Dynamic Collaboration and Secure Access of Services in Multi-Cloud Environments

Muhammad Kazim

A Thesis Submitted in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

College of Engineering and Technology

May 2019

Declaration

The study outlined in this dissertation was carried out in the College of Engineering and Technology at the University of Derby, under the supervision of Professor Lu Liu. This is to declare that the work stated in this thesis was done by the author unless otherwise is indicated, and no part of this thesis has been submitted in a thesis form to any other university or similar institution. Parts of this thesis have previously appeared in the papers listed in the list of publications.

Muhammad Kazim

University of Derby

2019

Acknowledgements

First and foremost, I am extremely grateful to my supervisor Professor Lu Liu for his support and guidance throughout my PhD research. He has been a great mentor and a source of motivation during my PhD journey. I will also like to thank my co-supervisor Dr. Shao Ying Zhu for her guidance in my research. I am also thankful to my previous supervisor Dr. David Evans (Late) who greatly helped me in my first year of research at Derby. Moreover, I will like to thank the College of Engineering and Technology, University of Derby for funding my PhD thesis.

I will also like to acknowledge and say special thank you to my parents, wife, brother, sisters and friends who have provided me great support during my four years of research especially my parents who have been a source of inspiration throughout my life.

List of Publications

1. M. Kazim, L. Liu, “A Framework for Orchestrating Secure and Dynamic Access of IoT Services in Multi-Cloud Environments”, *IEEE Access Journal*, 2018. [IF: 3.557]

2. M. Kazim, S. Y. Zhu, “A Survey on Top Security Threats in Cloud Computing,” *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 6, no. 3, 2015.

3. M. Kazim, D. Evans, “Threat Modelling for Services in Cloud Computing”, *10th IEEE International Symposium on Service-oriented System Engineering (SOSE),* Oxford, 2016.

4. M. Kazim, S. Y. Zhu, “Virtualization Security in Cloud Computing”, *Guide to Security Assurance for Cloud Computing,* Springer, 2015.

5. F. Ahmad, M. Kazim, A. Adnane, A. Awad, “Vehicular Cloud Networks: Architecture, Applications and Security Issues”, *8th IEEE/ACM International Conference on Utility and Cloud Computing, UCC,* Cyprus, 2015.

6. F. Ahmad, M. Kazim, A. Adnane, “Vehicular Cloud Networks: Architecture and Security”, *Guide to Security Assurance for Cloud Computing,* Springer, 2015

Abstract

The cloud computing services have gained popularity in both public and enterprise domains and they process a large amount of user data with varying privacy levels. The increasing demand for cloud services including storage and computation requires new functional elements and provisioning schemes to meet user requirements. Multi-clouds can optimise the user requirements by allowing them to choose best services from a large number of services offered by various cloud providers as they are massively scalable, can be dynamically configured, and delivered on demand with large-scale infrastructure resources. A major concern related to multi-cloud adoption is the lack of models for them and their associated security issues which become more unpredictable in a multi-cloud environment. Moreover, in order to trust the services in a foreign cloud users depend on their assurances given by the cloud provider but cloud providers give very limited evidence or accountability to users which offers them the ability to hide some behaviour of the service.

In this thesis, we propose a model for multi-cloud collaboration that can securely establish dynamic collaboration between heterogeneous clouds using the cloud on-demand model in a secure way. Initially, threat modelling for cloud services has been done that leads to the identification of various threats to service interfaces along with the possible attackers and the mechanisms to exploit those threats. Based on these threats the cloud provider can apply suitable mechanisms to protect services and user data from these threats. In the next phase, we present a lightweight and novel authentication mechanism which provides a single sign-on (SSO) to users for authentication at runtime between multi-clouds before granting them service access and it is formally verified. Next, we provide a service scheduling mechanism to select the best services from multiple cloud providers that closely match user quality of service requirements (QoS). The scheduling mechanism achieves high accuracy by providing distance correlation weighting mechanism among a large number of services QoS parameters.

In the next stage, novel service level agreement (SLA) management mechanisms are proposed to ensure secure service execution in the foreign cloud. The usage of SLA mechanisms ensures that user QoS parameters including the functional (CPU, RAM, memory etc.) and non-functional requirements (bandwidth, latency, availability, reliability etc.) of users for a particular service are negotiated before secure collaboration between multi-clouds is setup. The multi-cloud handling user requests will be responsible to enforce mechanisms that fulfil the QoS requirements agreed in the SLA. While the monitoring phase in SLA involves monitoring the service execution in the foreign cloud to check its compliance with the SLA and report it back to the user. Finally, we present the use cases of applying the proposed model in scenarios such as Internet of Things (IoT) and E-Healthcare in multi-clouds. Moreover, the designed protocols are empirically implemented on two different clouds including OpenStack and Amazon AWS. Experiments indicate that the proposed model is scalable, authentication protocols result only in a limited overhead compared to standard authentication protocols, service scheduling achieves high efficiency and any SLA violations by a cloud provider can be recorded and reported back to the user.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Overview

This chapter presents the motivation, context, aim and objectives of this research. These include introducing the concept and benefits of multi-clouds, as well as the key challenges in achieving multi-cloud communication and security. Moreover, this chapter also details the key research contributions of this research, including the development of a novel model for achieving dynamic and secure collaboration among multi-clouds. Finally, this chapter presents an overall organisation of this thesis.

1.2 Research Context

Cloud computing is a technology that offers various services ranging from infrastructure to storage, computation, software and application over the internet. However, the variety and proliferation of services offered by the cloud provider raises several challenges. These challenges include portability issues of SaaS on various IaaS and PaaS platforms, interoperability of distributed SaaS applications on various cloud platforms, PaaS dealing with the heterogeneity of protocols to support cloud service interactions, and the requirement of geodiverse platforms [1]. The concept of multi-clouds was introduced to solve these challenges. A multi-cloud environment is dependent on multiple clouds, so a user can be reliant on cloud services from AWS, Microsoft, or OpenStack which are communicating. Among all cloud environments including public, private, hybrid, heterogeneous and hybrid clouds the co-operation between multi-clouds has been desired by many cloud users and executives. A survey from 451-Microsoft [2] mentions that organizations are increasingly seeking cloud providers that can deliver a wide range of service, and be their brokers as a single point of contact through which they can access services from other providers. It is predicted that by the end of 2019 90% of the UK businesses will be using at least one cloud service [3].

The motives for multi-cloud for cloud provider can vary from dealing with a peak in service requests, having backup servers to diminish downtime scenarios and enhancing its own offers to get a market competitive edge. The main incentive for organizations in multi-clouds can be to optimise cost by having better access to services, and the ability to act as an intermediary to provide access to resources. While users can be tempted to use multi-clouds as it gives them benefits like consume services not delivered by their own cloud provider, using services irrespective of location and share resources with users in other clouds. In this thesis, we refer to cloud making a connection request to an external cloud as a "local cloud". While a "foreign cloud" is referred to as a cloud to which the user needs access and collaboration has to be established with it.

Multi-clouds offer greater agility, innovation, and more intense collaboration and they are predicted to become an industry norm, however, managing service orchestration is still an open issue [4]. In a multi-cloud environment, providers spread out their services across multiple cloud providers which changes the traditional cloud landscape. Therefore, advanced development frameworks are required that can reduce companies time-to-market and to keep cloud services running smooth. Along with the service orchestration issues in multi-clouds, many security concerns are also related to their adoption and application.

Cloud providers in a multi-cloud environment spread out their services across multiple clouds which changes the traditional cloud landscape and brings more security challenges. Cyber-attacks, in general, have been on the rise in the past few years, and the adoption of multi-clouds provides attackers with an even larger attack surface to gain access to sensitive data, applications, services and infrastructure [5]. This poses a threat not only to companies and enterprises using the cloud but also to government and security organizations using cloud computing. The parties involved in compromising cloud can vary from hackers to cloud administrators as well as malicious users, service providers and cloud providers.

1.3 Problem Description

Single cloud data centre which is the standard cloud computing model can pose several challenges to providers and users such as unavailability of service to thousands of customers if a single service goes down, and lower throughput due to high traffic. Multiple clouds provide the ability to run workloads on best-suited platforms, avoiding the need to migrate legacy applications and creating redundancy to avoid vendor lock-in [6]. Multi-clouds offer a way for dynamic collaboration between various clouds in a way that there is no former agreement between participating clouds and collaboration is established at runtime according to requirements [7].

Service orchestration is a key challenge in multi-clouds. The collaborating multi-clouds are hosted by different companies and their heterogeneous policies, security rules and internal network setup can be really different. The existing federated authentication solutions across multi-clouds are not suitable for dynamic collaboration as they are expensive, while some solutions require credential conversion across different cloud realms. Centralised protocols with fixed collaboration relationships are not suitable due to delay in processing large number of requests and huge costs incurred due to this delay. Therefore, a new scheme is required that could allow dynamic multi-party collaborating among different clouds.

Authentication between multi-clouds is the initial stage in setting up communication across them. Collaborating multi-clouds are independent and usually belong to different security realms which makes authentication a very complex problem. Having different security policies and the need of credential conversion across different realms requires calling a chain of middleware services to perform authentication. The basic authentication solutions that exist for traditional networks fail to meet the need for a dynamic collaboration of clouds and services in multi-cloud. This research focuses on providing a novel mechanism for dynamic authentication between multi-clouds in varying security realms by setting up multi-party collaboration sessions that neither require credential conversion nor the series of invocations to setup authentication in advance to clouds interactions. This leads to improving the scalability of the system by handling a large number of requests is less time and thus incurring lower costs compared to the state-of-the-art authentication protocols.

After a service has been moved to a foreign cloud, the cloud users have no mechanism to verify that the service they are using is trustworthy and neither do they have insights on what is happening with their data being handled by services. In order to trust the cloud services, users depend on their assurances given by cloud provider. Cloud providers give very limited evidence or accountability to users which offers them the ability to hide some behaviour of the service. These issues necessitate the need to develop solutions for multi-clouds that can facilitate multi-cloud collaboration and provide guarantees to the user that the software or service running on a foreign cloud node satisfies client quality of service (QoS) and does not violate the agreed service level agreements (SLAs). Therefore, a key challenge is to develop solutions for multi-cloud that can enable the users to select most suitable service provider among multiple foreign clouds according to their requirements through scheduling and to aid resource provisioning and placements in the foreign cloud. The scheduling algorithm must support efficient discovery according to the characteristics of services advertised by foreign clouds and user QoS requirements with high accuracy, and also select an optimum service in case there is no match of advertised services with QoS requirements. Moreover, another key challenge is to ensure that services in foreign cloud are compliant with the service level agreement (SLA) between user and cloud provider. While the user is accessing service in foreign cloud, the functional and non-functional QoS requirements defined in the SLAs should be managed by the proposed system.

