Towards a standardized multiview oriented architecture SOA integrating hybrid Cloud

Rida Boukour, Ahmed Ettalbi

IMS Research Team, ADMIR Laboratory, ENSIAS, Rabat-IT Center Mohammed-V University in Rabat, Morocco

Summary
Service-oriented architecture ("SOA") is one of the interesting models that facilitate the integration between many heterogeneous applications. However, this architecture is limited due to its inability to adapt to the users specific needs and profiles. In our previous work, we have proposed an extension of the architecture SOA called Multiview Service-Oriented Architecture ("MVSOA") that meets the adaptation requirements. To realize such architecture, we have introduced components, which transform the architecture SOA to a MVSOA. These components are introduced only into the client side architecture. In the present paper, we have outsourced one of the MVSOA architecture components: the Multiview service generator, using the hybrid Cloud, in order to have a standardized public MVSOA architecture available for any consumer of services.

Key words:
Architecture, SOA, MVSOA, Multiview, Cloud Hybrid, Generator, Service

1. Introduction

A SOA-oriented architecture is a software architecture based on a set of simple services. They are developed through getting inpired by the company's business processes. A service is a stand-alone component that provides business functionalities to other applications or services. These services represent the basic functions of corporate functionality. They communicate with each other via buses or the Internet, this is called Web Service. Exchanges can be synchronous or asynchronous. Companies are enriched with services enabling it to respond quickly and flexibly to market demands. Indeed, they correspond to a business process that can be shared at company level. This makes it possible for changes within the company's strategic and tactical decisions.
In general, it can be said that SOA is an approach that can integrate the different application components of an information system and manage their interactions. It seeks to facilitate the integration of new applications within the IT structure of a company, by optimizing its exchanges and its functioning. This approach relies on the reorganization of applications through services, which correspond to communication interfaces. The advantage here is that these services are standardized and can therefore be easily interpreted by other applications sharing the same standards while being potentially exploitable by all the systems of the SI environment. Some services expose data, and others consume data, composing the basis of SI exchanges in an SOA architecture.
Despite all these advantages, the SOA architecture has limitations since it does not take into account the profile and needs of the user to interact with the service. To solve this problem, we have proposed in a previous work an extension of the SOA architecture called MVSOA ("MultiView SOA") based on the SOA architecture and multiview services.
In this work, we will first present definitions of the SOA architecture, its concept, and the producor and consomator of this architecture. Section 3 focus on the Cloud Computing concept, we outline in this section definition, caracteristics, types and service models of cloud computing. Then, in our approach presented in section 4, we will see how to obtain the MVSOA architecture from a SOA model which has already been studied in our previous work [1] [2]; next we will state in detail the different steps to obtain the standardized MVSOA architecture. Finally we have used Hybrid Cloud to encompass our standardized MVSOA architecture in order to profit from the benefits that the cloud brings.

2. Service Oriented Architecture (SOA)

2.1 Definition of SOA architecture

Service-oriented architecture is a new vision for the IT systems. The latter is no longer described as a set of applications but as a set of services. Therefore, rather than focusing on an architecture application based on technical constraints, the service-oriented architecture proposes to divide the functionalities of an application into business services that can be reusable in other applications. By focusing on services, applications are aggregated to provide richer and more meaningful business processes.
There are different definitions for the service concept or for service-oriented architectures. Among the main ones, the W3C is presented [12]:
"A service-oriented architecture is a set of components that can be invoked, and whose interface descriptions can be published and discovered".

Manuscript received December 5, 2017
Manuscript revised December 20, 2017

A service-oriented architecture (“SOA”) is based on three common fundamental actors that are the service provider, the consumer and the registry. Figure 1presents these components and the interaction between them [3]:
- The service provider specifies a service and wants to make it discoverable by consumers. To do this, it saves it in a service registry which is called Universal Description Discovry and Integration (“UDDI”).
- The service registry sets out the specification of the services available to any consumer looking for a specific service.
- After negotiating with the register, the consumer gets one or more references to the site of the service complied with the sought contract. The consumer is then able to invoke them.

