Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

MODELING AND SIMULATION AS A CLOUD SERVICE: A SURVEY

Erdal Cayirci

Electrical Engineering & Computer Science Department
University of Stavanger
Stavanger, 4036, 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 2011). There are already M&S software offered as cloud services 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, Valipour 2009) 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 – automatic provisioning of computing capabilities*
- *Broad network access –access through standard mechanisms via thin or thick client platforms*
- *Resource pooling –multitenant model*
- *Rapid elasticity –elastically and automatically provisioned and released capabilities*
- *Measured service – measurement transparency for both the providers and consumers of services.*

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, and provided solutions for many of them. 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

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

related challenges of MSaaS are investigated in Section 4. Accountability, risk and trust are often related to security and especially privacy in 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. 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. 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 users noticing it because a VM can be migrated (i.e., snapshotting and teleportation) from one host machine to another while they are running. 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. However, hardware virtualization offers many advantages as listed above for cloud service providers (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. 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). 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. Not only hardware and software licenses but also software processes (i.e., a program that runs) can be shared, 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) shown in Figure 1 to the users (Internet 2013):

- Infrastructure as a Service (IaaS): computers and other resources like data storage
- Platform as a Service (PaaS): 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): application software including the underlying infrastructure and platform as required

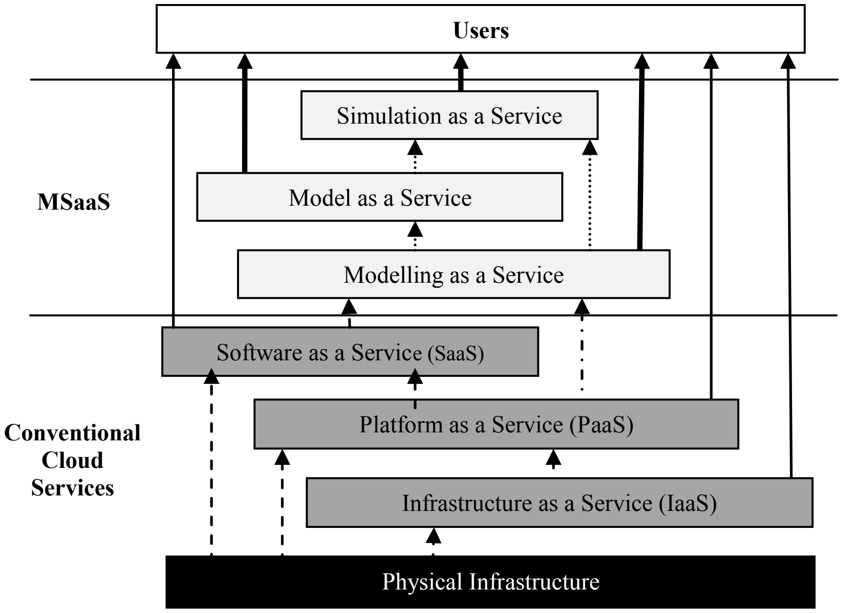

Users
MSaaS
Simulation as a Service
Model as a Service
Modelling as a Service
Conventional Cloud Services
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Physical Infrastructure