1.4 Aim and Objectives

The aim of this research is to design and develop a novel mechanism that ensures secure and dynamic collaboration across the multi-clouds by managing their orchestration, authentication, and scheduling on the basis of service level agreements.

The primary objectives of this research are as follows:

1. To investigate and identify the current state-of-the-art mechanisms in a multi-cloud collaboration, orchestration, authentication and SLA management.
2. To design and assess a comprehensive threat model for the cloud services in the face of untrusted cloud nodes, malicious users, operating systems, and applications operating on services.

3. To propose and implement a novel protocol that can be used to achieve secure authentication and authorisation for multi-cloud collaboration with improved performance.
4. To design and develop an efficient and dynamic service scheduling algorithm for the selection of cloud services in the multi-cloud scenarios based on the partial or closest matching of QoS attributes.
5. To design and develop a service level agreement (SLA) based mechanism to ensure that the service execution in a foreign cloud complies with the negotiated SLA parameters.

1.5 Research Contributions

The major contributions of this research are the following:

a. We developed a threat model for services in the cloud. It provides identification of the threats to cloud services and the methodologies that could be used by attackers to exploit those threats. This model can be used to determine threats for key service functionalities including authentication, data computation and data storage in relation to the cloud architecture and service interfaces. This threat model can be used to determine potential threats in relation to a foreign cloud architecture in which user will be accessing services.
b. A novel model is proposed for establishing dynamic collaboration in multi-clouds so that users can securely access services in foreign clouds as per their requirements. The protocols to support the proposed model and the functionalities of various model

components that enable secure and dynamic multi-cloud collaboration are presented with experimental results.

c. To facilitate dynamic authentication between multi-clouds we have proposed lightweight protocols and techniques based on single-sign-on (SSO) property. These authentication protocols have been formally verified using BAN Logic [8] and achieve better performance than traditional authentication protocols.
d. The proposed MCC model has been extended to support efficient scheduling among multiple clouds by doing service selection according to partial or closest matching of user quality of service (QoS) requirements with high accuracy.
e. The service level agreement (SLA) based mechanisms in MCC have been developed to facilitate secure service execution in a foreign cloud. This includes setting up SLA negotiation, enforcement and monitoring which helps in negotiating required service QoS parametres, enforcing mechanisms in a foreing cloud that can ensure that the functional and non-functional requirements agreed in the SLA can be satisfied, and monitoring service execution in the foreign cloud and reporting back any case of SLA violation to the user.

1.6 Thesis Organisation

This thesis is organized as follows:

- Chapter 2 presents a detailed review of the state-of-the-art research and existing methodologies related to multi-cloud collaborations and security to identify the research gaps, critical issues and limitations in the existing techniques of multi-cloud communications. Moreover, a comprehensive literature review is presented in the context of issues such as authentication, service selection and service level agreement (SLA) management in multi-clouds.

- Chapter 3 presents a detailed analysis of security threats to services in cloud computing. These security threats are analysed along with the methodologies that can be used to exploit these threats. A conceptual analysis is performed to how identified threats can affect various functionalities of services in cloud. Moreover, a generic model is proposed that can be used to determine possible threats, attacks and their impact on a specific service functionality.
- Chapter 4 presents a model for setting up secure collaboration between multi-clouds according to user requirements. In this framework, a novel authentication mechanism is proposed that is efficient, scalable and can authenticate users from a foreign cloud without prior agreement between communicating clouds. The proposed protocols for communication and authentication are formally and empirically verified.
- Chapter 5 extends the model proposed for multi-cloud collaboration to achieve an end to end solution for secure communication. Initially, a scheduling mechanism is proposed for efficient service selection that is used to choose best multi-cloud environment that can satisfy user quality of service (QoS) requirements among multiple providers. After selecting a foreign cloud provider, service level agreements (SLAs) are negotiated between the client and foreign cloud provider, and provider implements techniques that can facilitate service execution according to the agreement, and finally monitoring is performed to check if the service execution in foreign cloud satisfies client requirements agreed in the SLA.
- Chapter 6 summaries the research carried out and major contributions of this research work. This chapter also highlights the future research direction of this research.

- Appendix A shows the use cases of the proposed model and explains its applications in domains such as Internet of Things (IoT) and E-Healthcare in multi-clouds. Moreover, we present the benefits of using proposed model with each of these applications. The definitions of key terms used in this thesis are described in appendix B.

Chapter 2 Literature Review

2.1 Overview

This chapter presents the literature review on cloud computing and security issues pertaining to the domain of multi-clouds. A comprehensive survey of existing research mechanisms for multi-cloud collaboration, authentication, service selection and service level agreement management (negotiation, enforcement and monitoring) is presented. Furthermore, we highlight the limitations with existing research solutions and discuss the critical requirements for this research work.

2.2 Cloud Computing

Cloud computing – an emerging popular paradigm; is a gradual evolution of various interlacing technologies, [9, 10] enabling an organization's ubiquitous access to shared and globally distributed pools of higher-level computing services which are accessible exactly when they are needed. The aim of cloud computing is to better utilize the distributed resources; remotely placed together, to grab maximum throughput and to combat large-scale computational problems. The main features of could computing are visualization [11], scalability [12], interoperability [13], fail-over mechanism [14], quality of services [15] and its delivery/deployment models (public, private and hybrid etc.) [16]. Cloud computing offers services in terms of infrastructure, platform and software (IaaS, PaaS and SaaS respectively); being based on SOA offering everything as service (XaaS).

2.2.1 Service Models

Cloud computing offers on-demand access to the combined capacity of remote, shared and globally distributed resources for a pay-per cycle basis [17]. The cloud service architecture comprises of three main service models; Infrastructure as a Service, Platform as a Service and Software as a service (IaaS, PaaS and SaaS respectively) [18]. These models provide increased abstraction between the cloud client and pool of shared resources by offering an on-demand access exactly in need.

2.2.1.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service refers to the deployment of computing infrastructure; hardware (servers, storage and networks) with their associated software (OS, Virtualization tools and filesystem) as a Service to the cloud client. It avoids the capital expenditure for having set up an extensive infrastructure rather using it all together at the time of need on pay per cycle basis. The main advantages of IaaS are cost-effectiveness, on-demand availability, secure environment provision (especially for digital forensics and malware analysis), suitability for multi-platforms organizations (Computer, Mobile accesses), portability and interoperability of infrastructure. Amazon Secure Storage Service (S3) and Web Services Elastic Compute Cloud (EC2) [19], ServePath's GoGrid [20], Rackspace Cloud [21], Oracle Cloud Computing [22], GigaSpaces [23], RightScale [24] and Nimbus [25] are examples of IaaS.

2.2.1.2 Platform as a Service (PaaS)

Platform as a Service refers to the deployment of on-demand software development environment for the developers as a service. It includes; toolkits, developing environments and distribution and payment channels. In PaaS, computing platform, operating system, programming language execution environment, database management system and web servers are offered as service. The programmers can develop and run their applications on a cloud based platform without the cost and complexity of buying the underlying hardware and software. The main advantages of PaaS services are; deployment of an updated development environment, reduced cost, fully tested development toolkits and faster uptime. The example of PaaS offerings are Google App Engine [26], Microsoft Azure [27] and Oracle Cloud Platform [28].

2.2.1.3 Software as a Service (SaaS)

Software as a Service refers to the on-demand availability of application software and databases for the cloud users, charged on a pay-per-use basis or using a subscription fee. The cloud users do not need to deal with the underlying complexity or place where the applications run rather they are provided with software as a service exactly when they need it. Similar to IaaS and PaaS, SaaS also provides scalability, reduced cost, easy handling, secure application usage, fast and portable work, and interoperability. The maintenance, updating, upgrading, security are purely provider's concerns.

2.2.2 Security in Cloud Computing

Cloud computing emerged as a popular paradigm due to its high-quality services with low cost and enhanced performance. Its widespread adoption due to its flexible infrastructure and ease of access is a hallmark of its prevalence. Besides its service diversity utilities, security has become a major concern which aggravates the issues related to users' privacy as well as cloud's quality of service (QoS). In this section, a detailed overview on the security of cloud services models is presented.

2.2.2.1 Software as a Service (SaaS) Security

In Software as a Service (SaaS), the cloud provider is solely responsible for security provision while a cloud user is just supposed to work on the application [29-31]. Therefore, the cloud provider must prevent multiple users from seeing each other's data [32]. The traditional on-premises systems stored the user's data in the same promises i.e. physically secured but when it comes to cloud computing, the user's data is stored on the provider's cloud. Even in private clouds, the data is stored with other users' data (at the same place).

As SaaS applications are deployed over the web, web-security [33] becomes a major concern in SaaS security. The users are mostly concerned with confidentiality, integrity and availability of the SaaS applications. There are certain tests and security measures that can help to validate the data security in SaaS and prevent the cloud from issues including XSS, Cross-site request forgery (CSRF), Access control issues, OS issues, Cookies and hidden fields manipulations, SQL injections, Insecure storage and configuration. In Amazon Web Services [19], the network layer must combat the traditional security attacks [34] of Man-in-the-middle, IP-Spoofing, Port scanning and spoofing, ARP tunnelling and spoofing, Phishing and Botnet execution. For users' data privacy various encryption and privacy preserving techniques been proposed for clouds.

2.2.2.2 Platform as a Service (PaaS) Security

Platform as a Service (PaaS) offers deployment of on-demand software development environment without capital expenditure of setting up and maintaining the underlying hardware and software layers [35]. Being a cloud service model, it depends mainly on a secure and reliable web browser and networking. PaaS security has two sub-layers; the security of deployed users' applications and the security of PaaS itself [36]. PaaS providers are meant to secure the platform software stack; runtime engine running the users' applications.

PaaS has major data security concerns associated with it. PaaS third-party related security inherits could include Mashups' issues [37] as PaaS offers these third-party web-service components alongside traditional programming languages [35-38]. As PaaS users' community comprises of main developers, so a cloud-based secure application development is a major security concern for them and System Development Life Cycle (SDLC) [39, 40] must be followed during application development.

