MODELING AND SIMULATION AS A CLOUD SERVICE: A SURVEY

Erdal Cayirci

Electrical Engineering & Computer Science Department
University of Stavanger
Stavanger, Norway

ABSTRACT

Modelling and simulation as a service (MSaaS) is defined, and the differences between MSaaS and Software as a Service are clarified. MSaaS architectures and deployment strategies are surveyed. The top threats to cloud computing and MSaaS, the other security challenges and technical requirements are explained. Accountability, risk and trust modelling are related to each other and also to security and privacy. Those notions and their relations are presented. MSaaS composition in multi-datacenter and/or multi-cloud scenarios is also elaborated on.

1 INTRODUCTION

Modelling and simulation as a service (MSaaS) has attracted many researchers recently. Virtualization and cloud computing have already been used as infrastructure and platform both for military and civilian modelling and simulation (M&S) (Cayirci 2009b) (Cayirci 2011). There are already M&S software offered as cloud service in the Internet. However, to the best of our knowledge, a definition of MSaaS that is agreed by everyone and clarifies the distinction between MSaaS and Software as a Service (SaaS) (Armbrust 2010) (Garg 2011) (Hwang 2011) (Valipour 2009) (Zhang 2011) is still not available in the literature.

MSaaS is a model for provisioning modelling and simulation (M&S) services on demand from a cloud service provider (CSP), which keeps the underlying infrastructure, platform and software requirements/details hidden from the users. CSP is responsible for licenses, software upgrades, scaling the infrastructure according to evolving requirements, and accountable to the users for providing grade of service (GoS) and quality of service (QoS) specified in the service level agreements (SLA). The National Institute of Standards and Technology (NIST) lists five essential characteristics of cloud computing as the following (Internet 2013):

- **On-demand self-service** – Users can provision computing capabilities, such as server time and network storage, as needed, automatically without requiring human interaction.
- **Broad network access** – Capabilities are available over the network and accessed through standard mechanisms via thin or thick client platforms (e.g., mobile phones, tablets, laptops, workstations, etc.).
- **Resource pooling** – Computing resources are pooled to serve multiple consumers using a multitenant model. Different physical and virtual resources are dynamically assigned and reassigned according to consumer demand.
- **Rapid elasticity** – Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
- **Measured service** – Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.
These characteristics introduce better utilization, ease in technical administration and therefore cost reduction. They also imply a big paradigm shift in computing and a long list of challenges related to both its ecosystem and technical requirements. Academia and industry have put considerable effort to tackle with those challenges for almost a decade. Therefore, most of them are either solved or being solved at least in theory.

In this paper, we survey the literature related to the research challenges and available solutions for MSaaS. First, we clarify the terminology, differences and relations among virtualization, cloud computing, infrastructure, platform and software as a service (IaaS, PaaS and SaaS, relatively) in Section 2. We categorize and explain the MSaaS architectures based on the state of the art MSaaS deployments in Section 3. Security and privacy related challenges of MSaaS are investigated in Section 4. Accountability, risk and trust are often related to security and especially privacy in the literature. However, they have a larger scope. An insight into accountability, risk and trust is provided in Section 5. Service composition is an essential enabler for effective MSaaS. Trustworthy service composition, interoperability and weaknesses of the current simulation federation technologies with respect to MSaaS requirements are explained in Section 6. We conclude our paper in Section 7.

2 VIRTUALIZATION, CLOUD COMPUTING AND MSaaS

Virtualization and cloud computing are two notions often mixed. They are related but not the same. Virtualization is generally accepted as an enabler for cloud computing, which is also arguable, because virtualization is not a necessity for cloud computing. Differences between virtualization and cloud computing, as well as the differences between MSaaS and SaaS are clarified in this section.

![Figure 1: Type 1 and 2 hypervisors](image)

Hardware virtualization provides an abstraction from the underlying hardware, and is used for creating virtual machines that act like separate computers with their operating system (OS) on a single physical machine. This allows running multiple virtual machines (i.e., guest machines) with different operating systems over an actual machine (i.e., host machine). The software or firmware that runs in host machines to create and to manage guest machines is called as hypervisor or virtual machine monitor. There are two types of hypervisors: Type 1 (i.e., also called bare metal/native) hypervisors run directly on host machine hardware. Type 2 (i.e., also called hosted) hypervisors run on a operating system, which makes guest machines the third layer over the actual hardware as shown in Figure 1. Hardware virtualization typically classified as full, partial and paravirtualization. In full virtualization, all hardware is simulated by the hypervisor, and therefore the operating system and software of the guest machine can run on hypervisor without any modifications. When hypervisor partially simulates the actual hardware, the guest software
may need to be modified. Paravirtualization does not simulate any part of the actual hardware, but allows guest machines run in host machine in their isolated places. However, software in the guest machines needs to be modified for the specifics of the host machine when paravirtualization is used.

