Summer 9-4-2014

A SWOT ANALYSIS OF THE SOFT-COMPUTING PARADIGMS SOA/SOC/CLOUD COMBINATION (C-SOA) IN SOFTWARE DEVELOPMENT

Natalia, Kryvinska
University of Vienna, Vienna, natalia.kryvinska@univie.ac.at

Christine Bauer
Institute for Management Information Systems, Vienna University of Economics and Business, Vienna, Austria., chris.bauer@wu.ac.at

Christine Strauss,
University of Vienna, Vienna, christine.strauss@univie.ac.at

Michal, Gregus,
University in Bratislava, Bratislava, Michal.Gregus@fm.uniba.sk

A SWOT ANALYSIS OF THE SOFT-COMPUTING PARADIGMS SOA/SOC/CLOUD COMBINATION (C-SOA) IN SOFTWARE DEVELOPMENT

Complete Research

Kryvinska, Natalia, University of Vienna, Vienna, Austria, natalia.kryvinska@univie.ac.at

Bauer, Christine, Vienna University of Economics and Business, Vienna, Austria, chris.bauer@wu.ac.at

Strauss, Christine, University of Vienna, Vienna, Austria, christine.strauss@univie.ac.at

Gregus, Michal, Comenius University in Bratislava, Bratislava, Slovakia, Michal.Gregus@fm.uniba.sk

Abstract

Due to their technological complexity, traditional software development paradigms are not appropriate to face the challenges in the modern Web 2.0 world. Having the ability to adapt rapidly to the fast changing Web in an open environment, and challenged by the need for applications to be flexible, Service-Oriented Architecture (SOA), Service-Oriented Computing (SOC), and more recently Cloud Computing are becoming more and more popular. SOC/SOA and Cloud Computing share many drivers, such as enterprise portfolio and cost reduction. Both approaches are complementary and are expected to become the core of IT-based projects and/or businesses. Thus, this paper discusses the opportunities and challenges of Soft-Computing Paradigms, which are evaluated based on a SWOT analysis. For illustration, we also present a case of SOC/Cloud (C-SOA) based on the real-world application of Amazon Services.

Keywords: Cloud Computing, Service-Oriented Architecture (SOA), Service-Oriented Computing (SOC), Strengths Weaknesses Opportunities Threats (SWOT), Web Services, C-SOA.

1 Introduction

The shift from traditional to rather novel soft-computing paradigms in recent years has triggered extended discussions among scholars as well as industrial experts (Kryvinska et al., 2011b). Gellersen and Gaedke (1999) claimed that traditional software development approaches typically have four phases: analysis, design, implementation and maintenance/evolution. Since the introduction of the Internet, experts were looking for some kind of structured approach for software development and maintenance for and on the Internet. "The Web is defined as an information medium rather than an application platform" (Huang and Mak, 2003, p. 41) and therefore it is frequently subject to changes. The paradigm of Object-Oriented Computing (OOP) came up in the 1980's, as a result of this problem. The major advantage of OOP is the sharing and reusing of resources. Rine and Bhargava (1992) claimed that it was the best paradigm available, as software could be maintained longer. OOP enabled to alter and align Web applications faster and better on required changes than older paradigms such as structured programming. However, OOP is rather considered evolutionary than revolutionary, similar to other paradigms before and after OOP (Rine and Bhargava, 1992). This is the main reason why it did not emerge any overall state-of-the-art approach for software development on the Internet; and the development relies heavily on the individual skills of the developers and their best practices (Huang and Mak 2003).

Due to the fact that software engineering has "inherently no silver bullet" (Brooks, 1987, p. 2) in the evolution of technology and business trends, the Service-Oriented Architecture (SOA) and Service-Oriented Computing (SOC) evolved around 2005 (Huhns and Singh, 2005). SOC is a design paradigm to build a composition (in form of computer software) out of independent distributed services in SOA, which consists of three parts: a provider of applications, a consumer, and a registry. In SOA, (Web) services represent business functionalities, principles, concepts, and applications (Auer et al., 2011; Kryvinska et al., 2008). These are built as individual software components, which can be reused for different purposes among different applications. As a consequence, talking about SOC requires also talking about SOA (Huhns and Singh, 2005). Cloud computing, which has become popular in the last few years, shares many of the drivers as in SOC/SOA, such as enterprise portfolio and cost reduction (Mladenow et al., 2012). Raines (2009) states that SOC/SOA and cloud computing are complementary approaches, which bear benefits but also risks. The application fields are manifold and include, for instance, contextual digital signage (Bauer et al., 2011), damage prevention (Strauss et al., 2009), or tracking of high-value items such as arts (Bauer et al., 2013).

Thus, this paper is structured as follows: First we discuss related work on the soft-computing paradigms SOC/SOA and cloud computing. Then we evaluate these approaches based on a SWOT analysis. For an illustration, we then discuss these approaches with an exploration of a real-world example.