2.2.2.3 Infrastructure as a Service (IaaS) Security

Infrastructure as a Service (IaaS) offers deployment of infrastructure; virtual hardware (servers, networks and storages) and their associated software as an on-demand service [39]. IaaS users can run any software with full management and better control over the allocated resources [41] especially in VMs [42]. They are solely responsible for security policies configurations [43] but the underlying complex infrastructure is controlled by cloud providers. IaaS; being an integral part of the cloud platform has some security issues. The main feature of IaaS is infrastructure-virtualization as it offers IaaS users to manipulate (copy, share, migrate, create and roll-back) with VMs [44, 45] but it is vulnerable for attackers due to extra layer; needs to be secured [46]. VM security is as important as physical machine security [47]. Security is a major challenge as more entry-points are created in VMs [48]. VMs share the system resources on the same server; security of each VM may be compromised in this regard if there is a malicious VM [46] as VMs communication is possible using cover channels and bypassing VMM rules [45-47].

2.3 Multi-Clouds

After a decade of progress, cloud computing has appeared as the buzzword on the IT landscape due to its widespread adoption and architectural flexibility. With the advent of the internet, computing grew up swiftly from standalone to distributed, to cluster, to grid and then to cloud. The evolving and emerging cloud computing trends and directions have been discussed in discussed in research [49]. The multi-clouds have been stated as the evolving computing architecture. Multi-clouds are considered as a single heterogeneous architecture offering on-demand services from multiple cloud providers [50]. Traditionally, the multi-clouds were meant to leverage resources from widespread data centres but gradually applications were hosted to utilize resources [51, 52].

2.3.1 Benefits and Strength

With multi-clouds, a user can distribute a single workload between two IaaS providers, or can place single workload on one of IaaS providers and backup on the other; the user is free to transit between them. Multi-clouds have various benefits over single clouds. Various cloud providers have worldwide data centres and a single provider cannot have data centres in every administrative region and country [53], rather there are some legal geographical requirements for data storage as well. Therefore, to gain access across such widespread data centres multiple clouds providers can be utilized.

For single clouds various service-outages' cases have been reported. A popular among them was the Amazon's data centre failure after which Amazon encouraged the use of multiple data centres for fault tolerance [54]. Outsourcing services from multiple clouds and the ability to freely workload transition can facilitate a cloud user avoiding the vendor lock-in and dependencies. A cloud user can go freely elsewhere if a provider does not suit him in terms of policy or pricing [54]. Multi-clouds utilization ensures better overall quality of services as in terms of load-balancing, cost-reduction, flexibility, interoperability and scalability.

2.3.2 Obstacles and Challenges

Multi-clouds are being utilized and adopted very swiftly. However, certain obstacles need to be combatted. For example, common APIs facilitating multi-clouds are supposed to responsible for resources offered by different providers. The network and storage abstractions; different across the providers, let the multi-cloud adoption fit for each application instead of using a generic one. The price and billing mechanisms and policies are mostly different across providers, so multi-cloud adoption needs to put hard efforts to develop a multi-cloud application. Same is the case with management tasks (e.g., load balancing, fault tolerance, resource management etc.). Libcloud2 and jClouds are examples of APIs alleviating these challenges.

2.3.3 Types of Multi-Clouds

Multiple clouds can be categorized as hybrid and multi-clouds. A multi-cloud can be a hybrid cloud by combining either public and private clouds or IT infrastructure. A hybrid cloud basically combines two cloud deployment models; public and private [51, 52]. Hybrid clouds are used to cater sensitive data [55]. However, there are certain challenges in setting up hybrid clouds including latency, network topologies and bandwidth [56]. A multi-cloud is a single heterogeneous structure bringing various cloud services from different providers under the same umbrella. This encourages services and resources provision, applications' interoperability and portability and vendor lock-in avoidances [53, 57]. Multi-clouds are further classified into sub-types that will be discussed in detail in Multi-clouds – collaborations.

2.3.4 Multi-Clouds Collaborations

Multi-clouds – collaboration or federation refers to the integration of multiple clouds providers' services under the same collaborative or federated umbrella. Cloud computing is being evolved day by day through intensive researches. This evolution can be understood through various subsequent stages of how multiple clouds collaboration evolved. Initially, at stage 1, there were only monolithic clouds; independent proprietary cloud architecture based services, then at stage 2, there were vertical supply chain clouds/ hierarchal multi-clouds, and at stage 3, there are horizontal federated clouds/ cross cloud federation [58]. The terms inter-clouds, multi-clouds and clouds federation have been mostly used interchangeably in the literature. More preciously, the term 'cloud of clouds' is referred for them.

Various standardization bodies have defined federation as following; ENISA; which is known as the European network and information security agency, introduces a Cloud Federation as an integration; made up by combining two or more clouds [59]. The OPTIMIS project introduces multi-clouds and cloud-federation separately as if a service provider takes services from Infrastructure provider from another cloud; it is federation as the IPs can share resources among them horizontally and if service provider accesses two infrastructure providers separately, it is called multi-cloud [60]. This project suggests that for efficient decision-making QoS and risk management are key factors. QoS can include the process of deciding whether to allocate certain resources to a service such that best quality is achieved but resources are not wasted. OPTMIS project does not discuss the scheduling for negotiating discovery and allocation of resources to other cloud providers via central repository.

The Reservoir project [61] says relatively smaller and medium providers can't participate in cloud-service provisioning because of non-interoperability so the disparate providers should federate for utility provision. If RC denotes reservoir cloud and RS denotes reservoir site then it can be said that structurally the RC comprises several RSs handled by various infrastructure providers (IPs). Each reservoir site having resources further divided into Virtual Execution Environments (VEE). Service applications can simultaneously utilize virtual execution environments' hosts from various RSs. It focuses on need of SLAs and infrastructure monitoring but does not focus on the business requirements for multi-clouds and how to establish security and trust.

The Contrail research project [62] introduces a service level agreement centred federated approach for Clouds with an aim to reduce users' burden by eliminating providers' lock-in and to increase the cloud services' efficiency both vertical and horizontal integrations. However, its documentation does not describe the constraints that could be specified by users in an SLA.

The BonFIRE project [63] targets finding the possible integrations between network and service infrastructures. The BonFIRE project has addressed the extension of current clouds to federated clouds with heterogeneous virtualized Resources. They have developed a catalogue of standards and procedures to interconnect multi-cloud environment with advanced facilities for the purpose of a controlled networking. It also offers low level networking tools for monitoring statistics to cloud users. However, for testbeds VMs of different sizes use 100% of CPU meaning that the performance of resources might vary a lot between testbeds due to heterogeneous hardware.

The mOSAIC project [64] introduces the service requirement's specifications in terms of cloud ontology through an innovative API. Its implementation offers portability, vendor independence, APIs for application development using multi-cloud services. In [65], Buyya et al. have proposed a market-oriented cloud architecture and discussed the possibilities of global cloud exchanges. An extension of the proposed work in [53], has offered a just-in-time, scalable, federation-oriented and an opportunistic multi-clouds services provisioning architecture; known as InterCloud. Furthermore, they have set a catalogue of various research issues to present a market-oriented approach for interCloud offerings. From this security and runtime management of cloud resources can be identified as key issues.

In [66], a business-oriented cloud federation model has been proposed for Real-time Online Interactive Application (ROIA). In this model, multiple independent infrastructure providers may easily collaborate smoothly for QoS assured ROIA services through scalable IT infrastructure. The scalability and security issues related to cloud services provision have been discussed with a business-oriented perspective. The model used an additional business layer that could ensure the QoS assured services provision that reduces that scalability of the system.

In [67], authors discussed the few providers dominated over PaaS market causing adoption hurdles for multi-clouds due to lock-in issues; responsible non-portable and non-interoperable data applications. To cater them, a novel user-centric broker solution Cloud4SOA for multi PaaS; has been proposed to ease multi-cloud collaborations. It focuses on using ontologies to give interoperability among cloud providers but does not include the collaboration strategies for secure and scalable runtime access to multi-cloud resources.

2.4 Multi-Clouds Security

The widespread adoption of cloud computing has increased swiftly in many organizations. The high-quality computing services, on-demand shared pools of distributed resources, pay-as-you-go fashioned reduced costing model have made it a very popular paradigm. These so flexible and diverse services along with the layered architecture of cloud computing have great attractions for attackers, so security is a major concern in cloud computing. The potential risks and threats have increased as cloud computing evolved into multi-clouds computing. As various cloud service providers exchange services and resources, so security has become a highly addressable and significant concern in this regard. To cater this, various cloud security researchers have proposed a variety of techniques and approaches. We will review in detail the literature regarding authentication mechanisms, monitoring, service level agreements and secure collaborations in the next few sections.

2.4.1 Multi-clouds Authentication

Authentication in the perspective of cloud computing; is a fine-grained process of validating or verifying the identity of a cloud stakeholder; service user or subscriber. As cloud computing evolved to multi-cloud collaborations and services provisioning from multiple clouds providers is being utilized so the need of authentication has emerged significantly as it directly impacts on the reliability, security and privacy of the cloud services subscriber or users [68-70]. To address this significant need for multi-cloud collaboration, various researcher around the globe have proposed some state-of-the-art authentication mechanisms for multi-cloud federations based on identity management, access control mechanisms, cryptography approaches and various other strategies. Here we present the literature overview for authentication mechanisms in multi-cloud collaboration security.

In [71], Hassina Nacer et al. have developed a fully distributed and decentralized authentication model for catering dynamic authentication issues, that were complicated and time consuming, between various homogeneous and heterogeneous organizations. The proposed model basically provides two-way authentication with three-party key generation and a distributed certificate authority for the purpose of fulfilling the security requirements for web-services delivered over the internet. The author has proposed ontology-based authentication protocol annotation to overcome authentication mechanism heterogeneity and a conflict resolution algorithm to cater policy heterogeneity. As a proof of concept, a simulation design and a prototype web service have been designed. The problem with this work is that it has been done over two assumptions; various semantic web services have been saved into different trust circles and they have no underlying bottleneck problem. Moreover, it is only designed to address web services collaborations.

In [72], Noureddine et al. have enhanced the lightweight and user-centric authentication protocol OAuth to OAuth 2.0 to solve cloud federation challenges. The OAuth has already been used as a simple identity management protocol. The author proposed two modifications in this regard. The first modification needs a pre-established trust provision by synchronizing an authorization table between the authorization server, resource server and client. The second modification induced referral parameters into OAuth so the trust federation among different authorization servers can prevail by referring requests. The work can be extended to a single authentication server as a cloud-identity provider, but a key challenge is that it requires pre-established trust provision between multi-cloud entities.

In [73], Celesti et al. have proposed a three-phase authentication model for cross-cloud federation problems after in-depth analysis. Furthermore, a cross cloud federation manager has been developed and authentication agent using the SAML CCAA-SSO profile was designed. The performance of the work needs to be analysed by the evaluation of authentications, IdP enrolments, real testbeds or simulations.

