exchange participants have equivalent information. This might not be the case in each relationship, meaning that asymmetries can exist. For example, a supplier could provide information about their supply network, while an OEM may give information on downstream retailers or end customers. Asymmetries could occur because desired information is simply not generated or collected. For instance, a supplier of natural materials like wool might not have a system to collect real-time information about shorn sheep. Lastly, equivalent information might not interest one participant, e.g., while OEMs are interested in complying with social and environmental standards and documenting records, the same interest might not fully apply to suppliers. In the scope of research limitations, a deep single-case study might be affected by a sampling error resulting from a problem-driven approach and limitation to only two industry partners and only five users.

Funding Open Access funding enabled and organized by Projekt DEAL.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12599-024-00893-4.

References

Ahmed W, Omar M (2019) Drivers of supply chain transparency and its effects on performance measures in the automotive industry: case of a developing country. Int J Serv Oper Manag 33(2):159–186. https://doi.org/10.1504/IJSOM.2019.10022079)

Alboaie S, Cosovan D (2017) Private data system enabling self-sovereign storage managed by executable choreographies. In: Chen LY, Reiser HP (eds) Distributed applications and interoperable systems. Springer, Cham, pp 83–98. https://doi.org/10.1007/978-3-319-59665-5_6

Aldoori J (2019) The impact of supply chain collaboration on performance in automotive industry: empirical evidence. J Ind Eng Manag 12(2):241–253. https://doi.org/10.3926/jiem.2835

Amoore L (2016) Cloud geographies: computing, data, sovereignty. Progr Hum Geogr 42(1):4–24. https://doi.org/10.1177/0309132516662147

Blau PM (1964) Justice in social exchange. Sociol Inq 34(2):193–206. https://doi.org/10.1111/j.1475-682X.1964.tb00583.x

Boehm B (2003) Value-based software engineering: reinventing. ACM SIGSOFT Soft Eng Not. https://doi.org/10.1145/638750.638775

Braun D, Guston DH (2003) Principal-agent theory and research policy: an introduction. Sci Publ Policy 30(5):302–308. https://doi.org/10.3152/147154303781780290

Britannica (2023) Sovereignty. https://www.britannica.com/topic/sovereignty. Accessed 9 Oct 2023

Cachon GP, Fisher M (2000) Supply chain inventory management and the value of shared information. Manag Sci 46(8):1032–1048. https://doi.org/10.1287/mnsc.46.8.1032.12029

Campbell J (2019) FBI says hackers are targeting us auto industry. https://edition.cnn.com/2019/11/20/politics/fbi-us-auto-industry-hackers/index.html. Accessed 5 Aug 2024

Cash JI, Konsynski BR (1985) IS redraws competitive boundaries. Harv Bus Rev 63(2):134–142

Catena-X (2024) Landing page of Catena-X. https://catena-x.net/de/. Accessed 16 Jul 2024

Chandra Kruse L, Seidel S (2017) Tensions in design principle formulation and reuse. In: Proceedings of the 12th international conference on design science research in information systems and technology, Karlsruhe

Chandra Kruse L, Seidel S, Gregor S (2015) Prescriptive knowledge in IS research: conceptualizing design principles in terms of materiality, action, and boundary conditions. In: Proceedings of the 48th Hawaii international conference on system sciences, Hawaii. https://doi.org/10.1109/HICSS.2015.485

Choudhury V (1997) Strategic choices in the development of interorganizational information systems. Inf Syst Res 8(1):1–24. https://doi.org/10.1287/isre.8.1.1

European Commission (2020) A European strategy for data. COM 66

Corbin J, Strauss A (2014) Basics of qualitative research: techniques and procedures for developing grounded theory. Sage

Cowley S (2018) Big red flag: automakers' trade secrets exposed in data leak. https://www.nytimes.com/2018/07/20/business/suppliers-data-leak-automakers.html. Accessed 11 Nov 2022

Cross N (2001) Designerly ways of knowing: design discipline versus design science. Des Issues 17(3):49–55. https://doi.org/10.1162/074793601750357196

Doran D (2004) Rethinking the supply chain: an automotive perspective. Supply Chain Manag Int J 9(1):102–109. https://doi.org/10.1108/13598540410517610

Eisenhardt K (1989) Building theories from case study research. Acad Manag Rev 14(4):532–550. https://doi.org/10.2307/258557

Eitel A, Jung C, Brandstädter R, Hosseini SH, Bader S, Kühnle C, Birnstill P, Brost G, Gall M, Bruckner F, Weißenberg N, Korth B (2021) Usage control in the international data spaces. Position Paper Version 3.0, International Data Space Association

Fassnacht M, Benz C, Heinz D, Leimstoll J, Satzger G (2023) Analyzing barriers to data sharing among private sector organizations: combined insights from research and practice. In: Proceedings of the 56th Hawaii international conference on system sciences, Hawaii

