Leveraging Standard Software from the Cloud with Service-Oriented EAM

Helge Buckow, Hans-Jürgen Groß, Gunther Piller,
Norbert Stumpf, Oliver F. Nandico, Johannes Willkomm, Alfred Zimmermann

SOA Innovation Lab e.V.
Workstream SOA and Standard Platforms
c/o Deutsche Post AG
Charles-de-Gaulle-Straße 20
53113 Bonn

info@soa-lab.de, helge_buckow@mckinsey.com, hans-juergen.gross@daimler.com,
gunther.piller@fh-mainz.de, norbert.strumpf@deutschebahn.com,
oliver.f.nandico@capgemini.com, johannes.willkomm@msg-systems.com,
alfred.zimmermann@reutlingen-university.de

Abstract: The SOA Innovation Lab has investigated the use of standard software packages in a service-oriented context. As a result, a method for developing a service-oriented enterprise architecture with custom and standard software has been obtained. It starts on enterprise level with the identification of domains where both the SOA paradigm and standard software are of relevance. Here SOA capabilities of products from different vendors can be evaluated within a dedicated maturity framework. After pre-requisites and dependencies between distributed components are determined, a high-level architecture can be developed. Currently this approach is extended to address also cloud computing operations of standard software.

1 Introduction

The growing complexity of IT landscapes is a challenge for many companies. A large number of standard software packages, individual software solutions, legacy applications, and different infrastructure components lead to high cost and limited responsiveness to new business requirements. In areas where flexibility or agility is important, service-oriented architecture (SOA) is the current approach to organize and utilize distributed capabilities. Here, the use of well defined and independent services from standard software packages poses some challenges. With software-as-a-service with on-demand operation as part of the anticipated IT-landscape, this becomes even more complex.

Nowadays, ERP application suites often dominate the enterprise architecture application layer and the associated automation of business processes in a rather monolithic, proprietary way [Gr07]. Disadvantages of standard application platforms - whether used on-premise or on-demand - include potential difficulties when fitting to individual business processes. In addition, their limited agility after first customization provides obstacles for the adaption to changed business needs and flexible product or service extensions [Cs96].

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

When referring to software-as-a-service solutions, they typically lack of individual customization possibilities. In addition, they show the shortcomings of standard software packages: Limited scope and less performance as compared to their best of breed or custom developed counterparts. Additionally, the integration with other leading systems may be a serious issue and challenge for open system environments, which need to support end-to-end business processes.

Some of these limitations can be overcome with the help of service-oriented architecture. SOA is an IT architecture paradigm that utilizes services as fundamental, flexible, and interoperable building blocks for both, structuring the business and for developing applications. SOA is a business oriented architecture style, often based on best of breed technology for agnostic business services, delivered by applications in a business-focused granularity [Gr07]. An introduction into fundamental SOA concepts, technologies, and case studies can be found e.g. in [Er05, KBS05].

After a brief introduction into cloud terminology and basic challenges, we summarize our approach for the design of a service-oriented enterprise architecture, starting from the domain map of an enterprise [Bu+10a]. We then outline current investigations, which address the extension of our approach to cloud computing offerings.

2 Our View on Cloud Computing

Following the definition of NIST [MG09] we distinguish between different types of cloud computing:

- **Software as a Service** (SaaS) describes applications which offer automated business functions over the internet. Salesforce.com, myERP.com, SAP Business ByDesign, Oracle OnDemand and RightNow are examples for existing SaaS offers.
- **Platform as a Service** (PaaS) describes applications which offer technical platforms for developing web-applications and –services. They may also provide infrastructure for supporting the whole software life cycle.
- **Infrastructure as a Service** (IaaS) stands for the provisioning of computer infrastructure as a service. Customers can buy servers, operating software, storage and network equipment as a service.
- **Business Process as a Service** (BSaaS) extends SaaS by providing complex automated functions that might cover an organization as a whole.

According to the BITKOM/Deloitte survey [BD11], there are six major goals for companies, when using cloud computing:

- Higher flexibility
- Cost reduction
- Risk reduction - concerning large investments
- Dependency reduction -to the IT department
- Faster distribution of functionality
- Extension of functionality

ISBN 978-3-88579-286-4

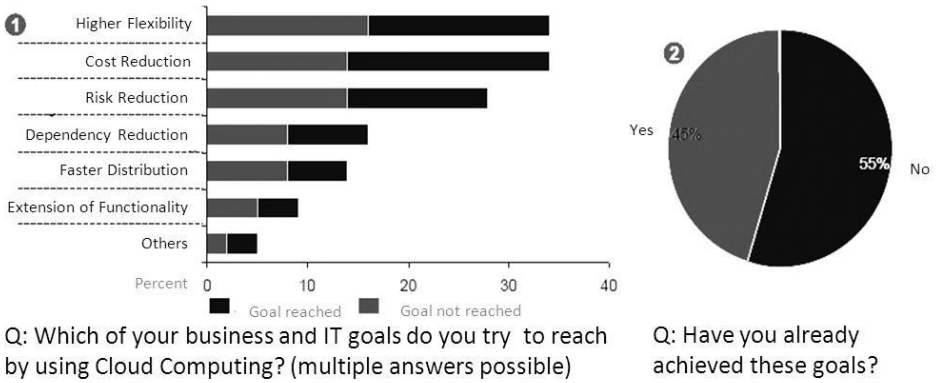

Figure 1. Business and IT goals when using services of the cloud.Picture from [BD11].

