Information Systems: Transforming the Future

24th Australasian Conference on Information Systems, 4-6 December 2013, Melbourne

Proudly sponsored by

ACIS 2013 Principal Sponsor

Towards an Electronic Marketplace for Bricks-and-Mortar Services

Peter Schenkel, Philipp Osl, Hubert Oesterle
Institute of Information Management
University of St. Gallen
St. Gallen, Switzerland
Email: {firstname}.{lastname}@unisg.ch

Abstract

Service provision of bricks-and-mortar services (e.g. cleaning, gardening) poses several challenges to the consumer. Finding a service provider as well as ordering and coordinating the service provision, requires intensive interaction between consumer and service provider. Due to the regional anchoring of these services, they are, to a large extent, provided by small- and medium-sized enterprises (SMEs). This poses additional challenges to the consumer: the market is fragmented and processes differ across service providers and industries. This problem is well-solved for tangible goods: consumers buy goods from different sellers via one marketplace (such as Amazon marketplace, eBay, etc.) and a seller-independent process. For services a similar consumer support is lacking. In this paper we address the gap from a consumer's perspective by proposing a software architecture that integrates standard applications and modules to support the consumer process. While the work is still in progress, first practice applications demonstrate the artifact's usefulness and viability.

Keywords

BPMN 2.0, software architecture, interaction modeling, service business, electronic marketplace

Introduction

Online marketplaces gained a tremendous increase in popularity over the last years. For consumers it is nowadays possible to buy any product from many different sellers using electronic marketplaces. The consumer benefits from uniform processes (i.e. secure payment), a large portfolio of products, intelligent product search and user ratings on products and sellers. A transparent order processing (e.g. dispatching notification) allows the user to track his order until the final delivery of the product. The consumer process for buying products online is well-understood and comprehensively supported. Shifting the focus from products (tangible goods) to services, the situation is slightly different. Traditional concepts to find a service provider include the yellow pages (a list of service providers in a specific region). The consumer is solely supported in his attempt to find service providers. State-of-the-art electronic service marketplaces (such as myhammer.de, redbeacon.com, taskrabbit.com, etc.) go further by offering means to contact service providers as well as to rate the service provision afterwards. In contrast to tangible goods, the provision of bricks-and-mortar services requires extensive coordination between consumer and service provider. From a consumer perspective a service marketplace would be favorable which supports time-consuming and error-prone interaction (e.g. appointment coordination) with service providers as well as the possibility to gain access to a large service portfolio via uniform processes.

Since one of the first theoretical approaches to electronic markets in 1987 (Malone et al. 1987), the research area of electronic markets steadily developed and considerably contributed to the proliferation of electronic marketplaces in practice to match supply and demand in a globalized economy. A retrospective interview with Malone (Wigand 2011) about the development of electronic marketplaces (EMs) over the last 20 years reveals the ongoing attention EMs receive in science. Recent literature reviews (Alt and Klein 2011; Wang et al. 2008) show the increasing importance of EMs, both in science and practice. With the emergence of Business-to-Consumer (B2C) online intermediaries (e.g. eBay, Amazon marketplace), marketplaces advanced from simple match-making platforms to fully integrated web shop systems (Langdon et al. 2000; Treutner and Ostermann 2011), offering extensive support for sellers (e.g. checkout, inventory management, payment) and for consumers (e.g. standard processes, secure payment, product search). (Leitner and Grechenig 2007) identified consumer centricity as a key success factor for B2C marketplaces. Predominantly the research in this area focuses on marketplaces for tangible goods or digital services without addressing the special challenges arising when selling bricks-and-mortar services (e.g. house cleaning, personal training). In absence of a commonly used term for such services, we refer to all services which are provided manually in accordance or cooperation with the consumer as "bricks-and-mortar" services. In practice, the few existing electronic service marketplaces cover the consumer process rather rudimentarily than comprehensively.

Neither practical nor scientific approaches explicitly deal with the specific problems evolving from this particular application domain. Due to their regionally bound character, services are offered and provided predominantly by small- and medium-sized enterprises (SMEs). This results in a large number of service providers per region, leading to market fragmentation and ultimately to increased complexity for consumers in terms of finding and comparing service offers as well as a multiplicity of processes among different service providers and different services. In addition, Information and Communication Technology (ICT) for SMEs is less standardized and more diverse in comparison to applications for large enterprises (LEs) (Stockdale and Standing 2004). Consequently, we identify process and application heterogeneity as a main technical challenge for the implementation of such a service marketplace, given the fact that 89,5% of all services e.g. in Switzerland are provided by SMEs with 10 employees or less (Bundesamt für Statistik 2013).

