



# Platform-based collaboration in digital ecosystems

Fabian Aulkemeier<sup>1</sup> · Maria-Eugenia Iacob<sup>1</sup> · Jos van Hillegersberg<sup>1</sup>

Received: 31 March 2018 / Accepted: 4 March 2019  
© The Author(s) 2019

## Abstract

Organizations operating in a network require strong information technology support to successfully collaborate. In a business service network, the links between partners must enable quick connect and disconnect relationships, in order to harness market opportunities. Scholars claim that information technology platforms are necessary to enhance this quick connect capability and to transform business networks into digital ecosystems. However, the state of the art in inter-organizational collaboration relies on static collaboration patterns between individual partners, and current systems are engineered without interoperability in mind. In this study, we follow a design science approach to promote the concept of platform-based collaboration. More precisely, we propose an architecture for an inter-organizational platform that facilitates the provisioning of collaboration services. Furthermore, we present a prototype in the context of e-commerce as a means of evaluating the proposed design. We conclude that the platform approach is beneficial in achieving increased flexibility in business-to-business collaboration.

**Keywords** Business networks · Inter-organizational collaboration · Cloud service platform · Canonical data model · E-commerce

**JEL classification** L86 Information and Internet Services · Computer Software

## Introduction

In the era of smart supply chains, the requirements for flexible business operations are evolving. Processes must respond rapidly to changes in the environment and require real-time and dynamic optimization. For example, cyber-physical systems allow for the tracing and analysis of the status of goods in real time, and enable ad hoc decision making (Monostori et al. 2016). Furthermore, by applying agile development methods, systems can be quickly adapted over short cycles, adding another dimension of flexibility through the continuous evolution of their capabilities (Porter and Heppelmann 2014).

However, flexibility in collaborating with suppliers and partners involves additional complexity. There is a broad consensus that tight integration and commitment to long-term partnerships is required in order to achieve working inter-organizational processes (Ramanathan and Gunasekaran 2014). Examples of such loyal relationships can be found in various domains (Chick et al. 2014). Governance mechanisms that define the roles, responsibilities and processes in the business network must be in place (Chandra and van Hillegersberg 2015). Thus, a key requirement for effective inter-organizational processes appears to be the existence of well-established and static communication channels between the IT systems of all partners. However, the downside of this practice is related to the limited possibilities for collaborating in an ad-hoc manner; as a consequence, companies are missing out on the opportunities offered by engagement in dynamic collaboration relationships (Fawcett et al. 2015).

To develop agile business processes in this setting, the network and its participants need to improve their capability to quickly connect to (and disconnect from) each other (van Heck and Vervest 2007). The time and cost required to engage with new business partners are core aspects of the concept of quick connect capability (QCC) (Koppius and van de Laak 2009). From an information technology (IT) perspective,

---

This article is part of the Topical Collection on Research Advances in Multi-Sided Platforms

---

Responsible Editor: Nizar Abdelkafi

✉ Fabian Aulkemeier  
f.m.aulkemeier@utwente.nl

<sup>1</sup> Industrial Engineering & Business Information Systems, University of Twente, Enschede, The Netherlands

QCC depends on the capability of applications and of the IT architecture to reduce the effort associated with adopting new services. The limitations of current IT systems are often considered to be a reason for the limited agility of business networks (van Hillegersberg et al. 2012). Few of the application components of collaborative processes were constructed with interoperability requirements in mind. More specifically, few of these components were designed based on standards such as the data models and protocols of the network they should operate in.

Environments that allow open, flexible and demand-driven collaboration have been referred to as digital ecosystems, and these are often associated with digital platforms (Boley and Chang 2007; Sørensen et al. 2015). Van Heck and Vervest (2007) and van Hillegersberg et al. (2012) propose platform-based approaches to simplify inter-organizational collaboration. Although existing platforms facilitate the task of linking the endpoints of systems within and across organizational boundaries, current practice still requires the implementation of dedicated integration artifacts that contain both the business logic, and the model transformations required for system interaction (Aulkemeier et al. 2016). Such artifacts link two or more systems and are deployed onto one of the systems or onto an integration platform, which then acts as a mediator between those systems. This type of approach has a number of limitations, as indicated below:

- A dedicated collaboration artifact needs to be built for each and every system involved in the collaboration;
- Each integration artifact typically focuses only on a single specific business process or use case;
- Any changes to the business processes or integrated systems usually requires the re-engineering of integration artifacts;
- Design decisions and subsequent changes to the integrated system can lead to conflicts between business partners;
- The division of responsibility for the integration artifact and the resources required for its operation can lead to conflicts between the business partners.

Thus, the overhead involved in building dedicated integration artifacts can lead to delays in establishing working business collaborations. Furthermore, the need to build integration artifacts may strain the relationship between partners. We can conclude that the practice of building integration artifacts on top of existing information systems leads to inflexible collaboration, despite the use of existing integration platforms.

To overcome these limitations and to achieve a better QCC, an architecture is required that obviates the need for dedicated integration artifacts to allow collaboration between network partners. Hence, the research question that we address in this paper is how to build a platform architecture that eliminates a user's need for dedicated integration artifacts when engaging

in a new collaboration with other participants in a business network.

The novelty and main contribution of the platform architecture put forward in this paper is support for collaborative services that encapsulate the logic required for specific collaboration scenarios. The essential difference between the earlier mentioned dedicated integration artifacts, and such a collaborative service is a higher level of pluggability; in other words, a collaborative service can be consumed 'out of the box' (whenever needed) while an integration artifact must be tailored to a specific combination of existing information systems. The main benefit of the collaborative service is the ability to engage in business collaboration in an ad hoc manner, and thus to establish a higher QCC for its users.