Gaia-X (2024) https://gaia-x.eu/who-we-are/vertical-ecosystems/. Accessed 16 Jul 2024

Goel V, Pirolli P (1992) The structure of design problem spaces. Cogn Sci 16(3):395–429. https://doi.org/10.1016/0364-0213(92)90038-V

Goldkuhl G (2004) Design theories in information systems: a need for multi-grounding. J Inf Technol Theor Appl 6(2):59–72

Gregor S, Hevner AR (2013) Positioning and presenting design science research for maximum impact. MIS Q 37(2):337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Gregor S, Jones D (2007) The anatomy of a design theory. J Assoc Inf Syst 8(5):312–335. https://doi.org/10.17705/1JAIS.00129

Gregor S, Chandra Kruse L, Seidel S (2020) The anatomy of a design principle. J Assoc Inf Syst 21(6):1622–1652. https://doi.org/10.17705/1jais.00649

Gregor S (2009) Building theory in the sciences of the artificial. In: Proceedings of the 4th international conference on design science research in information systems and technology, Malvern. https://doi.org/10.1145/1555619.1555625

Gu L, Ding X, Deng RH., Xie B, Mei H (2008) Remote attestation on program execution. In: Proceedings of the 3rd ACM workshop on scalable trusted computing, New York, pp 11–20. https://doi.org/10.1145/1456455.1456458

Hellmeier M, von Scherenberg F (2023) A delimitation of data sovereignty from digital and technological sovereignty. In: Proceedings of the 31st European conference on information systems, Kristiansand

Hellmeier M, Pampus J, Qarawlus H, Howar F (2023) Implementing data sovereignty: requirements & challenges from practice. In: Proceedings of the 18th international conference on availability, reliability and security, New York. https://doi.org/10.1145/3600160.3604995

Hesse T-M, Paech B (2016) Documenting relations between requirements and design decisions: a case study on design session transcripts. In: Daneva M, Pastor O (eds) Requirements engineering: foundation for software quality. Springer, Cham, pp 188–204

Hevner A, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105. https://doi.org/10.2307/25148625

Holland CP (1995) Cooperative supply chain management: the impact of interorganizational information systems. J Strateg Inf Syst 4(2):117–133. https://doi.org/10.1016/0963-8687(95)80020-Q

Hummel P, Braun M, Tretter M, Dabrock P (2021) Data sovereignty: a review. Big Data Soc. https://doi.org/10.1177/2053951720982012

ISO/IEC/IEEE (2018) Systems and software engineering: life cycle processes – requirements engineering. Vernier

Jamshidi P, Pahl C, Mendonça N, Lewis J, Tilkov S (2018) Microservices: the journey so far and challenges ahead. IEEE Softw 35:24–35. https://doi.org/10.1109/MS.2018.2141039

Jarke M, Otto B, Ram S (2019) Data sovereignty and data space ecosystems. Bus Inf Syst Eng 61(5):549–550. https://doi.org/10.1007/s12599-019-00614-2

Jarke M (2020) Data sovereignty and the Internet of Production. In: Dustdar S et al (eds) Advanced information systems engineering. Springer, Cham, pp 549–558

Johnston HR, Vitale MR (1988) Creating competitive advantage with interorganizational information systems. MIS Q 12(2):153–165. https://doi.org/10.2307/248839

Jussen I, Möller F, Schweihoff J, Gieß A, Giussani G, Otto B (2024) Issues in inter-organizational data sharing: findings from practice and research challenges. Data Knowl Eng 150:102280. https://doi.org/10.1016/j.datak.2024.102280

Jussen I, Schweihoff J, Möller F (2023) Tensions in inter-organizational data sharing: findings from literature and practice. In: IEEE 25th conference on business informatics. https://doi.org/10.1109/CBI58679.2023.10187530

Kern J, Wolff P (2019) The digital transformation of the automotive supply chain – an empirical analysis with evidence from Germany and China: case study contribution to the OECD TIP Digital and Open Innovation project

Khurana M, Mishra P, Singh AR (2011) Barriers to information sharing in supply chain of manufacturing industries. Int J Manuf Syst 1(1):9–29. https://doi.org/10.3923/ijmsaj.2011.9.29

Klein S, Poulymenakou A, Riemer K, Papakiriakopoulos D. Gogolin M, Nikas A (2005) IOIS and interfirm networks-interdependents and managerial challenges. In: Eom SB (ed) Inter-organizational Information Systems in the Internet Age, IGI Global, pp 170–213

Kukutai T, Taylor J (2016) Data sovereignty for indigenous peoples: current practice and future needs. In: Kukutai T, Taylor J (eds) Indigenous data sovereignty. ANU Press, pp 1–22