The main contribution of this work is to analyze approaches in literature and practice and to develop an advanced electronic Service Management Platform (SMP) for the specific requirements of brokering services within an SME environment. We approach this goal from a consumer centric perspective as advocated in (Alt and Puschmann 2012). The relevance of this research topic is described in (Winter et al. 2012).

This research is part of the Independent Living project that aims to increase the independency of people, especially senior citizens living at their familiar home environment. Within this project, consumer requirements have been investigated over a period of two years and the proposed architecture is implemented and finally tested in a productive environment. The work delivers insights into the technical feasibility of such a platform and into sustainability of the solution in practice as initial point for further research on this topic. The structure of the paper strictly links to the applied research methodology. In this section we describe the application domain and motivate the development of an artifact (software architecture) for electronic service management platforms. The next section introduces the applied design-driven research framework. The section *Related Work* gives an overview of relevant scientific streams with impact on the architecture development and existing solutions in practice and their respective limitations. In the next step the *objectives of the solution* are derived, serving as the basis for architecture development. In the section *Design and Development*, the software architecture, based on the requirements, is introduced alongside the underlying development principles. In the *Demonstration* part, the implementation of the architecture for a given application domain is described. The paper closes with preliminary insights from the pilot application and a conclusion of the work.

The Research Framework

Following our objective to design an IT artifact (SMP), we pursue a Design Science Research (DSR) instead of a behavioral research approach because of two reasons: (1) we design an innovative artifact, which does not exist currently. (2) Coherently, we classify the current state-of-the-art in electronic service marketplaces as the observed phenomenon, aiming to improve the status-quo by identifying scope for improvements (Österle et al. 2010). Both arguments contradict the objectives of behavioral research which assumes the existence of a phenomenon and seeks to explain or predict human behavior.

To ensure compliance with the requirements of DSR, we follow the research guidelines proposed by (Hevner et al. 2004) by transferring them to the given application domain as described in Table 1. We operationalize our research by applying the process model (DSRP) as described by (Peffers et al. 2006).

Table 1: Design science guideline compliance according to (Hevner et al. 2004)

Guideline	Description
Design as an Artifact	Two artifacts are designed within the research process. A software architecture (type: model) and, as part of the evaluation phase, an instantiation of the architecture (type: instantiation)
Problem Relevance	Lacking comprehensive support through service marketplaces of the consumer process for bricks-and-mortar services.
Design Evaluation	The architectural design is *technically validated* by implementing the architecture for the described business context and executing predefined test cases. The final phase of evaluation is a *controlled experiment* to study the qualities of the artifact in a real application. This phase is still ongoing

Research Contributions	The proposed architecture can be applied to a class of problems with similar characteristics: ▪ integration of heterogeneous applications/processes ▪ continuously changing processes ▪ m:n interaction support The designed and instantiated artifact serves as a model for practitioners. Methodically this project advocates a rigorous consumer driven approach to develop software architectures.
Research Rigor	By strictly following the DSRP as described, a rigorous research process is ensured.
Design as a Search	Prior to the artifact instantiation requirements are iteratively collected over a period of 12 months while continuously adapting the architecture. Theoretical foundation is taken from different relevant literature streams and the analysis of solutions in practice.
Communication of Research	The findings of this research address technical as well as economic topics and are disseminated by scientific publications.

We identify a design gap (*Problem Identification & Motivation*) by analyzing the status-quo of electronic service marketplaces from a consumer perspective: the consumer process is well understood and supported by electronic marketplace for tangible goods. This raises the question why no similar, comprehensive support of the consumer process exists for bricks-and-mortar services.

We derive the *objectives of the solution* starting from a consumer perspective and, consequently, translating consumer requirements into technical requirements for the software architecture. Based on the consolidated consumer requirements, an architecture is proposed *(Design & Development)* which is capable of meeting the defined requirements. The software architecture is implemented to proof the practical relevance of the solution. A special requirement is to develop and implement a system which goes further than a demonstrator and which is applicable in a productive environment *(Demonstration)*. This allows to prove the solution in a pilot setting and to continuously feed consumer requirements back into the development cycle. *Evaluation and Communication* is still ongoing work. The iterative consolidation of user requirements and the development of the architecture took around 2 years. Since March 2013 the system is live and first evaluation results are expected by end of the year.

Related Work

There exist several scientific streams with impact on this work.

Modeling service choreographies and interaction modeling is a major research area which contributes to the proposed software architecture. Decker and Barros propose a BPMN extension for interaction modeling (Decker and Barros 2007). Existing conceptual approaches to model service choreographies (Decker, Kopp, and Barros 2008; Decker, Kopp, Leymann, et al. 2008) have not proven practical relevance, yet. Modeling interaction processes is a central objective of the solution allowing the support of interaction between consumer and service provider.