The intended outcome of this study is an architecture, in the form of a design artefact hypothesized to provide a solution for an existing problem, namely the incapacity of existing collaboration architectures to facilitate ad hoc interoperability in business networks. The most common approach when researching this type of problem is to follow a design science research methodology (DSRM). In this study, we combine two DSRMs that support the construction and evaluation of the design artifact. Peffers et al. (2007) propose a *process-oriented* DSRM that draws upon other established work in the field, and which is suitable for research in collaboration with industry partners that need to be involved in particular steps in the research process. The second DSRM we incorporate was proposed by Verschuren and Hartog (2005) and which can help to formulate intermediate outcomes such as the goals and requirements of the artifact. The formulation of these outcomes makes the design process more reliable, since it includes the evaluation of the plan in comparison with the final artefact. Figure 1 shows the five steps in the DSRM proposed by Peffers et al. (2007), and maps them to the stages of design science (DS) research proposed by Verschuren and Hartog (2005). The functional specification of the design is represented by an architecture model, and the implementation is represented by a prototype. The design and implementation of the prototype were carried out in collaboration with industry partners in the domain of e-commerce. While the evaluation of these two artifacts is a vital step in the method, the knowledge gained, is primarily associated with the two artifacts that are product of the design process as pointed out by Cross (2006).

In the next section, we outline the state of the art in collaboration architectures and describe the goals for an architecture that addresses the aforementioned limitations. Following this, we discuss the requirements that will be used to evaluate the final prototype. Its functional specifications and implementation are described in the two subsequent sections. In the final sections, we evaluate the proposed artifacts and present conclusions and directions for future work.

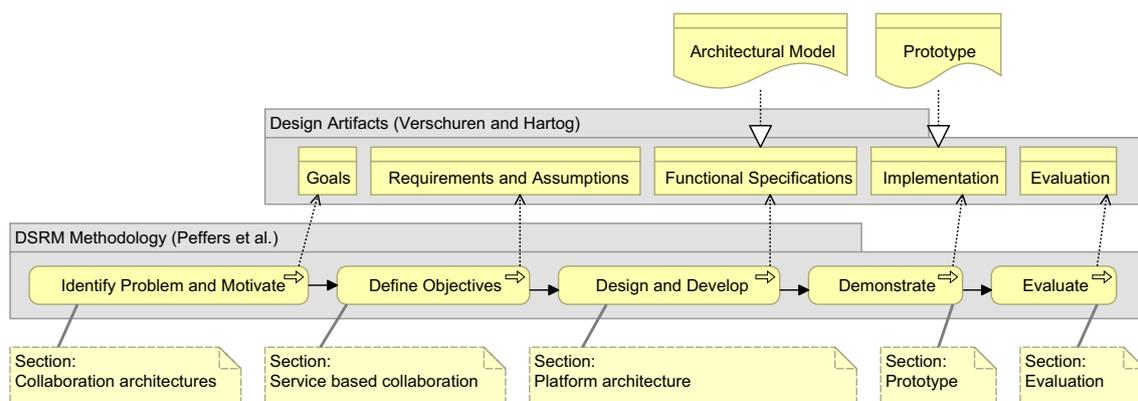


Fig. 1 Research design

## Collaboration architectures

In order to assess the predominant inter-organizational IT architectures, we present a review of the state of the art in collaboration architectures. This allows us to identify the limitations of the extant IT collaboration architectures. Furthermore, according to Verschuren and Hartog (2005), problem identification and motivation in DS should also result in the formulation of a small set of goals, and we discuss this in the second part of this section.

Since the early days of IT-enabled inter-organizational collaboration and trade, the heterogeneity of the information systems involved has been identified as a major challenge. In particular, incompatible representations of the relevant information and semantic differences between the systems used by trading partners are major obstacles, and can reduce the expected benefits of IT-driven collaboration (Lincke and Schmid 1998). To overcome these issues, various approaches have been proposed that are based on the core concepts of mediation (Wiederhold 1992) and federation (Busse et al. 1999) of information systems. The common denominator of these approaches is the recognition of the necessity of introducing of semantic and syntactic industry data standards that would help businesses to address with the issue of heterogeneity.

## State of the art

The goal of mediation is to transform data from various sources, in order to make it accessible via a unified structure. Handschuh et al. (1997) propose various models for mediating product catalogs that can help business partners obtain a standardized view of product information, and thus facilitate transactions. The issue of whether or not the mediation should be handled by an intermediary has been raised out by Sen and King (2003). According to Palmer and Johnston (1996), the benefits of intermediaries are additional security and virtual marketplaces that can help businesses to initiate new partnerships. Mediation that is handled by so-called brokers is particularly interesting for markets with a large number of

participants, as these can increase trust and reduce transaction costs (Bichler et al. 1998).

The implementation of mediating technologies such as electronic data interchanges, requires a stable and long-term relationship between partners. Reimers (2001) claims that Extensible Markup Language (XML) technologies are more suitable for the ad hoc connection of services, due to their self-descriptive nature. However, these data standards are limited to the exchange of transaction information, and are not suitable for supporting an integrated architecture for agile collaboration (Vujasinovic et al. 2010).

Web services have been introduced to solve the problem of static communication patterns between dedicated systems (Papazoglou 2003). The use of common internet protocols provides some advantages over the use of proprietary business-to-business networks; however, the most promising mechanisms in the context of service-oriented architectures, such as automatic service discovery and consumption through Universal Description, Discovery and Integration (UDDI), have not found their way into practice. The mere use of web services does not solve the inflexibility problem: integration artifacts still need to be built, including the mapping of data, serialization and deserialization of messages, interface testing, monitoring and maintenance. A more sophisticated approach to reducing the engineering efforts that are required to connect web service endpoints involves semantic web services, the goal of which is to allow machines to understand the ontological meaning of data, and let them act in a more autonomous fashion (Pellegrini 2017). As yet, semantic web services have not been widely adopted outside of academic studies or prototypical implementations, and hence have not yet solved the agility problem in inter-organizational collaboration.

Another concept from the domain of enterprise application integration is the canonical data model (CDM) (Hohpe and Woolf 2003). Originating from a messaging paradigm, a CDM promotes the idea of increased interoperability by applying unified data models across collaboration partners. However, the use of CDMs has not been widely reflected in research to date, and there is a lack of validated CDM

development methods in particular. Other domains however, provide concrete construction methods for unified data models, such as for data analytics in data warehousing (Prat et al. 2006) or ontologies for semantic systems (Aussenac-Gilles et al. 2000). However, no concrete methods exist, that describe how to build a unified data model for systems to support inter-organizational transactions.