In [74], Polzonetti et al. have presented a security framework implementing a centralized access control for authentication and authorization functions provision to cloud based web services. The framework leverages SPID infrastructure to uniquely recognize citizens and enterprises through pre-issues identifiers from identity providers. The proposed framework does not provide users' management as it uses SPID. It basically provides authentications through federation. The framework being an interface; authenticates remote SPID complaints and shows the results to web applications or services hosted on the cloud. It has two OpenAM platform based modules; access manager and IdP proxy and finder. It works in three approaches as; policy agent for compatible application and services, a reverse proxy for outdated application and services, and OAuth/OpenID connect protocol. This work can be extended by inducing an attribute authority service.

In [75], Demchenko et al. have proposed a federated access control model using Federated Identity Management (FIDM) framework. It can also be supported by the trusted third-party entities, for example, Cloud Service Broker (CSB). The model further defines the intercloud federation framework (ICFF) that is a part of the previously proposed general intercloud architecture framework (ICAF). The research discusses components of the distributed federated multi-domain authentication and authorization infrastructure and provides various federated identity management scenarios and architecture patterns.

In [76], Celesti et al. have put forward a SAML SSO profile for the establishment of trusted interdomain communications under three-tier cloud architectures applied in various CLEVER-based clouds. Cross-Cloud federation indicates the trust-context establishment between various cloud provider platforms in different administrative domains and places. Federation encourages clouds' interdomain communications. The federation is set up in three phases; discovery, match-making and authentication. In this work, the author focused on authentication phase for CLEVER intra-domains secure interactions but do not provide measures of the scalability of their solution.

In [77], Celesti et al. have discussed the privacy, security and federation issued in detail in the context of the federation of clouds then presented an authentication architecture to cater the problems and issued faced in identity management in the context of InterCloud. Furthermore, it has been investigated how the proposed architecture can be applied to manage the desired authentication level among clouds to establish a federation.

In [78], the authors have proposed a novel framework for access control mechanism in the perspective of IaaS cloud environment. A hybrid access control model/mechanism of the type-enforcement access control and the role-based access has been proposed. Furthermore, to assign the permissions dynamically for virtual machines, a permission-transition model has been designed. An access control mechanism based on VMM arbitrates in a fine-grained manner; the virtual machines' requests to the underlying resources. A VMM-enabled access control mechanism has been put forth for relating intra-virtual machines communication channels. The author further implemented iHAC in iVIC IaaS cloud platform. This research recommends access control to be moved in hypervisor and taken away from network which places large overhead on the model.

In [79], authors have proposed a novel authentication and authorization model for cloud services. The proposed model supports all the features that are intended for authorization services provisioning. That features can be listed as hierarchical RBAC, federation, path-based object hierarchies and multi-tenancies. The author further discussed in detail the implementation and architecture of the proposed model and highlighted it in terms of scalability.

2.4.2 Multi-Clouds Secure Collaboration

The collaboration or federation of multiple clouds refers to integration or combination of multiple clouds service providers' working into a single and integrated unit of clouds. Cloud computing paradigm has evolved considerably, and this evolution is of very much interest for the potential attackers. With the advent of the internet, the computing grew up swiftly from standalone to distributed and from distributed to cluster and grid and then to clouds. In the paper [50], the authors have discussed in detail the various evolving and emerging cloud computing trends and directions proposed or presented so far. The multi-clouds or inter clouds collaborations or federations have been considered as the evolving computing architectures.

Multi-clouds or inter-clouds federations have been considered as a single and a heterogeneous architecture that offers on-demand cloud services from multiple cloud services providers to the cloud users or customers on pay as you go fashion with high quality services at reduced costs [53]. To manage this all the security and privacy have become a significant concern in this regard. Let us have a broad overview of the literature in the context of multi-clouds secure collaborations.

In the paper [80], authors have proposed a novel collaborative framework combining software defined networking with service function chaining to maximize the collaboration among various SSFs to cater large-scale security threats and attacks. As visualization has opened a new era of security and privacy on the network landscape and security attacks and threats are being evolved and emerged, their increasing diversity and size urge to make a collaborative solution more resilient to combat them. Collaboration among security services functions (SSFs) is needed and expected to be essential to 'security as a service' layer. Furthermore, the author discussed a framework allowing security services functions (SSFs) from various domains to dynamically control the resources allocated. This collaboration framework launches a distributed large attacks mitigation system in a scalable and dynamic manner. This work incurs low overheads among its compared techniques or frameworks.

Standards such as ISO 27000 and NIST-FISMA can aid the cloud service providers to increase security and maintain the customers trust. But the problem with these standards is that they are still not fully capable of dealing the full complexities and underlying complications of cloud computing platforms rather security frameworks are needed for cloud secure collaborations as well. To combat this situation, the authors of [81], have presented a novel cloud computing security management framework. The proposed framework has been aimed at FISMA standards' aligning with the cloud computing platform. It can help to enable the cloud customers and cloud service providers to be security certified. The proposed framework has been based on improvements in collaboration among cloud service providers and cloud service customers/consumers in maintaining the state of security of the cloud based hosted services. It has been built on security standards assisting in the automation of security management process. However, the security categorization of service provider in this paper is qualitative and does not take user requirements into account.

In the paper [82], the authors have presented a novel model checking scheme that can work as a management service tool to verify the multi domain cloud policies. Their proposal was based on generic model checking by the National Institute of Standards and Technology (NIST) and further interlaced with role based access control (RBAC) reasoning. The existing authentication approaches and techniques in grid computing based systems are able enough to verify and detect only the redundancies and conflicts among multiple policies. But the issue is; the latter could not be capable of overcoming the risk of legitimate user access in multi domain cloud computing systems. Furthermore, a formal definition of the proposed technique and its security properties have been provided that needs to be verified in multi domain cloud computing systems.

In the paper [83], authors have first reviewed the existing security landscape of cloud orchestration and emphasized the current literature's knowledge gaps. The orchestration is a term referred to automated arrangements, services, management of complex systems and coordination among multiple computing systems. Authors provide a security threat model by listing the security assumptions and elaborating the actual attack surface. The authors of the paper have discussed deeply the building blocks of the cloud orchestrators for the deployments of multi-cloud federations. In the same way, the authors have proposed and presented a novel security architecture in this regard along with various security and privacy enablers for cloud computing orchestrators in the context of federated cloud deployment. The proposed security architecture has the potential to improve the security of cloud orchestrators but needs to be extended to address orchestrations in multi-cloud collaborations.

2.4.3 Multi-Clouds Monitoring

A major advantage of cloud computing is that it offers the high level quality of services (QoS) as the whole cloud's story revolves around services. The quality of services is improved effectively through monitoring and it is an effective approach for improving various service features, for example, software optimization, performance evaluation and auditing, profiling, etc. The flexible, diverse and elastic nature of cloud computing needs a strong monitoring mechanism, however, monitoring cloud at runtime is a challenging issue.

To cater this, the authors in [84] have proposed a novel model for monitoring the cloud in its runtime. The model was named as RMCM that has provided a representation of running cloud by paying attention towards common monitoring concerns. On the base of the proposed model RMCM, the authors have implemented a robust and efficient framework for monitoring that was able enough to maintain trade-off monitoring capabilities and overheads incurred on runtime through the management of motoring related facilities. The proposed model is focused on presenting raw monitoring data in a more intuitive form so that user requests can be handled more efficiently. Although it improves existing security metrics based on service model information, research in this thesis focuses on providing customised methodology to map metrics to the cloud system.

In [85], the authors have proposed, developed and validated a cross layer multi-cloud application (CLAMS) as a framework based on services. The proposed services based framework was capable of performing various tasks including quality of services monitoring of the components of the application (any database, server deployed) and paying visibility the QoS of individual applications. The author conducted the extensive experiments on the real-world multi-clouds environments, and the results acknowledged that the CLAMS outperformed among its counterparts. However, this monitoring approach has not been verified for selecting services to suggest their usability along with service selection framework.

In [86], the author has presented an architecture of monitoring that is configurable automatically and activated on the base of a signed service level agreement (SLA). These type of monitoring architectures combine various security-related monitoring tools (may be developed or hired on ad hoc base). As the data grows and is spread over multiple environments, the complexity of risk assessments has become an uncontrollable challenge to deal with. In the paper [15], the authors have presented a model for assessing the security risks for the distributed business processes in a multi-cloud environment. However, it is only a mapping study to understand security risks.

Policy management has been a significant concern in dealing with the security of heterogeneous environments. In the paper [38], the authors have pointed towards the lack of security and trust in this new infrastructure model. The authors have designed and presented a robust security policy model that has deployed business processes for a cloud based infrastructure. The main purpose of this security policy model was to generate an appropriate and suitable security policy for a cloud based infrastructure in a dynamic way. As the new progress in the domain were carried on, the researchers in [87] present a security monitoring framework that is able to deal with the particularities of cloud computing in an enough adequate way. The authors have proposed a detailed and open ISG framework; named ISGcloud that has dealt the security monitoring linked at the cloud service lifecycle. However, this work also focuses on limited set of security governance while we focus on measuring QoS properties of services through security monitoring within multi-clouds.

2.4.4 Multi-Clouds SLA Assurance

The widespread adoption cloud computing has increased swiftly in many organizations. SLA – service level agreement, in a general context; as name states, is an agreement between a service provider and subscriber that the pre-defined service level of quality, availability, reliability and responsibility would be maintained. In cloud computing as the paradigm has evolved from SOA, there is SLA needed between service provider and cloud client. As the cloud architecture further evolved to multi-cloud collaboration, the SLA assurance became a major concern to guarantee the quality of services. So, it is now necessary to mention the right usage level of a service and its conditions into a contract; Service Level Agreement (SLA). More formally, in an SLA a service-level provider management negotiates, agrees upon and finally documents the agreed service-goals with clients of organization and, afterward, monitors and produces reports for service providers ability to deliver the agreed level of service [88]. Researchers have proposed many secure SLA assurance frameworks and research works in the context of multi-clouds federations. Here we present the literature overview of secure service level agreement assurance in regard of a multi-cloud collaboration context.

In [89], authors have developed a secure service level agreement ontology based framework. The framework can be used for purposes, for example, to understand the agreement of security provision from a cloud provider, to audit that either the compliances from a provider are in accordance with federal regulations or not, and to make negotiations for desired security levels. The authors extend this work to secure service level agreement in WSAG4J5 based on Agreement. WSAG4J5 is the Java implementation based on this work of WS-Agreement. This extension would help in designing and implementing service level agreements for particular services, validating, monitoring and accounting etc.