The *integration of heterogeneous applications* and, subsequently, *the integration of heterogeneous processes* are of special importance in an SME environment. (Lind and Goldkuhl 2005) propose a "Business phase matrix" to align process variants to different business transaction phases, strongly depending on product characteristics in an 1:n relationship between seller and buyer. Therefore, this concept especially applies to marketplaces where a large number of sellers and buyers cooperate. The lack of standardization is one of many barriers for SMEs to adopt ICT, which aggravates the situation (Stockdale and Standing 2004) and leads, among other factors, to a large number of heterogeneous applications among SMEs (Hoyer 2008). A large number of scientific contributions argue that document standards such as ebXML, RosettaNet, UN/EDIFACT (Liegl et al. 2010), communication standards such as SOAP, interface descriptions (e.g. WSDL) or description standards (e.g.

USDL) solve this problem (Dumas 2005). Various studies disagree, stating that standards itself do not solve inter-organizational cooperation problems (here: the connection between platform and service provider backend application). Despite existing standards, (Beck et al. 2002) state that a key challenge for SMEs is to adopt ICT in general, advocating easy-to-use and easy-to-adopt software for SMEs. A comparative analysis by (Buonanno et al. 2005) shows that out of the 153 interviewed companies with less than 50 employees only 14 use ERP systems to manage their business. We therefore conclude that maturity, both in applications and processes, is still a major challenge for marketplaces with predominantly SMEs as service providers. (Lu et al. 2006) found out that industry standards are only one success factor of inter-organizational cooperation among several other factors.

Finally, there is considerable work in the area of *software architectures for B2B transaction support* (Gionis et al. 2007; Jiang 2009; Stemmer et al. 2011) providing input for our work. In recent years most approaches in the software architecture domain focus on cloud applications, advocating Software-as-a-Service as a "revolution in business" (Münzl et al. 2009) and as state-of-the-art software architecture design paradigm (Bateman and Wood 2009; GFI Software 2010; Holtkamp 2010; Münzl et al. 2009; Stemmer et al. 2011). Except for a few rare implementations (Holtkamp 2010) most architectural approaches stay on a conceptual level without proof-of-concept in reality, yet, contributing valuable input to the artifact design.

For each field of research there are mature concepts, which are transferred in the new domain and incorporated in the proposed architecture. The main contribution of this work is not to advance concepts and solutions in each single field of research but to combine concepts and transferring them into a new application domain. To tackle the requirements we apply paradigms such as service-oriented architecture (SOA) as it has found application in similar systems, addressing the integration of heterogeneous applications. BPMN as process modeling has been widely used to design interaction processes and with BPMN 2.0 the tasks of process designing and process execution have been merged by allowing direct execution of processes by following standardized process design patterns. To make the interaction process accessible as a service we use similar concepts as used in cloud-oriented architecture patterns.

We contribute to the body of knowledge by implementing the concepts and proof their viability through demonstration and evaluation.

Objectives of the Solution

The requirements for an SMP are derived by analyzing customer needs, based on the principles of customer orientation as described in (Kagermann et al. 2010). Kagermann advocates customer orientation as a key challenge and success factor for companies. This implies support of the customer process comprehensively in all phases of the value creation by following a set of principles including "Everything", "Everywhere", "Anytime", "One-Stop", "Segment-of-One", "One-face-to-the-customer" and "Anyhow". The principles are described in the context of the given application domain below.

"Everything" means it is essential that an SMP supports different kinds of services as well as the complete service usage process, from finding the right service to payment, in order to serve the customer's needs comprehensively. Most solutions in practice have to deal with a trade-off between continuous consumer process support and a large service portfolio. There is no solution yet, which is capable to handle the complexity of continuous process support as well as a large variety of services. Supporting a large variety of services supra-regional comes to the cost of massive complexity to connect service providers. Assuming *m* service providers per region and a coverage of *n* regions, results in *m*n* different service provider backend instantiations which need to be integrated in the consumer process. Instead of reducing complexity by reducing the variety we reduce the scope from supra-regional to regional and therefore to a manageable complexity. The software architecture must be able to handle different process variants and different service provider backend applications in a dynamic business environment *(R1)*. The consumer does not want to be bound to a certain point of sale to order services but wants to order services "*Everywhere"* using the device which suits him best (*"Anyhow")*. Therefore the system must be capable to offer a standard compliant interface to connect different sorts of frontend technologies (e.g. mobile frontends) *(R2)*. The "*Anytime"* aspect gains special importance in an SME environment where availability of a contact person at a service provider is not self-evident, since there is usually no dedicated contact person (e.g. a contact person has to fulfill services as well and might be busy at the time of contact) . A 24/7 availability of the system is a basic requirement. This requires asynchronous messaging as core concept for a human-to-human interaction *(R3)*. A comprehensive support of the consumer process is essential ("*One-Stop"*). Extensive interaction between consumer and service provider is a key aspect of service provision (a service is provided in accordance and/or cooperation with the consumer). The one-stop principle is only partly realizable by integrating the service provider in the consumer process. However, extensive interaction is a constitutive characteristic of bricks-and-mortar services and can only be supported, but not avoided (*R4)*. A service is provided in accordance and/or cooperation with a consumer and is therefore a personalized product ("*Segment-of-One"***)**. Thus, this principle is inherently followed in this domain. From a consumer's perspective processes