## Goals

From this analysis of the extant literature we can conclude that modern inter-organizational collaboration within business networks is generally realized through dedicated intermediated or point-to-point integration artifacts, allowing organizations to exchange information required for a specific business transaction. The concept of mediating CDMs, can only be found in ERP systems within organizations, and has been neglected in the recent debate on inter-organizational interoperability.

In order to establish the requirements for a new design, Verschuren and Hartog (2005) state that a number of *goals* must be formulated. In our case, we aim at an improvement of the QCC of organizations within a business network. As Koppius and van de Laak (2009) show, the QCC consists of three factors: *quick connect*, *quick complexity*, and *quick disconnect*. A suitable solution for supporting a pluggable inter-organizational collaboration should therefore achieve the following goals:

- G1** Individual services can be adopted quickly (quick connect).
- G2** Complex inter-organizational functionality can be handled through the use of appropriate collaboration services (quick complexity).
- G3** Disconnecting individual services or partners will not affect any remaining services or collaborations (quick disconnect).

## Platform-based collaboration

While Peffers et al. (2007) propose defining the objectives of a solution by rationally inferring them from the goals, the more rigorous DS process of Verschuren and Hartog (2005) involves a two-step approach: the first step defines a set of requirements, while the second step carries out an evaluation of the plan. In the following section, we put forward a set of requirements for an architecture that meet the goals described in the previous section. We then carry out a plan evaluation which provides grounded arguments for the contribution of each requirement to the different goals.

## Requirements

In the previous sections, we argued that the use of a mediating CDM could help achieve the goal of improved QCC. However, the goal of connecting various services through the use of a common information system like a CDM has become unpopular, as it runs the risk of creating a tight coupling between services, that in turn impedes the agility and evolution of individual services (E. Evans 2004). To address this issue, we aim to separate of the evolving components in the system from the stable components, an issue that is commonly addressed by the introduction of a platform. Baldwin and Woodard (2009) define a platform as a set of stable components that support variety and evolvability in a system by constraining the linkages among the other components. Through the use of platform design, it is possible to introduce a common information system across services, ideally without affecting their evolvability.

The initial design task for the platform is to identify a domain model that is suitable for use as a CDM to create a single information space for the business network (Heath and Bizer 2011). In the context of data warehousing, Prat et al. (2006) differentiate between two approaches for the construction of a unified data model: in the first, the data model results from combining the source systems, and in the second, the model is based on the requirements for its use. Due to the unpredictability of the systems collaborating in the network and their data needs, the first of these methods is less viable. In the case of analytical systems, the goals for the use of the model are the reporting needs of the decision support system. The equivalent goal in a transactional system is support for processes taking place in the domain of the platform and collaboration services. Thus, the completeness of the data model with regard to a reference process model of the domain is crucial. However, the goal of supporting a maximum number of use cases runs the risk of leading to a system that is too complex to adopt; for example, the data exchange standards mentioned above, are difficult to comply with, due to their complexity (Damsgaard and Truex 2000). We therefore consider a good balance between completeness and simplicity to be an important factor in the construction of a model.

Intermediaries offer another way of reducing complexity by outsourcing the collaboration logic. The resulting collaboration services can be maintained by service providers, which act as domain experts for particular collaboration scenarios (Aulkemeier et al. 2017; Sherer and Adams 2001). Another advantage of relying on intermediaries is the higher possibility of discovering new business opportunities (Giaglis et al. 2002).

One means of overcoming the conflicting goals of completeness and simplicity is extensibility. In a paper on configurability of cloud-based solutions, Nitu (2009) notes that ‘there will be some unique features in the database of each

tenant,' and proposes a data model that 'meets the most common requirements of the tenants with an option to add the tenant's specific data requirements like adding additional fields to a table.' This is in line with the general trend towards less rigorously structured data repositories. For example, NoSQL databases make use of flexible structures rather than a static relational data scheme. The possibility of defining ad hoc structures in addition to the given data model is a way of increasing the usability and acceptance of the CDM. In practice, we propose to stick to the core entities and a limited number of attributes in the definition of the data model, and to leave the possibility of defining additional attributes for each entity to the user of the system. The requirements for the architecture can be summarized as follows:

- R1** Achieve agility by separating the stable components from the evolving ones.
- R2** Provide a CDM that is suitable for data federation among services.
- R3** Reach a good balance between completeness and simplicity through extensibility in the domain model.
- R4** Allow intermediaries to offer collaborative services.
- R5** Allow business partners to discover new services and new business partners.
- R6** Provide a means of granting intermediaries access to shared information.

## Plan evaluation

According to Verschuren and Hartog (2005), a plan evaluation should be carried out in order to ensure that the requirements are valid sub-goals and that they contribute to the achievement of the overall design goal. In the following, we describe how the design requirements contribute to each of the design goals. An overview of the mapping is shown in Fig. 2.

The use of shared resources (R1, R2) has various benefits for the goal of improved collaboration (G1, G2). Firstly, it will reduce the effort of mapping data models of distributed data repositories. The data quality also increases, due to the limited redundancy of data across systems (Dromey 1995). Finally, a unified resource repository allows centralized business monitoring and exception handling by defining business rules for the enterprise or network wide-data (Bajec and Krisper 2005). Certain events can then raise exceptions, which can then be handled by a service or control tower. Normal business events can also be handled on a more global scale, are suitable for real time processing instead of scheduled events. This is in line with the current trends in social media, where the pattern is changing from request-and-reply to real-time data-streaming APIs. The separation of agile and stable components (R1) is also required to limit the effects on the overall system when outdated components are replaced (G3).