Running server software on virtual machines is very common nowadays. Many large organizations and corporate prefer virtualization also for desktops. In desktop virtualization, VM for each desktop is run on central host machines. A user can access to a desktop VM (i.e., a guest machine for desktop) by various types of hardware, typically thin clients specifically designed for this purpose. The main benefits of server or desktop virtualization can be listed as centralization of administrative tasks, higher scalability, and better resource utilization. Software can be upgraded, and hardware can be maintained without the user noticing it because a VM can be migrated (i.e., snapshotting and teleportation) from one host machine to another while they are running. The same approach can be used for backups and disaster recovery procedures. All these provide highly increased efficiency and flexibility.

Virtualization can also be used by individuals in their personal hardware, such as a laptop. For example, an individual can run both MS Windows and Linux as VM in a laptop. Moreover, multiple VM with the same operating system but different settings can be run in the same host laptop for various applications.

As it is already clear, hardware virtualization is definitely not only for cloud computing. Cloud computing does not necessitate the use of virtualization either. Cloud service providers (CSP) can use operating systems and software directly running on actual hardware without any virtualization. However, hardware virtualization offers many advantages as listed above for CSP.

Cloud computing takes the advantages listed above to the next level. It promises delivering computation and data management as a service on demand. In principle, cloud hides all the complexity of the underlying architecture and infrastructure (e.g., communications, networks, hardware, software, etc.), and the users do not need a specific hardware for receiving services from it.

As explained in the Introduction, on demand self service and measured services are among the characteristics of cloud computing, which enable utility computing that means users pay for services as they use (i.e., pay per use) but not fixed prices as when they buy a new hardware or software. This is in essence like utility services (i.e., electricity and water) and therefore called as utility computing. It promotes rapid elasticity for users and better utilization of resources. The same hardware and software are shared by many people. Therefore, less number of technical administrators is needed, which reduces the costs. Software licenses can also be shared. Please note that a debate in industry on software licenses issue in clouds continues. Cloud seems advantageous for users. However, the profits of the software companies may be affected because of that, which may lead drop in quality and supply. Therefore, there needs to be an agreeable point established among software suppliers and CSP. Not only hardware and software licenses but also software processes (i.e., a program that runs) can be shared in cloud computing, which means a program serves to multiple users at the same time (i.e., multi-tenancy). Self configuration and optimization mechanisms automatically decide which user processes will use which computational and memory resources, and which user data will be stored in which data center. Similarly self healing mechanisms make planning, preparation and execution of automatic recovery procedures from failures.

Apart from utility computing, cloud computing is often linked also to the other forms of computation, such as, mainframe computing, grid computing, autonomic computing and high performance computing. All of these are related and paved the way for cloud computing. Cloudlet computing is yet another term introduced recently in the literature (Huerta-Capena 2010)(Satyanarayanan 2009)(Shi 2012) especially for mobile computing. Cloudlet computing is used for offloading mobile applications from resource limited smart phones to resource richer devices in the vicinity, and still avoid relatively long propagation delays, which may imply high communications costs. We will not elaborate on those ways of computation or their differences from cloud computing in this paper.

Having the capabilities explained above, a CSP can provide three basic service types (i.e., service models) to the users (Internet 2013):
Infrastructure as a Service (IaaS): This is the most basic service. CSP provides computers (physical or virtual machines) and other resources such as data storage.

Platform as a Service (PaaS): This is the second level, which includes not only physical environment but also computing utilities, such as operating systems, programming languages, database management systems and web servers.

Software as a Service (SaaS): CSP provide application software as a service, that basically includes also the underlying infrastructure and platform as required.

There are also many other service types proposed in literature or offered by CSP, such as, Network as a Service (NaaS), Trust as a Service, Reputation as a Service, Authentication as a Service, Authorization as a Service (Laborde 2013). They are basically derivations of PaaS and SaaS in various combinations and forms. Modelling and Simulation as a Service (MSaaS) can be perceived as one of these derivatives. At the first glance, it is not easy to see the difference between SaaS and MSaaS, because MSaaS is in essence a special form of SaaS. The inter-relations between MSaaS and conventional cloud services, i.e., SaaS, PaaS and IaaS are depicted in Figure 2. We consider three types of MSaaS: modelling as a service, model as a service and simulation as a service. Users may use any type of MSaaS and store the results for later use in the CSP that provides these services, in another CSP or in their own personal or corporate environment. They may develop models by using modelling as a service, use previously developed models to run simulations in their own environment or run simulations by using simulation as a service from a CSP.

![Figure 2: The inter-relations of cloud services including MSaaS](image)

### 3 MSAAS ARCHITECTURES

In the previous section, we explain virtualization, cloud computing and MSaaS. In this section, we focus on their architectural components and deployment scenarios. A cloud has two ends: front-end and back-end. The front-end is where the user interfaces are. That is the only part of a cloud visible to the users, and should not need any special hardware. The user sees the back-end as a cloud without knowing any details about its internal architecture.