Figure 1: The inter-relations of cloud services including MSaaS

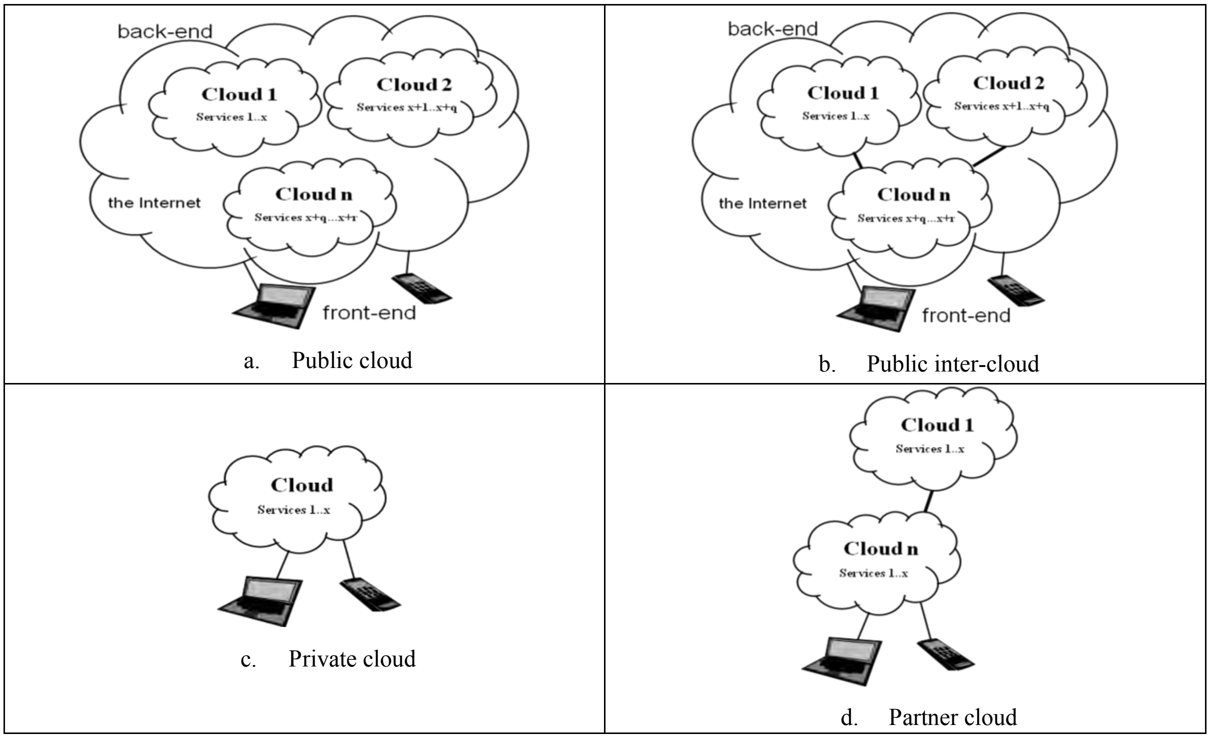
There are also many other service types introduced in literature, such as, Network as a Service (NaaS), Trust as a Service, Authorization as a Service (Laborde 2013). These are derivations of PaaS and SaaS in various combinations and forms. Modelling and Simulation as a Service (MSaaS) can be perceived as one of these derivatives. MSaaS is in essence a special form of SaaS. The inter-relations between MSaaS, SaaS, PaaS and IaaS are depicted in Figure 1. 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, in another CSP or in their own enterprise. They may develop models by using modelling as a service, use previously developed models to run simulations in their enterprise or run simulations by using simulation as a service.

3 MSAAS ARCHITECTURES

Let's elaborate on the architectural components and deployment scenarios of MSaaS. 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 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 cloud architecture can be introduced in various forms listed below and depicted in Figure 2:

- 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 form an **inter-cloud** (i.e., cloud of clouds), also called as service mash-ups or multi-clouds.
- A **private cloud** provides services to an organization through an intranet.
- A community cloud may open its services not to public but to a community of interest from multiple organizations through the Internet or a special inter-organizational network.
- Private clouds can be connected to each other to form a **partner cloud**.
- **Hybrid clouds** are any combination of the clouds listed above.



a. Public cloud

b. Public inter-cloud

c. Private cloud

d. Partner cloud

Figure 2: Various forms of clouds

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 in the Internet.
- Federated standalone MSaaS applications: Standalone MSaaS applications can be federated. These applications can be from the same data center or multiple data centers.
- Composed MSaaS: Not standalone applications, but software modules and data can be offered as an MSaaS that can be integrated into a composite service.
- Automatically composed MSaaS: As the technology matures, composed MSaaS can become automatically discoverable and composable with each other.