should always be alike and all information regarding the process should be universally available ("*One-face-to-the-Customer"*), no matter what kind of service is ordered from which service provider. In a fragmented market as in the given application domain, this means to shift complexity from the consumer to the platform to offer a unified frontend user interface. (*compare R1*). Additionally, following this principle requires a central point of contact for the consumer. This means to integrate means for the marketplace operator to support the respective business model *(R5)*. The consumer should be able to access the marketplace offers by his device of choice ("*Anyhow"*) to initialize the consumer process.

An SMP is an independent business entity and therefore needs an own *user management (R6)* system. There is an additional set of non-functional requirements (Medvidovic et al. 2000) which need to be considered for the platform architecture. To ensure *scalability (R7)*, the architecture follows the paradigm of service oriented architectures (SOA). Applications are loosely coupled by encapsulating and publishing the functionalities via stateless web services. This enables load balancing and the replacement of single modules to keep track with technical improvements. For evaluation in practice, the software architecture must implement an adequate level of communication *security (R8)*. To conclude, the previously derived requirements are translated into technical requirements as in Table 2.

Table 2: Translation consumer and general requirements to technical requirements

Requirement	Short description	Technical Requirement
R1	process configuration and adaption	▪ configurable processes (TR1) ▪ directly executable process designs for continuous change implementation (TR2)
R2	flexible frontend interfaces	▪ standard web service interfaces for communication (TR3)
R3	process execution	▪ process state persistence for asynchronous messaging (TR4)
R4	flexible backend interfaces	▪ process must be capable to communicate with various third party applications (TR5)
R5	platform operator backend	▪ integration of an ERP component to support respective business models of marketplace operators (TR6)
R6	user management	▪ integration of standard user management components (TR7)
R7	service oriented architecture	▪ component coupling via web services (TR8)
R8	security	▪ state-of-the-art authentication and encryption (TR9)
		▪

Design and Development: Software Architecture

It is intended to use as many standard applications as possible to keep development and maintenance costs as low as possible. The architecture follows the paradigms of service oriented architectures (SOA) and cloud computing as Raines describes (Raines 2009). SOA aims to integrate diverse systems in an architecture by loosely coupling applications via service specifications, such as REST or SOAP. This allows easy use of existing services and modules To enable m:n communication between consumers and service providers, interaction processes must be inherently publicly accessible for the involved parties. The process engine therefore acts as a platform to execute processes which are offered as a service in the cloud (Vaquero et al. 2009).

The derived requirements are aligned to applications which are capable to cover the functional scope. Table 3 shows the alignment of requirements to applications. Figure 1 shows the modular architecture including interaction patterns.

Table 3: Alignment requirement to module

Technical Requirement	Application
▪ configurable processes (TR1) ▪ process state persistence for asynchronous messaging (TR4) ▪ process must be capable to communicate with various third party applications (TR5) ▪ state-of-the-art authentication and encryption (TR9) ▪ component coupling via web services (TR8)	Process Engine
▪ standard web service interfaces for communication (TR3)	Data Service
▪ directly executable process designs for continuously implementing changes (TR2)	Process Repository
▪ integration of an ERP component to support respective business models of marketplace operators (TR6)	ERP
▪ integration of standard user management components (TR7)	User Management

