International Journal of Research in Engineering, Science and Management
Volume-2, Issue-2, February-2019
www.ijresm.com | ISSN (Online): 2581-5792

Current Cloud Computing and Service-Oriented Cloud Computing Architecture (SOCCA): Security Issues and Benefits

Nasrin Munshibhai Shah
Lecturer, Department of Computer Engineering, G. P. Jalna, Jalna, India

***Abstract*: Service Oriented Architecture (SOA) SOA is an architectural pattern that guides business solutions to create, organize and reuse its computing components. SOA allows application components to provide services to other components via communications protocol. This paper gives an overview survey of current cloud computing architectures, discusses issues that current cloud computing implementations have and proposes a Service-Oriented Cloud Computing Architecture (SOCCA) so that clouds can interoperate with each other. Furthermore, the SOCCA also proposes high level designs to better support multi-tenancy feature of cloud computing. The SOCCA is a 4-layer architecture that supports both SOA and cloud computing. SOCCA allows an application to run on different clouds, migration of applications and interoperate with each other. SOCCA provides high support of multi-tenancy feature of Cloud. SOCCA allows application development across multiple Clouds.**

***Keywords*: IaaS, PaaS, SaaS, SOCCA.**

1. Introduction

A Hierarchical View of Cloud Computing Most of the current clouds are built on top of modern data centers. It incorporates Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and provides these services like utilities, so the end users are billed by how much they used. Figure 1shows a hierarchical View of Cloud Computing. Current Cloud Computing and Service-Oriented Cloud Computing Architecture (SOCCA): Security issues and benefits. Infrastructure as a Service:Built on top of data centers layer, IaaS layer virtualizes computing power, storage and network connectivity of the data centers, and offers it as provisioned services to consumers. Users can scale up and down these computing resources on demand dynamically. Typically, multiple tenants coexist on the same infrastructure resources .Examples of this layer include Amazon EC2, Microsoft Azure Platform.

A. Platform as a service

PaaS, often referred as cloud ware, provides a development platform with a set of services to assist application design, development, testing, deployment, monitoring, hosting on the cloud. It usually requires no software download or installation, and supports geographically distributed teams to work on projects collaboratively. Google App Engine, Microsoft Azure, Amazon Map Reduce/Simple Storage Service are among examples of this layer.

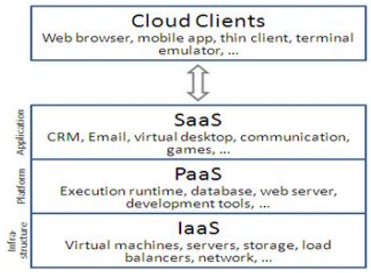

Fig. 1. Hierarchical view of cloud computing

B. Software as a service

In SaaS, Software is presented to the end users as services on demand, usually in a browser. It saves the users from the troubles of software deployment and maintenance. The software is often shared by multiple tenants, automatically updated from the clouds, and no additional license needs to be purchased. Features can be requested on demand, and are rolled out more frequently. Because of its service characteristics, SaaS can often be easily integrated with other mashup applications. An example of SaaS is Google Maps.

C. Data centers

This is the foundation of cloud computing which provides the hardware the clouds run on. Data centers are usually built in less populated areas with cheaper energy rate and lower probability of natural disasters. Modern data centers usually consist of thousands of inter-connected servers.

D. Deployment models of cloud computing

Cloud computing is providing developers and IT departments with the ability to focus on what matters most and avoid undifferentiated work like procurement, maintenance, and capacity planning. As cloud computing has grown in popularity, several different models and deployment strategies have emerged to help meet specific needs of different users. Each type of cloud service, and deployment method, provides you with different levels of control, flexibility, and management. Understanding the differences between

Infrastructure as a Service, Platform as a Service, and Software as a Service, as well as what deployment strategies you can use, can help you decide what set of services is right for your needs.

1) Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management essentially ", the economic model that makes cloud computing such an intriguing concept".

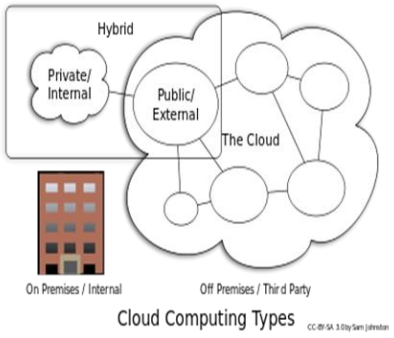

Fig. 2. Cloud computing types

2) Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free. Technically there may be little or no difference between public and private cloud architecture. However, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and "Azure Express Route" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

3) Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers [2]. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service. Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services. Hybrid cloud adoption depends on a number of factors such as data security and compliance requirements, level of control needed over data, and the applications an organization uses.