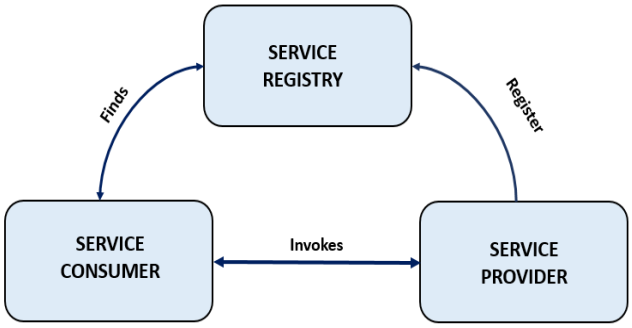

Fig1. Service Oriented Architecture.

Unlike three-tier applications that are designed as a presentation, application logic, and database format, SOA separates the design according to the model: presentation, synchronous interaction, processes, services, and databases.

2.2 Characteristics of the SOA architecture

SOA is a dynamic architecture since the service providers and consumers can join and leave the architecture during execution. SOA also provides an answer to the problems currently faced by enterprises in terms of reusability, interoperability and reduction of coupling between the different systems that make up their information system.
Data Services of SOA architecture meet the following criteria:
• Weakly coupled: traditional applications include in their code the business data of the company. They are completely linked to the systems for which they were designed. This constraint involves the difficulty of any modification request, whether it is about access to data, management rules or presentation rules. A weak coupling allows a splitting of the business aspects of the code which will allow a simple reconfiguration of the processes when the business functions evolve.
• Distributed: the services that make up the applications can be physically distributed not only across different systems within the enterprise, but also beyond.
• Evocable and publishable: services must be invokable and publishable, regardless of the used systems
The service-oriented approach greatly promotes adaptability. This approach is increasingly used, in particular in e-commerce (“Business to Business”) or in the integration of enterprise applications.

2.3 SOA Implementation

Service-oriented architectures were created before the invention of their name. For example, in 1991, the first version of the CORBA standard, (“Common Object Request Broker Architecture”), was published [4]. CORBA normalizes calls made between objects that are possibly distributed on a network and possibly written in different languages. To do this the IDL standard (“Interface Definition Language”) is used to describe how to access an object. CORBA then specifies how to switch from the DDL contract to each supported language. This definition matches that of SOA, where objects are services.
Although CORBA is still commonly used, SOAs have enjoyed a significant boost in their popularity through web services technologies. To date, web services constitute the implementation of SOA references.
Hence, this type of service is mainly used in the examples of the present paper. According to the OASIS Group, “A web service is a software component described by WSDL that can be accessed by standard network protocols such as SOAP and HTTP”. The WSDL language, which stands for “Web Service Description Language”, is an XML-based language which describes a service and answers three questions: What does the service do, how to use the service, and where is the service?
Additionally, the access to a WSDL document is sufficient to be able to interact with the service described by the latter. The WSDL specification has been developed and is maintained by the W3C group. It constitutes a standard in the industry due to its rigor and independence from the used technology platform. Several tools allow automatic code generation to communicate with a service simply by using its WSDL document [4].
Typically, in an enterprise, these documents are grouped together in one or more central registers that can be consulted by developers and possibly by partner companies.

2.4 SOA Advantages

First, SOA has all the advantages of client-server architecture. That is to say a greater modularity; one component can easily be replaced by another. They also offer the reuse of components, better evolution possibilities, greater fault tolerance and easier maintenance.
It is very easy in SOA to add new services without disturbing the existing ones as well as having them evaluated. These services are reusable, so it is possible to build new applications using third-party services. For example, an

enterprise ERP can be created using SAP services ("Systems, Applications and Products for data processing"). The services are used in accordance with the needs of the companies and the updating of these services is carried out in a totally transparent way for the consumers.
SOAs have n-tier architecture that is the separation of the data presentation from the processing and the database. For a client/server architecture, the presentation of the data as well as the processing are carried out by the client and the server used for the database. However, in n-tiers, each server can have a unique function. For example, if we take 3-tier architecture, we have the terminals for the http queries that manage the presentation part of the human-machine interface. We have an application server to manage the processes and a data server that manages data access.
N-tier architecture presents the first IT infrastructure for cooperative work. The main advantage is centralized processing at the server level because services are stored on servers. It also enables simpler management of data consistency and integrity [5].