In [90], the authors have introduced a novel approach named SPECS. The SPECS approach helps to offer various mechanisms to access security features that have been offered by CSPs, specify security requirements and to integrate the security services with cloud services to form security as a service approach. Moreover, the SPECS helps to negotiate, monitoring and enforcing various security parameters pre-specified in the service level agreement (SLA). The main benefit of SPECS is that it offers security assurance to clouds' end users for the cloud services provisioning and managing of agreed security parameters mentioned earlier in the service level agreement (SLA). The research for this thesis uses the concepts of SPECS for SLA assurance such as negotiation, monitoring and enforcement but uses light weight mechanisms at each stage for them to be used in a multi-party collaboration scenario across multi-clouds.

In [70], authors have investigated the ways of services selection and allocation in multi-cloud delivery model from the perspective of Software as a service provider (SaaS provider). Furthermore, authors have proposed a novel framework that assists the providers of SaaS in finding the appropriate infrastructure services (IaaS) as per their need or satisfaction levels. Moreover, the complete framework, service selection and allocation to detect either there is any SLA violation are described. However, for multi-cloud collaboration there is a requirement to propose a complete framework that offers service selection and allocation to users and detects if there is any SLA violation.

2.5 Applications of Multi-Clouds

2.5.1 Meta-Scheduling using Multi-Clouds

Meta-Scheduler is a term; referred to a broker or a central scheduler that has a core purpose to enforce or establish a wide policy over the distributed resources. It has been frequently discussed in various grid computing literature review. The key features of a meta-scheduler are negotiation, job scheduling, dispatching and management of resources etc. As various resource ownerships led to different topologies, in multi-clouds architectures the meta-scheduler works as a broker that assign a job, discard a job, put the job in the queue (if there is already a job in provision).

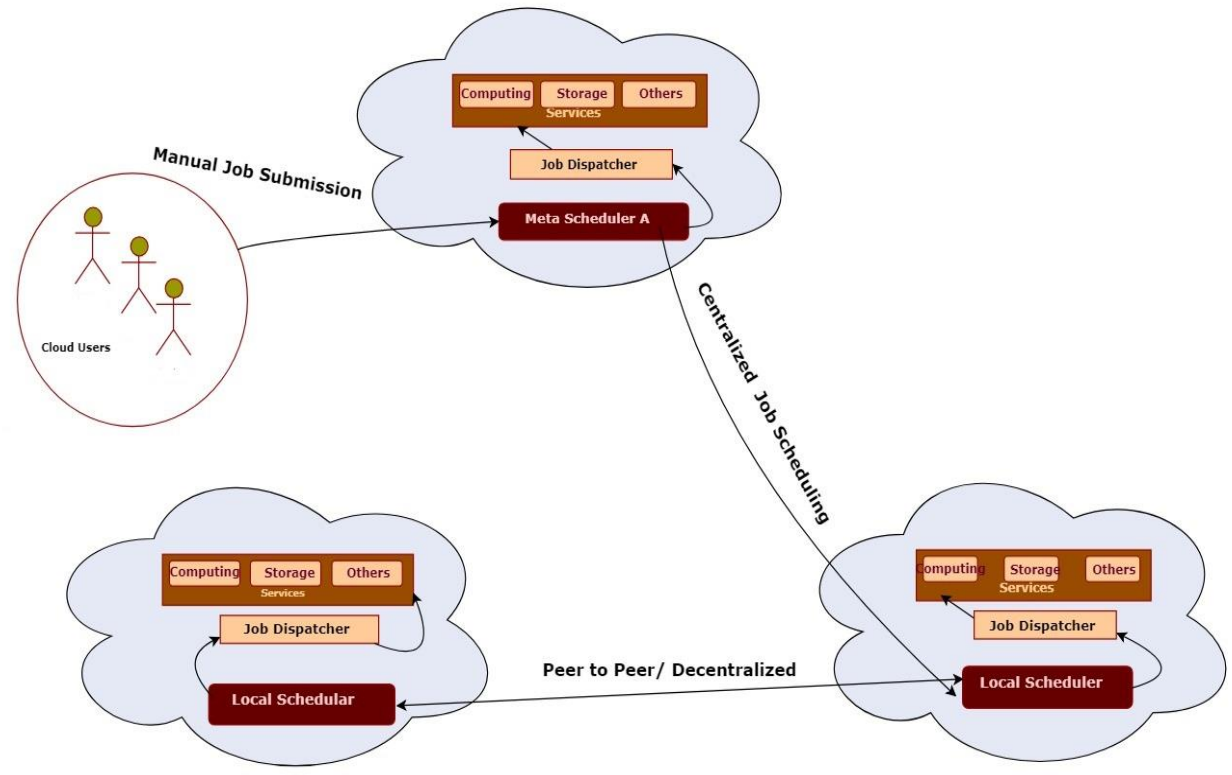


Figure 2-1 Meta-scheduling use case

Meta-Scheduling architecture or topologies are of two types; centralized scheduling and peer-to-peer /decentralized scheduling. In centralized scheduling, the job-scheduling is carried out by a central instance (meta-controller) that is entitled to maintain information of all the resources. When a job is submitted from the cloud user to cloud, it is shifted to meta-scheduler.

The meta-scheduler checks whether there is already another job in progress or not, if there is, the meta-scheduler puts the job in a waiting queue or shifts it to other cloud's local scheduler in case of multi-clouds. If there is no job already in progress, then the coming job is shifted to job-dispatcher that dispatch the job as per its request to the desired resource.

In peer-to-peer or decentralized meta-scheduler, each cloud has its own control (a decentralized control) that shifts the job to its peers (horizontal peers) and so on. Figure 2.1 shows a high-level use case diagram of both scheduling (centralized and decentralized) in multi-clouds. In a meta-scheduler use case, actors are the cloud users; that submit a job, cancel a job, list jobs and make a query about a specific job or service. The resources are the cloud services resources that can be related to either service layer (SaaS, PaaS or IaaS) comprising of applications, data storage, servers, computing and other miscellaneous services provided by clouds. While the requirements can be user-identity, security measures, service-level-agreements (SLAs) and federated identity [91-93].

The key goals of the scheduling algorithms are to minimise the execution time, costs and improve the scalability of the overall system [94]. To minimise the scheduling cost of load distribution among clouds a divisible load theory (DLT) based solution was developed in [94]. Authors demonstrate their algorithm supports the multi-QoS scheduling but do not test the communication overheads which can affect the system performance significantly. Moreover, most scheduling algorithms are static in the sense that they assume number of virtual resources and cloud facilities (pricing, availability) do not change over time [95]. However, a multi-cloud collaboration scenario requires a dynamic scheduling algorithm in which virtual resources and provider conditions can change depending on the collaborating clouds.

2.5.2 E-Healthcare using Multi-Clouds

Security and privacy have become the significant concerns in such organizational collaborations where sensitive data of individuals are dealt, stored and processed. The remote geographically dispersed data centres store such data by multi-clouds collaboration where one cloud's services are aggregated with that of another's to give an integrated and heterogeneous single service provision. The security and privacy become of utmost importance when the data is health-related.

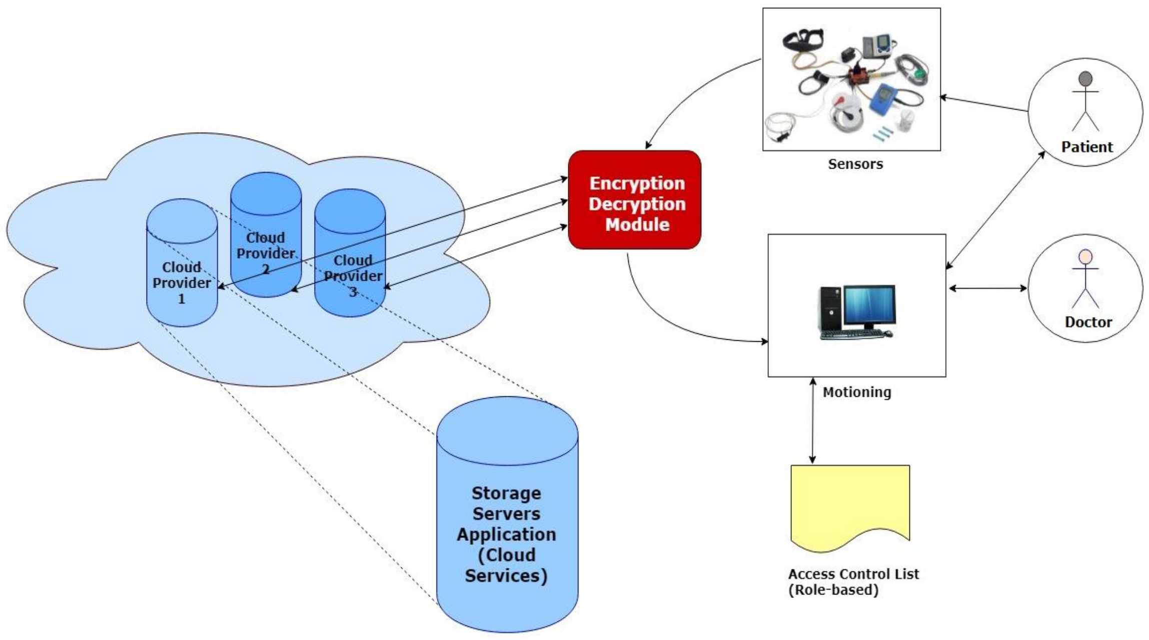

Sensors
Patient
Encryption Decryption Module
Cloud Provider 2
Cloud Provider 1
Cloud Provider 3
Doctor
Motioning
Storage Servers Application (Cloud Services)
Access Control List (Role-based)

Figure 2-2 Use case for Sensors E-health

There are various acts and legislations in this regard to preserve the privacy of patients' data, for example, HIPAA act. So, for preserving such sensitive data the various research works have been proposed so far. With the advent of smart cities, smart phones, smart homes and smart everything, the electronic heath (E-Health) have become a new fashion to electronically collect, store and retrieve health-related records of a patient for treatment and research purpose. The data deal in this regard is termed as 'Electronic Health Records' (EHRs) that can be electronically collected, stored and retrieved. They may be a form of medical imaging, video of ultrasound (for example echocardiography of heart), simple prescriptions or advice, medical history or any sort of medical test documents or reports etc. [96-98]. Nowadays, wireless sensor networks (WSNs) [99] are being used to collect electronic health records directly from the patient even without a doctor. These sensors are the main components to boost up the E-health records collection automatically.