Kumar K, van Dissel HG (1996) Sustainable collaboration: managing conflict and cooperation in interorganizational systems. MIS Q 20(3):279–300. https://doi.org/10.2307/249657

Kumar RS, Pugazhendhi S, Muralidharan C, Murali S (2018) An empirical study on effect of information sharing on supply chain performance – the case of Indian automotive industry. Int J Logist Syst Manag 31(3):299–319. https://doi.org/10.1504/IJLSM.2018.095821

Lee HL, Padmanabhan V, Whang S (1997) Information distortion in a supply chain: the bullwhip effect. Manag Sci 43(4):546–558. https://doi.org/10.1287/mnsc.43.4.546

Lee HL, So KC, Tang CS (2000) The value of information sharing in a two-level supply chain. Manag Sci 46(5):626–643. https://doi.org/10.1287/mnsc.46.5.626.12047

Lee JS, Pries-Heje J, Baskerville R (2011) Theorizing in design science research. In: Jain H et al (eds) Service-oriented perspectives in design science research. Springer, Heidelberg, pp 1–16

Lee H, Kim MS, Kim KK (2014) Interorganizational information systems visibility and supply chain performance. Int J Inf Manag 34(2):285–295. https://doi.org/10.1016/j.ijinfomgt.2013.10.003

Legner C, Eymann T, Hess T, Matt C, Böhmann T, Drews P, Mädche A, Urbach N, Ahlemann F (2017) Digitalization: opportunity and challenge for the business and information systems community. Bus Inf Syst Eng 59(4):301–308. https://doi.org/10.1007/s12599-017-0484-2

Lempinen H, Rossi M, Tuunainen VK (2012) Design principles for inter-organizational systems development – case Hansel. In: Peffers K, et al (eds) Design science research in information systems. Advances in theory and practice. Springer, Heidelberg, pp 52–65

Lyytinen K, Damsgaard J (2011) Inter-organizational information systems adoption – a configuration analysis approach. Eur J Inf Syst 20(5):496–509. https://doi.org/10.1057/ejis.2010.71

Ma Z (2017) Digital rights management: model, technology and application. China Commun 14(6):156–167. https://doi.org/10.1109/CC.2017.7961371

Maedche A, Gregor S, Morana S, Feine J (2019) Conceptualization of the problem space in design science research. In: Tulu B (ed) Extending the boundaries of design science theory and practice – 14th International conference on design science research in information systems and technology, Worcester. Springer, Cham, pp 18–31. https://doi.org/10.1007/978-3-030-19504-5_2

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Markovsky B (1985) Toward a multilevel distributive justice theory. Am Sociol Rev 50(6):822–839. https://doi.org/10.2307/2095506

Mayer RC, Davis JH, Schoorman FD (1995) An integrative model of organizational trust. Acad Manag Rev 20(3):709–734. https://doi.org/10.2307/258792

Mentzer JT, DeWitt W, Keebler JS, Min S, Nix NW, Smith CD, Zacharia ZG (2001) Defining supply chain management. J Bus Logist 22(2):1–25

Möller F, Guggenberger T, Otto B (2020) Towards a method for design principle development in information systems. In: Hofmann S et al (eds) Designing for digital transformation. Co-creating services with citizens and industry. Springer, Cham

Möller F, Chandra Kruse L, Schoormann T, Otto B (2022a) Design principles for boundary spanning in transdisciplinary design science research. In: Drechsler A, et al (eds) The transdisciplinary reach of design science research. Springer Cham, pp 42–54

Möller F, Schoormann T, Strobel G, Hansen M (2022b) Unveiling the cloak: kernel theory use in design science research. In: Proceedings of the 43rd international conference on information systems, Kopenhagen

Mullarkey MT, Hevner AR (2019) An elaborated action design research process model. Eur J Inf Syst 28(1):6–20

Nunamaker J, Briggs RO (2012) Toward a broader vision for information systems. ACM Trans Manag Inf Syst 2(4):1–12

OASIS (2013) eXtensible Access Control Markup Language (XACML) Version 3.0: OASIS Standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html#_Toc325047170. Accessed 11 September 2022

Opriel S, Möller F, Burkhardt U, Otto B (2021) Requirements for usage control based exchange of sensitive data in automotive supply chains. In: Proceedings of the 54th Hawaii international conference on system sciences, Hawaii

Opriel S (2021) Austausch sensibler Informationen in Liefernetzwerken der Automobilindustrie, Praxiswissen Service

Otto B, Jarke M (2019) Designing a multi-sided data platform: findings from the international data spaces case. Electron Mark 29:561–580. https://doi.org/10.1007/s12525-019-00362-x

Otto B (2022) The evolution of data spaces. In: Otto B et al (eds) Designing data spaces: the ecosystem approach to competitive advantage. Springer, Cham, pp 3–15. https://doi.org/10.1007/978-3-030-93975-5_1