2 The Definitions

2.1 SOA Basics

As it is defined in Huhns and Singh (2005), the architecture of service-based applications contains three main parts: provider, registry/broker, and requestor. These parts interact using publish, find, and bind operations. The service provider is an organization that provides access to a Web service and publishes the service description in a service registry run by a broker. The requestor finds the service description in a service registry and uses the information in the description to bind to a service. In a Service-Oriented Architecture (SOA), the service registry provides a centralized location for storing service descriptions with following key standards: Find - UDDI (Universal Description Discovery and Integration) (Erenkrantz, 2004); Bind - SOAP (Simple Object Access Protocol), simple XML-based protocol to let applications exchange information over HTTP to access a Web services (Erenkrantz, 2004); Publish - WSDL (Web Services Description Language), document written in a XML, specifies the location of the service and the operations the service offers (Wei and Blake, 2010).

2.2 Service-Oriented Computing – the SOA Follower

"Service-Oriented Computing (SOC) is an emerging computing paradigm whose main goal is to support the development of distributed applications in heterogeneous environments" (Crasso et al., 2011). Deploying SOC enables building software systems by assembling together distributed functionalities.

This new paradigm is rooted in object-oriented and component-based software development. Its primary goal is to enable developers to build networks of interoperable and collaborative applications, regardless of these applications' programming languages and the platforms they run on. This goal is achieved through the use of independent computational units, which are called "services" (Fantechi et al., 2012).

In addition, SOC brings in software qualities, which are of major importance. With its planned reuse approach, SOC furthers rapid, high quality development of software applications. By using existing, already tested software elements, the time needed to build an application is reduced and its overall quality is improved (Chollet et al., 2012; Arbab, 2012).

SOC uses SOA services to develop applications in the fast changing world of the Internet. These services are independent entities and can be "used in a platform independent way" (Papazoglou et al., 2006). The requester can compound them, as he or she wants, as long as these services follow certain standards. These standards (e.g., UDDI) have been developed by major computing companies (such as BEA, IBM, Microsoft, Amazon, etc.), which have been moved towards the paradigm of SOC and SOA (Tsai et al., 2006).

2.3 Cloud Computing to Prompt

Cloud computing is booming in recent years, and the IT industry shifts more and more to the cloud, although cloud computing is not a new technology. Rather it gained acceptance in recent years; but actually the cloud-computing paradigm was first mentioned in 1997 (Chellapa, 1997).

Cloud computing is the distribution of applications and/or hardware provided as services over the Internet and in data centers (e.g., servers) (Armbrust et al., 2010; Erdogmus, 2009), and is often referred to as "the Cloud". Cloud Computing exists in three forms (levels of services): Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) (NIST-2011). Definitions in literature are heterogeneous and the separation lines between the levels of services are drawn differently. Dillon et al (2010) define these three services as follows:

- *SaaS (Software as a Service)* - Software that is offered over the Internet, available to the end consumer as and when wanted;
- *IaaS* (Infrastructure as a Service) - Organization outsources the equipment for operations, including storage, hardware, servers and networking;
- *PaaS (Platform as a Service)* - Organizations provide infrastructure on which software developers can build new applications or extend existing ones.

A federal cloud is defined "as the deployment and management of multiple external and internal cloud computing services to match business needs. A federation is the union of several smaller parts that perform a common action" (Rouse, 2011). As can be seen from Figure 1, hybrid architecture can be used additionally to the federal cloud. Thereby, a service provider may be in a cloud or not, and a SOA/SOC may involve any combination of clouds and non-clouds (Barry and Dick, 2013).

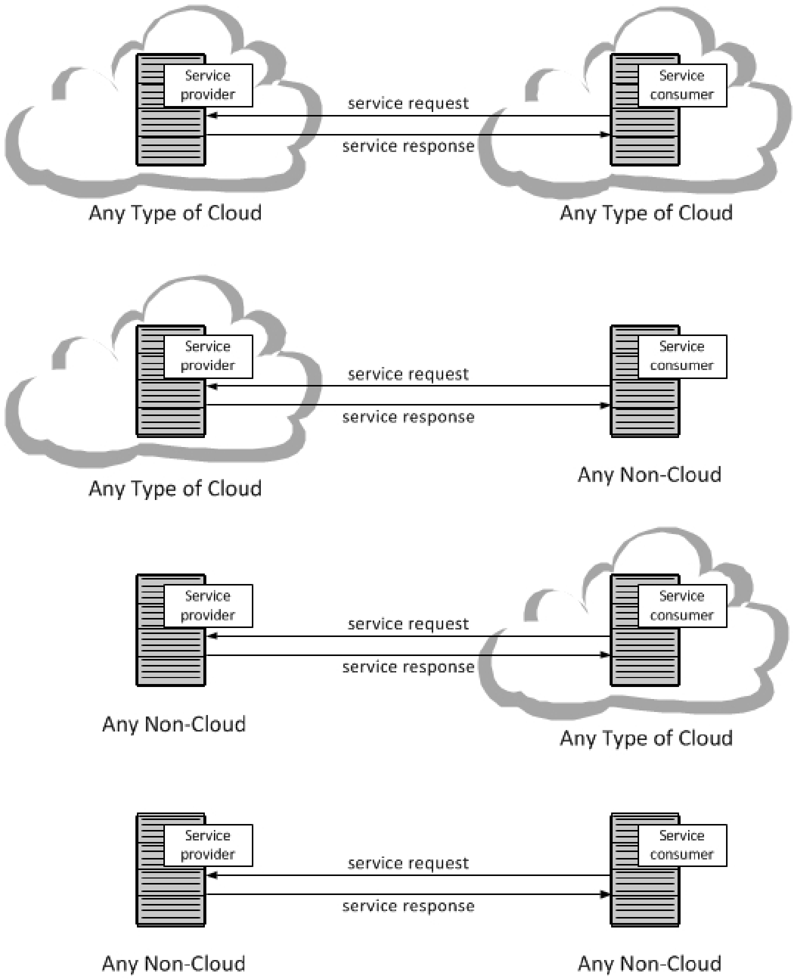

Figure 1. SOA, SOC and Cloud working together (source Barry and Dick, 2013).