To understand the whole mechanism, we have illustrated a use-case in figure 2.2, in which a patient and a doctor are the main actors. The health records are encrypted through that secret cryptography scheme and the encrypted form of records are sent to multi-clouds providers so that they may store the encrypted data. The access control lists have been maintained on role based access control (RBAC). Therefore, the access control list validates the doctor or patient role to the computer and the request is forwarded to the encryption-decryption module that decrypts the intended electronic health records and provides it to the user computer. E-health is no doubt in trends nowadays, but there can be some security, privacy, reliability and availability issues either in data-in-transit, sensors and other perspectives.

2.6 Summary

Due to increased security issues in multi-cloud domain, threats to cloud services need to be analysed to develop a solution providing greater awareness to correlate the services being offered with customer requirements. The threat model can act as a trusted advisor to the proposed system for mitigating possible attacks by identifying required security features such as compliance and monitoring. The state-of-the-art techniques on establishing collaboration between multi-clouds are focused on providing cloud broker mechanisms as architectural solutions that can connect heterogeneous clouds offering on-demand services from multiple cloud providers. Various APIs and solutions are designed that address specific issues such as a service provider accessing infrastructure of a separate cloud to share resources, integration of network and service infrastructures, and multiple independent infrastructure providers sharing physical resources. The existing security standards such as ISO 27000 and FISMA do not cater the complexities of multi-party service interactions for the establishment of collaborative paths between session partners across multi-clouds. Other proposed mechanisms either do not take user collaborative requirements into account [81], have not been formally tested on multi-clouds [82] or only focus on specific orchestration issues such as authentication or scheduling rather than offering an end to end dynamic collaboration.

The current literature on multi-cloud collaboration also focuses on providing authentication mechanisms by which multi-cloud providers can authenticate each other dynamically. The dynamic authentication process between multi-clouds could be highly complex and time-consuming since intermediate authentication paths need to be created at runtime to dynamically covert credentials from different security realms. Most techniques offer a one-time authentication solution to multi-clouds so that they can authenticate each other and share infrastructure resources. In case, where hundreds of users might need to access services based in another cloud and each of them must be authenticated separately such solutions are not feasible. Existing research either focuses on having pre-established trust provision between multi-cloud entities, has scalability problems in dealing with large number of requests, or the proposed systems lack the implementation details for testing their scalability on a real cloud. Moreover, it has been identified that the multi-cloud authentication establishment may require time-consuming activities for credential exchange, verification and session establishment. Therefore, a lightweight solution is required that can handle credentials generation, exchange and coordinate authentication across multiple heterogeneous clouds independent of their own authentication mechanisms and with limited performance overhead to provide an accelerated connection to market. Along with the experimental validation, formal analysis of the proposed model needs to be done to prove its security in an end to end multi-cloud collaboration scenario.

From the state of the art techniques focused on service level agreement (SLA) management it can be understood that the clouds' success, trust, and reliability depends significantly on the ability of clouds service providers of fulfilling the agreed level of services they have promised in the service level agreements (SLAs). There is a limited research that offers secure SLA assurance frameworks in the context of multi-clouds collaborations. Most of these techniques focus on providing solutions to ensure SLA guarantees that service provisioning in cloud is according to agreed mechanisms, and not on the enforcement of SLAs during runtime. There are a lack of frameworks for SLA management lifecycle in multi-clouds which necessitates the requirement to develop light weight mechanisms at each stage of SLA management for them to be used in a multi-party collaboration scenario across multi-clouds. Moreover, runtime QoS monitoring approaches in clouds focus on limited set of infrastructure monitoring, focus on a specific business model, and there is a lack of mechanisms that provide a customised methodology to map metrics to the cloud system [84]. Moreover, providing mechanisms that can define and generalize the acceptable threshold of SLA violations in terms of functional and non-functional QoS requirements of users can highly benefit SLA management in multi-cloud collaborations.

Chapter 3 Threat Modelling for Services in Cloud

3.1 Overview

Due to the dynamic nature of cloud services, many enterprise level security policies, standards and practices cannot be implemented in cloud which leads to different security threats. These threats can be exploited by various attackers to compromise the cloud services. In this chapter, threat modelling for cloud services has been done by considering various attackers such as hackers, malicious administrators, malicious users and service providers. After describing various threats to services, methodologies to exploit those threats have been presented. Moreover, the generalization of threat model has been done to determine the threats related to a specific service functionality for various attackers in cloud.

3.2 Threat Modelling

The critical assets of cloud such as services and data can be compromised to gain access to sensitive data, applications, services and infrastructure. Moreover, compromising the services can lead to the misrepresentation of data, manipulating data processing results, failing to provide advertised services, and/or performing actions against the user consent as well as against the service level agreement between the cloud provider and user. In order to secure cloud environment, it is critical to a have a complete understanding of various threats and attacks that can compromise various cloud operations.

An approach called threat modelling can help to identify and address the security issues associated with a process. It optimises the security architecture by identifying vulnerabilities, analysing the possible threats, and defining countermeasures that can be used to prevent a threat. This can also be useful in designing new security mechanisms while taking threats into consideration. The key stages in threat modelling are identifying the assets, threats, attackers and mechanisms that can be used to exploit the threats [100]. The threat modelling approach can be applied to the cloud assets such as services to determine the possible threats and mechanisms to exploit those threats. Although threat modelling for web services and cloud infrastructure has been presented in literature, only limited research has been done to model threats for cloud services and the data they process.

Each service deployed in the cloud can have multiple instances running in different virtual machines in cloud simultaneously and used by different customers. These services process large amount of data ranging from public source to private and highly sensitive data. Threat modelling for services in this research leads to the identification of various threats, possible attackers and the mechanisms to exploit those threats. Some of these threats are specific to the misuse of cloud services but other threats come from exploiting various cloud resources such as infrastructure, virtual machines, networks, operating systems and applications. Similarly, attackers who exploit the threats can vary from outside attacker (hacker) to malicious cloud administrator, service provider and cloud users. The mechanisms used by attackers to exploit threats have also been explained along with the severity level of threats and their possible effects on cloud assets. The generalization of threat modelling for various service functionalities has been done that determines the threats for any specific service functionality such as data processing for possible attackers in the system.

3.3 Service Functionalities

In this section the generic properties of services are explained. Each service has different interfaces that are used for its various functionalities. The general functionality that each service must have is data storage, data processing, data transfer in network and authentication.

3.3.1 Authentication

This stage involves authenticating the users before giving them access to the service. Most of the service providers use application programming interfaces (APIs) to provide services, and APIs accept tokens for authentication. Since cloud services can be accessed using different devices such as mobile phone and PC, strong authentication should be used. Enterprises should use standard such as Security Assertion Markup Language (SAML) and Web Services Federation (WS)-Federation [101] to authenticate users before giving them access to cloud services.

3.3.2 Data Processing

In this stage data can be viewed, accessed, updated, and used with or without modification. Examples of data processing stage are performing a transaction on the data, or using it in a business process.

3.3.3 Data Storage

In this stage data is stored in database or storage repository in cloud. The key features of cloud storage are durability, availability, performance and security. Data storage also includes archiving and storing backup.

3.4 Threats to Services in Cloud

Threat can be defined as any event that is undesirable due to its malicious nature. Data and services are vulnerable to different threats in cloud. In this section the threats to data and services in cloud are explained. After describing various threats the methodologies presented in existing literature to exploit those threats have also been analysed. Moreover, we also list the assets affected by specific attacks.

3.4.1 Data Breach and Data Loss

Data breach is defined as the leakage of sensitive customer or organization data to the unauthorized user. It can have a huge impact on the operations of an organization resulting in the loss of finance, trust and customers. Similarly, data loss is the second most critical threat in cloud computing that can have a very negative affect on the operations of any enterprise.

3.4.1.1 Methodology

Data breach threats originate from the flaws in infrastructure, application designing and insufficiency of AAA (authentication, authorization, and audit) controls [102]. Y. Zhang et al. used cross VM side channel attack to extract cryptographic keys of other VMs on the same system and can access their data [103]. While, data loss mostly happens due to data deletion and corruption, loss of data encryption key, faults in cloud infrastructure (computing resources), or natural disasters. Similarly, data loss can also occur due to malware attacks have also been targeted at cloud applications, operating systems and services resulting in data destruction.

3.4.2 Evading Provenance

Provenance is the metadata that describes the history of data. Cloud provider or data owner can define different policies in terms of information flow properties, such as read, write, forward, excerpt, and paste data, and they may use provenance to keep track of those properties. Evading provenance affects data and such threats are mostly exploited by outside attackers.

3.4.2.1 Methodology

Data can be manipulated by hackers, insiders or malicious users by evading the provenance or violating the policies using the untrusted operating system and applications [104]. Similarly, outside hackers can manipulate the shared hardware resources of infrastructure (e.g., CPU caches, branch target buffers, network queues, etc.) to access the confidential information from running services which require confidentiality [105].

3.4.3 Malicious Service Threats

In a cloud environment, users generally have no mechanisms to trust the services they are running and depend on the trust assurances provided by the service provider. The service provider or insider can launch a malicious service in cloud, and the users who trust the service provider can run that service. This affects both the cloud users and the cloud provider because the service may become non-compliant to the service level agreements between user and provider. This damages the reputation and credibility of the service provider. The major assets affected by such attacks are cloud service and data.

3.4.3.1 Methodology

The malicious service can be deployed in cloud that can manipulate the user data, fail to provide the advertised services, modify the data processing results, store additional data for harmful purposes, and perform other unadvertised activities without the user permission [106, 107]. These services are usually deployed by the service provider intentionally but certain issues such as service designing, bugs in program code or malware can also compromise the cloud services. Attackers can also exploit security misconfigurations in cloud to compromise the security of user’s data, services and the whole infrastructure.

3.4.4 Malicious Administrator Threats

Cloud administrators are the employees of the cloud provider who have root access to the virtual machines and services running inside cloud. They can misuse their privileges to launch different attacks inside cloud which lead to the compromise of cloud services and data security.

3.4.4.1 Methodology

The cloud administrator can install and execute different software inside the user’s virtual machine to perform various attacks. These attacks include hardware, virtualization, operating systems, and application attacks. Similarly, the physical access to the hardware provides the ability to the administrator for launching more precise attacks that can manipulate user data and services. Such attacks include tampering with hardware and cold boot attacks. Cold boot attack is usually done by an administrator with physical access to a machine who restarts the physical machine to extract encryption keys from the running operating system [108, 109]. Moreover, administrators can also install arbitrary software on the cloud node to access data on that node, perform man-in-the-middle attacks on the hosted system, eavesdrop the network [110], update virtual machine drivers to vulnerable instances, and copy service data [111].