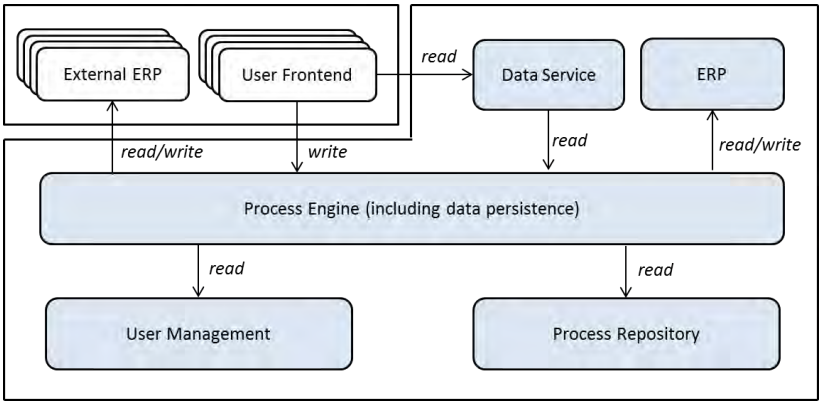

Figure 1: SMP - modular architecture and interaction permissions

The *process engine* is responsible for executing process definitions stored in the *Process Repository*. The engine is the core of the architecture. It ensures encrypted interfaces for the communication with third party applications and integrates required applications in the process. The process engine offers Software-as-a-Service for users on the highest level and, in combination with the process repository, an application framework (Platform-as-a-service), where platform operators are able to design and change processes for the higher layer. The integrated *Enterprise-Resource-Planning (ERP)* module supports the business model of the platform operator. Therefore each economically relevant transaction is handed over to the ERP module for further processing. The *data service* offers atomic web services to read data. This is important to synchronize user frontends which cannot be served via push mechanisms. The platform acts as an independent electronic marketplace, allowing an m :n interaction between consumers and multiple service providers. A *user management* is a mandatory module for authentication and user master data.

Demonstration and Evaluation: Architecture Implementation

We divide the artifact evaluation in two parts: as a first evaluation we validate the software architecture by implementing. The implementation demonstrates the practicability of the developed architecture. The functionality of the implementation is evaluated in a pilot setting. The second ongoing part of the evaluation covers the acceptance of the system by practitioners.

A major objective for the implementation is to use standards as well as scalable, open source applications. For applications (e.g. ERP), which need to comply with certain business standards and frequent updates due to regulations, a proprietary, but mature application is preferred since costs for adapting open source ERPs and maintenance are unpredictable. To secure communication with external applications we choose Transport Layer Security protocol (TLS), which is widely used to secure communication channels. To persist process definitions we chose BPMN 2.0, a business process notation (Chinosi and Trombetta 2012; White 2004) which is designed to be directly executable and a widely used standard for process design (Rücker 2012). The choice of this notation influences the choice of the process engine. Figure 2 shows the implementation of the architecture. The applications and frameworks are described below. Compared to “traditional” monolithic software architectures, a

component based modular architecture offers more flexibility regarding performance, application replacement and testing (Martens et al. 2010).

Process engine. (Rücker 2012) analyzed existing process engines, based on BPMN 2.0, proposing a set of technical and economic criteria to consider regarding the choice of an appropriate process engine. Following a design science approach we rate easy extensibility and flexibility as pivotal criteria. We therefore chose the activiti framework as basis for further development (Activiti.org 2013). This is an open-source lightweight process engine with a vivid development community. First benchmarks proof the performance of the activiti framework (Barrez 2012) in practice. To match the defined requirements, we extended the activiti framework by components to integrate a user management and means to persist transactional business objects within a process. This is implemented using an Object-Relational-Mapping (ORM) framework. To ensure database independence, as well as maturity of the framework we use hibernate as ORM (Hibernate.org 2013). For business object as well as process state persistence we use a MySQL database, since it is a well-performing open source database which can be easily upgraded to a clustered version as well for better scalability. Activiti runs on Apache Tomcat, which is a mature and well-established server framework.

*ERP***.** Even though there are many cloud ERP systems available we decided to use SAP Business ByDesign as ERP cloud solution. SAP has a long-time experience in complex business systems and scalability as one of their core concepts. On top, SAP offers a comprehensive software development kit (SDK) to develop customer specific solutions. This component is not available under open-source license, but continuous product support is crucial for economically relevant systems as well as high availability outweighs the cost argument.

User Management. Following the principle of data isolation, we use a single purpose user management system, clearly separating user data from transactional data. We use the Lightweight Directory Access Protocol (LDAP) for user management, which is optimized for fast reading access and a widely used standard. We use an LDAP implementation from Apache (http://directory.apache.org/).

Process Repository. Process definitions are stored in MySQL database. Read and write operations on process definitions are less performance critical (read access at each process start, write access only when designing/changing process definitions). There are no special requirements on the process repository. We use a MySQL database as process repository to reduce the number of different data storage technologies to one.