2.4 The Relationships between Cloud Computing and SOC/SOA

Cloud Computing and SOC/SOA are independent approaches; still, they may be used as complementary activities (Figure 2). Cloud Computing is a broad term for any Web service, which offers the entire "traditional IT stack" (Raines, 2009), such as software, hardware, and applications. SOC/SOA, instead, focuses mainly on software services (Raines, 2009; Kryvinska et al., 2011a). In order to be combinable and benefit from each other, both approaches need to follow the Web service standards (i.e., UDDI, WSDL, SOAP).

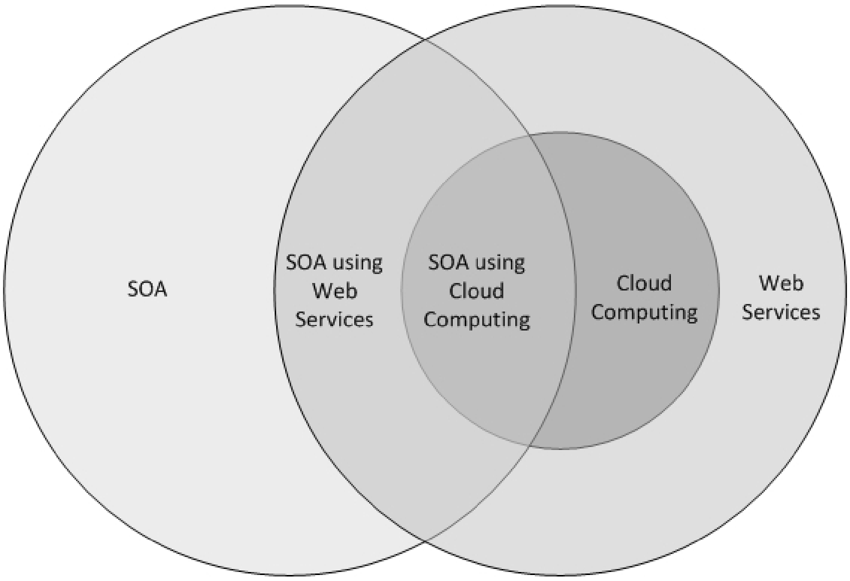

Figure 2. Venn diagram of relationships between Web Services, SOA, and Cloud Computing (source Barry and Dick, 2013).

3 The SWOT Analysis

The idea behind a SOC and Cloud Computing combination (which is referred to as C-SOA) is to make business easier and offer outsourcing solutions for enterprises (Figure 3). However, using SOC within "the cloud" environment does not only bring benefits, but also challenges arise. Based on a SWOT analysis, we evaluate this combination. The SWOT analysis is a strategic planning method (cf. Hill and Westbrook, 1997; Weihrich, 1982). Thereby, SWOT is an acronym representing strength, weaknesses, opportunities, and threats, which characterize the dimensions along with the entities or situations to be analyzed. Based on the analysis for each dimension, respective strategies are derived. Compared to other strategic planning methods, a particular strength of the SWOT analysis is that it considers both internal (strengths, weaknesses) as well as external (opportunities, threats) dimensions (Weihrich, 1982).

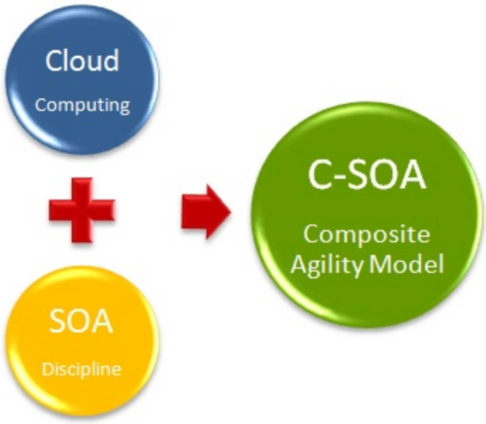

Figure 3. Guarding against Cloud silos with C-SOA (source Shaheen, 2012).

While strategic planning has a long history in military, predominantly product-oriented companies have extensively used the SWOT analysis (Weihrich, 1982). Only later this analysis' value was recognized for service-oriented businesses and ideas (e.g., Bernroider, 2002; Andrews, 2009).

The SWOT analysis is summarized in Table 4; details can be found in the following subsections.