3.4.5 Virtualization threats

Virtualization is the abstraction to provide virtual interfaces similar to the underlying hardware. It enables a single system to run multiple instances of isolated virtual machines or containers. The basic components involved in virtualization are the virtual machine manager (hypervisor), virtual machines (VMs) and virtual disk images. Security mechanisms for physical systems are not always applicable to the virtual systems, therefore, these components are vulnerable to different threats in cloud computing. Moreover, with virtual machine migration data and services are transferred to another physical node which can have a different security policy and network topology that may result in different security issues. Cloud users can run malicious programs in their VMs to gain root privileges and access others users and data [112]. Cloud users can run malicious programs in their VMs to gain root privileges and access others users and data.

3.4.5.1 Methodology

Virtual Machine Escape is a type of attack in which the attacker can run an arbitrary script on the guest operating system to get access to the host operating system [113]. This provides the attacker root access to the host machine. Running arbitrary code that can also lead to bypassing data tracking [114]. Another attack to gain root access to host operating system is done by installing malicious hypervisor such as BLUEPILL rootkit on the fly [115]. An attacker with privileged access can access the data from physical storage drives, install malicious software on the cloud nodes, extract data from secure cloud nodes, copy virtual machines or disks to steal user data, bypass data tracking and extract useful information without getting tracked [116, 117]. Moreover, an attacker with root privileges can also eavesdrop on the data transferred in the network by sniffing, spoofing or man-in-the-middle attacks. Due to malicious sniffing and spoofing over the network traffic rates can be monitored, and cryptographic keys can become vulnerable [118].

3.4.6 Network Threats

The network plays a key role in the communication between cloud resources and between the users and the cloud. Without proper network security controls, new attack vectors can arise that can have a malicious effect on cloud assets such as data and services. Network threats exist because of the vulnerabilities in software, applications, services, web servers, and misconfiguration of cloud nodes. This can lead to compromise of user data privacy with attacker having access to network traffic, hijacking of user account and services, and unavailability of resources. Network threats can impact on both the data as well as service security and can be launched by outside attacker or malicious users.

3.4.6.1 Methodology

Hackers can exploit the platform level vulnerabilities in the cloud to launch network attacks. SQL Injection, phishing, fraud, Cross Site Scripting (XSS), botnets, and software vulnerabilities such as buffer overflow can result in the hijacking of user account and service hijacking [115]. Hijacking user account or services is the stealing of user credentials to access his account or computing services. By compromising a service, hackers can execute a denial of service (DOS) attack that consumes cloud resources such as computing resources and network bandwidth to make them unavailable to legitimate users. Distributed Denial of Service (DDOS) attack is a variant of DOS attack in which multiple network sources are used for consuming cloud resources [119].

3.5 Generalization of Threat Modelling

The threat model shown in figure 3.1 can be used to determine the threats and attacks on various services for a specific attacker in the cloud. The attacks can vary for different services and possible attackers in the system. In a case where there is no service running in cloud, then even if there are attackers, no threats to services exist. However, if there is a service running in cloud with different interfaces for various functionalities than it is vulnerable to different threats. By using the threat model presented in the chapter, possible threats and attacks on a service can be determined for different service functionalities. The details of the possible threats to services, the methodologies to exploit those threats and the specific service functionalities affected by those threats are shown in table 3.1.

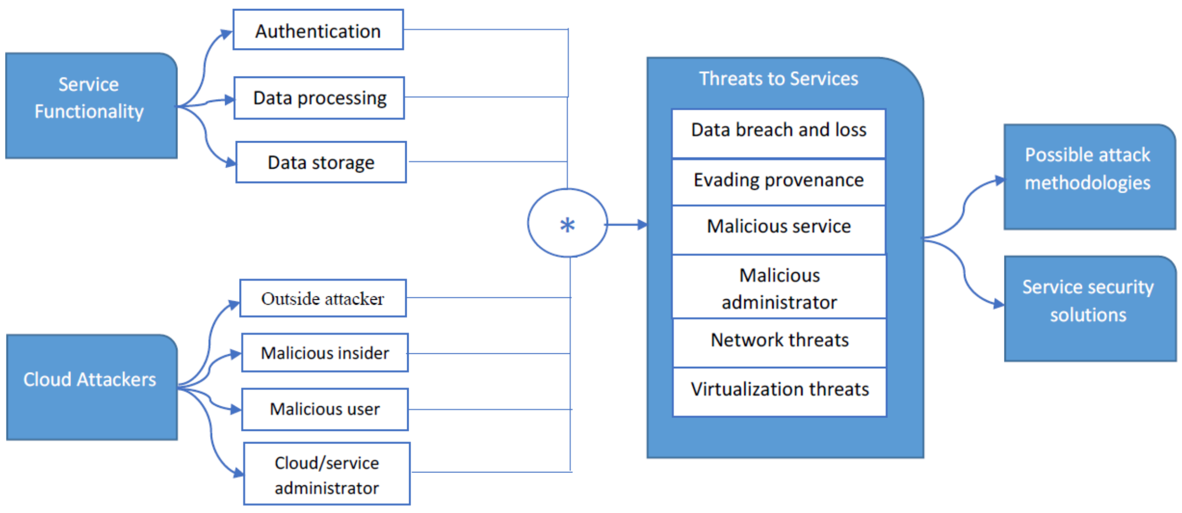


Figure 3-1 Threat model for services in cloud computing

3.5.1 Case Study

Consider a cloud system based on OpenStack is hosted in a data centre. Suppose a user X leases a virtual machine in cloud and runs a Microsoft Word (MW) service in it. User X can access, view, read and write data to MW. After using the service, user wants to store it in cloud. In order to do the threat analysis for this service we can use our threat model. When the user is accessing, viewing, reading or writing the service MW, he is using the data processing functionality (F1) of service. Similarly, when a user saves the file on cloud server he uses data storage functionality (F2) of service. The possible threats for service MW functionalities (F1 and F2) can be separately determined for various possible attackers.

For this case, consider that malicious administrator is the attacker in the system. From the threat model in table 1, it can be seen that service is vulnerable to different threats such as evading provenance (T1) and malicious service (T2). Threat T1 can affect both the functionalities F1 and F2 of service MW. The threat exploitation methodology that T1 can use to affect F1 and F2 is "violation of policies by users using untrusted OS and applications", and T1 can affect F1 by "manipulating shared physical resources". Whereas, threat T2 can affect functionality F1 of service MW. The threat T2 can affect functionality F1 by "launching a malicious service", "malicious node giving untruthful data processing results" and "exploiting session misconfigurations".

3.5.2 Generalization

The basic criteria to determine the threats on services as mentioned in the above example can be generalized to figure out the possible threats for service functionalities in the cloud. The threats for any service functionality are the cross product of service functionality and possible attacker in the system. Therefore, this threat model can be used to determine the possible threats for a service functionality in a foreign cloud architecture and can also help in implementing security features to mitigate those threats. The summary of the possible threats to services, the methodologies to exploit those threats and the specific service functionalities affected by those threats are shown in table 1.

3.6 Summary

With the increase in cloud usage, the number of services running in cloud and data processed by those services has been rapidly increasing. As a result, cloud services have been targeted by various attackers. In this chapter, threat modelling approach has been presented to determine the threats for cloud services. Along with the important threats to services, this chapter also presents the methodologies to exploit those threats. Different possible attackers who can exploit specific threats have also been identified. The summary of the possible threats, attackers and the mechanisms to exploit those threats has been shown in a table. Threat modelling of services can be generalized to determine the threats for specific service interfaces. This generalized approach for determining threats for specific service functionalities can be used to determine a system for automated reasoning in which threats for each service interface are listed so that possible security mechanisms for those interfaces can be implemented. As a user access services in a foreign cloud during multi-cloud collaboration, this model can be used to determine specific threats for that services in relation to the foreign cloud architecture and service interfaces.

Threat	Possible Attackers	Threat Exploitation Methodology	
		Methodology	**Service Functionality Affected**
Data breach and loss	Outside attacker	Data deletion and corruption	Data storage
		Insufficient AAA controls	Authentication
		VM side channel attack	Data processing
		Malware attacks	Data processing, data storage
Evading provenance	Malicious insiders, malicious users	Violation of policies by users using Untrusted OS and applications	Data processing, data storage
		Manipulating shared physical resources	Data processing
Malicious service	Service provider, Malicious insider	Launching a malicious service	Data processing
		Malicious service node giving untruthful dataflow processing	Data processing
		Exploiting security misconfigurations including session, protocol configurations and varied security policies	Authentication, data processing
Malicious administrator	Cloud administrator	Software execution at customer VM	Authentication, data processing
		Sysadmin can login remotely to any machine running outside trusted machine	Authentication, data processing, data storage
		Extracting data from insecure cloud nodes	Data processing
		Rebooting a node and installing malicious software to eavesdrop on network	Authentication, data processing
Virtualization threats	Malicious user, outside attacker	VM Escape, BLUEPILL, Eavesdropping	Data processing
		Arbitrary code execution	Data processing
		Infecting user OS with arbitrary malware	Authentication, data processing
		Cross tenant attacks such as sniffing and spoofing	Data processing
		Privilege escalation attacks	Data processing
Network threats	Malicious user, outside attacker	Network attacks (Phishing, SQL-Injection, Cross-site scripting, DOS, DDOS)	Data processing

Table 3-1 Summary of the threat modelling

Chapter 4 Dynamic Collaboration and Authentication Model for Multi-Clouds

4.1 Introduction

In a multi-cloud environment, users access services across multiple cloud providers which changes the traditional cloud landscape. However, as discussed in the previous chapters very limited research has been done to support services deployment and access across multi-clouds. Therefore, advanced development frameworks are required that can offer services orchestration across multi-clouds and reduce companies time-to-market to keep cloud services running smoothly. A significant challenge associated with the adoption of multi-clouds is the lack of solutions and frameworks that can facilitate its usage.

Despite the widespread adoption of multi clouds and recent advances in this research context, the establishment of a secure dynamic collaboration between heterogeneous clouds participating in the multi-cloud infrastructure is still an open research problem. Standards such as NIST-FISMA and ISO 27000 do not cater the complexities of underlying cloud platforms. The framework aimed at aligning with FISMA standards is qualitative and does not take user requirements into account. Moreover, the frameworks based on NIST do not provide formal definition of proposed techniques in multi-cloud scenario. Furthermore, mechanisms for automated orchestration such as [83] do not offer end to end solution for multi-clouds and the performance and security evolution of proposed solutions is not discussed to prove their effectiveness.