The back-end includes various components (i.e., infrastructure and platforms) loosely coupled to each other through a mechanism that allows elasticity. Please note that a cloud (i.e., a CSP) typically maintains...
Cayirci

multiple datacenters remotely located from each other. A datacenter is a facility that houses server pools and infrastructure to store, to process and to communicate large volume of data. The architecture illustrated in Figure 3 can be introduced in various forms listed below:

- **A public cloud** is a CSP that provides cloud services to public over the Internet.
- Services provided by multiple public clouds may compose more sophisticated services, which forms an **inter-cloud** (i.e., cloud of clouds). This is sometimes also called as service mash-ups or multi-clouds.
- **A private cloud** provides services to an organization through an intranet.
- Cloud services may be open not to public but to a community of interest from multiple organizations through the Internet or a special inter-organizational network. This type of clouds is called as **community cloud**.
- Private clouds can be connected to each other to form a cloud of private clouds, called **partner clouds**.
- **Hybrid clouds** are any combination of the clouds listed above.

![Figure 3: Basic components of a cloud architecture](image)

Please note that, NIST defines only public, private, community and hybrid clouds as the deployment models (Internet 2013). MSaaS can be a service offered by any type of clouds (Johnson 2013). MSaaS can be designed in one of the following forms:

- **Standalone MSaaS applications**: Standalone applications, such as business process modelling and supply chain simulation (Rosetti 2012), are already available as MSaaS from public clouds in the Internet. They are mostly offered as web based services. Several militaries and international military organizations are also making preparations to start serving standalone combat models such as Joint Conflict and Tactical Simulation (JCATS) and Joint Theater Level Simulation (JTLS) (Cayirci 2009a) as MSaaS from their private clouds.
- **Federated standalone MSaaS applications**: Standalone MSaaS applications can be federated. These applications can be from the same data center or multiple data centers. When the services are from multiple data centers, formation of an inter-cloud or partner cloud may be required. We call the federations of standalone MSaaS applications shown in Figure 5.a as Type 0 or Type 2 federations. Categories of MSaaS federations are better clarified in Section 6.
- **Composable standalone MSaaS**: Not standalone applications, but software modules (e.g., weather, line of sight, etc) and data can be offered as an MSaaS. They can be composed to a service or their results can be inputted to a standalone MSaaS application.
- **Automatically composable MSaaS**: As the technology matures, MSaaS can become automatically discoverable and compatible with each other. This may allow clouds to compose sophisticated
MSaaS by integrating several composable MSaaS as illustrated in Figure 5.b. We call these kinds of MSaaS federations as Type 1 or Type3 federation. In Figure 5, both of the Type 2 and 3 federation examples are shown in an inter-cloud architecture. Please note that federations can be served also by private clouds (i.e., from one or multiple data centers in a single private cloud).

**Figure 4:** Various forms of clouds

- a. Public cloud
- b. Public inter-cloud
- c. Private cloud
- d. Partner cloud

**Figure 5:** Examples for Type 2 and 3 MSaaS federations

- a. An example for Type 2 MSaaS federation
- b. An example for Type 3 MSaaS federation
Militaries are planning to follow a stepped approach towards Type 3 federated MSaaS architecture. For example, the United States (US) Joint Staff (JS) plans to start serving JCATS as a standalone MSaaS application over their private cloud first, and later to decompose JCATS service modules and to create an environment for Type 3 MSaaS federations (Weter 2012). This architecture is also called as service oriented cloud. We expect that service oriented private clouds will join to service oriented partner clouds (i.e., the architecture that allows Type 3 federations) through gateways.

4 SECURITY THREATS TO MSaaS

In 1961, Baran categorized the communications networks into three broad classes as illustrated in Figure 6: centralized, decentralized, and distributed. He also defined survivability in his papers (Baran, 1964) as “the percentage of stations surviving a physical attack and remaining in electronic connection with the largest single group of surviving stations.” Based on this definition, he also concluded that centralized and decentralized architectures are vulnerable. The redundant links make the distributed architecture more survivable. In his papers, Baran considers a network of several hundreds of stations that can survive physical attacks. Although, the Internet has grown much larger scales, ArpaNet and later the Internet have evolved following the distributed network concept as described by Baran. The first impact of cloud computing is that it changes the physically distributed Internet architecture towards a functionally decentralized network. Although the network stays distributed, the stations will depend on data centers. This simply implies a less survivable architecture. That is a big security consideration especially nowadays, because cyber attacks are getting more common. The cloud providers tackle this issue by maintaining multiple data centers located remotely and employing self organization/healing mechanisms. Nevertheless, these techniques do not completely eliminate the risks due to the vulnerabilities in de-centralized architecture, and even create new security challenges. On the other hand, cloud computing also introduces some security advantages because it allows protection of data and software at fewer physical machines and platforms engineered and secured more carefully by experts. The bottom line is that cloud computing introduces new security challenges or exacerbates some already existing, as well as it introduces a few advantages for security.