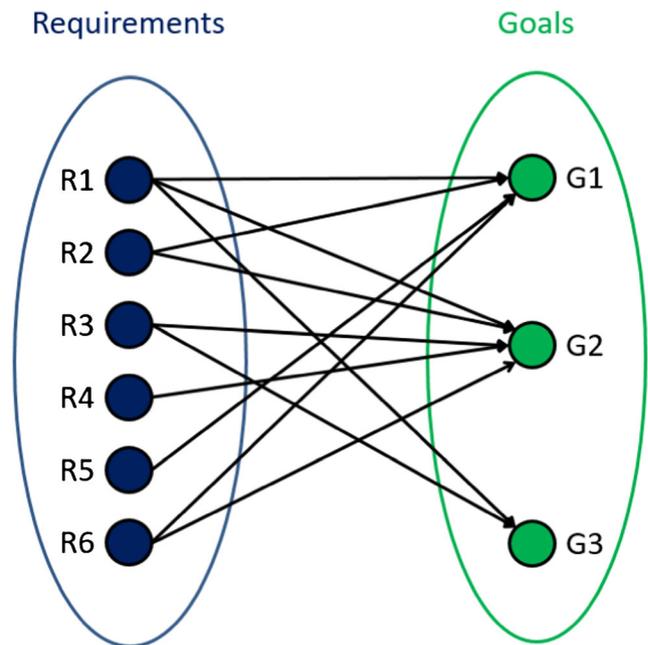


Fig. 2 Mapping of the design requirements and design goals

A robust data model is required (R3) in order to guarantee long-term use for future services (G3) and to support a maximum number of business scenarios (Saltor et al. 1991) such as inter-organizational collaboration services (G2). According to Palmer and Johnston (1996) these collaborations should be handled by intermediaries (R4). The adoption of new services (G1) can be further improved by the provisioning of discovery services (R5), as pointed out by Bichler et al. (1998). Finally, the intermediary need to be able to gain access to the resources of the business partners (R6) in order to allow for inter-organizational services with QCC (G1, G2).

## Platform architecture

In DS, the specification of the design artifact is based on the requirements, and forms the basis for its implementation (Verschuren and Hartog 2005). In this case, the functional specification is for a platform architecture that can achieve the goal of improved QCC among business partners. Since our application domain is e-commerce, we focus on the description of an architecture for a network of business partners in online retail.

The ArchiMate notation is used to formally specify our architecture model. The purpose of an ArchiMate model is to identify and relate IT and business concepts (The Open Group 2016). It is a suitable way of discussing the components and actors involved, and of highlighting the interaction mechanisms in the platform.

## Ecosystem

There are several approaches and architectures exist that follow a CDM-based approach to inter-organizational integration. Kleeberg et al. (2014) present various cloud-based integration scenarios. Among several more visionary solutions, the authors mention ‘EAI-in-the-cloud’ which ‘leads to a situation where more enterprise resources are being exposed to off-premise access or moved to the cloud. This situation opens novel opportunities for supporting B2B-transactions. [...] A straight forward example is cross-enterprise data sharing by means of a common cloud-based data store.’ Wlodarczyk et al. (2009) have put forward the idea of an industrial cloud that ‘should be controlled by an organization in form of e.g. [a] special interest group’. However, no architecture or solution is presented that goes beyond a high-level vision of such a solution.

Figure 3 shows that the high-level ecosystem of the architecture involves the three main roles: a *Platform Provider*, a (platform) *Client* that implements and offers platform-based business services, and a (service) *User*. High levels of dominance and trust in the e-commerce market may attract more service providers, and is likely to give a have a higher chance of achieving a critical mass of users (D. S. Evans and Schmalensee 2010). We can see from the model that according to the mediated approach discussed earlier, the *Platform* embodies *Federated Data* in the form of a CDM. Following the reference model for federated systems (Busse et al. 1999; Sheth and Larson 1990), the *Business Service* makes use of the platform’s CDM and controls only the *Component Data*.

## Platform

The platform consists of multiple application components that rely on two different data object types (Fig. 4). A resource set data object is assigned to each user of the platform, containing core business data for e-commerce such as orders, customer data and product information. The metadata contain extensions to the core data and the access privileges for the various clients. The data object definition and management component provide access to the data and is used by the clients. The billing component

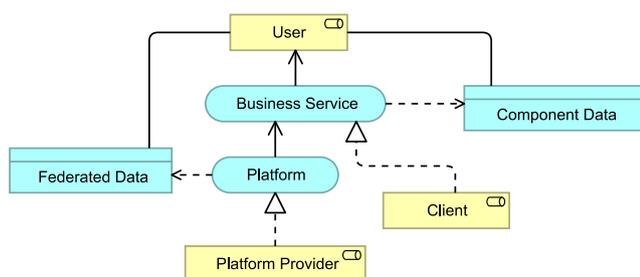


Fig. 3 Ecosystem of the architecture

and the service marketplace facilitate the mutual retrieval and payment over the various platform services.

The platform provides two interfaces, a web user interface (web UI) that is used to access the marketplace, and billing component. The core data can also be exposed via the web UI to facilitate administration tasks involving the core data. The API or Web Service interface is used by the client to grant authorization and to access the core data.

## Platform implementation

An instantiation of a design allows us to evaluate the feasibility and effectiveness of the functional specifications. This instantiation may be a realization (immaterial artifact) or a materialization, may meet all or only some of the specifications, or may simply be a mock-up (Verschuren and Hartog 2005). In our case, we aim to achieve a complete materialization of the platform, in order to evaluate its ability to support the implementation of services that help to achieve the goal of QCC. In this section, we describe the prototype of the platform, and in the following section, we discuss the implementation of services to evaluate the platform.

## Canonical data model

The resource set is one of the core components of the platform and is based on a CDM of the business domain. According to Saltor et al. (1991), expressiveness is a critical success factor for CDMs: ‘A CDM must have an expressiveness equal or greater than any of the native models of the component DBs that are going to interoperate, in order to capture the semantics already expressed with the native models. Moreover, it should support additional semantics made explicit thru a semantic enrichment process.’