Strengths	**Weaknesses**
• integration over time • SOA standards available (i.e., UDDI, SOAP, WSDL) • rapid services composing and orchestration • services re-use • managing services heterogeneity • rapid elasticity • high scalability • fault tolerance • distributed functionality • on-demand self-service • resources pooling • managing complex software intensive systems • rapid resources reconfiguration	• lack of standardization • lack of interoperability • lack of portability • communication of software components of several cloud service providers insufficient • additional workload • more complicated workflow • "dynamically accessible" versus "dynamically discoverable" • high availability required • no cloud is 100% reliable • changing providers may imply redevelopment of solutions • high dependence on provider • the service level has not yet evolved
Opportunities	**Threats**
• potential of standards, agreements, and rules • federal clouds • intermediary layer • rapid service deployment may increase interoperability • introduction of a Q-Cloud Service • integration of standards of ontologies in Web services • dynamic/rapid building complex system-of-systems on demand	• availability of cloud not perfectly 24/7 • constant merging and relocating of companies hampers seamless interoperation of web services • trust issues • messages could be intercepted or inferred by the competition or anybody who can use this information • security isolation

Table 4. The SWOT Analysis.

3.1 Strengths

SOC/SOA and Cloud computing may be used in combination as complements or in separation as independent solutions. This is a strength as they have not to be introduced at the same time but may be integrated over time (Wei and Blake, 2010).

There are already *Open API* platforms (e.g., the SUN Open Cloud Platform) that support federal clouds, which define key resources (via HTTP and JSON) like Cloud, Virtual Data Center, Cluster, Virtual Machine, etc., especially when they cooperate with other projects like Eucalyptus (Dillon et al., 2010).

There are already some standards available from SOA (i.e., UDDI, SOAP, WSDL) that C-SOA may build on. The Eucalyptus Project, for example, follows the approach of UDDI in the SOC/SOA paradigm, by starting the partnership with Amazon (Dillon et al., 2010).

Due to the re-use approach, rapid service composing and resources reconfiguration is possible. C-SOA allows for rapid elasticity and high scalability as required by a service consumer, and profits from resources pooling, due to the cloud-computing ingredient. Further strengths are the integration and management of a heterogeneous and complex service landscape.

Overall, we can summarize that C-SOA combines the strengths of SOA and cloud computing, which are widely discussed in scientific literature and do not need further elaboration at this place.

3.2 Weaknesses

The management of cloud services, which is inherent in C-SOA, implies an additional workload (Tsai et al., 2010) and also concerning the infrastructure, there are certain limitations, which may interfere or complicate the workflow (Tsai et al., 2010).

Currently, Web services are rather "dynamical accessible" than "dynamical discoverable" with the current standards (e.g., WSDL or SOAP). Problems are identified in the detailed information of semantics, which are provided by individuals. For example if someone looks for tax preparation software in the US, and types in a code like "xx.xx.xx.xx.xx", the right information for offered tax service software could be parsed out or not, depending on the provided semantics (Dogac et al., 2002). Accordingly, while for SOA some standards already exist (i.e., UDDI, SOAP, WSDL), there is a lack of standardization for C-SOA (Wei and Blake, 2010) (Dillon et al., 2010). This weakness also goes in hand with the lack of well-defined semantics in XML and the movement of web services beyond their key standards (UDDI, WSDL and SOAP) (Yu et al., 2008).

As a result of the lack of standardization, the interoperability of the cloud services maintains a problem. Dillon et al. (2010), for instance, argues that SaaS providers use different application domains like ERP, CRM, etc. and experts at KDD09 panel in 2009 (Zeller et al., 2009) raised the issue that data mining, which is also a SaaS, requires standardization to operate properly.

The lack of well-running interoperations and service deployment also cause a lack of portability, as SOC depends on these (Wei and Blake, 2010).

Due to the lack of standardization between service providers, also the communication of software components of several cloud service providers is currently insufficient (Wei and Blake, 2010). Accordingly, there is a need of cloud service providers to unify (Wei and Blake, 2010).

Furthermore, SOA systems frequently require a high availability. However, loud computing is still struggling with server outages, which make the combination of the two challenging. The recent example of Amazon's EC2 hosting service has shown that not only the cloud provider itself suffers under availability problems, but also companies, which rely on the respective cloud services (Hesseldahl, 2012). In addition, from time to time, server updates of the cloud provider require a shutdown. In short, no cloud is 100% reliable as they suffer from human failures and technical problems (Dillon et al., 2010).

Additionally, cloud solutions are not always designed for cross-cloud usage (Wei and Blake, 2010). The problem is and if an organization changes, it is not unlikely that – the despite a lot of time was spent on developing applications in the cloud –the solution has to be redeveloped at the new cloud provider (Tsai et al., 2010).

As a consequence, to lower the additional costs when moving to another, provider cloud user is often stuck with one provider because the migration from one cloud to another is hard to undertake (Tsai et al., 2010). Accordingly there is strong dependence on a single cloud provider, as only a switch to other solutions of the very same cloud provider seems feasible (Tsai et al., 2010). And even the connectivity between cloud services of the same provider still have to be optimized, which requires novel approaches as current approaches do not suffice (Wei and Blake, 2010).