4 SECURITY THREATS TO MSAAS

In 1961, Baran categorized the communications networks into three broad classes as 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.*" He also concluded that centralized and decentralized architectures are vulnerable. The redundant links make the distributed architecture more survivable. 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 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 of centralization, and create new security challenges. On the other hand, cloud computing also introduces some security advantages because it allows protection of data at fewer locations secured more carefully by experts.

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) listed the following as the top security threats to cloud computing (CSA 2010):

- Abuse and nefarious use of cloud computing: By abusing relative anonymity behind cloud registration and usage models, malicious activities can be conducted with relative impunity.
- 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.
- 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.
- 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.

The list above gives the top threats, and it is not exhaustive. 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 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:

- **Privacy**: Users must rely on the CSP administration for the protection of their privacy and security of their proprietary 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.
- **Anonymity and Traffic Analysis**: Not only the private data owned by a particular user, but also the anonymity may need to be protected. In addition, CSP should prevent users from unauthorized analysis of the network traffic to derive information about the results of a simulation based study.
- **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. When a system is hacked and/or fails, the impact is much bigger comparing to distributed architecture.
- **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.
- **Large Databases and High Number of Clients**: Huge number of users, larger databases and higher number of processes create new opportunities for denial of service attacks. 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 intruders, bugs, covert channels and bypasses.
- **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, the users of clouds are more sensitive to DOS attacks because they are highly dependent on the centralized resources.
- **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. In addition to these, CSP may not know in which physical server the data and processes reside at a given time due to self-organization and healing algorithms.
- **Vendor Lock-in:** A cloud user may end up as being highly dependent on a vendor for a service.

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. 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. 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 secure flow control and sanitization. The cloud specific challenges of flow control and sanitization are explained in (Cayirci 2011).

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 (Ezell 2010, Kaplan 1980). 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 complicated issue and increases the risk for users.

Accountability should not be treated as an issue related only to security but also QoS and 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.

In (Mayer et al. 1995, 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 to 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 (Kandukuri 2009), Service Level Agreement (SLA) is identified as the only way that the accountability and auditability of a CSP is clarified. 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 to recover from catastrophic failures.

- Regulatory compliance: CSP complies with security regulations, certified 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 2013a), 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.

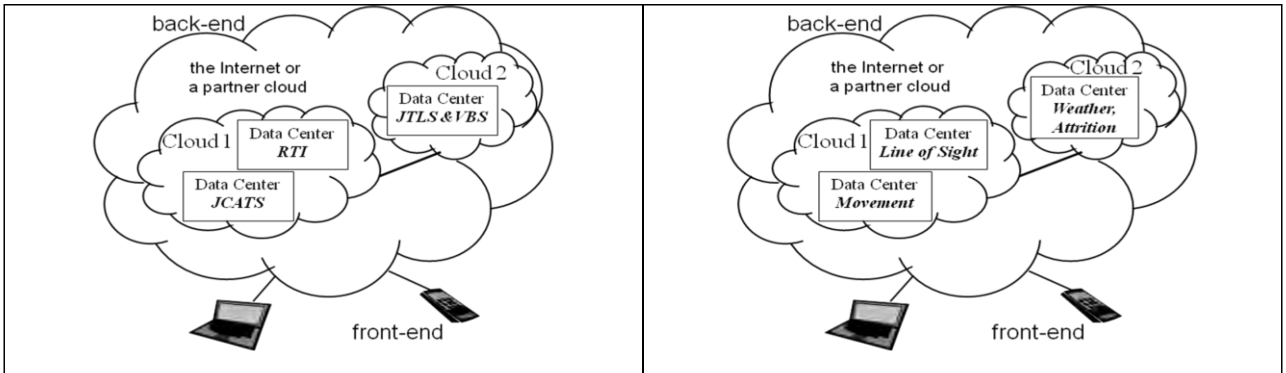
Before elaborating on the schemes for MSaaS composition, we first would like to clarify the following terms: federation, service mash-up, multiple cloud service, inter-cloud service and composite service. These terms are being used interchangeably in the literature, although they may have different meanings 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, Singhal 2013, 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 in Table 1:

- Type 0: Federation of standalone applications located in the same datacenter (Toosi 2011)
- Type 1: Composite MSaaS of service modules located in the same datacenter (SOA)
- Type 2: Federation of standalone applications from multiple datacenters (Cayirci 2013b)
- Type 3: Composite MSaaS of service modules from multiple datacenter (Cayirci 2013b).