2. Literature survey

Cloud computing has been cited as ‘the fifth utility’ (along with water, electricity, gas, and telephone) where by computing services are readily available on demand, like other utility services available in today’s society [Buyya,Yeo, Venugopal, Broberg, and Brandic, 2009]. This vision is not essentially new. Dating back to 1961, John McCarthy, retired Stanford professor and Turing Award winner, in his speech at MIT’s Centennial, predicted that in the future computing would become a ‘public utility’ [Wheeler and Waggener, 2009].

It could be argued that cloud computing has begun to fulfill this vision of computing on demand. The first step of studying research into cloud computing is to clarify the concept. Among the various definitions, the one by the NIST (National Institute of Standards and Technology) has gained recent recognition and popularity. For the purpose of this study, the NIST definition of cloud computing is adopted to facilitate the following discussions. The NIST further suggests that a cloud computing model should be composed of five essential characteristics, three service levels, and four deployment models.

A. Definitions of Cloud Computing

- A style of computing where massively scalable IT-related capabilities are provided as a service across the Internet to multiple external customers. Gartner [Plummer, Smith, Bittman, Cearley, Cappuccio, Scott, et al., 2009]
- A pool of abstracted, highly scalable, and managed infrastructure capable of hosting end-customer applications and billed by consumption. Forrester [Staten, 2008]

- The illusion of infinite computing resources available on demand, thee limitation of up-front commitments by cloud users, and the ability to pay for use of computing resources on a short-term basis as needed.

B. Security issues in cloud.

Although there are many benefits to adopting Cloud Computing, there are also some significant barriers to adoption. One of the most significant barriers to adoption is security, followed by issues regarding compliance, privacy and legal matters. Because Cloud Computing represents a relatively new computing model, there is a great deal of uncertainty about how security at all levels (e.g., network, host, application, and data levels) can be achieved and how applications security is moved to Cloud Computing. That uncertainty has consistently led information executives to state that security is their number one concern with Cloud Computing. Traditional security mechanisms such as identity, authentication, and authorization are no longer enough for clouds in their current form Security controls in Cloud Computing are, for the most part, no different than security controls in any IT environment. However, because of the cloud service models employed, the operational models, and the technologies used to enable cloud services, Cloud Computing may present different risks to an organization than traditional IT solutions. Unfortunately, integrating security into these solutions is often perceived as making them more rigid.

C. Software-as-a-service (SaaS) security issues

SaaS provides application services on demand such as email, conferencing software, and business applications such as ERP, CRM, and SCM. SaaS users have less control over security among the three fundamental delivery models in the cloud[4]. The adoption of SaaS applications may raise some security concerns.

1) Application security

These applications are typically delivered via the Internet through a Web browser .However, flaws in web applications may create vulnerabilities for the SaaS applications. Attackers have been using the web to compromise user's computers and perform malicious activities such as steal sensitive data .Security challenges in SaaS applications are not different from any web application technology, but traditional security solutions do not effectively protect it from attacks, so new approaches are necessary. The Open Web Application Security Project (OWASP) has identified the ten most critical web applications security threats .There are more security issues, but it is a good start for securing web applications.

2) Multi-tenancy

SaaS applications can be grouped into maturity models that are determined by the following characteristics: scalability, configurability via metadata, and multi-tenancy. In the first maturity model, each customer has his own customized instance of the software. This model has drawbacks, but security issues are not so bad compared with the other models. In the second model, the vendor also provides different instances of the applications for each customer, but all instances use the same application code. In this model, customers can change some configuration options to meet their needs. In the third maturity model multi-tenancy is added, so a single instance serves all customers This approach enables more efficient use of the resources but scalability is limited. Since data from multiple tenants is likely to be stored in the same database, the risk of data leakage between these tenants is high.

3) Data security

Data security is a common concern for any technology, but it becomes a major challenge when SaaS users have to rely on their providers for proper security. In SaaS, organizational data is often processed in plaintext and stored in the cloud. The SaaS provider is the one responsible for the security of the data while is being processed and stored. Also, data backup is a critical aspect in order to facilitate recovery in case of disaster, but it introduces security concerns as well. Also cloud providers can subcontract other services such as backup from third-party service providers, which may raise concerns. Moreover, most compliance standards do not envision compliance with regulations in a world of Cloud Computing. In the world of SaaS, the process of compliance is complex because data is located in the provider's datacenters, which may introduce regulatory compliance issues such as data privacy, segregation, and security, that must be enforced by the provider.