Security and privacy are perhaps the biggest concern for the potential users before they buy the cloud computing offer. Therefore, there is a great interest in its security challenges and how to tackle with them. Cloud Security Alliance (CSA) is one of the organizations, and brings many stakeholders together to identify the threats to cloud computing and to promote the efforts to solve them. CSA listed the following as the top security threats to cloud computing (CSA 2010):

- Abuse and nefarious use of cloud computing: Both IaaS and PaaS providers are targeted with this
category of threats. By abusing relative anonymity behind cloud registration and usage models, malicious activities can be conducted with relative impunity. Future areas of concern include password and key cracking, distributed denial of service attacks (DDOS), botnet (i.e., robot network) command and control, launching dynamic attack points, hosting malicious data, building rainbow tables and CAPTCHA solving farms.

- Insecure interfaces and APIs: The security of cloud services depends upon the security of interfaces and APIs used by the users to access them.
- Malicious insiders: The malicious insider threat is amplified because of the abilities that a CSP employee can have.
- Shared technology issues: Guest operating systems may exploit the weaknesses of hypervisors to gain inappropriate level of control on the underlying platform. This is a threat that can be effective mainly against IaaS providers.
- Data loss or leakage: This is an increased threat in cloud computing due to the number and complexity of risks and challenges.
- Account or service hijacking: This is also an increased threat because it may create a base for an attacker to organize more sophisticated attacks possible in cloud architectures.
- Unknown risk profile: Since most of the underlying architecture, infrastructure and platform details are hidden from the users, it is not easy for the users to estimate the risks associated with them.

The list above gives the top threats, and it is not an exhaustive list. There are also other efforts for the identification and classification of security threats. In (Subashini 2011) (Jensen 2009), the security issues are surveyed with a structured and layered approach as the risks related to data security, network security, data locality, data integrity, data segregation, data access, authentication and authorization, data confidentiality, web application security, data breaches, vulnerability in virtualization, availability, backup, identity management and sign on process. Several papers investigate the security risks of cloud computing based on the cloud service models (i.e., IaaS, PaaS and SaaS) (Sandikkaya 2012). Additional vulnerabilities are introduced also by virtualization, which is a common technology preferred by CSP. These vulnerabilities can be exploited by various types of attacks, such as the following (Badger 2012)(Jasti 2010)(Lin 2011):

- VM Hopping: A VM may gain access to another VM at the same host machine.
- VM Mobility: When a VM is moved from one machine to another due to optimization or healing purposes, an adversary may use the differences in the security policies applied at different host machines.
- VM Sprawl: The number of VMs may grow rapidly due to bugs in VM management algorithms.
- VM Escape: A malicious code in a VM may gain access in hypervisor, which may be disastrous for a cloud.
- VM Hijack: By exploiting multi-tenancy, an adversary can gain access to the configuration files, through which another VM is accessed.

In the rest of this section, we elaborate on some of those risks that have specific dynamics related to MSaaS (Cayirci 2009b):

- **Privacy**: Users must rely on the CSP administration for the protection of their privacy and security of their proprietary data. The users may want to keep their data not accessible by any other user or may be willing to share it with a set of other users. In some cases the users may make their data releasable to the others only temporarily. Therefore, it may be desirable to use own data that is secured locally while receiving MSaaS, and sharing these data only with the users approved by the owner of the data. Data segregation, which requires that reliable encryption is always available, is also an issue related to privacy. Encryption brings up the requirement for a secure, efficient and practical key distribution scheme. It is not easy to design a secure key distribution scheme for such a dynamic and flexible environment.
• **Anonymity and Traffic Analysis**: Not only the private data owned by a particular user, but also the anonymity of the users may need to be protected. In addition, CSP should prevent users from unauthorized analysis of the network traffic to derive some information about the nature and the results of the simulation study. In some cases, CSP may need to ensure its users that it will not attempt to analyze their traffic.

• **Single Point to Attack and Single Point of Failure**: Although the centralization of services increases security by reducing the size of infrastructure to protect, that also creates points of gravitation for cyber and physical attacks. Services in a cloud can be a very attractive target for hackers. Moreover, when a system is hacked and/or fails, the impact is much bigger comparing to distributed computation approaches. Therefore, clouds require comprehensive intrusion prevention, detection and response techniques and fault tolerance measures. Both clustering and para-virtualization techniques are naturally fault tolerant. Nevertheless there are key services, and when one of them is compromised, all elements in the cloud can be affected.

• **Fate Sharing**: A large number of users share the same ecosystem although they may be from completely different background. Generic security policies may not fit the profiles of all subscribers. Moreover, treatment to a subset of users may affect all. For example, an FBI raid to a data-center in Texas for a fraud investigation related to a set of companies caused considerable damage in many companies not related with the investigation in 2009.

• **Large Databases and High Number of Clients**: The centralization of services also reduces the probability of configuration errors since there is no need for local system administrators. Therefore, at the first glance it looks like the points that can be exploited by the hackers are less comparing to the conventional approach. However, a cloud typically has huge number of users, much larger databases and a much higher number of processes. This creates new opportunities for denial of service attacks. For example a single malicious user that uses multiple identities, i.e., sybil attack, can attempt to consume as much system resources as possible. The cloud can be accessible from many different points by many users using generic and simple client devices, it is therefore not an easy task to detect an intruder. Huge databases, high number of users and services also make the detection of bugs, covert channels and bypasses very difficult tasks. Therefore, each module, component and their contents should be carefully verified and accredited before putting into service. This may increase the time required to modify a cloud or adding a new piece of data or software into it.