2.5 SOA Disadvantages

The implementation of an SOA implies high cost both financial and human. A team of design experts and several teams are needed to develop and administer the various services. In the ideal case, business activity should be oriented toward services. Thus, the design of the information system is an initial critical step. If the functioning of the business is service oriented then it is difficult to use an SOA and therefore the operating cost will be high. Indeed, SOAs have a limited interest if the company does not base its processes on the exploitation of the services. Furthermore, it is necessary to design adapted workflows. Moreover, it is difficult to migrate from a monolithic architecture to SOA architecture without efficient prior study [5].

3. Cloud Computing

3.1 Definition

Like many computer terms, the term cloud can lead to confusion among the users community. In fact, cloud computing refers to the computing on the Web. If we use the Web, we are a cloud user. Online banking, on-line retail stores and on-line betting are common examples of Cloud Computing [6].
Here are some essential points to understand Cloud Computing.
• Users access cloud applications via a web browser.
• No preloaded software is required on their computers.
• They should not buy cloud applications. Access is typically free (for example, for web-based messaging such as hotmail.com), payable at the transaction (for example, for merchant sites such as amazon.com) or to the service (for example, for information services Like wsj.com).
The Cloud can also provide technological infrastructure services, such as data storage and processing capacity. The IT department can use cloud services as part of the business continuity plan by storing data outside the physical boundaries of the organization [6].
Some activities require significant processing capabilities, such as predicting the price of a commodity or simulating crash tests. If these activities are not much requested, it makes more sense to rent the capacity of a third party rather than buying a very expensive technology, which will quickly become obsolete and will be largely underused.

3.2 Cloud Computing characteristics

Cloud Computing Services have characteristics that distinguish them from other technologies [7]:
- In general, Cloud Computing users do not own the computing resources they use. The servers they operate are hosted in external data centres.
- Services are provided on a pay-per-use or subscription model basis. - The resources and services provided to the client are often virtual and shared by multiple users.
- Services are provided via Internet.
These features make Cloud Computing a new option that gives users the ability to access software and IT resources with the desired flexibility and scalability and at very competitive costs.

3.3 Cloud Computing Forms

Public Cloud

The public cloud can be used nationally or globally. The infrastructure is then made available to several categories of the public not a dedicated organization. These audiences rent services in proportion to the use they make of them. The distinction between "public" and "private" is therefore not related to the sectors of the economy concerned [8].
The advantages of this type of cloud are linked to the large-scale use, which multiplies the beneficial effects of mutualisation. Users only pay operating expenses ("OPEX"); they do not have to bear the initial investment costs. However, the public cloud encounters strong reluctance about sovereignty issues and the fact that the infrastructure is not locally or at least nationally hosted.
In addition, states are attempting, with varying degrees of success, to use nationwide solutions to counter Google's and Amazon's public cloud offerings.

Private Cloud
When large entities (like companies or communities decide to launch by themselves their clouds and become their own suppliers, it is a private cloud. They not only benefit

from the services of a cloud operator, but also they provide themselves the service. This means that the infrastructure is made available to a single entity [8]. The private cloud requires a high investment because the organization that hosts the infrastructure bears all the investment and operating costs. This system generates many advantages, such as the possibility to adapt services as much as possible to the needs of the organization (customization), greater security, some control and legal simplicity. Nevertheless, it generates capital investment expenditure ("CAPEX"). This type of arrangement mostly applies to large groups that mutualise the infrastructure of their subsidiaries or companies.