4) Accessibility

Accessing applications over the internet via web browser makes access from any network device easier, including public computers and mobile devices. However, it also exposes the service to additional security risks. The Cloud Security Alliance has released a document that describes the current state of mobile computing and the top threats in this area such as information stealing mobile malware, insecure networks (Wi-Fi), vulnerabilities found in the device OS and official applications, insecure marketplaces, and proximity-based hacking.

5) Platform-as-a-service (PaaS) security issues

PaaS facilitates deployment of cloud-based applications without the cost of buying and maintaining the underlying hardware and software layers.As with SaaS and IaaS, PaaS depends on a secure and reliable network and secure web browser. PaaS application security comprises two software layers: Security of the PaaS platform itself (i.e., runtime engine), and Security of customer applications deployed on a PaaS platform. PaaS providers are responsible for securing the platform software stack that includes the runtime engine that runs the customer applications. Same as SaaS, PaaS also brings data security issues and other challenges that are described as follows:

6) Third-party relationships

Moreover, PaaS does not only provide traditional programming languages, but also does it offer third-party web services components such as mashups. Mashups combine more than one source element into a single integrated unit. Thus,

PaaS models also inherit security issues related to mashups such as data and network security. Also, PaaS users have to depend on both the security of web-hosted development tools and third-party services.

7) Development life cycle

From the perspective of the application development, developers face the complexity of building secure applications that may be hosted in the cloud. The speed at which applications will change in the cloud will affect both the System Development Life Cycle (SDLC) and security. Developers have to keep in mind that PaaS applications should be upgraded frequently, so they have to ensure that their application development processes are flexible enough to keep up with changes. However, developers also have to understand that any changes in PaaS components can compromise the security of their applications. Besides secure development techniques, developers need to be educated about data legal issues as well, so that data is not stored in inappropriate locations. Data may be stored on different places with different legal regimes that can compromise its privacy and security.

8) Underlying infrastructure security

In PaaS, developers do not usually have access to the underlying layers, so providers are responsible for securing the underlying infrastructure as well as the applications services[7]. Even when developers are in control of the security of their applications, they do not have the assurance that the development environment tools provided by a PaaS provider are secure. PaaS applications and user's data are also stored in cloud servers which can be a security concern as discussed on the previous section. In both SaaS and PaaS, data is associated with an application running in the cloud. The security of this data while it is being processed, transferred, and stored depends on the provider.

9) Infrastructure-as-a-service (IaaS) security issues

IaaS provides a pool of resources such as servers, storage, networks, and other computing resources in the form of virtualized systems, which are accessed through the Internet. Users are entitled to run any software with full control and management on the resources allocated to them. With IaaS, cloud users have better control over the security compared to the other models as long there is no security hole in the virtual machine monitor. They control the software running in their virtual machines, and they are responsible to configure security policies correctly. However, the underlying compute, network, and storage infrastructure is controlled by cloud providers. IaaS providers must undertake a substantial effort to secure their systems in order to minimize these threats that result from creation, communication, monitoring, modification, and mobility. Here are some of the security issues associated to IaaS.

D. Virtualization

Virtualization allows users to create, copy, share, migrate, and roll back virtual machines, which may allow them to run a variety of applications. However, it also introduces new opportunities for attackers because of the extra layer that must be secured. Virtual machine security becomes as important as physical machine security, and any flaw in either one may affect the other. Virtualized environments are vulnerable to all types of attacks for normal infrastructures; however, security is a greater challenge as virtualization adds more points of entry and more interconnection complexity. Unlike physical servers, VMs have two boundaries: physical and virtual. Virtual machine monitor. The Virtual Machine Monitor (VMM) or hypervisor is responsible for virtual machines isolation; therefore, if the VMM is compromised, its virtual machines may potentially be compromised as well.

The VMM is a low-level software that controls and monitors its virtual machines, so as any traditional software it entails security flaws. Keeping the VMM as simple and small as possible reduces the risk of security vulnerabilities, since it will be easier to find and fix any vulnerability. Therefore data storage and virtualization are the most critical and an attack to them can do the most harm. It also describes the threats that are related to the technology used in cloud environments, and it indicates what cloud service models are exposed to these threats. We put more emphasis on threats that are associated with data being stored and processed remotely, sharing resources and the usage of virtualization.