Cloud computing has specific characteristics that require additional criteria to be applied when developing an enterprise wide service-oriented architecture, since it is a melting pot of internet technologies, virtualization strategies and standardization strategies. Although in principle, service-oriented architecture supports the integration of cloud computing components.

3 Our Approach: Definition and Weighting of Domain

When developing enterprise architecture under the SOA paradigm, a domain map is the central artefact, which structures and organizes the needed capabilities. It uses the principles of information hiding, separation of concerns, and tight/lose coupling. Since one major goal of SOA is the increase of reusability and flexibility by respecting these principles in business and IT, domain maps provide the corresponding topographic base. Methods for the identification of domain maps are described in several publications, e.g. [En+08]. However, more work needs to be done, if an architect wants to identify areas, where SOA and the use of service from software packages, may they be cloud or on-premise solutions, make sense. Here we have introduced the following steps for refinement [Bu+10a]:

- **Define degree of differentiation and sharing** on domain level
- **Refine domains** by applying the following patterns
    - **Modularization**: Here one domain is split into two. This pattern is often applied to domains where the need for differentiation cannot be clearly marked.
    - **Generalization**: This pattern is applied to domains that have some, but not all capabilities in common. The common capabilities are generalized and form a new, centralized domain. They now can be used in a shared way.
    - **Aggregation**: This pattern is usually applied to a number of domains that share their low need for differentiation. Here two or more domains are merged. In this case the architect reduces potentially unnecessary complications due to needless loose coupling.
- **Finalization** and buy in from all stakeholders

Once an architect has defined a domain map he will need to characterize the domain map and provide information on a more fine-grained level. The goal is to develop a to-be architecture, which marks clearly those areas where the usage of standard software or software-as-a-service packages makes sense. In order to achieve this, the architect needs to classify each domain according to its business needs. Figure 2 gives an example for this categorization for a typical domain for sales (for details see [Bu+10a]).

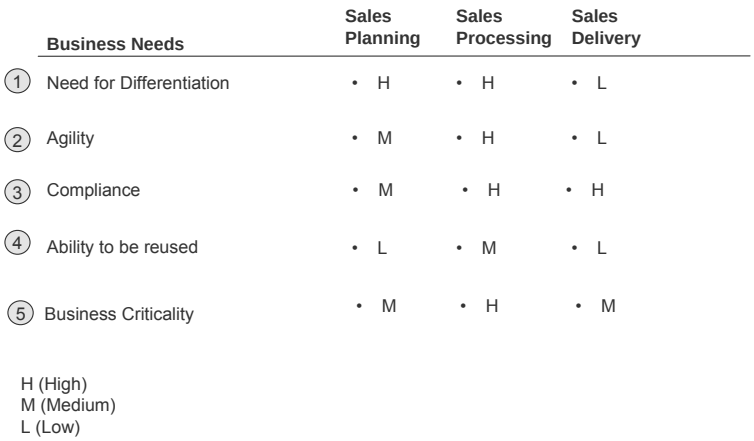

Figure 2. Illustrative example for categorizing sub-domains according to their business needs [Bu+10a].

For an appropriate classification domains have to be iteratively refined and classified until agility and differentiation is of low relevance, or the domains represent capabilities that correspond to exactly one role and one goal. Automating isolated fine-granular services with standard software or cloud solutions usually will not bring additional benefits, because dealing with built-in dependencies of software packages is more expensive than automating these services in a custom-built way.

After the architect has iterated through these steps, he will have to identify concrete software packages for domains, for which agility and differentiation are of low relevance. On this basis the architect will develop a to-be architecture, in which fine-grained services on function or data level enhance the software packages with required functions and where software packages offer services that can be combined to higher-level processes.

4 EAM Aspects on Cloud Computing

So far our discussion has focused on the use of standard software packages in heterogeneous IT-Landscapes, considering mainly solutions which are operated on premise. Most of the arguments from [Bu+10a] - which we have summarized above - are still valid, when cloud computing solutions are used in selected domains.

The following illustration gives an overview of the different and most important aspects that have to be addressed when using services of the cloud.

ISBN 978-3-88579-286-4

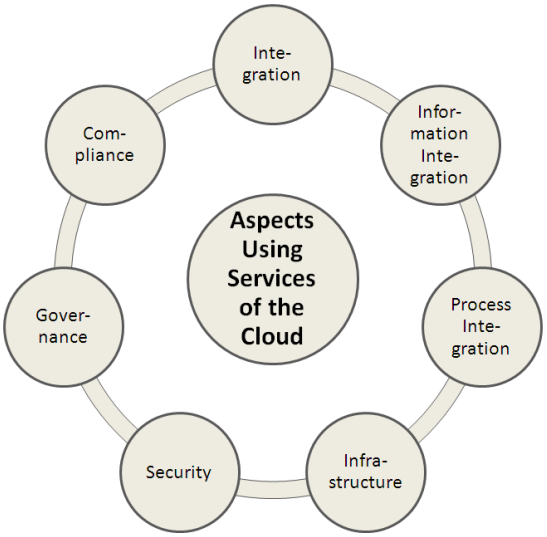

Figure 3. Most important aspects when using services of the cloud.