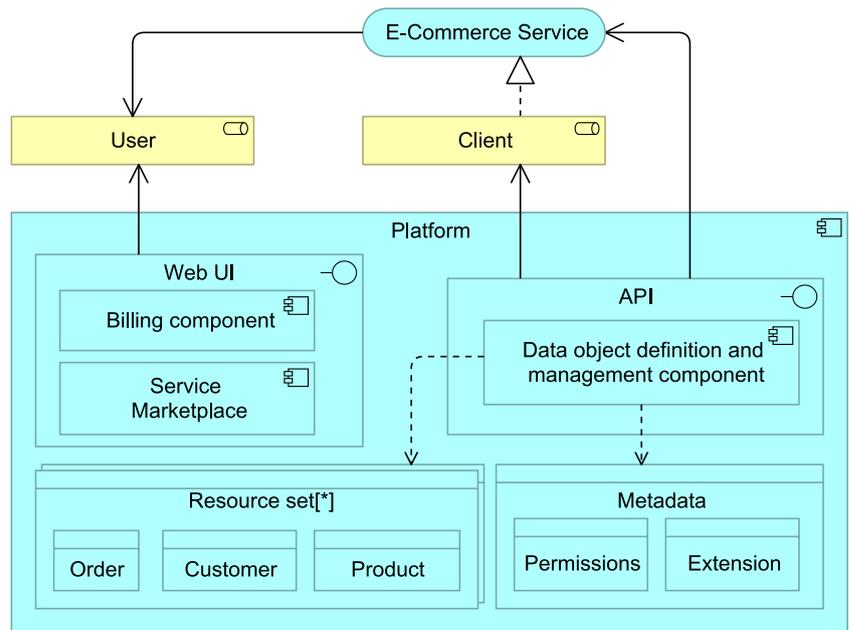
To adapt the cycle proposed by Saltor et al. (1991), we start from a native model of one service component and extend the model by successively adding more services. For the prototype, we stopped the enrichment cycle after the implementation of a service-based end-to-end product return process.

## Metadata model

The heterogeneity of components that rely on the CDM can be addressed through meta-information (Busse et al. 1999). We therefore introduced two types of metadata into the prototype: client-related metadata and infrastructure-related metadata.

The capability to allow custom data model extensions to be introduced is the main purpose of client-related metadata, as it allows client components to define additional attributes for each resource on the platform (extensions). The definition of new resource attributes is carried out declaratively at design time through the web UI. For example, a client application for temperature-controlled transport planning can enrich the

**Fig. 4** Platform components and interfaces



product entity with information on ambient thresholds, information that is too specific for the CDM. The approach described here dissolves the rigid structure of the data model and can be adjusted to any use case within the context of the platform. Furthermore, the client can determine whether the scope of the additional attributes should be private or shared with other clients.

The use of infrastructure-related metadata resolves various requirements related to data ownership and mutual data access (permissions). Allowing to configure the visibility and access to data by business partners. Through the use of access tokens, similar to the case of state-of-the-art social media platforms, users can grant access to their resources to other clients of the platform (Hardt 2012).

### Interfaces

The platform prototype provides two interfaces: a computing interface for platform client access, and a user interface for various data administration purposes.

The computing interface is a REST API with two endpoints for client authorization and access, and a number of endpoints for business resources. Table 1 contains a description of the endpoints and the supported methods. The resource endpoints are available for every resource of the CDM and allow access to single resources or entire collections via filter specification.

The web UI of the prototype shown in Fig. 5 has four main functions that allow the platform users to maintain resources, clients and services:

- The business resource administration function allows users to access their resource set. Users can view, update, and delete resources and metadata, such as additional attributes created by clients.
- The service administration function allows the user to edit service subscriptions, search for and subscribe to new services, and to retrieve billing and payment information for services.
- Since the user can be both a service consumer and a provider at the same time, the client administration function

**Table 1** Platform prototype - API endpoints

Endpoint	Method	Description
/authorize	PUT GET	Redirect a user to the web page to authorize access to resources
/token	GET	Exchange and refresh tokens that give access to resources
/<resource>	GET	Access and modify resources. Single resources can be accessed by their ID and collections of resources can be filtered through parameters. Each request requires an access token. Additional attributes that are defined in the meta-model by the client are seamlessly integrated into the response.
	PUT	
	POST	
	DELETE	

## Resource Overview

Name	Product
Instances	5 (Show)
Attributes	10 (6 Default / 4 Custom)

### Default Attributes

Name	Description	Data Type	
Product Name	Name of the product	String	<a href="#">Details</a>
Description	Detailed product description	String	<a href="#">Details</a>
Dimension Length	Product length in cm	Float	<a href="#">Details</a>
Dimension Width	Product width in cm	Float	<a href="#">Details</a>
Dimension Height	Product height in cm	Float	<a href="#">Details</a>
Weight	Product weight in gramms	Float	<a href="#">Details</a>

### Custom Attributes

Client	Name	Description	Data Type	Scope	
Cross-selling	CS_Enabled	Enabled for cross selling	Boolean	Private	<a href="#">Details</a>
Cross-selling	CS_Comission	Comission for cross-selling in percent	Integer	Private	<a href="#">Details</a>
Shop Frontend	SF_Brand	Product Brand	String	Public	<a href="#">Details</a>
Shop Frontend	SF_AvailableDate	First date of availability	Date	Public	<a href="#">Details</a>

Fig. 5 Prototype web UI – business resource administration

allows the user to set up new clients, which can then be subscribed by other users. This includes setting up of custom attributes and the retrieval of subscriber information, including invoicing.

- The documentation function offers access to user documentation and documentation for service providers, including API documentation.

## Evaluation

In the previous sections, we outlined the requirements, functional specification and implementation of a collaborative platform. As part of the plan evaluation, we have already discussed how the platform's requirements match the overall goals, which are related to improved QCC among business network partners. The DSRM proposed by Peffers et al. (2007) required instantiation of the artifact to observe how well it supports the objectives. Similarly, Verschuren and Hartog (2005) suggest a product evaluation in order to assess the extent to which the goals have been achieved. In order to evaluate the platform's capabilities in terms of allowing QCC among business partners, we consider an e-commerce

business case. In conjunction with the industry partners involved in this study, we designed and implemented a platform prototype and various services; this allowed us to verify the platform's capabilities in terms of enabling retailers to enter into a dedicated business relationship without prior agreement. In the remainder of this section, we present the prototype of a platform-based collaborative sales service and assess its effectiveness in achieving its three goals.

## Collaborative sales service

A collaborative sales service is a platform client that facilitates cross-selling transactions. Our study involved several stakeholders from the Dutch online retail sector. Their explicit need for a platform that could facilitate platform-based internal and external processes, and this led to the current joint project. Cross-selling was chosen since it is an important issue for collaboration in e-commerce.