Furthermore, although there is a possibility for organizations to outsource to different cloud service providers, this would, however, cause additional complexity (Dillon et al., 2010) and, thus, would limit the benefits of the C-SOA approach.

Another obstacle is the service level agreement (SLA), as the service level has not yet evolved as far as it needs to be (for customers). Especially if a business is subject to constant changes, this causes problems, as SLA supports tend to be static (Tsai et al., 2010).

3.3 Opportunities

The various approaches and models are suggested by scientific literature point to more or less the same opportunity: The potential of standards, agreements, and rules.

Service discovery could, for instance, be realized through federal clouds (Wei and Blake, 2010). In addition, scientific literature suggests also several opportunities with respect to federal clouds:

Integrating an intermediary layer between cloud consumers to manage the various interactions of APIs with providers is one opportunity (Dillon et al., 2010). A prominent example is OpenNebula, which is an open-source project. Its mission is to provide a solution for managing and organizing data centers and IaaS clouds (OpenNebula.org, 2012). Another solution for a layer is to integrate high-level application requirements for low-level cloud resources (Sun et al., 2012). Consumer needs are used to specify high-level application requirements. Then they get translated into high-level infrastructures, which can be translated onto low-level descriptions of cloud resources.

Another opportunity lies in the standardization between cloud providers. To gain positive impact out of federal clouds, companies may unify and standardize (Dillon et al., 2010). The Eucalyptus Project is a positive example (https://www.eucalyptus.com/); it follows the approach of UDDI in the SOC/SOA paradigm, by starting the partnership with Amazon. This kind of approach will be driven forward as more partnerships with the “big players” in the business emerge (Dillon et al., 2010).

Furthermore, Wei and Blake (2010) suggest that the cloud community could learn a lesson from UDDI and the data base community. These promising examples might entice the cloud community to support federal clouds and benefit from the opportunities, which lie beyond them (Wei and Blake, 2010).

A further opportunity lies in a rapid service deployment; in fact, cloud computing is anticipated to increase their interoperability by adding new features (Wei and Blake, 2010). Additionally, as more and more public and private organizations penetrate into the cloud, this could result into a massive increase in cross-cloud deployment (Wei and Blake, 2010).

Nathuji et al. (2010) strongly suggests the introduction of a Q-Cloud Service (a cloud backup service), which “is designed to provide assurances that the performance experienced by applications is independent of whether it is consolidated with other workloads” (Nathuji et al., 2010). Q-Clouds ease performance interference by 31% and the system utilization is improved by 35% (Nathuji et al., 2010). Improvements in the cloud community like the Q-Cloud Service support the Service-Oriented Computing in a Cloud Computing environment, to benefit from synergy effects between them. This synergy can help to build a powerful architecture for the enterprise agility (Figure 5).

The integration of standards of ontologies in Web services could improve services and start “the way for automated composition and seamless interoperation” (Yu et al., 2008). Wei and Blake (2010) claim that cloud services may support this by deriving ontological information from the information (in form of data), which is stored in the cloud. Additionally, an agent-mediated ontology generation from co-located information makes it possible that Web services have computer readable languages and work like the metadata schema – with better descriptive capacity (Yu et al., 2008).

An opportunity to increase the security level may be achieved with individual or community solutions.

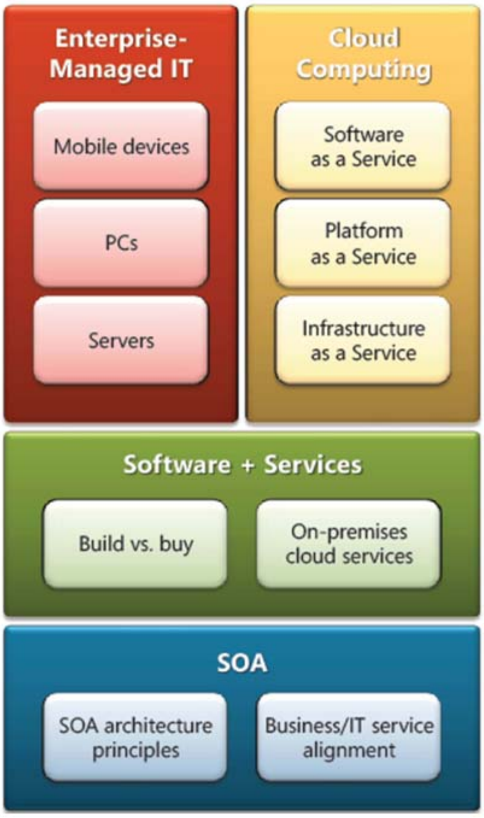

Figure 5. SOA, SOC and Cloud for Enterprise Agility (source MSDN, 2013).

For example, using Amazon's EC2 requires the consumer to fill out forms and list addresses, which will be checked by Amazon and only authorized lists are in the cloud. Furthermore Amazon collaborates with Spamhaus, an anti-spam organization, to gain a high security level (Chen et al., 2010) or Alliances have been built (Cloud Security Alliance Guidance, 2011).

3.4 Threats

As availability of the cloud is limited, this challenge may hinder a widespread adoption of cloud computing (Wei and Blake, 2010).