• **Denial of Service (DoS) Attacks in Medium Access Control (MAC) and Higher Layers of Networking Protocols**: Malicious intermediate nodes in the routes between the users/clients and centralized services can degrade the service quality. Although this kind of attacks is not specific to cloud computing and most of the services are run in a centralized server pool and therefore the communications are more restricted within a server pool, the users of clouds are more sensitive to DOS attacks because they are highly dependent on the centralized resources. In partner clouds, these risks are even higher. Resource centralization also makes the organization of such attacks easier. Examples for this kind of attacks are given in (Cayirci. 2009b).

• **Self-configuring, Self-optimizing, Self-monitoring and Self-healing Procedures**: Cloud computing requires algorithms for self configuration, self optimization, self monitoring and self healing. These processes may create opportunities to exploit for security attacks because of two reasons: First their implementation may have some bugs, and a hacker can use those bugs to gain access to a service. Second, a hacker may make these processes misbehave to degrade the services or to gain access to a service. For example a malicious user may change some system variables to show a system resource as busy, and make a load balancing algorithm assign no task to the system resource, which is available in reality. Another fact that exacerbates the security risks in a cloud is that a CSP may not know in which physical server the data and processes of a user reside at a given time due to self-organization and healing algorithms.
Cayirci

- **Vendor Lock-in:** A cloud user may end up as being highly dependent on a vendor for a service. When this service is for an important business process and based on some proprietary solutions, it may not be easy to change the vendor.

All the security challenges explained above are common challenges for cloud services. In addition to those, MSaaS, especially international military MSaaS architectures, has another major challenge, which is multi-level security (MLS). When single level security is provided, only users that have a security clearance equal to or higher than the security classification of a cloud can access the services of that cloud, and data that has higher classification level cannot be processed in the cloud. Cloud procedures may allow changing the security level of the cloud from time to time. However, a cloud can have only a single security classification level at a time. There can be virtual clouds in a single private or partner cloud with different security classifications. Each of these virtual clouds can be perceived as separate clouds that require separate servers, i.e., both hardware and software (i.e., user segregation). This approach can be called multiple single level security (MSL), and seems the only practical option in the beginning. It is also possible to connect multiple virtual clouds with different security classification through mail guards and firewalls that apply strict flow control mechanisms (e.g. data diodes).

Benefits of an MSaaS can be fully achieved when true multi level security (MLS) is realized. That means all users with different clearances can access a cloud, and an automated security mechanism can guarantee the following:

- A user cannot access a service that has higher security classification than his/her clearance. Please note that a service can be software, platform, infrastructure or data in MSaaS environments.
- A process can read and write an object if it has a classification level equal to the classification of the object.
- A process can read an object with a classification label of a lower level than its own clearance.
- A process cannot write to an object with a lower classification level to prevent leakage.
- A process cannot read or write to an object that has higher classification level, which is also related to the first item in this list.

A reliable flow control mechanism is required in order to meet these requirements. That can be achieved by labeling each data item, service and user with its security classification and clearance, and by implementing procedures for the automated security mechanism based on these labels (i.e., data and user segregation). Service labeling is a major challenge when it is an MSaaS because an MSaaS is characterized by huge number of users and very large databases. Clearance management for the users in such a dynamic environment with so many users is not an easy task.

Efficient sanitization techniques allow a reader to see parts of a document or data, which has security classification lower or equal to his/her clearance, although the classification of the overall document is higher. Sanitization is almost a “mission impossible”. First, sanitization requires an intelligence to decide which parts of a service cannot be accessed by a particular user. Second, it also requires an effective and scalable implementation for high number of users and large databases. Please note that some of the services in MSaaS, such as live and virtual simulations in military MSaaS are real time services and have very stringent latency constraints. Moreover, utilities in some applications make this task harder. For example, some documents may keep editing information to be capable for undoing changes later. Therefore, the parts deleted during sanitization can be undone if the mechanism misses those utilities.

There are also other risks related to sanitization. Information kept within a simulator includes for instance models, attributes and values. In relation to simulation, new factors complicate the problem of sanitization:

- The value as such of a particular object may be unclassified (e.g. a geographical position of a simulated entity), but derived values may be classified under certain conditions. For example, velocity of the object can be derived from its position updates. The average velocity may be unclassified; however, the breaking capabilities or turn rates of an aircraft (when avoiding threats) may be classified.
- Combinations of unclassified values may disclose classified data. For example, position infor-
mation of a strike package in a military simulation provides details about the doctrines that are used for specific operations.

- Data rates as such may provide classified information.

