

SOA: Trends and Directions

George Feuerlicht^{1,2}
Shyam Govardhan¹

¹Faculty of Engineering and Information Technology,
University of Technology, Sydney
Level 2, Building 10, 235 Jones Street,
Broadway 2007, Australia
email: jiri@it.uts.edu.au, shyamg@uts.edu.au

²Department of Information Technologies
Prague University of Economics
W. Churchill Sqr. 4
130 67 Prague 3, Czech Republic

Abstract: *While many organizations have adopted SOA there are recent indications that not all organizations are willing to make substantial investments in new skills and technologies required for the transition to SOA in the current economic climate. The recent emergence of Cloud Computing is continuing the trend of delivering enterprise applications and IT infrastructure in the form of externally sourced services, providing an alternative to on-premise solutions. The convergence of Cloud Computing and Web 2.0 is redefining the very basis on which the computer industry has operated for decades, challenging some of the basic SOA assumptions and principles. In this paper we discuss the synergies between the above technology trends and consider the likely impact of these trends on enterprise computing.*

Keywords: SOA, Cloud Computing, SaaS, IaaS, PaaS, Web 2.0

1. Introduction

As a result of an increasingly competitive business environment during the present economic downturn, organizations are looking for ways to minimize Information Technology (IT) costs and at the same time offer more sophisticated products and services. In many organizations, this effort is hindered by the current business structures and the supporting IT infrastructure. Typically computing resources are organized according to specific business functions they support, resulting in separate *application silos*. This situation leads to duplication of application functionality, limited ability to share information, high integration costs and significant underutilization of computing resources. Furthermore, such enterprise computing environments tend to be excessively complex and inflexible making it difficult for organizations to respond to changing business requirements in a timely manner.

Service-Oriented Architecture (SOA) has been widely promoted by analysts and IT vendors as the architecture capable of addressing the business needs of modern organizations in a cost-effective and timely manner. Perceived SOA benefits include improved flexibility and alignment between business processes and the supporting enterprise applications, lower integrations costs (in particular for legacy applications), and numerous other advantages. Although SOA can play an important role in inter-enterprise business-to-business (B2B) applications, SOA is primarily regarded as an intra-enterprise architecture used for internal integration [1].

SOA adoption was initially driven by the emergence of Web Services and related technologies and the need to provide a more effective enterprise computing architecture. Although the precise impact and level of adoption of SOA is difficult to ascertain (partly because of the widely differing definitions of SOA) it was estimated by Gartner Research in September 2008 that most European and North American companies have either already adopted SOA or are planning to adopt SOA over the next 12 months [2]. However, it is important to note that the same report has identified that some companies are not willing to make the substantial investment required for the transition to SOA in the current economic climate. Some experts argue that SOA has failed to deliver on its promised benefits and that most business people no longer believe in SOA [3]. Part of the problem is that SOA has become an *umbrella* term and a *silver bullet* solution for most IT problems, and this expectation is impossible to meet in practice.

So what is SOA and what is its relevance to enterprise computing today? While there is no widely accepted definition of SOA, most experts agree about the basic principles underlying the SOA paradigm [4, 5] that include:

- (i) Services are the basis SOA abstraction - business function are externalized as services via well defined and stable interfaces specified with a service contract
- (ii) Services are loosely coupled - services are designed to be mutually independent and orthogonal to their environment (i.e. the context they operate in)
- (iii) Services are autonomous - services have independent software development cycles, explicit (administrative) boundaries over which services operate (i.e. organization boundaries, and interfaces between service provider and service consumer)
- (iv) Stateless operation – services should be stateless to minimize resource consumption and to simplify failure recovery
- (v) Quality of service (QoS) characteristics that include security, transactions, performance, etc. are explicitly defined for each service

SOA is best described as a set of architectural concepts and principles that include systems development methods, techniques and related technologies that enable the implementation of service-oriented distributed enterprise applications. Although the concept of SOA has been evolving over time, it remains focused on intra-enterprise computing and does not fully address the challenges associated with consuming externally provided services. To be effective in today's enterprise computing environments, the scope of SOA needs to be extended to encompass different types of service models, including SaaS (Software as a Service), IaaS (Infrastructure as a Service), Platform as a Service (PaaS) and services that represent the Web 2.0 environment. These important service-oriented computing trends have been evolving in parallel with SOA, and have now reached a stage of maturity and adoption that make SaaS, IaaS, PaaS and Web 2.0 essential components of enterprise computing.

In this paper we identify and discuss two key trends that impact on the future direction for SOA: Cloud Computing (section 2) and Web 2.0 (section 3). In the concluding section (section 4) we argue that SOA must incorporate both Cloud Computing and Web 2.0 to enable organizations to take advantage of this new computing environment and remain relevant to enterprise computing in the future.