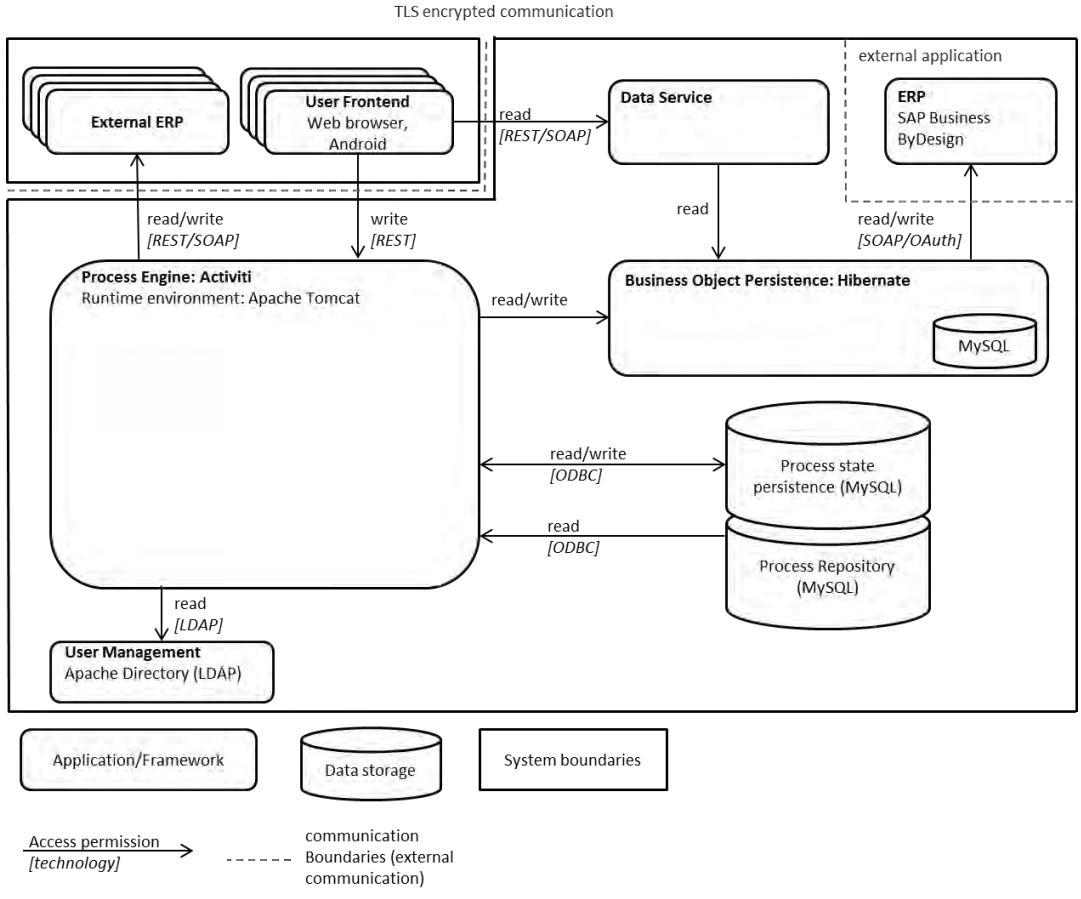

Figure 2: Implementation of the proposed software architecture

To demonstrate the viability of the system implementation we launched a pilot in March 2013 in "Weil der Stadt", a city in Germany. This pilot currently covers 195 different services with 35 connected service providers and 30 customers in the first evaluation phase. We expect insights regarding feasibility, usability, performance and acceptance (both, for consumers and for service providers) after the first evaluation phase by the end of 2013. The insights are fed back in the next refinement iteration of the development. Coordination and communication costs between consumer and service provider can be significantly reduced due to the 24/7 availability of the system and the asynchronous communication (compare synchronous communication via telephone): a service provider can answer a request upon availability, whereas the consumer can send requests to the service provider anytime.

The system enables consumers to request services (gardening, cleaning, shopping) in "Weil der Stadt" via smartphone and web shop by indicating the preferred service provider and a preferred execution date. The requests are processed by the system and sent to the appropriate service provider via email or web frontend. The service provider can either confirm the date or propose alternative dates to the consumer. The process is designed to escalate unanswered requests (within a certain timeframe) to the platform operator for resolving with the service provider. A confirmed request is persisted as order in the platform ERP, enabling the platform operator to pool consumer invoices on a monthly basis and handling supplier liabilities. Depending on the operator's business model, the consumer can receive either one consolidated invoice from the platform operator (operator as market-maker), allowing margins for the operator, or be directly billed by service providers (operator as match-maker), allowing revenues via transaction fees. The consumer and service provider maintains a contract with the platform operator.