Figure 6 shows how the final service is embedded in the platform ecosystem which consists of numerous transaction services and collaboration services offered by service providers in partnership with retailers. While transaction services are used to reflect the internal processes of the retailer, the collaboration services are built on the same backend and

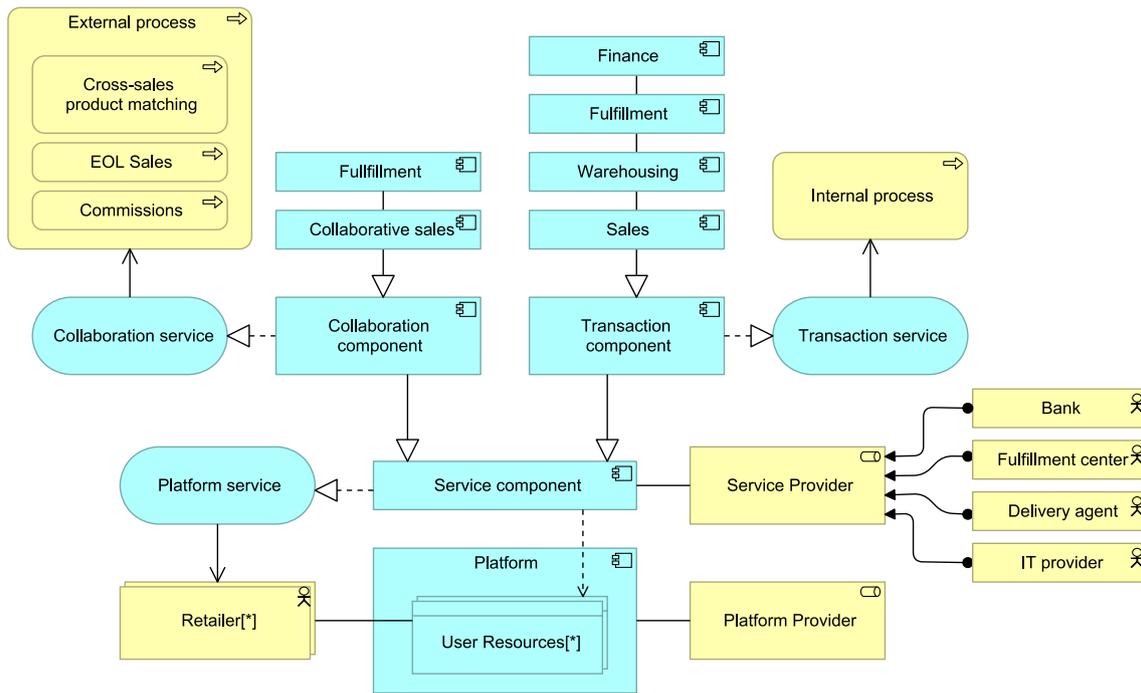


Fig. 6 Platform architecture for a retail business network

reflect inter-organizational processes. Earlier versions of the platform architecture relied on individual backends for collaboration and transaction components. The evaluation of the resulting prototype in conjunction with the partners revealed the QCC shortcomings of the services, and lead to the modification of the architecture.

The service component reflects various processes that are relevant to collaborative sales transactions. These include end-of-life (EOL) product sales, which are crucial in short-cycle product sectors. The goal is to reduce stock and minimize the sales margin loss by selling products that are nearing the end of their lifecycles through partner channels. The second opportunity for sales through partners is known as cross-sales,

where a matching of product is carried out, resulting in an advertising of matched products during checkout. Figure 7 shows the web application for the service, which allows the manual matching of products qualifying for cross-selling. Since the service has full access to the customer, product and stock information held by the partners, the implementation of a smart product matching algorithm is possible (Xiao and Benbasat 2007).

Finally, any collaborative sales transactions results in a commission process that takes care of the settlements between the partners. The collaboration service handles all aspects of the inter-organizational collaboration scenario in an end-to-end way. These collaborative sales processes are included in

Cross selling configuration		
Product	Partner	Product
Apple iPad 9.7	Technosurance...	2 years extended tablet warranty
Samsung Galaxy Tab	Technosurance...	2 years extended tablet warranty
LG 24" Washer/Dryer Combo	Applinstall	Home appliance delivery and install
GE Space Maker Electric Laundry Center	Applinstall	Home appliance delivery and install

Fig. 7 Mapping products between retail partners

the design; other collaboration services may involve collaborative fulfillment processes that allow the processing of orders through partners.

## Product evaluation

The collaborative sales service is used to evaluate the platform in terms of the three goals G1–G3. If multiple retailers rely on the platform to handle resources such as products and orders, the provider of the collaborative service does not need to map the data models of these partners, since they share the same data schema. The same applies to the reduced efforts required to adopt the collaboration service. All platform users share the same information system, and no further implementation work is required to access all their data, thus allowing them to outsource the complexity of the external process logic to the service provider. Finally, none of the transaction services relies on the collaboration service, resulting in a low coupling between them.

An evaluation of the architecture and prototype was carried out at each iteration of the design cycle. In the first iteration, the state-of-the-art architectures used by the industry partners were assessed, and their shortcomings were documented. Next, the platform architecture was outlined and assessed in workshops conducted with individual consortium members, including a service provider in the field of retail logistics and an IT solution provider in the same sector. The complete platform architecture and the prototype were shared with the entire consortium and adjusted based on their feedback. Finally, the implementation of the collaborative sales service was assessed with three different partners, in individual workshops with between three and six participants. The expert panel was questioned regarding the three goals of this design cycle (G1–G3). More precisely, the members were asked to provide their opinion on ‘how the use of the platform affects the adoption of the service’ (G1), ‘how the encapsulation of the collaboration logic as a service helps to facilitate complex inter-organizational logic’ (G2), and ‘how the architecture affects the coupling between services and the agility of the solution’ (G3). In the following, we outline the conclusions from this questioning.

## Service adoption

The platform prototype was constructed to include four different transaction services of an e-commerce process. All the data model entities required for the collaborative use case (i.e. products, orders, customer, and stock) were reflected in the CDM, thus indicating that the construction of a complete unified data model was successful with reasonable effort. The simplicity of the data model was achieved by limiting the attributes of the entities. Through the mechanism of extension, the services can include commission rate in the product entity.

Compared to earlier federated architectures, the functionality of each service can be developed individually and thus offered as a cloud-based service, for example.