Otto B, Steinbuß S, Teuscher A, Lohmann S, Bader S, Birnstil P, Böhmer M, Brost G, Cirullies J, Eitel A (2019) Reference architecture model. Version 3.0. International Data Spaces Association

Park J, Sandhu R (2004) The UCONABC usage control model. ACM Trans Inf Syst Secur 7(1):128–174. https://doi.org/10.1145/984334.984339

Polatin-Reuben D, Wright J (2014) An internet with BRICS characteristics: data sovereignty and the balkanisation of the internet. In: 4th USENIX workshop on free and open communications on the internet

Prat N, Comyn-Wattiau I, Akoka J (2015) A taxonomy of evaluation methods for information systems artifacts. J Manag Inf Syst 32(3):229–267. https://doi.org/10.1080/07421222.2015.1099390

Pretschner A, Hilty M, Basin D (2006) Distributed usage control. Commun ACM 49(9):39–44

Richter H, Slowinski PR (2019) The data sharing economy: on the emergence of new intermediaries. Int Rev Intellect Prop Compet Law 50(1):4–29. https://doi.org/10.1007/s40319-018-00777-7

Saeed KA, Malhotra MK, Grover V (2011) Interorganizational system characteristics and supply chain integration: an empirical assessment*. Decis Sci 42(1):7–42. https://doi.org/10.1111/j.1540-5915.2010.00300.x

Saldaña J (2015) The coding manual for qualitative researchers. Sage

Sarabia-Jácome D, Lacalle I, Palau CE, Esteve M (2019) Enabling industrial data space architecture for seaport scenario. In: IEEE 5th World Forum on Internet of Things, pp 101–106. https://doi.org/10.1109/WF-IoT.2019.8767216

Schoormann T, Stadtländer M, Knackstedt R (2021) Designing business model development tools for sustainability – a design science study. Electron Mark. https://doi.org/10.1007/s12525-021-00466-3

Seddon PB, Scheepers R (2012) Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. Eur J Inf Syst 21(1):6–21. https://doi.org/10.1057/ejis.2011.9

Seidel S, Chandra Kruse L, Székely N, Gau M, Stieger D (2017) Design principles for sensemaking support systems in environmental sustainability transformations. Eur J Inf Syst 27(2):221–247. https://doi.org/10.1057/s41303-017-0039-0

Sein MK, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. MIS Q 35(1):37–56. https://doi.org/10.2307/23043488

Shapiro C, Varian HR (1998) Information rules: a strategic guide to the network economy. Harvard Business School Press, Boston

Straub D, Welke R (1998) Coping with systems risk: security planning models for management decision making. MIS Q 22:441–469. https://doi.org/10.2307/249551

Veit D, Clemons E, Benlian A, Buxmann P, Hess T, Kundisch D, Leimeister JM, Loos P, Spann M (2014) Business models: an information systems research agenda. Bus Inf Syst Eng 6(1):45–53. https://doi.org/10.1007/s12599-013-0308-y

Ven K, Verelst J, Mannaert H (2008) Should you adopt open source software? IEEE Softw 25(3):54–59

Venable J, Pries-Heje J, Baskerville R (2012) A comprehensive framework for evaluation in design science research. In: Peffers K et al (eds) Design science research in information systems. Advances in theory and practice. Springer, Heidelberg, pp 423–438

Venkatesh V, Thong JYL, Xu X (2012) Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. MIS Q 36(1):157–178. https://doi.org/10.2307/41410412

Vernadat FB (2009) Enterprise integration and interoperability. In: Nof SY (ed) Springer handbook of automation. Springer, Heidelberg, pp 1529–1538. https://doi.org/10.1007/978-3-540-78831-7_86

vom Brocke J, Winter R, Hevner A, Maedche A (2020) Accumulation and evolution of design knowledge in design science research – a journey through time and space. J Assoc Inf Syst 21(3):520–544. https://doi.org/10.17705/1jais.00611

Wall J, Lowry P, Barlow J (2016) Organizational violations of externally governed privacy and security rules: explaining and predicting selective violations under conditions of strain and excess. J Assoc Inf Syst. https://doi.org/10.17705/1jais.00420

Wand Y, Weber R (1995) On the deep structure of information systems. Inf Syst J 5(3):203–223

Wiengarten F, Humphreys P, McKittrick A, Fynes B (2013) Investigating the impact of e-business applications on supply chain collaboration in the German automotive industry. Int J Oper Prod Manag 33(1):25–48. https://doi.org/10.1108/01443571311288039

Zrenner J, Möller FO, Jung C, Eitel A, Otto B (2019) Usage control architecture options for data sovereignty in business ecosystems. J Enterp Inf Manag 32(3):477–495. https://doi.org/10.1108/JEIM-03-2018-0058