5 RISK, TRUST AND ACCOUNTABILITY IN MSAAS

The increased list of vulnerabilities and security threats exacerbate the risks that CSP and users have to take. The literature on risk is extensive with a very large scope of application areas. Therefore, we will not attempt to survey all the literature but refer to (DHS 2008), (Ezell 2010), (Kaplan 1980) and (Voeller 2010). In the seminal paper by Kaplan and Garrick, the distinctions between uncertainty, hazard and risk are clarified, and the absolute and perceived risk notions are explained. Risk analysis is defined as “an attempt to envision how the future will turn out if a certain course of action or inaction is taken” (Kaplan 1981). Three questions are answered during a risk analysis:

- A scenario \( s_i \) (i.e., What can go wrong?)
- The probability \( p_i \) of \( s_i \) (i.e., the probability that the scenario is realized)
- The consequence \( x_i \) of \( s_i \)

Hence, the risk \( R \) is a set of triplets that answers these three questions (i.e., \( R=\{s_i, p_i, x_i\}, i=1, 2, ..., N \) for \( N \) scenarios (i.e., \( N \) represents the number of all possible scenarios) (Kaplan 1981).

The risk that a cloud user has to accept is higher than a CSP. CSP usually keep the locations of their server farms and data centers confidential from users. Additionally, CSP have to prioritize the issues to solve when risks are realized. These uncertainties increase risk (Kaplan 1981) and imply that the users have to trust CSP. A user has to rely on the autonomic procedures of CSP for managing the infrastructure appropriately according to the users’ security dynamics, treating the users’ issues in a timely manner, detecting, recovering and reporting the security incidents accurately. Therefore, CSP have to be accountable to their users, and in many cases the users should be able to transfer their accountability to their CSP. However, since we expect that CSP may use services by the other CSP and even private clouds may be linked to partner clouds, the transfer of accountability may end up at a CSP whose accountability does not mean anything to the end user. It is clear that the nested nature of clouds makes accountability an extremely sophisticated issue and increases the risk for users.

Accountability should not be treated as an issue related only to security but also quality of service (QoS) and grade of service (GoS). The centralization of resources and sharing them increase the utilization. However, shared resources may be congested from time to time. Congestion control, service differentiation, user differentiation and prioritization are complex challenges especially for large clouds with high scalability requirements. The users need to be assured that their GoS and QoS requirements are fulfilled and their operations are not hampered due to congested cloud resources. Providing such an assurance, measuring and guaranteeing QoS/GoS are not trivial tasks.

The bottom-line is that accountability and trust are concepts required to be realized before potential users embrace cloud services. Therefore, “trust” has been extensively studied in the literature recently (Aljazzaf 2012) (Pearson 2012) (Rashidi 2012), and “trust as a service” is introduced to cloud business model.

Definition of trust can be a starting point for modeling it. In (Mayer et al. 1995) and (Roussaeau 1998), trust is defined as “the willingness of a party to be vulnerable to the action of another party based on the expectation that the other will perform a particular action important to the trusting party, irrespective of the ability to monitor or control the trusted party”. This definition does not fully capture all the dynamics of trust, such as the probabilities that the trustee will perform a particular action and will not engage in opportunistic behavior (Pearson 2012). There are also hard and soft aspects of trust (Wang 2008)(Singh 2009)(Osterwalder 2001). Hard part of trust depends on the security measures, such as authentication and encryption, and soft trust is based on things like brand loyalty and reputation. In (Ryan 2011), the authors introduced not only security but also accountability and auditability as elements which impact user’s trust in cloud computing, and can be listed among the hard aspects. In (Kandukiri 2009), Service Level Agreement (SLA) is identified as the only way that the accountability and auditability of a
CSP is clarified and therefore a CSP can make users trust them. The conclusion is that “trust” is a complex notion to define.

In (Rashidi 2012), the user trust to a CSP is related to the following parameters:

- **Data location**: Users know where their data are actually located.
- **Investigation**: Users can investigate the status and location of their data.
- **Data segregation**: Data of each user are separated from the others.
- **Availability**: Users can access services and their data pervasively at any time.
- **Privileged user access**: The privileged users, such as system administrators, are trustworthy.
- **Backup and recovery**: CSP has mechanisms and capacity to recover from catastrophic failures and not susceptible for disasters.
- **Regulatory compliance**: CSP complies with security regulations, certified for them and open for audits.
- **Long-term viability**: CSP has been performing the required standards for a long time.

The authors (Rashidi 2012) statistically analyze the results of a questionnaire answered by 72 cloud users to investigate the perception of the users on the importance of the above parameters. According to this analysis, backup and recovery produces the strongest impact on user’s trust in cloud computing followed by availability, privileged user access, regulatory compliance, long-term viability and data location. Their survey showed that data segregation and investigation have weak impact on user’s trust on cloud computing.

Chief information officers perceives the barriers for cloud adoption (Pearson 2012) as vendor lock-in (i.e., to be dependent on a vendor), cloud performance and availability, security and challenges in integrating internal and external services. According to another survey among 264 non information technology executives (non-IT) and 462 information technology executives, the barriers are security, regulatory risks, business case, adapting business processes, interoperability, lack of awareness, adjusting policies and building skill sets (Pearson 2012). These barriers are important in trust modelling because they are why the potential users trust or do not trust a CSP.