According to the partners, a single collaboration scenario usually consists of several transactions, each involving multiple interfaces. In the commonly used web service approach, each interface requires a mapping and message processing logic on each side of the collaboration. In the use case of the collaborative sales service, we encountered three different transactions that are handled by individual components of the service: the mapping of products, the sales transaction, and the handling of the commission fees. Thus, for each sales collaboration six different integration artifacts must be built and maintained. The practitioners underlined the benefits of using the platform-based collaboration service to reduce the size of new collaboration projects, which included the expectation of a faster time-to-market of inter-organizational processes a reduced project effort.

## Complexity

In the same way as internal processes, inter-organizational processes have their own complex business logic. However, these processes must be implemented by agreement between the partners and must have a footprint in the IT systems operated by different parties. Conflicts over the process logic and responsibilities for implementation across disparate systems often cause tension and delay in new collaboration projects. Predefined collaboration scenarios implemented by a service provider were perceived as a suitable approach to reduce the time-to-market and project risk. Furthermore, these packaged collaboration scenarios can be based on best practices in the sector, this was perceived as added value, since the expertise of intermediaries reduces the need to ‘reinvent the wheel’ for each new collaboration project.

During the assessment, it was highlighted that the capabilities of the platform-based collaboration service go beyond those of existing solutions. These services rely on a unified information system and can act as a cockpit for monitoring and controlling the inter-organizational processes. For example, a collaborative sales service can make use of these capabilities and give each retailer insights into the current top-selling goods from their partners. During the matching of products, a retailer can see which products have been selling best over the last few hours and match a special offer to that product. Building this complex inter-organizational functionality is straight-forward, since all the data required are available from the canonical data component. The realization of business rules and exception-handling scenarios have been raised as potential use cases for the platform.

Another benefit of the CDM is the increase in data quality. In solutions that rely on data from multiple repositories, the replication and synchronization of these sources is often a cause of poor

data quality, and this problem intensifies over time if single updates are not carried out properly into all of the systems. Since the service relies on a single data source, which also happens to be the basis for other operational systems of the two shops, no data replication is required. Thus, there is no risk of reducing data quality inside the cross-selling component.

## Decoupling

Agile collaboration between business partners requires both quick connection and loose connections, which provide the capability to switch between partners. Thus, the separation between primary internal services and external services, such as the collaborative service, was perceived as an advantage of the platform approach over the current architectures used by the industry partners. It is possible to disconnect the service with no effect on other functional components. The tight coupling between services that is often linked to the use of shared databases does not exist in this case, as none of the services owns the core data model. However, the need to rely on the core data model may limit the flexibility of the service provider due to the constraints imposed by the platform.

## Conclusion and future research

In this paper, we surveyed the state-of-the-art literature concerning collaboration architectures and pointed out the shortcomings in current theory and practice. We argue that a platform with federated architecture and a CDM can improve the agility of the collaboration between members of a business network and we propose a corresponding platform architecture and a prototype for the online retail sector that stems from our collaboration with industry partners. The assessment of a platform-based collaboration service shows that the goal of improved QCC can be achieved.

However, there are certain limitations associated with the platform approach worth mentioning. For a two-sided platform there is an inherent need for a critical mass of service providers and service users in order to create sufficient value for the other side (D. S. Evans and Schmalensee 2010). The research consortium suggested that the platform should be developed as a joint project with diverse participants to ensure a successful launch. In fact, they consider the availability of a minimum set of transaction services covering mission-critical processes as the main factor in the success of the platform. Reducing the scope of the platform could also help in achieving this goal. One of the industry partners with a background in IT service operation also aims to adopt this platform architecture in one of their projects, within the domain of payment and invoicing.

Another limitation is the architecture's strong focus on system interoperability, which comes at the cost of constraints in terms of service development. In current systems, the ownership of the data schema stays with the implementer of the

application component, an approach that simplifies the design and realization of that functionality. Giving up part of this ownership to the platform will lead to more restrictions when building new services. Furthermore, the proposed approach requires strong commitment from the partner in the business network towards a particular platform which might lead to higher risks.

The main focus of the study is on demonstrating the benefits of the platform approach in terms of collaboration. However, we believe that the platform architecture can offer additional advantages for its users. More precisely, we argue that this platform can provide a means of sourcing IT services of any kind from external service providers in a pluggable way. As a result, new business functionality can be adopted with minimal efforts. Since the backend is already in place, functional components can be easily plugged into the existing application landscape. According to Włodarczyk et al. 2009 this can be particularly interesting for small and medium enterprises or start-ups, as it offers low entrance costs to specific IT services and requires little to no IT expertise. In fact, one of the collaboration partners which acts as an intermediary in customs handling aims at migrating one of their offerings from a web service to a platform-based pluggable service that complies with the principles of the proposed collaboration service.

The scope of this study has been limited to the design of the platform architecture and its instantiation in the form of a prototype. Future research should focus on the services that need to be built for the platform in order to achieve its full benefits. This includes technical issues such as the support of application frameworks, the additional complexity of building distributed solutions, and side-effects of the architecture, for example its impact on application performance. Furthermore, the organizational implications that may impact the success of platform adoption must be researched and discussed in more detail. In addition to the general skepticism of organizations towards moving business resources to cloud platforms, the willingness to share resources within a business network must be examined.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Aulkemeier, F., Paramartha, M. A., Iacob, M.-E., & van Hillegersberg J. (2016). A pluggable service platform architecture for e-commerce. *Information Systems and e-Business Management*, 14(3), 469–489.
- Aulkemeier, F., Iacob, M.-E., & van Hillegersberg J. (2017). An architectural perspective on service adoption: A platform design and the case of pluggable cross-border trade compliance in e-commerce. *Journal of Organizational Computing and Electronic Commerce*, 27(4), 325–341.