E. Issues with current clouds

Current cloud computing has following characteristics:

Users are often tied with one cloud provider: Even though up-front cost for a cloud computing deployment is reduced and long term lease is eliminated, much effort and money is spent on developing the application for a specific cloud platform which makes it difficult to migrate the same application onto a different cloud[6]. Often, migration simply may mean redevelopment. For example, applications deployed on Amazon EC2 cannot be migrated easily due its particular storage framework.

F. Computing components are tightly coupled

This can be clearly explained using an analogy. Suppose one wants a new computer, this person has the choices of either buying a ready-to-use computer from a manufacturer (buying) or purchasing the components separately and building the computer in a DIY style (building). The advantages of building over buying include wider selection of components, flexibility to customize, and cheaper cost . However, as the computing resources over the internet, current cloud implementations do not allow this kind of flexibility. If a customer opts to use Amazon S3 storage service, he is then stuck with other cloud computing services Amazon provides, such as EC2, Elastic Map Reduce.

G. Lack of SLA supports

Currently, SLA is an obstacle that prevents wide adoption for cloud computing. Cloud computing infrastructure services such as EC2 are not yet able to sign the SLA needed by companies

that want to use cloud computing for serious business deployment [11].

H. Lack of multi-tenancy supports

Multi-tenancy can support multiple client tenants simultaneously to achieve the goal of cost effectiveness. Currently, one has three types of multi-tenancy enablement approaches: virtualization, mediation and sharing To achieve the full potential of multi-tenancy, three issues remain to be solved [12]:

- Resource sharing: To reduce the hardware, software and management cost of each tenant.
- Security isolation: To prevent the potential invalid access, conflict and interference among tenants.
- Customization: To support tenant-specific UI, access control, process, data, etc.

I. Lack of flexibility for user interface

UI is an important part of the application, and user experience can be a major evaluation factor for a business application. However, cloud/SaaS users are limited with UI choices because UI composition frameworks.

3. Service oriented cloud computing architecture (SOCCA)

A. Cloud computing and SOA

SOA and cloud computing are related, specifically, SOA is an architectural pattern that organize and reuse its computing components, while cloud computing is a set of enabling technology that services a bigger, more flexible platform for enterprise to build their SOA solutions. In other words, SOA and cloud computing will coexist, complement, and support each other. There have been several initiatives at attempting bridging SOA and cloud computing.

B. Layered architecture of SOCCA

1) Individual cloud provider layer

This layer resembles the current cloud implementations. Each cloud provider builds its own data centers that power the cloud services it provides. Each cloud may have its own proprietary virtualization technology or utilize open source virtualization technology, such as Eucalyptus. Similar to Market-Oriented Cloud Architecture proposed in, within each individual cloud, there is a request dispatcher working with Virtual Machine Monitor and Service/App Governance Service to allocate the requests to the available recourses. The distinction from current cloud implementations is that the cloud computing resources in SOCCA are componentized into independent services such as Storage Service, Computing Service and Communication Service, with open-standardized interfaces, so they can be combined with services from other cloud providers to build a cross-platform virtual computer on the clouds. In order to achieve maximum interoperability, uniform standards need to be implemented. For example, SQL is de facto standard for RDBMS data management, and many database vendors have their own implementations. A cloud version of SQL needs to be defined, so data manipulation logic of an application that works on one cloud can also work other clouds. A distributed computing framework standard to unify all different implementations of Map/Reduce is also in need for the same reason.

C. Cloud ontology mapping layer

Cloud providers might not conform to the standards rigidly; they might also have implemented extra features that are not included in the standards. Cloud Ontology Mapping Layer exists to mask the differences among the different individual cloud providers and it can help the migration of cloud application from one cloud to another. Several important ontology systems are needed:

- *Storage ontology:* It defines the concepts and terms related to data manipulation on the clouds, such as data update, date insert, data delete, and data select, etc.
- *Computing ontology:* It defines the concepts and terms related to distribute computing on the clouds, such as Map/Reduce Framework.
- Communication Ontology: It defines the concepts and terms related Communication Schema among the clouds, such as data encoding schema, message routing.