Although, cloud computing aims at encapsulation these aspects, there are still many questions to be answered, e.g.:

- On demand self service characteristics raise questions how standard platform provider support the provisioning process and how to keep control over the possibly adapted system environment.
- Cloud computing requires large network bandwidth, which calls for a respective provider offering - in particular if we look into mission critical applications.
- As cloud computing claims to go beyond a mere virtualization, we have to take different success factors into consideration.
- Different domains require a different degree of elasticity which has to match the offering of standard software and cloud providers.
- As services are measured by consumption, license models for standard software have to be adapted - e.g. allowing usage based billing.

These aspects of cloud computing are focus of a current research project of the SOA Innovation Lab. The corresponding topics are analyzed by combining methods for enterprise architecture management and service-oriented architecture with properties of cloud solutions. Results are then validated by matching them against different use cases from partner companies of the SOA Innovation Lab.
As emphasized in Figure 3, an investigation of cloud computing covers many dimensions. The SOA Innovation Lab has started an explorative study on several of them. In particular we focus on the analysis of cloud capabilities for standard software. Our goal is, to find out, in which circumstances it makes sense to use standard cloud components in a heterogeneous IT environment. For this purpose we have developed a questionnaire for software vendors and cloud service providers.
To ensure a systematic approach, we have based the areas of our assessment on an extension of the architecture maturity model SOAMMI [Bu+10b]. The top level

ISBN 978-3-88579-286-4

structure of SOAMMI is organized by five architecture domains adapted from TOGAF [To09]: Architecture Strategy and Management, Business Architecture, Information Architecture, Application Architecture, Technology Architecture, Service & Operation Architecture, Architecture Realization. For an assessment of service management processes in the cloud we include elements of ITIL [Ta+07]. Typical areas of investigation address key benefits of cloud services as discussed e.g. in [BD11]. In particular we distinguish questions on: on demand services, broad network access, resource pooling, rapid elasticity and measured services. Furthermore, we differentiate between IaaS, PaaS and SaaS services models, as well as private, public, hybrid and community cloud scenarios.

As a start we have explored the current opinion of partner companies of the SOA Innovation Lab. First findings – which sketch the spectrum of our investigation – are:

- **Differentiation**: Even more than for on-premise standard software packages, software-as-a-service solutions shall only be used in domains where the need for differentiation is minimal. The main reason for this is that individual extensions are much harder to achieve and operate than in on-premise situations.
- **Sharing**: For business services with a high degree of standardization, software-as-a-service solutions can be used to achieve high degree of efficiency. Furthermore, costs of shared services can be made transparent and allocated appropriately to business units.
- **Integration**: Integration is currently a serious challenge for software-as-a-service solutions. For end-to-end business processes, IT support through mixed solutions - with on-premise and on-demand components - is not favored for complex integration processes. Examples are scenarios where process control is handed over from on-premise-components to software-as-a-service solutions and back, or processes which require complex read and write transactions between software packages running in different operation modes.

Literature

[BD11] BITKOM, Deloitte: Cloud Computing in Deutschland – Ergebnisse einer Umfrage von BITKOM und Deloitte (2011).

[Bu+10a] H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, A. Zimmermann: Method for Service-Oriented EAM with Standard Platforms in Heterogeneous IT Landscapes. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010), Paderborn 2010, LNI, P-160 (2010)

[Bu+10b]H. Buckow, H.-J. Groß, G. Piller, K. Prott, J. Willkomm, A. Zimmermann: Analyzing the SOA ability of Standard Software Packages with a dedicated Architecture Maturity Framework, In: Entwicklungsmethoden für Informationssysteme und deren Anwendung, Karlsruhe (EMISA2010), LNI, P-172, (2010)

[Cs96] CSC: Standardsoftware und geschäftliche Flexibilität. Foundation Bericht 107 (1996)

[En+08] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J. Richter, M. Voß, J. Willkomm: Quasar Enterprise. dpunkt.verlag (2008)

[Er05] T. Erl: Service Oriented Architecture. Prentice Hall (2005)

[Gr07] A. Grigoriu: An Enterprise Architecture Development Framework. Trafford Publishing (2007)

[KBS05] D. Krafzig, K. Banke, D. Slama: Enterprise SOA. Prentice Hall (2005)

[MG09] P. Mell,T. Grance: NIST Definition of Cloud Computing. Version 15, 10-7-09, National Institute of Standards and Technology, Information Technology Laboratory (2009)

[Ta+07] S. Taylor (ed.): ITIL Livecycle Suite. The Stationery Office Ltd. (2007)

[To09] TOGAF Version 9. The Open Group Architecture Framework (2009)

ISBN 978-3-88579-286-4