Hybrid Cloud

The hybrid cloud uses both private clouds and public ones for different jobs. For example, a company can increase its efficiency and reduce its costs by using public cloud services for standard and non-sensitive operations, and rely on a private cloud for other, more critical or specific applications or operations. Hence, the Hybrid cloud remains a flexible solution that brings together the advantages of the public cloud (costs, scalability) and those of the private one (security, personalization).

3.4 Services model in cloud computing

Despite the fact that cloud computing evolves over time, it is still divided into three service categories: infrastructure as a service ("IaaS"), platform as a service ("PaaS") and software as a service ("SaaS"):

- Software as a Service: providing Internet applications as a subscription service, the data is also stored on a server of the SaaS operator. This is the visible part of cloud computing for the end users, who no longer need to install the application on their workstation, and who access their account via the Web, on a totally secured environment.

Software as a Service is a concept consisting of proposing a subscription to software rather than purchasing a license. With the development of Information and Communication Technologies, more and more SaaS offers are being made through the web. There is no need to install a desktop or client-server application. This concept, which emerged in the early 2000s, continues that of the application service provider [9].

- Platform as a Service: is the evolution of the idea of SaaS. It refers to the providing of an environment for the development and exploitation of software on the Internet. The PaaS also brings together the developer part and the Cloud Computing system. It provides basic functions, so that the developer, for example, does not have to worry about user management or availability issues [9].

- Infrastructure as a Service: goes even further. Under this name, a hardware infrastructure was designated and rented on demand: storage, virtual machines, OS, etc. In this case, the user can, on request, have a processing capacity for any type of application.

Another kind of services that can be provided by a Cloud provider is data. This kind of service, to which we are interested in our work, is called Data as a service ("DaaS") and can be described as below:

- Data as a Service: is the cloud service that enables enterprises to access remote databases for data reading and writing. The DaaS is a data storage service that provides users with very high integrity guarantees (no loss or alteration of data, daily backup policies, monthly and annual weekly policies). Since there can be huge amounts of data (several gigabytes / terabytes), DaaS providers (hundreds of worldwide) apply rental rates based on volume, type of data transferred, policy and frequency Archiving. In some cases, clients are charged according to the amount of data they use, including both consulting and storing [10].

4. OUR APPROACH

4.1 Standardized Multiview SOA

As shown in our previous work ([1], [2]), the multiview SOA is obtained thanks to the integration of the customer-side Multiview service generator [11] as shown in this diagram (Fig. 2):

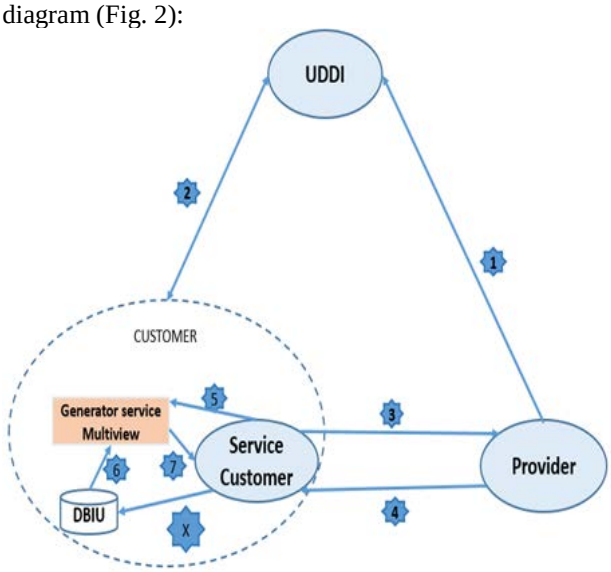

Fig. 2 Multiview SOA.