The system is evaluated in two phases to proof the flexibility of the system in practice. In the first, currently ongoing, evaluation, the match-maker scenario is evaluated. All transactions regarding payment are between consumer and service provider. The platform operator is responsible for quality management and earns a transaction fee.

Preliminary results show that the implemented system is well accepted by service providers as an additional sales channel. This is reflected by a growing number of proactive requests from service providers in the region to be listed on the platform as well as recommendations concerning additional providers to be added. To keep the acceptance threshold as low as possible, service providers have been connected to the system via email without any deep integration into their existing systems which we consider as a major argument to attract service providers in this first evaluation phase. Consumer adoption and usage is still reserved. We therefore put special effort in promoting the concept among consumers. Focusing on two specific target groups for which we offer pinpoint service portfolios, show first effect regarding consumer usage.

The second evaluation, starting in January 2014, evaluates the market-maker scenario where the platform operator acts as independent business unit, buying services from service providers and selling with an extra charge to the consumer. In this case, the platform operator is not only responsible to ensure the quality of the services but also to handle invoicing and payments.

Conclusion & Further Work

In this work we developed a software architecture for bricks-and-mortar service marketplaces by deriving platform requirements strictly based on consumer needs. The consumer requirements are translated into technical requirements in due consideration of context-specific factors. The technical requirements are aligned to functional clusters (modules). Consequently, an architecture is proposed, which integrates modules and standard applications to cover the required functional scope. Following the DSRP we implemented the architecture for demonstration purpose, using standard applications to cover the derived requirements and a service-oriented architecture to enable application interaction. For evaluation, a pilot application is still ongoing in a real world setting in the city "Weil der Stadt". This project is work in progress and further evaluation results of the platform are expected by the end of 2013. We expect that the evaluation provides insights on benefits and limitations of the system and further improvements of business processes for the next development iteration. Entering the market of rather small- than medium-sized companies with all underlying implications, we see this work as a first approach to tackle existing barriers (process and application heterogeneity) by a flexible, architecture with a process-driven integration of service providers and consumers. An electronic marketplace for regional bricks-and-mortar services removes huge market inefficiencies, for consumers as well as for service providers, and therefore could play a similar role as electronic markets for physical goods.

References

Activiti.org. 2013. “Activiti,” .

Alt, R., and Klein, S. 2011. “Twenty years of electronic markets research—looking backwards towards the future,” *Electronic Markets* (21:1), pp. 41–51.

Alt, R., and Puschmann, T. 2012. “The rise of customer-oriented banking - electronic markets are paving the way for change in the financial industry,” *Electronic Markets* (22:4), pp. 203–215.

Barrez, J. 2012. “The Activiti Performance Showdown,” .

Bateman, A., and Wood, M. 2009. “Cloud computing.,” *Bioinformatics (Oxford, England)* (25:12), p. 1475.

Beck, R., Weitzel, T., and König, W. 2002. “Promises and Pitfalls of SME Integration,” (1), pp. 567–583.

Bundesamt für Statistik. 2013. “Industrie und Dienstleistungen Panorama,” , pp. 1–9.

Buonanno, G., Faverio, P., Pigni, F., Ravarini, a., Sciuto, D., and Tagliavini, M. 2005. “Factors affecting ERP system adoption: A comparative analysis between SMEs and large companies,” *Journal of Enterprise Information Management* (18:4), pp. 384–426.

Chinosi, M., and Trombetta, A. 2012. “BPMN: An introduction to the standard,” *Computer Standards & Interfaces* (34:1), pp. 124–134.

Decker, G., and Barros, A. 2007. “Interaction Modeling using BPMN,” in *International conference on Business process management*, , pp. 1–12.

Decker, G., Kopp, O., and Barros, A. 2008. “An Introduction to Service Choreographies,” .

Decker, G., Kopp, O., Leymann, F., Pfitzner, K., and Weske, M. 2008. “Modeling Service Choreographies using BPMN and BPEL4Chor,” in *International conference on Advanced Information Systems Engineering*, , pp. 79–93.

Dumas, M. 2005. “A Critical Overview of the Web Services Choreography Description Language Alistair Barros,” (March), pp. 1–24.

GFI Software. 2010. “On-premise vs . Cloud-based Solutions,” , pp. 1–9.

Gionis, G., Mouzakitis, S., Janner, T., Schroth, C., and Koussouris, S. 2007. “Implementing Next Generation e-Business Platforms for Heterogeneous SME Environments,” in *Panhellenic Conference in Informatics*, , pp. 532–540.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. “Design Science in Information Systems Research,” *MIS Quarterly* (28:1), pp. 75–105.