2. Cloud Computing

In the drive to reduce ICT costs during the present economic downturn organizations are actively investigating alternatives to on-premise ICT solutions. Cloud Computing is a relatively recent terminology that is used to describe software and infrastructure services delivered from a remote data centre and in general includes SaaS, IaaS and PaaS models [6]. Cloud Computing services are

delivered on-demand and rely on virtualization technologies to provide highly scalable IT services that can be *rented* by client organizations on a subscription basis. Cloud Computing can be regarded as a continuation of a trend towards network computing advocated by the proponents of the Java computing model in the 1990s [7]. In the following section we discuss the various types of service models that constitute the Cloud Computing environment.

2.1. Software-as-a-Service

The SaaS model has its origins in the Application Service Provider (ASP) model that emerged in the late 1990s as an alternative to the traditional, *on-premise* enterprise software implementation approach. While the idea of software services is not new, recent studies indicate a significant increase in the adoption of SaaS [8]. The adoption of SaaS was initially driven by the need to reduce the cost of software ownership, but there are indications that the benefits of SaaS go well beyond cost-effective software delivery, enabling organizations to transform their business processes [9]. From the point of view of the service provider, the SaaS business model generates ongoing revenues in the form of subscription fees from each client organization. Importantly, the providers benefit from economies of scale as the incremental costs associated with adding new SaaS users is relatively low. The customers on the other hand benefit from minimizing (or eliminating altogether) the up-front costs associated with hardware and software acquisition, and at the same time avoid having to hire and train business analysts and technical support staff. There are a number of possible SaaS alternatives; the most common is the ASP-style model with the service provider hosting services for a large number of clients using a scalable multi-tenant architecture owned and maintained by the service provider. However, it is also possible for the client to host the software *internally* in a data centre within the client organization, with the service provider maintaining the applications from a remote location. Equally, it is possible for the client organization to host their internally developed applications externally on the provider's hardware infrastructure. Another, more recent SaaS model involves the client organization *self-hosting* their own software and providing software services to their partners, or subsidiaries in remote locations. Various combinations of these models are also possible, including situations where the service provider and the client participate in the development and delivery of software services in the form of a joint venture, sharing the benefits and the risks. Delivering SaaS applications to a large heterogeneous user base requires a specialized application architecture that facilitates the customization and integration of applications. Multi-tenant architecture is widely used for SaaS applications in order to support large user populations in a scalable manner. Multi-tenant architecture relies on resource virtualization to enable sharing of physical and logical resources among different end-user organizations, while at the same time ensuring security and confidentiality of information. Using database virtualization, different organizations are allocated logical (virtual) database instances that coexist within one physical database. This significantly reduces the administration overheads associated with maintaining the database (i.e. database backup and recovery, etc.), and at the same time provides a secure environment for individual user organizations.

Examples of successful SaaS applications are Salesforce.com CRM applications [10] and Google Apps [11]. Google Apps includes messaging via Gmail and Google Calendar and collaboration services via Google Docs, Google Video and Google Sites. Google also recently announced the Google Secure Data Connector (SDC) product [12], which allows access to corporate data residing behind a firewall. SDC enables the creation of web applications that have access to both private as well as public data in a secure manner by allowing the validation and authorization of requests.

The combination of recent business and technological developments has created a situation where delivery of enterprise applications in the form of SaaS services becomes both technically possible, and economically compelling. The SaaS model provides a viable alternative to the traditional on-premise approach for many application types today, and is likely to become the dominant method for delivery of enterprise applications in the future. It can be argued that the traditional on-premise model is unsustainable, and that in the future most enterprise applications will be delivered in the form of reliable and secure application services over the Internet or using private networks.

2.2. Infrastructure Services

Cloud Computing involves making computing, data storage, and software services available via the Internet, transforming the Internet into a vast computing platform. While Cloud Computing is a relatively recent concept popularized by companies such as IBM, Amazon.com, Google and Salesforce.com [13], the underlying technological solutions that include Cluster Computing [14], Grid Computing [15], Utility Computing [16] and Virtualization [17, 18] have evolved over the last decade. Recently, a number of providers including Amazon.com, Salesforce.com, Google.com, and Yahoo.com have introduced Cloud Computing infrastructure services. The range of infrastructure services is rapidly expanding and includes storage services: e.g. Amazon Scalable Storage Service (Amazon S3), compute services: e.g. Amazon Elastic Compute Cloud (Amazon EC2) and, database management services: e.g. Amazon SimpleDB [19], [20].