This generator makes it possible to obtain Multiview services through a mechanism for decomposing the services into sub-services according to the point of view and the access rights of the user. Then, the obtained sub-services are gathered so that they can be adapted according to the profile of each user. For example, if we take two users, a customer and a sales manager of a house of cars, each profile of these two users will see the services that are adapted

to their profile: the client can consult the services that corresponds to the prices of the cars, by contrast, he can not modify the prices of these cars which are modifiable only by the manager of house of cars.
As shown in our previous work ([1], [2]) and in Figure 2, we see that the Multiview service generator is only used in the client side in the MVSOA architecture. This work seeks to make the MVSOA architecture standardized publicly.
Our job is to outsource this Multiview service generator to standardize and publicize it so that all users using the MVSOA architecture can benefit from the Multiview services as shown in Figure 3 below:

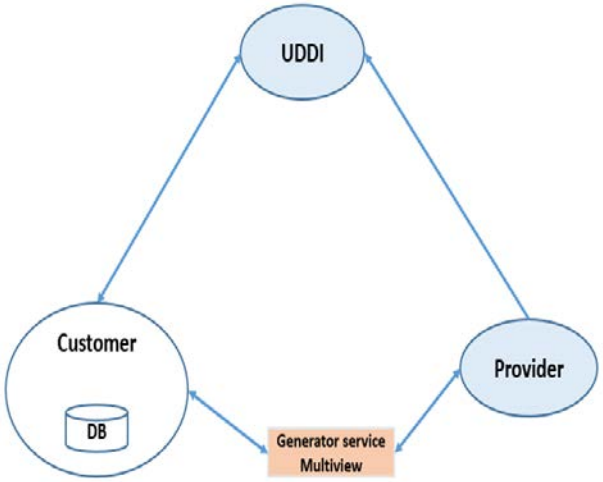

Fig. 3 Standardized MVSOA.

First, the service provider broadcasts the service description to UDDI, the service consumer needs a particular service, it addresses the UDDI, the UDDI responds by sending it the description of the requested service.
Once the description of the requested service is received by the service consumer, he uses the description to invoke it with the service provider. The description of the service will be identified by an ID for adaptation and identification reasons of the service applicant.
The service description will then be intercepted by the Multiview service generator before it arrives at the service provider. The role of this Multiview generator is for now to recover the ID, and then it will transfer the description of the service without the ID to the service provider.
The service provider responds to the multiview service generator by sending the requested service to it, then the generator recovers the service and then queries the client database (service applicant) using the ID recovery to get all information about the service applicant (login, password, profile, etc.).
Once the service requested by the service consumer, his profile and access rights are received by the multiview generator. The latter transforms the service received by the service applicant into Multiview service. Now the Multiview service generated by the Multiview service generator is able to meet the needs and profile of the service consumer, and finally the generator component of the Multiview service sends a Multiview service response to the service applicant.

4.2 Standardized Multiview SOA in Hybrid Cloud

It is generally known that hybrid cloud is an integrated cloud service using both private clouds and public ones to perform different functions within the same organization. While all types of cloud services are expected to offer some level of efficiency, to varying degrees, public cloud services are likely to be more cost-effective and more scalable than private clouds. Therefore, an organization can maximize its efficiency by using public cloud services for its non-sensitive operations and by relying on a private cloud when needed, ensuring that all its platforms are integrated harmoniously.
For our Multiview SOA, we propose to integrate the hybrid cloud to benefit both from the private cloud DaaS and the public cloud SaaS (Figure 4).

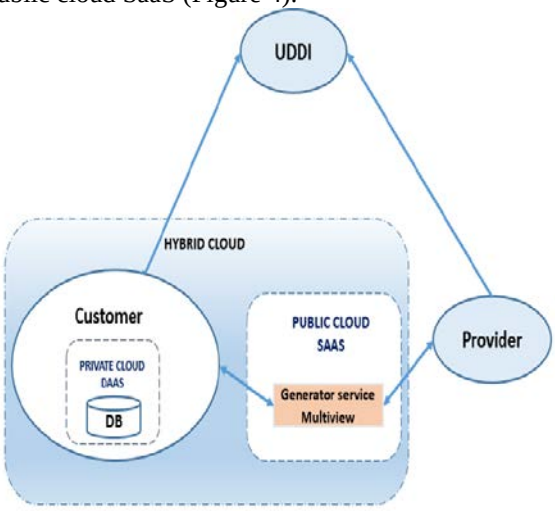

Fig. 4 Using the hybrid cloud for Dynamic MVSOA.