Hibernate.org. 2013. “Hibernate,” .

Holtkamp, D. B. 2010. “Cloud computing für den mittelstand am beispiel der logistikbranche,” , pp. 1–22.

Hoyer, V. 2008. “Modeling Collaborative e-Business Processes in SME environments,” *Journal of Information Science and Technology* (5:2).

Jiang, B. 2009. “Flexible Business Process Integration for Clusters of Small-Medium Sized Enterprises in Heterogenous Environment,” (4:4), pp. 315–322.

Kagermann, H., Österle, H., and Jordan, J. M. 2010. *IT-Driven Business Models*, John Wiley & Sons Inc., p. 220.

Langdon, C. S., Hall, B., and Shaw, M. J. 2000. “The Online Retailing Challenge : Forward Integration and E-Backend Development,” *ECIS* .

Leitner, P., and Grechenig, T. 2007. “NEXT GENERATION SHOPPING : CASE STUDY RESEARCH ON FUTURE E-COMMERCE MODELS,” (2005), pp. 312–316.

Liegl, P., Zapletal, M., Pichler, C., and Strommer, M. 2010. "State-of-the-art in business document standards," *2010 8th IEEE International Conference on Industrial Informatics* Ieee, pp. 234–241.

Lind, M., and Goldkuhl, G. 2005. "Designing business process variants," , pp. 27–28.

Lu, X.-H., Huang, L.-H., and Heng, M. S. H. 2006. "Critical success factors of inter-organizational information systems—A case study of Cisco and Xiao Tong in China," *Information & Management* (43:3), pp. 395–408.

Malone, T. W., Yates, J., and Benjamin, R. I. 1987. "Electronic Markets And Electronic Hierarchies," *ACM* (30:6), pp. 484–497.

Martens, A., Koziolek, H., Prechelt, L., and Reussner, R. 2010. "From monolithic to component-based performance evaluation of software architectures," *Empirical Software Engineering* (16:5), pp. 587–622.

Medvidovic, N., Taylor, R. N., and Society, I. C. 2000. "A Classification and Comparison Framework for Software Architecture Description Languages," *Transactions on software engineering* (26:1), pp. 70–93.

Münzl, G., Przywara, B., Reti, M., Schäfer, J., Sondermann, K., Weber, M., and Wilker, A. 2009. "Cloud Computing - Evolution in der Technik , Revolution im Business," , pp. 1–84.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., and Sinz, E. J. 2010. "Memorandum on design-oriented information systems research," *European Journal of Information Systems* (20:1), pp. 7–10.

Peffers, K., Gengler, C. E., Rossi, M., Hui, W., and Bragge, J. 2006. "The Design Science Research Process: A Model For Producing And Presenting Information," in *DESRIST*, , pp. 84–106.

Raines, G. 2009. "Service-Oriented Architecture (SOA) SERIES Cloud Computing and SOA," , p. 12.

Rücker, B. 2012. "Prozesse in Bewegung," *javamagazin* , pp. 86–95.

Stemmer, M., Holtkamp, B., and Königsmann, T. 2011. "Cloud-orientierte Service-Marktplätze Integrationsplattformen für Moderne Dienstleistungen und IT-Dienste," , p. 53.

Stockdale, R., and Standing, C. 2004. "Benefits and barriers of electronic marketplace participation: an SME perspective," *Journal of Enterprise Information Management* (17:4), pp. 301–311.

Treutner, M. F., and Ostermann, H. 2011. "Evolution of Standard Web Shop Software Systems : A Review and Analy- sis of Literature and Market Surveys," , pp. 8–18.

Vaquero, L. M., Rodero-merino, L., Caceres, J., and Lindner, M. 2009. "A Break in the Clouds : Towards a Cloud Definition," (39:1), pp. 50–55.

Wang, S., Zheng, S., Xu, L., Li, D., and Meng, H. 2008. "A literature review of electronic marketplace research: Themes, theories and an integrative framework," *Information Systems Frontiers* (10:5), pp. 555–571.

White, S. A. 2004. "Introduction to BPMN," *BPTrends*, , pp. 1–11.

Wigand, R. T. 2011. "20 Years of Research in Electronic Markets and Networked Business: An Interview with Thomas Malone," *Electronic Markets* (21:1), pp. 5–17.

Winter, A., Alt, R., Ehmke, J., Haux, R., Ludwig, W., Mattfeld, D., Oberweis, A., and Paech, B. 2012. "Manifest – Kundeninduzierte Orchestrierung komplexer Dienstleistungen," *Informatik-Spektrum* (35:6)Springer-Verlag, pp. 399–408.

COPYRIGHT