Furthermore, due to the constant merging and relocating of companies in the IT industry, seamless interoperation of web services is a key challenge (Yu et al., 2008), which tends to be difficult to overcome.

In addition, cloud consumers still have trust issues (reasonable or not) and providers risk that they may not acquire sufficient clients.

While the trust issues for cloud environment predominantly focus on data privacy, SOC security issues are based on messaging; messages could be intercepted or inferred by the competition or anybody who can use this information (Wei and Blake, 2010). The security challenge is generally a very crucial topic for IT. This makes it also very important for the two paradigms of cloud computing and SOC and might be the biggest threat, as malware, SPAM, or spy-software are improving constantly as well.

Moreover, Tsai et al. (2010) point to the phenomenon of "security isolation". Even if a provider may provide a safe cloud, this cloud may also be isolated from others, as there may be conflicts with other providers and interferences.

The “lose coupling” of services in SOC to a workflow, often contains more than one service provider. The issue is that most operational services are from one firm. This creates high dependence or the service providers need to manage the applications (build out of these services) on their own to keep the workflow constant (Wei and Blake, 2010).

In addition, as Tsai et al. (2011) point out, “… to provide valid and stable services for cloud users” is a significant threat for providers’ business success.

4 SOC and Cloud Combination Amazon Services

This section explores an example of SOC/SOA/Cloud usage by Amazon. We look at different Amazon Web Services (AWS) of Amazon (SaaS, PaaS and IaaS) and its partner network (APN). Amazon is seen as a big player in Web services and sets new standards with its network (Amazon Web Service, 2013). The big amount of information from Amazon and the problems to gain practical example as a member of the Amazon Partner Network (APN) suggests further research. Thus, this section demonstrates the benefits of the combination of SOC in a cloud-computing environment on the example of AWS.

4.1 Amazon Partner Network – Need for Standardization

Besides the vast offer of Amazon Web services the Amazon Partner Network (APN) offers huge benefits. In this Network are currently over 600 different technology partners like Adobe, IBM, SAP, and many more. If companies like to join the APN they must submit an application with standardized input fields, where prospects provide information helping Amazon including them in their Network (AWS Partner Network, 2013). When accepted in the Network the partner is subjected to the standards of the APN. This supports the interoperability and the cross-cloud infrastructure as Amazon has a registry (Amazon S3) of APIs with standard operations (Amazon Simple Storage Service, 2006).

4.2 Up Scaling Workflow and SLA at AWS

Every independent software/service vendor (ISV) at AWS certifies himself to support its customers. Amazon guarantees that every software development works in the APN. These developments need to be submitted to Amazon otherwise it will not work in the network. This guarantee of Amazon, tackles the problem of the need of high longer-standing workflows. The software development does not need to be tested against multiple platform configurations by the provider (AWS: White Paper, 2010). Amazon will take care of this, as it promises with its service level agreement. This might fulfill the claim of Tsai et al. (2010) of a dynamic solution as the ISV develop their own and new business needs and submit it to Amazon that configures it. Of course this will take its time, but is a good approach to shrink the obstacle of the service level agreement and support the dynamic business development (Tsai et al., 2010). To fight the problem of redundancy with a new storage option called Reduced Redundancy Storage of the Amazon S3 registry, which stores data in multiple places to overcome the problem of a single server blackout. Data is saved on several servers to guarantee 99.99% availability (Amazon Simple Storage Services, 2006).

4.3 Security of Amazons Web Services

Amazon is labeled with different third party security certificates. For example the SOC 1 or SOC 2 reviews, according to SSAE 16 and ISAE 3402 standards. Furthermore, AWS received the ISO 27001 certificate and was confirmed for the data security standards DSS and PCI. Additionally, Amazon has experience in data security, thanks to years of experience in its own product selling, but if a customer or partner still has security issues he can encrypt its data or software additionally (AWS: Risk and Compliance, 2012).

5 Conclusions

"Today, many organizations strive to cope with rapid market changes, such as evolving customer requirements and new business processes" (Wei and Blake, 2010).

In this paper, we have deliberated the prospects and encounters of Soft-Computing Paradigms and their interworking. In an SWOT analysis, we evaluated the current situation of a combination of SOA and cloud computing (C-SOA). Despite many strengths of this concept, the analysis revealed many challenges that are yet to be overcome. Still, the main opportunities to overcome these challenges seem too lie in the establishment of standards, agreements, and rules. As companies have to cope with rapid market changes, we are convinced that this situation will drive the establishment of standards.

For an illustration, we have also presented a case of C-SOA based on the real-world application of Amazon Services. We could also show that Amazon Web Services benefit from some opportunities of SOC in a cloud-computing environment as it has managed to integrate standards and structure. Still, note that this paper does not claim that the example of Amazon is the only or best solution for the future; rather we hope that this illustration contributes to eliminating a lot of the concerns of this concept.

Future work should emphasize on the current weaknesses of C-SOA and opportunities to overcome these. Particular research and development has to be put into the establishment of standards among service providers.

References

Amazon Simple Storage Services (2006). Amazon S3. Amazon Simple Storage Services, http://docs.amazonwebservices.com/AmazonS3/latest/API/Welcome.html, Retrieved on 2014-03-21.