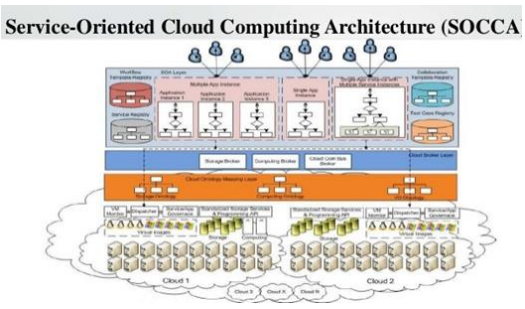

Fig. 3. Layered Architecture of SOCCA

D. Cloud broker layer

Cloud brokers serve as the agents between individual cloud providers and SOA layer. Each major cloud service has an associated service broker type.

E. SOA layer

This layer fully takes the advantages of the existing research and infrastructure from traditional SOA. Many existing SOA frameworks, such as CCSOA, UCSOA, GSE and UISOA can be integrated into this layer. The fundamental difference of the SOA layer of SOCCA from traditional SOA is that the service providers no longer host the published services anymore. Instead, they publish the services in deployable packages, which can be easily replicated and redeployed to different cloud hosting environments. Application developers can decide which clouds they want to these services to run based a set of criteria. SOA layer of SOCCA allows more flexibility than traditional SOA; it further separates the roles of service

providers and cloud providers, and the service logics and its running environments.

F. Application development on SOCCA

1) Service package

Service providers of traditional SOA develop the logic of a service and provide its running environment. In SOCCA, services are published as re-deployable packages, namely service package. A service package contains the following required/ optional information and files

2) Compiled code:

If service providers only use the standard APIs and protocols, a single version of complied code is enough; if service providers optimize the performance of their services by utilizing some platform unique APIs and features, complied code for each platform is needed.

3) Source code

This is optional. It is useful to help its user to understand the service better, also gives the freedom to its users to tweak the services to accommodate their specific requirement.

4) Configuration file

Services might use external basic services. For example, a computing intensive scientific service which also uses a lot of storage might deploy its computing logic on a cloud that provides high performance computing power, but use the cheaper storage service provided by another cloud. This requires a configuration file which specifies the external service's locations, partner link, etc. This can also be achieved in a BPEL manner. However, since basic services such as storage services, have a widely adopted standards, and are frequently used, so it is more efficient to handle in a database connection configuration file style.

G. Advantages & disadvantages

Advantages

- A new multi tenancy pattern single instance and multiple service instance becomes possible with SOCCA.
- SOCCA allows application to run on other cloud and interoperate with each other.
- SOCCA support easy migration of application from one cloud to another.
- SOCCA provides better support for Multi tenancy feature.

Disadvantages

- SOCCA does not provide any modeling language to map different language sets as different clouds support language.
- Instances for the same service cannot live on multiple clouds.
- SOCCA does not provide any modeling language to map different language sets as different clouds supports different language sets.
- Instance for the same service cannot live on multiple clouds.

4. Conclusion

This paper presented an overview on current cloud computing and service-oriented cloud computing architecture (SOCCA): Security issues and benefits.

References

[1] Tsai, Wei-Tek, Xin Sun, and Janaka Balasooriya. "Service-oriented cloud computing architecture." Information Technology: New Generations (ITNG), 2010 Seventh International Conference on. IEEE, 2010.
[2] Cloud Computing: benefits, risks and recommendations for information security D Catteddu - Web application security, 2010 – Springer.
[3] CCOA: Cloud computing open architecture LJ Zhang, Q Zhou - Web Services, 2009. ICWS 2009. IEEE 2009.
[4] Cloud computing: G Boss, P Malladi, D Quan, L Legregni, H Hall - IBM white paper, 2007 - files.spogel.com.
[5] Cloud computing: state-of-the-art and research challenges Q Zhang, L Cheng, R Boutaba - Journal of internet services 2010.
[6] The management of security in cloud computing S Ramgovind, MM Eloff, E Smith - Information Security for South 2010.
[7] Addressing cloud computing security issues D Zissis, D Lekkas - Future Generation computer systems, 2012
[8] Study on cloud computing security, "DG Feng, M Zhang, Y Zhang, Z Xu - Journal of software, 2011.
[9] http://www01.ibm.com/software/solutions/soa/newsletter/nov09/article_soaandcloud.html
[10] http://www.opengroup.org/soa/source-book/socci
[11] Parascale. [Online]. http://www.parascale.com
[12] Elastra. [Online]. http://www.elastra.com
[13] Appirio. [Online]. http://www.appirio.com