Cloud Computing relies on a highly scalable infrastructure that uses statistical multiplexing to significantly increase resource utilization from the low levels prevailing in a typical data center, estimated between 5%-20%. According to some estimates multiplexing leads to cost reduction by a factor of 5-7, with additional cost savings possible by placing Cloud Computing data centers in geographical locations such that the cost of electricity, labor, property purchase, and taxes are minimized. The result of these savings is that Cloud Computing services can be offered at low cost, for example Amazon EC2 Services offer 1.0- GHz x86 ISA instance for 10 cents/per hour, and Amazon S3 storage charges are \$0.12 to \$0.15 per gigabyte-month [6].

2.3. Platform Services

An example of Cloud Computing platform service is Google App Engine [21]. Google App Engine for Java [22] is a Cloud Computing platform for Java developers that makes it possible to rapidly develop and deploy Java and J2EE applications to the *cloud* without having to provision servers, configure databases and server-side technologies. Google App Engine offers services that include authentication, authorization, data persistence and task scheduling. The Google Plugin for Eclipse simplifies the development, debugging, testing and deployment of applications, by integrating these tasks with the Eclipse development framework [23], [24]. The inclusion of the App Engine and Google Web Toolkit (GWT) SDKs within the plugin makes it possible for developers to create advanced AJAX user interfaces as well as backend services using the Java programming language. The support of standard technologies such as Java Servlets, Java Server Pages (JSP), Java Data Objects (JDO), Java Persistence API (JPA), etc. allows developers to reuse their existing skills, making it relatively easy to port existing applications to the cloud [25].

3. Web 2.0

Another important trend impacting on SOA today is the emerging Web 2.0 environment [26]. Web 2.0 is a set of economic, social and technology trends that collectively form the basis for the next generation of Internet computing. Web 2.0 involves extensive user participation and benefits from the network effect as the number of participants reaches tens of millions [27]. The Web 2.0 platform relies on light-weight programming solutions based on AJAX technologies (i.e. JavaScript and related technologies) and the REST interaction style (i.e. XML over HHTTP) [28], avoiding the complexities of SOAP Web Services. An important facet of Web 2.0 is extensive user participation and social networking, enabled by easy-to-use Web interfaces and popular application platforms (e.g. Myspace – <http://www.myspace.com/>, and Facebook – <http://www.facebook.com/>).

The proliferation of publicly available Web APIs and AJAX-enabled websites [29] is transforming the Web into a platform for collaborative development and information sharing. Web 2.0 is characterized by specialized, publicly available data sources (e.g. mapping data, weather information, etc.) accessible via light-weight APIs (Application Programming Interfaces) that can be used to create

value-added services in the form of Mashups [30]. Mashups create Web-based resources that include content and application functionality through reuse and composition of existing resources. The SOA framework should provide facilities to integrate SaaS applications with Mashups taking advantage of the extensive array of services available in the form of programmable Web APIs, so that, for example, a travel application can utilize the Google mapping service to indicate the location of hotels and other amenities. Also, importantly the AJAX programming environment enables the creation of thin-client SaaS applications with rich client-side functionality (e.g. movable windows, user entry validations, etc.). Web 2.0 and related technologies are redefining the Web into a programmatic environment with thousands of different APIs externalized by Internet companies such as Amazon.com, Google, and eBay, and facilitating new types of Internet-scale applications.

4. Conclusions

Most large organizations today manage highly complex and heterogeneous enterprise computing environments characterized by the coexistence of several generations of architectural approaches and related technologies. As a result, enterprise computing is associated with high cost of ownership and low return on investment. The high cost of IT solutions combined with the fast evolution of hardware and software necessitating frequent and costly upgrades has led some observers to conclude that information technology does not provide competitive advantage to organizations that make substantial investment in IT, and in some cases can detract from the core business in which the organization is engaged [31], [32]. In the drive to reduce IT costs during the present economic downturn, organizations are actively investigating alternatives to on-premise IT solutions. The SaaS and IaaS subscription models are particularly attractive to SMEs (Small and Medium Enterprises) as little or no upfront capital expenditure is required. However, the adoption of these models introduces its own set of problems as organizations need to be able to effectively integrate and manage externally sourced services from a range of different providers and incorporate these services into their existing enterprise IT architecture. This situation is further complicated by enterprises adopting Web 2.0 technologies resulting in new socio-economic environments that further blur the boundaries of the enterprise. The emerging enterprise computing scenarios involve organizations interacting with multiple external providers of various types of IT services, as well as with partner organizations within the context of supply chains and other types of collaborative situations. As the adoption of Cloud Computing accelerates, it is likely that most enterprise computing needs will be addressed by external providers of software and infrastructure services, and internally organizations will be mainly concerned with service integration and with managing the interface with service providers.