Amazon Web Services (2013). AWS Technology Partner. Amazon Web Services, https://aws.amazon.com/solution-providers/isv, Retrieved on 2014-03-21.

Andrews, C. (2009). SWOT: The Evolution Of IT Service Providers To Business Technology Competitors: The Business Technology Offerings Of Accenture. Capgemini, Cognizant, and IBM. Forrester Research, Cambridge, Massachusetts.

Arbab F. (2012). Will the Real Service Oriented Computing Please Stand Up? In Proceedings of the 7th international conference on Formal Aspects of Component Software (FACS 2010), Guimarães, Portugal, October 14-16, 2010, LNCS-6921, 2012, pp. 277-285.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of Cloud Computing. Communications of the ACM, 53(4), 50-58.

Auer, L., Kryvinska, N., Strauss, C. and Belov E. (2011). Software-based Business Applications/Tools to Assess Complex SOA Investments - The Cross-Vendor Comparative Analysis. In Proceedings of 5th International Workshop on Engineering Complex Distributed Systems (ECDS-2011), in conjunction with CISIS-2011, June 30th-July 2nd, 2011, Seoul, Korea, pp. 397-401.

AWS Partner Network (2013). Become an APN Technology Partner. AWS Partner Network, https://aws.amazon.com/solution-providers/isv/new, Retrieved on 2014-03-21.

AWS Risk and Compliance (2012). Amazon Web Service Risk and Compliance. AWS Risk and Compliance, http://media.amazonwebservices.com/AWS_Risk_and_Compliance_Whitepaper.pdf, Retrieved on 2014-02-20.

AWS White Paper (2010). Software as a Service on AWS. AWS White Paper, http://d36cz9buwru1tt.cloudfront.net/SaaS_whitepaper.pdf, Retrieved on 2014-03-19.

Barry, D. K. and Dick, D. (2013). Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide. 2nd Edition. Morgan Kaufmann Publishers, http://www.service-architecture.com/index.html, Retrieved on 2014-03-20.

Bauer, C., Dohmen, P., and Strauss, C. (2011). Interactive Digital Signage: an Innovative Service and its Future Strategies. . In Proceedings of the First International Workshop on Frontiers in Service Transformations and Innovations (FSTI-2011), in conjunction with EIDWT 2011, IEEE Computer Society Digital Library, 2011, 137-142.

Bauer, C., Strauss, C., Stummer, C., and Trieb, A. (2013). Context-aware services in cooperative value chains: a key player-centered approach. Scalable Computing: Practice and Experience, 14(3), 139-153.

Bernroider, E. (2002). Factors in SWOT Analysis Applied to Micro, Small-to-Medium, and Large Software Enterprises: an Austrian Study. European Management Journal, 20(5), 562-573.

Brooks, F. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. Computer, 20(4), 10-19.

Chellappa, R. (1997). Intermediaries in cloud-computing: A new computing paradigm. In Proceedings of INFORMS Intermediaries in Electronic Markets.

Chen, Y., Paxson, V., and Katz, R. H. (2010). What's new about cloud computing security? University of California, Berkeley Report No. UCB/EECS-2010-5 January.

Chollet, S., Lestideau, V., Maurel, Y., Gandrille, E., Lalanda P. and Raynaud, O. (2012). Practical Use of Formal Concept Analysis in Service-Oriented Computing. In Proceedings of the 10th International Conference on Formal Concept Analysis (ICFCA 2012), Leuven, Belgium, May 7-10, 2012, LNCS-7278, pp 61-76.

Cloud Security Alliance Guidance (2011). Version 3.0. 2011. https://cloudsecurityalliance.org/research/security-guidance/, Retrieved on 2014-02-05.

Crasso, M., Zunino, A. and Campo, M. (2011). Research Review - A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. Journal of Database Management, 22(1), 102-132.

Dillon, T., Wu, C., and Chang, E. (2010). Cloud Computing: Issues and Challenges. In Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 27-33.

Dogac, A., Laleci, G.B., and Cingli, Y. (2002). Exploiting web service semantics: Taxonomies vs. ontologies. IEEE Data Engineering Bullettin, 25(4), 10-16.

Erdogmus, H. (2009). Cloud computing: Does nirvana hide behind the nebula? Software, IEEE, 26(2), 4-6.

Erenkrantz, J. R. (2004). Web Services: SOAP, UDDI, and Semantic Web. Institute for Software Research, Report UCI-ISR-04-3.

Fantechi, A., Gnesi, S., Lapadula, A., Mazzanti, F., Pugliese, R., and Tiezzi, F. (2012). A Logical Verification Methodology for Service-Oriented Computing. ACM Transactions on Software Engineering and Methodology, 21(3):16.

Gellersen, H-W., and Gaedke, M. (1999). Object-Oriented Web Application Development. IEEE Internet Computing, 3(1), 60-68.

Hesseldahl, A. (2012). Amazon's Cloud is down again, taking heroku with it. http://allthingsd.com/20121022/amazons-cloud-is-down-again-taking-heroku-and-github-with-it/, Retrieved on 2014-02-05.

Hill, T. and Westbrook, R. (1997). SWOT analysis: It's time for a product recall. Long Range Planning, 30(1), 46-52.