- Aussenac-Gilles, N., Biebow, B., & Szulman, S. (2000). Revisiting ontology design: A method based on corpus analysis. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 172–188). Springer.
- Bajec, M., & Krisper, M. (2005). A methodology and tool support for managing business rules in organisations. *Information Systems*, 30(6), 423–443.
- Baldwin, C. Y., & Woodard, C. J. (2009). The architecture of platforms: A unified view. In A. Gawer (Ed.), *Platforms, Markets and Innovation* (pp. 19–44). Edward Elgar.
- Bichler, M., Beam, C., & Segev, A. (1998). Services of a broker in electronic commerce transactions. *Electronic Markets*, 8(1), 27–31.
- Boley, H., & Chang, E. (2007). Digital ecosystems: Principles and semantics. In *Digital EcoSystems and Technologies Conference* (pp. 398–403). Cairns.
- Busse, S., Kutsche, R.-D., Leser, U., & Weber, H. (1999). *Federated information systems: Concepts, terminology and architectures*. Technische Universität Berlin.
- Chandra, D. R., & Hillegersberg, J. van. (2015). The governance of cloud based supply chain collaborations. In *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 1608–1612). Singapore.
- Chick, S. E., Huchzermeier, A., & Netessine, S. (2014). Europe's solution factories. *Harvard Business Review*, 92(4), 111–115.
- Cross, N. (2006). *Designerly ways of knowing*. Springer London.
- Damsgaard, J., & Truex, D. (2000). Binary trading relations and the limits of EDI standards: The procrustean bed of standards. *European Journal of Information Systems*, 9(3), 173–188.
- Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on Software Engineering*, 21(2), 146–162.
- Evans, E. (2004). *Domain-driven design: Tackling complexity in the heart of software*. Addison-Wesley Professional.
- Evans, D. S., & Schmalensee, R. (2010). Failure to launch: Critical mass in platform businesses. *Review of Network Economics*, 9(4).
- Fawcett, S. E., McCarter, M. W., Fawcett, A. M., Webb, G. S., & Magnan, G. M. (2015). Why supply chain collaboration fails: The socio-structural view of resistance to relational strategies. *Supply Chain Management: An International Journal*, 20(6), 648–663.
- Giagliis, G. M., Klein, S., & O'Keefe, R. M. (2002). The role of intermediaries in electronic marketplaces: Developing a contingency model. *Information Systems Journal*, 12(3), 231–246.
- Handschuh, S., Schmid, B. F., & Stanoevska-Slabeva, K. (1997). The concept of a mediating electronic product catalog. *Electronic Markets*, 7(3), 33–35.
- Hardt, D. (2012). *The OAuth 2.0 Authorization Framework* (technical standard). Internet engineering task force.
- Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1), 1–136.
- van Heck, E., & Vervest, P. (2007). Smart business networks: How the network wins. *Communications of the ACM*, 50(6), 28–37.
- van Hillegersberg, J., Moonen, H., & Dalmolen, S. (2012). Coordination as a service to enable agile business networks. In J. Kotlarsky, I. Oshri, & L. P. Willcocks (Eds.), *The dynamics of global sourcing. Perspectives and practices* (pp. 164–174). Berlin Heidelberg: Springer.
- Hohpe, G., & Woolf, B. (2003). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley.
- Kleeberg, M., Zirpins, C., & Kirchner, H. (2014). Information systems integration in the cloud: Scenarios, challenges and technology trends. In G. Brunetti, T. Feld, L. Heuser, J. Schnitter, & C. Weibel (Eds.), *Future Business Software* (pp. 39–54). Springer Cham.
- Koppius, O. R., & van de Laak, A. J. (2009). The quick-connect capability and its antecedents. In P. H. M. Vervest, D. W. van Liere, & L. Zheng (Eds.), *The network experience* (pp. 267–284). Berlin Heidelberg: Springer.
- Lincke, D.-M., & Schmid, B. (1998). Mediating electronic product catalogs. *Communications of the ACM*, 41(7), 86–88.
- Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., & Ueda, K. (2016). Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2), 621–641.
- Nitu. (2009). Configurability in SaaS (Software As a Service) Applications. In *Proceedings of the 2Nd India Software Engineering Conference* (pp. 19–26).
- Palmer, J. W., & Johnston, J. S. (1996). Business-to-business connectivity on the internet: EDI, intermediaries, and interorganizational dimensions. *Electronic Markets*, 6(2), 3–6.
- Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE)* (pp. 3–12).
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77.
- Pellegrini, T. (2017). Semantic metadata in the publishing industry – Technological achievements and economic implications. *Electronic Markets*, 27(1), 9–20.
- Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 64–88.
- Prat, N., Akoka, J., & Comyn-Wattiau, I. (2006). A UML-based data warehouse design method. *Decision Support Systems*, 42(3), 1449–1473.
- Ramanathan, U., & Gunasekaran, A. (2014). Supply chain collaboration: Impact of success in long-term partnerships. *International Journal of Production Economics*, 147, 252–259.
- Reimers, K. (2001). Standardizing the new e-business platform: Learning from the EDI experience. *Electronic Markets*, 11(4), 231–237.
- Salto, F., Castellanos, M., & García-Solaco, M. (1991). Suitability of Datamodels as canonical models for federated databases. *ACM SIGMOD Record*, 20(4), 44–48.
- Sen, R., & King, R. C. (2003). Revisit the debate on intermediation, disintermediation and Reintermediation due to E-commerce. *Electronic Markets*, 13(2), 153–162.
- Sherer, S. A., & Adams, B. (2001). Collaborative commerce: The role of intermediaries in e-collaboration. *Journal of Electronic Commerce Research*, 2(2), 66–77.
- Sheth, A. P., & Larson, J. A. (1990). Federated database Systems for Managing Distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), 183–236.
- Sørensen, C., de Reuver, M., & Basole, R. C. (2015). Mobile platforms and ecosystems. *Journal of Information Technology*, 30(3), 195–197.
- The Open Group. (2016). *ArchiMate 3.0 Specification*. The Open Group.
- Verschuren, P., & Hartog, R. (2005). Evaluation in design-oriented research. *Quality and Quantity*, 39(6), 733–762.
- Vujasinovic, M., Barkmeyer, E., Ivezic, N., & Marjanovic, Z. (2010). Interoperable supply-chain applications: Message metamodel-based semantic reconciliation of b2b messages. *International Journal of Cooperative Information Systems*, 19(01n02), 31–69.
- Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer*, 25(3), 38–49.
- Włodarczyk, T. W., Rong, C., & Thorsen, K. A. H. (2009). *Industrial cloud: Toward inter-enterprise integration* (In *Cloud Computing* (Vol. E-commerce product recommendation agents, pp. 460–471)). Berlin Heidelberg: Springer.
- Xiao, B., & Benbasat, I. (2007). E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Quarterly*, 31(1), 137–209.