However, most architectural approaches, including SOA are primarily focused on modeling and managing on-premise IT resources (i.e. hardware and software components implemented within the boundaries of the organization) and do not satisfactorily address situations where infrastructure and enterprise applications are sourced externally in the form of SaaS and virtualized infrastructure services (IaaS). SOA was initially positioned as intra-enterprise architecture based on the principles of service-oriented computing with applications composed of loosely coupled services closely aligned with business processes they support and implement. Today, this basic premise on which SOA is built needs to be re-evaluated to address the challenges of computing environments that involve externally provided software and infrastructure services.

SOA, Cloud Computing, and Web 2.0 have important synergies and share a common service-based model. These technology trends are likely to converge into a single architectural framework at some point in the future, resulting in the unification of the intra-enterprise and inter-enterprise computing environments and creating new opportunities for enterprise applications and associated business models. This new architectural framework will need to address a number of important issues that include the integration of external and internal services, optimization, management and governance of complex service environments, design and modelling of internal and external services, and alignment of architectural and organizational service models.

References

1. Domingue, J., D. Fensel, and R. González-Cabero. SOA4All, Enabling the SOA Revolution on a World Wide Scale. in IEEE International Conference on Semantic Computing, 2008. Santa Clara, CA, USA.
2. Sholler, D., 2008 SOA User Survey: Adoption Trends and Characteristics. 2008. Gartner, p. 32.
http://www.gartner.com/DisplayDocument?ref=g_search&id=765720&subref=simplesearch
3. Manes, A., Thomas SOA is Dead; Long Live Services. 2009.
<http://apsblog.burtongroup.com/2009/01/soa-is-dead-long-live-services.html>
4. Eidson, B., Maron, J., Pavlik, G., Raheja, J., SOA and the Future of Application Development, in First International Workshop on Design of Service-Oriented Applications (WDSOA'05). 2005, IBM Research Report: Amsterdam, The Netherlands.
5. Papazoglou, M., Web Services: Principles and Technology 1ed. 2007: Prentice Hall. 0-32115-555-6.
6. Armbrust, M., et al. Above the Clouds: A Berkeley View of Cloud Computing. University of California, Berkeley, Tech. Rep 2009; Available from:
<http://scholar.google.com.au/scholar?q=above+the+clouds:+A+Berkeley&hl=en&lr=&output=search>.
7. Business Wire. Sun Sparks the Java Computing Revolution; Network Leader Maps Out the Future of Java's Effect on the Enterprise 1996 4 December 2007]; Available from:
http://findarticles.com/p/articles/mi_m0EIN/is_1996_Sept_23/ai_18698587.
8. ACA Research. Software-as-a-Service (SaaS) in Australia: Is it the Next Big Thing? 2007 4 March 2008]; Available from:
<http://www.anthonywatson.net.au/files/SAAS%20White%20Paper%20FINAL.PDF>.
9. Vorisek, J., Feuerlicht G. . Trends in the Delivery and Utilization of Enterprise ICT. in 17th Annual IRMA International Conference. 2006. Washington DC, USA.
10. Financial Insights IDC. On-Demand Customer Knowledge Management in Financial Services 2008 4 March 2008]; Available from:
http://www.salesforce.com/assets/pdf/analysts/idc_ondemand_financial_services.pdf.
11. Google Apps Sites website. Available from:
<http://www.google.com/apps/intl/en/business/collaboration.html#sites>.
12. Google Secure Data Connector (SDC) website. Available from:
<http://code.google.com/securedataconnector/>.
13. Buyya, R., C.S. Yeo, and S. Venugopal, Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, in Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications - Volume 00. 2008, IEEE Computer Society.
14. Oguchi, M., Research works on cluster computing and storage area network, in Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication. 2009, ACM: Suwon, Korea.
15. Yu, J. and R. Buyya, A taxonomy of scientific workflow systems for grid computing. SIGMOD Rec., 2005. **34**(3): p. 44-49.
16. Bhattacharya, J. and S. Vashistha, Utility computing-based framework for e-governance, in Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance. 2008, ACM: Cairo, Egypt.
17. Fong, L. and M. Steinder, Duality of virtualization: simplification and complexity. SIGOPS Oper. Syst. Rev., 2008. **42**(1): p. 96-97.
18. Miller, K. and M. Pegah, Virtualization: virtually at the desktop, in Proceedings of the 35th annual ACM SIGUCCS conference on User services. 2007, ACM: Orlando, Florida, USA.