In our case, the hybrid cloud can encompass the public cloud multiview service generator and the database that contains private cloud user information (privileges).
The advantage of the hybrid cloud in this standardized MVSOA architecture is to first benefit from the multiview service generator as a centralized public cloud that can be used by any SaaS service consumer and then to benefit from the private cloud which contains in our case the database in a DaaS service. The advantage of the private element of the hybrid cloud model is that it not only provides security where it is needed for sensitive operations, but also it enables it to meet the legal requirements for data hold and storage if necessary.

5. Conclusion

In the present paper, we have proposed a standardized Multiview Service-Oriented Architecture ("MVSOA"). Such architecture adapts according to the profile and the need of the service applicant. To do this, we have outsourced a component of the multiview SOA architecture that was on the consumer side of the service. This component is the Multiview Service Generator. The goal of this outsourcing is to make the MVSOA architecture standardized to enable any service user to receive services adapted to his profile. The multiview service generator module is the key element in this architecture, because of its ability to transform a simple service to a service capable of meeting the needs of the user.

In this work, the hybrid cloud in our standardized MVSOA architecture has been introduced. This introduction allows us to combine the public cloud that contains the multiview generator into SaaS service, and the private cloud to recover the Data access rights in database as a DaaS service.

In our future work, we intend to clarify in detail the steps that occur between the service consumer and the service provider of the standardized MVSOA architecture, in particular the operation of the ID identifier and its integration in the XML language.

We would like to introduce security in the standardized MVSOA architecture, and especially in the communication part between the SAAS multiview service generator and the DAAS database.

References

[1] R.Boukour, A.Ettalbi, Multiview SOA: extending SOA using a private Cloud Computing as SaaS, IEEE International Conference on Cloud Computing Technologies and Applications (CloudTech2015), June 2-4, Marrakesh, Morocco, http://www.macc.ma/cloudtech15/.
[2] R.Boukour, A.Ettalbi, Multiview soa: extending soa using a private cloud computing as SAAS and DAAS, International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.6, pp. 1-11, November 2015.
[3] H, Stefan, SOA : L'utilité organisationnelle, technique et financière de l'architecture orientée service, paper, Fribourg, Aoaut, 2013.
[4] A.Beaupré, Étude des architectures de sécurité orientées service, Mémoire, SHERBROOKE, September, 2012.
[5] A.Guillaume, Les Architectures Orientées Services (SOA), mémoire, calais, 2010.
[6] Cisco, innovate together, les base du Cloud Computing: revaloriser les technologies de l'information, May 2011.
[7] G.SIPA, Cloud Computing: Etude et tests, université montréal, p1037306, 2009.
[8] S.van Gaver, A.Fiorentino, M.Slitni, T. Bedouin, A.Alehyane, Bonne pratique pour la mise en place d'un cloud dans l'enseignement supérieur, Université numérique Paris ile-de-France, Paris, 2015
[9] http://icp.ge.ch/sem/cms-spip/spip.php?article962.
[10] F.Pêcheux, « Cloud Computing ou informatique dans les nuages », Encyclopædia Universalis [en ligne], February 2017.
[11] R.Boukour, A.Ettalbi and M.Nassar, Multiview Web Service: The Integration of The Notion of View And Point of View in The Web Services, International Journal of Computer Science and Network Security (IJCSNS),Vol. 14 No. 2 pp. 31-36, February 2014.
[12] Web Services Glossary, available at: http://www.w3.org/TR/ws-gloss

BOUKOUR RIDA PhD student systems engineering and models At the Higher National School for Computer Science and Systems Analysis (ENSIAS), Rabat. Administrator senior system

ETTALBI Ahmed Professor at Software Engineering Department of the Higher National School of Computer Science and Systems Analysis (ENSIAS) Rabat. His main research interests Object Modeling with Viewpoints, Software Architecture and Business Process Modeling architecture, Cloud Computing.