Let's clarify the difference between Type 2 and 3 by using a military MSaaS (Cayirci 2009, Tolk 2012) example depicted in Figure 3. Joint theater level simulation (JTLS), joint conflict and tactical simulation (JCATS) and virtual battle space (VBS2) are three widely used combat modelling software (Cayirci 2009). 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. 2009). Please note that the co-location of JTLS and VBS in Figure 3.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.

Table 1:. Types of MSaaS Federations

Nature of Federates	Intra datacentre	Inter datacenter
Standalone applications	Type 0	Type 2
Services composed by using SOA	Type 1	Type 3



a. An example for Type 2 MSaaS federation

b. An example for Type 3 MSaaS federation

Figure 3: Examples for Type 2 and 3 MSaaS federations

In Type 3, instead of standalone combat models, software modules that model 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 an 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 investigate 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.

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

Armbrust M., A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. 2010. A view of Cloud computing. *Communications of the ACM, vol. 53, no. 4, pp. 50–58*.

Badger L., T. Grance, R.Patt-Corner and J.Voas. 2012. Draft Cloud Computing Synopsis and Recommendations. *National Institute of Standards and Technology, Special Publication 800-146*.

Baran, P., 1964.On Distributed Communications Networks. *IEEE Transactions on Communications Systems*, Vol. 12, Issue 1, pp. 1-9.

Buyya R., R. Ranjan, R.N. Calheiros. 2010. InterCloud: Utility-oriented federation of Cloud computing environments for scaling of application services. *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10)*, pp. 13–31.

Cayirci E., and D. Marincic. 2009. *Computer Assisted Exercises and Training: A Reference Guide*. *John Wiley*.

Cayirci, E., C. Rong. Intercloud for Simulation Federations. 2011. *The Second International Workshop on Cloud Computing Interoperability and Services (Intercloud 2011)*.

Cayirci, E. 2013a. A Joint Trust and Risk Model for MSaaS Mashups. *Winter Simulation Conference (WinterSim 2013)*.

Cayirci, E. 2013b. Configuration Schemes for Modelling and Simulation as a Service Federations. *Simulation Transactions of the Society for Modelling and Simulation*. (to appear)

Cloud Security Alliance (CSA). 2010. Top Threats to Cloud Computing. *https://cloudsecurityalliance.org*.

Davis K. and Tolk A., "Observations on new developments in composability and multi-resolution modeling," Proceedings of the 2007 IEEE Winter Simulation Conference, December 2007.

Ezell, B.C., S.P.Bennet, D. Von Winterfeldt, J.Sokolowski and A.J.Collins. 2010. Probabilistic Risk Analysis and Terrorism Risk. *Risk Analysis*, Vol. 30, No. 4, pp. 575-589.

Jasti, A., P.Shah, R.Nagaraj and R.Pendse. 2010. Security in Multitenancy. *IEEE International Carnahan Conference on Security Technology*.

Hang, C-W. and M.P. Singh. 2011. Trustworthy Service Selection and Composition. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 6, No. 1, pp. 5:1–5:17.

Huerta-Capena G. and D. Lee. 2010. A virtual cloud computing provider for mobile devices. *ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond*.

Jensen M., J.Schwenk, N.Gruscka and L.L.Iacono. 2009. On Technical Security Issues in Cloud Computing. *IEEE International Conference on Cloud Computing*, pp. 109-116.

Johnson H. and Tolk A., "Evaluating the Applicability of Cloud Computing Enterprises in Support of Next Generation of Modelling and Simulation Architectures," *Spring Simulation Multi-Conference*, April 2013.

Kandukuri, B.R., R. Paturi V, A. Rakshit. 2009. Cloud Security Issues. *2009 IEEE International Conference on Services Computing*.