In (Cayirci 2013), a joint trust and risk model is introduced for MSaaS mash-ups. In this model, the real risk is defined as the risk that cannot be (or is not) eliminated by a CSP. If the part of the security and the service outage risk not eliminated by the CSP is lower than the user can take, then the cloud service is viable for the user. For this evaluation, the risk is perceived as the probability that a security threat is realized or the probability that a service outage occurred, and trust as the probability that the CSP can eliminate a security risk when realized or the probability that the CSP can recover from a service outage before it hampers the user’s operations. The probabilities for risk and trust are determined based on historic data. For trust negative and positive performances are differentiated and the freshness of the data is taken into account.

### 6 SERVICE COMPOSITION FOR MSAAS

As explained in Section 5, risks get higher and more difficult to analyze in nested cloud architectures (i.e., inter-cloud, service mash-ups and partner clouds). Composing an MSaaS from the services provided by multiple clouds is a challenging task. The risk and trust relations among the clouds and services contributing to a composite MSaaS are complex.

Before, elaborating on the schemes for MSaaS composition, we first would like to clarify the following terms: federation, service mash-up, multiple cloud service, inter-cloud service and composite service. These terms are being used interchangeably in the literature, although they may have different meanings in different context. We will use the term “MSaaS federation” for a composite MSaaS, and the term “federate” for each service that the federation is composed of. Federation has a different meaning in cloud computing from modelling and simulation (M&S). In cloud computing, the term “federation” is used not only for federating models but also for infrastructure or platforms, and therefore a federation may also mean a cloud service that integrates various resources in the form of IaaS (e.g., memory, processor time,
etc) from multiple datacenters (Buyya 2010)(Cayirci 2013b)(Toosi 2011). On the other hand, in MSaaS domain, federations integrate multiple MSaaS either in standalone application or service module form. We categorize MSaaS federations into four broad classes as the following:

- **Type 0**: Multiple MSaaS in the form of standalone applications located in the same datacenter are federated (Toosi 2011).
- **Type 1**: Multiple service modules located in the same datacenter are composed into a composite MSaaS following service oriented architecture (SOA).
- **Type 2**: Standalone applications from various datacenters that may belong to different clouds are integrated into a seamless software federation by an MSaaS provider (Cayirci 2013b).
- **Type 3**: Not standalone applications, but software modules and data located in multiple datacenters are brought together for dynamically configured service oriented software federations. The datacenters that provide service modules may belong to different clouds (Cayirci 2013b).

<table>
<thead>
<tr>
<th>Nature of Federates</th>
<th>Intra datacentre</th>
<th>Inter datacentre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone applications</td>
<td>Type 0</td>
<td>Type 2</td>
</tr>
<tr>
<td>Services composed by using SOA</td>
<td>Type 1</td>
<td>Type 3</td>
</tr>
</tbody>
</table>

Lets clarify the difference between Type 2 and 3 by using a military MSaaS (Cayirci 2009a)(Tolk 2012) example depicted in Figure 5. Joint theater level simulation (JTLS), joint conflict and tactical simulation (JCATS) and virtual battle space (VBS2) are three widely used combat modelling software (Cayirci 2009a). They work at different resolution levels. JTLS works in theater level to simulate scenarios with large units in very large areas. On the other hand, VBS2 is a very high resolution model that also provides three dimensional visualization services for relatively limited number of entities in a limited space. JCATS is in between of these two models. JTLS, JCATS and VBS2 can be integrated into a Type 2 federation by using a runtime infrastructure (RTI) as defined in high level architecture (HLA) (Cayirci. 2009a). Please note that the co-location of JTLS and VBS in Figure 5.a is just an example. Any of these federates can be co-located in a datacenter. Alternatively, services provided by RTI can be distributed in multiple datacenters.

In Type 3, instead of complete combat models, software modules that models different aspects of theatre are federated. In other words, services from multiple data centers are composed into a composite MSaaS. For example a software module in one datacenter can make the line of sight calculations, and another one from another datacenter computes the effects of weather on the results. Configuring a Type 3 federation is definitely a more challenging task.

For the realization of a Type 2 or 3 inter-datacenter federation, the first task is the configuration of the federation, which is the discovery and the selection of the services to be integrated for a given simulation task. When the federation is being configured, all the policy and physical constraints, performance expectations and interoperability requirements need to be satisfied. Among these, interoperability requirements (Davis. 2007) are special challenge for MSaaS. In an inter-datacenter, there can be multiple MSaaS that can provide the same service. Therefore, determining the federates interoperable with each other and selecting the set that fits best to the constraints and performance expectations is a new challenge. We call this challenge as MSaaS inter-datacenter federation (MIF) configuration (MIFC).

MIFC for Type 2 or Type 3 federations can be managed automatically by self configuration algorithms or manually. In any case, MIFC is proved as an NP Complete problem (Cayirci 2013b), which simply means that if the number of alternative MSaaS for each service module in a MSaaS federation or the number of service modules is high, it is not possible to find the optimum solution in a reasonable time. In (Cayirci 2013b) a scheme for the optimization and a heuristic for feasible solutions are introduced, and scalability of these schemes is analyzed.