Huang, G. Q. and Mak, K. L. (2003). Internet Applications in Product Design and Manufacturing. Heidelberg, Springer-Verlag.

Huhns, M.N., and Singh, M.P. (2005). Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing, 7(1), 75-81.

Kryvinska, N., Auer, L. and Strauss, C. (2008). Managing an Increased Service Heterogeneity in Converged Enterprise Infrastructure with SOA. International Journal of Web and Grid Services (IJWGS), 4(4), 440 – 460.

Kryvinska, N., Auer, L. and Strauss, C. (2011a). An Approach to Extract the Business Value from SOA Services. In Proceedings of Second International Conference on Exploring Services Science (IESS 1.1), 16-18 February, 2001, Geneva, Switzerland, Springer LNBIP-82, pp. 42–52.

Kryvinska, N., Strauss, C. and Auer, L. (2011b). Synergy of two Computational Intelligence Paradigms: SOA and Mobility for the Efficient Management of Resource-Intensive Applications on Constrained Devices. Chapter 14, book Ed.: N. Bessis and F. Xhafa, (SCI-352), Springer-Verlag Berlin Heidelberg, pp. 357-382.

MSDN - Microsoft Developer Network (2013) http://msdn.microsoft.com/en-us/architecture/aa699439.aspx, Retrieved on 2014-03-25.

Mladenow, A., Kryvinska, N. and Strauss, C. (2012). Towards Cloud-Centric Service Environments. The Society of Service Science, Journal of Service Science Research, Springer, 4(2), 213-234.

Nathuji, R., Kansal, A., and Ghaffarkhah, A. (2010). Q-clouds: managing performance interference effects for qos-aware clouds. In Proceedings of the 5th European conference on Computer systems (ACM), pp. 237-250.

National Institute of Standards and Technology US (2011). Definition of Cloud Computing, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, Retrieved on 2014-02-20.

OpenNebula.org (2012). About the OpenNebula.org Project, http://opennebula.org/about:about, Retrieved on 2014-02-05.

Papazoglou, M. P., Traverso, P., Dustdar, S., Leymann, F., and Krämer, B. J. (2006). Service-Oriented Computing: A research roadmap. http://drops.dagstuhl.de/opus/volltexte/2006/524/pdf/05462.SWM.Paper.524.pdf, Retrieved on 2014-02-07.

Raines G. (2009). Cloud Computing and SOA. http://www.mitre.org/work/ tech_papers/tech_papers_09/09_0743/09_0743.pdf, Retrieved on 2014-02-07.

Rine, D.C., and Bhargava, B. (1992). Guest Editor's Introduction: Object-Oriented Computing. Computer, 25(10), 6-10.

Rouse, M. (2011). Definition: Federated cloud. http://whatis.techtarget.com/definition/federated-cloud-cloud-federation, Retrieved on 2014-02-07.

Shaheen R. (2012) Guarding Against Cloud Silos With C-SOA. Cloud Industry Insights, July 24th, 2012, http://www.rackspace.com/blog/guarding-against-cloud-silos-with-c-soa/, Retrieved on 2014-04-02.

Strauss, C., Stummer, C., Bauer, C., and Trieb, A. (2009). A Networked Ubiquitous Computing Environment for Damage Prevention: A Decision Support Framework for the Insurance Sector. In Proceedings of 38th International Conference on Parallel Processing (ICPP-2009), International Workshop on Design, Optimization and Management of Heterogeneous Networked Systems (DOM-HetNetS'09), Vienna, Austria, September, 22-25 2009, 276-281.

Sun, Y., Harmer, T., and Stewart, A. (2012). Mapping High-Level Application Requirements onto Low-Level Cloud Resources. Journal of Software Engineering and Applications, 5(11), 894-902.

Tsai, W. T., Chen, Y., Bitter, G., and Miron, D. (2006). Introduction to Service-Oriented Computing, Report, Arizona State University.

Tsai, W. T., Sun, X. and Balasooriya, J. (2010). Service-Oriented Cloud Computing Architecture. In Proceedings of Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, 12-14 April 2010, pp.684-689.

Tsai. W.T., Zhong P., Balasooriya J., Yinong Chen, Xiaoying Bai, Elston J. (2011). An approach for service composition and testing for cloud computing. In Proceedings of Tenth International

Symposium on Autonomous Decentralized Systems (ISADS 2011), 23-27 March 2011, Tokyo & Hiroshima, pp. 631-636.

Wei, Y., and Blake, M. (2010) Service-oriented computing and cloud computing: Challenges and opportunities. IEEE Internet Computing, 14(6), 72-75.

Weihrich, H. (1982). The TOWS matrix: A tool for situational analysis. Long Range Planning, 15(2), 54-66.

Yu, Q., Liu, X., Bouguettaya, A., and Medjahed, B. (2008). Deploying and managing Web services: issues, solutions, and directions. The VLDB Journal, 17(3), 537-572.

Zeller, M., Grossman, R., Lingenfelder, C., Berthold, M., Marcade, E., Pechter, R., Hoskins, M., and Holada R. (2009). Open standards and cloud computing. KDD-2009 panel report, Paris, pp. 11-18.