Kapitsaki, G., D.A. Kateros, I.E.Foukarakis, G.N.Prezerakos, D.I.Kaklamani and I.S. Venieris. 2007. Service Composition: State of the art and future challenges. *16th IST Mobile and Wireless Communications Summit*, pp. 1-5.

Kaplan S. and B.J. Garrick. 1981. On The Quantitative Definition of Risk. *Risk Analysis*, Vol. 1, No. 1, pp. 11-27.

Laborde R., Barrere F. and Benzekri A., "Toward Authorization as a Service: A Study of the XACML Standard," *Spring Simulation Multi-Conference*, April 2013.

Lin, Z. 2011. Virtualization Security for Cloud Computing Services. *IEEE International Conference on Cloud and Service Computing*.

Mayer, R. C., J. H. Davis, and F. D. Schoorman. 1995. An integrative model of organizational trust. *The Academy of Management Review*, Vol. 20, No. 3, pp. 709–734.

Osterwalder, D. 2001. Trust Through Evaluation and Certification. Social Science Computer Review. Sage Publications, Inc., 19(1), pp 32-46.

Pearson, S. 2012. Privacy, Security and Trust in Cloud Computing. *Privacy and Security for Cloud Comptutting, S.Pearson and G.Yee (eds.), Computer Communications and Networks, Springer*.

Rashidi, A., and N. Movahhedinia. 2012. A Model for User Trust in Cloud Computing. *International Journal on Cloud Computing: Services and Architecture(IJCCSA)*,Vol.2, No.2.

Rosetti M. And Chen Y., "Cloud Computing Architecture for Supply Chain Network Simulation," *Winter Simulation Conference*, December 2012.

Rousseau, D., S. Sitkin, R. Burt, C. Camerer. 1998. Not so Different after All: a Cross-discipline View of Trust. Academy of Management Review, 23(3), pp 393-404.

Ryan, K. L. K., P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, B. S. Lee. 2011. TrustCloud: A Framework for Accountability and Trust in Cloud Computing. *2nd IEEE Cloud Forum for Practitioners (ICFP)*.

Sandikkaya, M.T. and A.E. Harmanci. 2012. Security Problems of Platform as a Service. *31st International Symposium on Reliable Distributed Systems*.

Satyanarayanan, M., P. Bahl, R. Caceres and N. Davies. 2009. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, Vol. 8, pp. 14-23.

Singh, S., C. Morley. 2009. Young Australians' privacy, security and trust in internet banking. In: Proceedings of the 21st Annual Conference of the Australian Computer-Human interaction Special interest Group: Design: Open 24/7.

Singhal, M., S. Chandrasekhar, G. Tingjian, R., S. Sandhu, R. Krishnan, G-J. Ahn, E. Bertino. 2013. Collaboration in Multicloud Computing Environments: Framework and Security Issues. *IEEE Computer Magazine February 2013*, pp 76-84.

Shi C., V. Lakafosis, M.H.Ammar and E.W.Zagura. 2012. Serendipity: enabling remote computing among intermittently connected mobile devices. *ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

Subashini, S. and Kavitha, V. 2011. A survey on security issues in service delivery models of cloud computing. *Elsevier Journal of Network and Computer Applications*, Vol. 34, Issue 1, pp. 1-11.

Tao, J., D. Franz, H. Marten and A. Streit. 2012. An Implementation Approach for Inter-Cloud Service Combination. *International Journal on Advances in Software*, Vol. 5, No. 1&2, pp 65-75.

Tolk, A. 2012. Engineering Principles of Combat Modeling and Distributed Simulation. John Wiley & Sons.

Toosi A.N., R.N.Calheiros, R.K.Thulasiram, R.Buyya. 2011. Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud Environment. *HPCC 2011*.

Wang, Y., K.-J. Lin. 2008. Reputation-Oriented Trustworthy Computing in E-Commerce Environments. Internet Computing, IEEE, 12(4), pp 55–59.

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 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 email is <erdal.cayirci@uis.no>.