Type 3 MSaaS federations exhibit special challenges which are more original for the M&S community. Our first observation is that technologies like distributed interactive simulation (DIS) and high level
architecture (HLA) do not suffice for integrating the federates in a Type 3 MSaaS federation because of several reasons. One of these reasons is that in Type 3 MSaaS federations an attribute of an entity can be updated based on the computations made by several federates. In DIS and HLA, an attribute of an entity is normally owned by a single federate. Another important observation with Type 3 federations is about “service fan-in and fan-out.” The number of federates in a large Type 2 federation is relatively reasonable (i.e., typically between 2 and 15 for military federations) comparing to Type 3 federations. For Type 3 federations, service fan-out is expected to be higher in the order of magnitude.

However, service composition as in Type 3 MSaaS federation concept has been extensively studied for the last decade, which may be very useful also for Type 3 MSaaS federation composition. The service composition schemes in the literature can be categorized based on different criteria, such as dynamic versus static, automatic versus manual or based on the method used. The later approach is taken in (Kapitsaki 2007) to classify the service composition techniques as following:

- Artificial Intelligence (AI) Planning: The schemes that fall in this category investigates the possible actions to take the system from its current state to the desired goal. AI Planning based methods are further categorized as finite state machines, situation calculus, hierarchical task networks and Petri Nets.
- Semantic Web Approach: Semantic approaches attempt not only to identify the structure of the messages exchanged among services but also to interpret their content. They are also further classified as semantic annotation, rule based approaches and knowledge based composition.
- Middleware Approach: This class relies on middleware that enables discovery and invocation of services and has the following sub groups: mobile agents, input/output dependency and policy based approaches.
- Others: Composition based on patterns and manual service composition based on modeling fall in the others category.

The detailed descriptions of all the sub categories listed above are in (Kapitsaki 2007).

Recently service composition in cloud computing context has also attracted many researchers. An implementation approach for inter-cloud service combination is introduced in (Tao 2012). The challenges and research questions for trustworthy service selection and composition are investigated in (Hang 2011). Although the research in this field has been extensive, pragmatic solutions are still missing. Cloud computing introduced many new notions, and therefore the technological gaps in the field of Type 3 MSaaS federation configuration are big. For example, due to self configuration, optimization and healing mechanism, services may migrate during execution. Therefore, new dynamic routing schemes, content (or information) centric networking and naming schemes, jitter resilient algorithms for real-time simulators, congestion control schemes for such environments and an analytical framework that models the dynamics of the architecture are needed.

7 CONCLUSION

MSaaS is an emerging approach for M&S following the latest trends in information technologies. It promises many advantages, such as rapid elasticity, ease in technical administration and licensing, better utilization, pay per use, and therefore considerable cost reduction. However, it also introduces many challenges including security, privacy, accountability, risk and trust management and service composition. Industry and academia have tackled with these challenges for almost a decade and solved many of them at least in theory. Therefore, we observe more and more M&S applications deployed as MSaaS and militaries start considering MSaaS as their next generation architecture. However, the number of unsolved challenges, most notably for some ambitious deployment plans, is not negligible, and more time is needed before especially service oriented MSaaS federations are realized.
ACKNOWLEDGEMENT

This work was supported by EU FP7 Accountability for Cloud and Other Future Internet Services (A4Cloud) Project.

REFERENCES


**AUTHOR BIOGRAPHY**

**ERDAL CAYIRCI** graduated from Army Academy in 1986 and from Royal Military Academy, Sandhurst in 1989. He received his MS degree from Middle East Technical University, and a PhD from Bogazici University both in computer engineering in 1995 and 2000, respectively. He retired from the Army when he was a colonel in 2005. He was an Associate Professor at Istanbul Technical University, Yeditepe University and Naval Sciences and Engineering Institute between 2001 and 2005. Also in 2001, he was a visiting researcher and lecturer at the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is currently Head, CAX Support Branch in NATO’s Joint Warfare Center in Stavanger, Norway, and also a professor in the Electrical Engineering and Computer Science Department of University of Stavanger. His research interests include cloud computing, modelling and simulation, security in ad hoc networks, sensor networks and mobile communications. Professor Cayirci has acted as an editor of the journals IEEE Transactions on Mobile Computing, AdHoc Networks (Elsevier Science) and ACM/Kluwer Wireless Networks. He received the “**2002 IEEE Communications Society Best Tutorial Paper**” Award, the “**Fikri Gayret**” Award from Turkish Chief of General Staff in 2003, the “**Innovation of the Year**” Award from Turkish Navy in 2005 and the “**Excellence**” Award in ITEC 2006. He is author of two textbooks, “Security in Wireless Ad Hoc and Sensor Networks” and “Computer Assisted Exercises: A Reference Guide,” both published by John Wiley & Sons. His email is <erdal.cayirci@uis.no>.