The basic authentication solutions that exist for traditional networks fail to meet the need for a dynamic collaboration of clouds and services in multi-cloud. Some authentication solutions assume pre-established trust provision that does not satisfy our runtime multi-cloud collaboration requirements. Single Sign On (SSO) based authentication profiles for SAML have been proposed for cloud collaborations but they induce comparatively higher overheads. Moreover, other centralized authentication solutions discussed in chapter 2 are not suitable for multi-clouds as they bring performance and security issues and the scalability of these solutions to handle large number of requests is not discussed. Therefore, this chapter provides a novel de-centralised model for the dynamic and secure collaboration between multi-clouds.

The proposed model called Multi-Cloud Collaboration (MCC) model provides a novel technique for the dynamic authentication which offers single sign on to users trying to connect to the foreign cloud. The authentication solution proposed in MCC achieves better performance compared to traditional security protocols such as SAML and Kerberos while maintaining security. In the next stage, an extension of service level agreements (SLAs) establishment is proposed which helps to ensure security while user data and request is being handled in the foreign cloud. The usage of SLAs in the proposed model ensures that best services and provider are selected to handle the user requests and secure collaboration between multi-clouds is setup.

The proposed model is based on the NIST cloud computing security architecture standard [120]. It satisfies the following conditions: i) Rapid provisioning by automated service deployment; ii) Mapping authenticated and authorised data and tasks onto VMs; iii) Monitoring the cloud resources, operations and performance; iv) Metering active user accounts to guarantee that security policies are always enforced; v) Maintaining the service level agreement (SLA) established between customers and service providers.

The functionalities of components responsible for multi-cloud collaboration in MCC are presented in detail. Moreover, the proposed collaboration protocols in our research are verified using the BAN Logic [121]. The experiments to consider the scalability and overhead of proposed techniques are performed on different cloud setups based on OpenStack [122]. These experiments indicate the proposed approach supports collaboration among a large number of services across multi-clouds and incurs a minor overhead. The major contributions presented in this chapter include the following:

1). A novel model MCC is presented for establishing multi-cloud collaboration. The protocols to support the model and the functionalities of its components responsible for multi-cloud collaboration are presented in detail. Moreover, it is discussed how the proposed model meets the multi-cloud requirements.

2). Dynamic and lightweight authentication protocols between multi-clouds have been presented. We propose a single sign on (SSO) authentication mechanism by which any cloud user in the local cloud can authenticate itself with the foreign cloud, and access required resources. Moreover, the proposed model uses service level agreement (SLA) mechanism to guarantee that best service provider is selected with respect to client requirements, and the service execution in the foreign cloud is according to the agreed SLA parameters.

3). The proposed authentication and collaboration protocols have been formally verified using BAN Logic [121], which is a logic used to reason about beliefs, encryption, and protocols.

4). Experiments have been carried out that show system is scalable and incurs only a minor overhead. Our results show that the proposed authentication mechanism performs better than traditional the authentication protocols like SAML [123] and Kerberos [124]. Moreover, the system can monitor any SLA violations by the cloud provider and detect them quickly.

4.2 Multi-Cloud Collaboration Requirements

Researchers have identified the usage of cloud according to three major categories. First is the monolithic environment that involves designing and developing platform independent applications. Second is the "Horizontal Expansion" which involves federated clouds which as described earlier involve setting up federation between multi-clouds, while third is the usage of multi-clouds as "Vertical Supply Chain" that includes cloud providers leveraging services from other clouds. This chapter describes establishing collaboration between heterogeneous cloud environments in "Vertical Supply Chain" configuration.

Multi-clouds provide a way for dynamic collaboration between various clouds as there is no former agreement between participating clouds and collaboration is established at runtime according to requirements. Moreover, in multi-clouds user has the knowledge of all connected clouds and is directly responsible to the provisioning of services from multiple clouds which can be more beneficial from customers and organizations perspective. Therefore, this work focuses on multi-clouds so that users and organizations can dynamically access services across various multiple cloud providers. As private clouds offer services to their clients using locally hosted infrastructure, using the proposed model can enable them to enlarge their own offers by supporting access to other cloud providers.

4.2.1 Collaboration Objective

The objective of multi-cloud collaboration is that the proposed model MCC should be able to handle maximum number of user requests from local cloud and successfully grant them access to required services in the foreign cloud.

To achieve this objective, we define an objective function for multi-cloud collaboration in which M cloud users (j = 1, 2 … M) in local cloud have requested access to S services (i = 1, 2 … N) in a foreign cloud. This can be formally defined as:

$$\mathbf{O\ (i,\ j) = Min}\ \sum_{i=1}^{N} \sum_{j=1}^{M} (R_{ij} - C_{ij})$$

In the above equation, N is the numbers of services requested and M is the number of users making requests respectively. R_{ij} is the required number of user’s requests for services to be granted while C_{ij} is the number of services that were actually granted by MCC. We want the objective function to be as low as possible, and in an ideal case it should be zero to show all user requests have been handled and they have been allocated services in foreign clouds.

4.3 Multi-Cloud Collaboration (MCC) Model

Based on heterogeneous requirements of multi-clouds, we propose a novel model MCC that can enable dynamic collaboration between users and services in multi-clouds. It acts as a possible marketplace between two clouds to set up communication, perform authentication, match SLAs, and handle security. Therefore, MCC has a greater awareness to correlate the services being offered with customer requirements and acts as a trusted advisor to the proposed system so that security features such as compliance and availability are considered. MCC also acts as a possible marketplace between two clouds to set up communication, match SLAs as well as negotiate and handle security.

The proposed model Multi-Cloud Collaboration (MCC) provides extended functionalities to organizations in which they can apply their business models. MCC has been designed with the goal to enhance secure multi-cloud collaboration in which cloud providers can easily apply their business model. It is based on an architectural solution that can be setup multi-cloud collaboration between any clouds irrespective of their underlying implementation. The architecture of the proposed model involves various components that have been implemented in each participating cloud to achieve the secure multi-cloud authentication and collaboration. These components involved in model design are authenticator for managing identity and authorization, and SLA coordinator for managing SLA negotiation, enforcement and authorization.

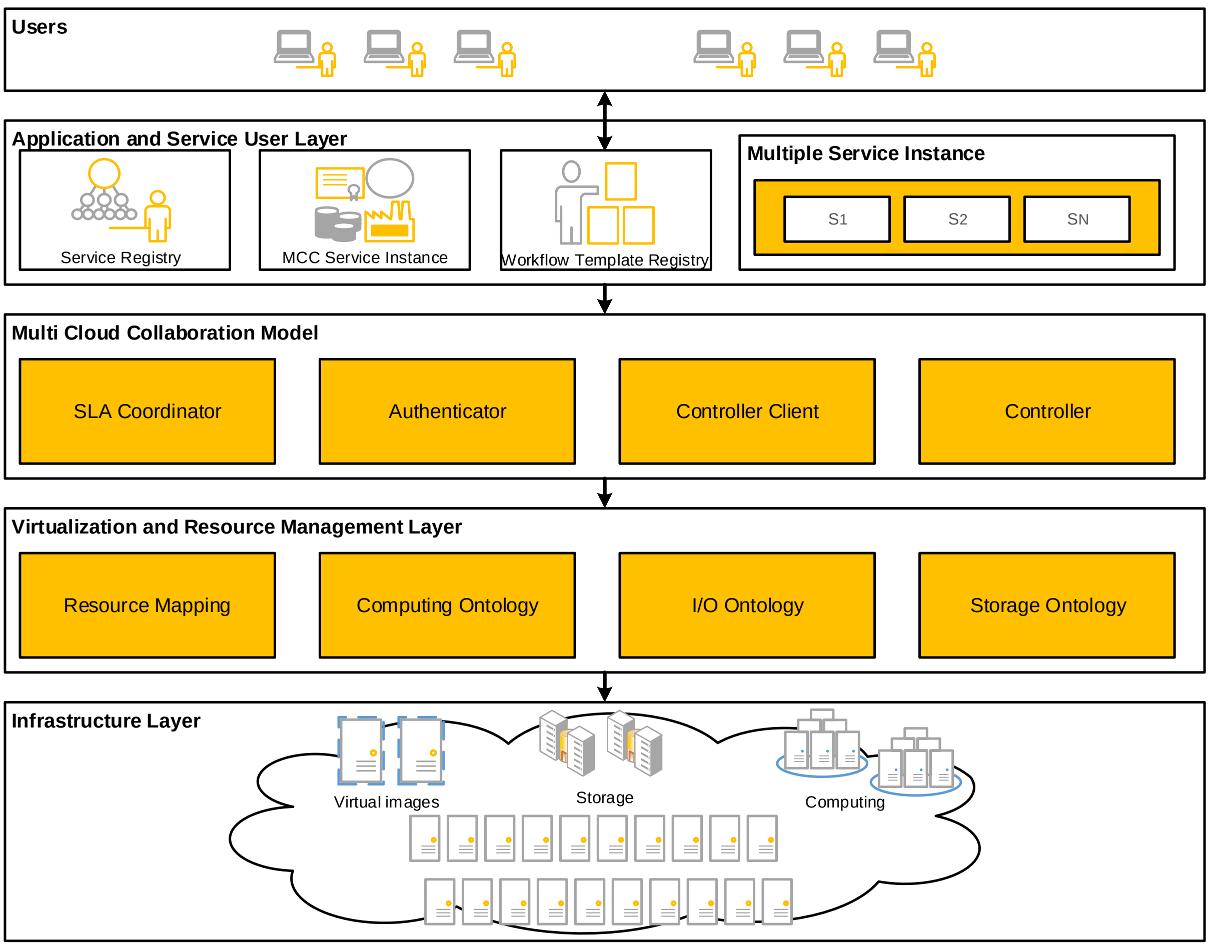


Figure 4-1 The proposed Multi Cloud Collaboration model architecture

The figure 4.1 displays various components in a MCC that are used to communicate and collaborate among multi-clouds. All the system components serve different functions which are described below. In this chapter, the communicating clouds are referred to as “local cloud” in which the user is located and “foreign cloud” to which user needs access and collaboration has to be established.

4.3.1 MCC Orchestration and Components

In this section, the details of MCC’s components and their functionalities are discussed. The protocols to support the model are also presented in this section. The details of how all components interact and MCC components integration with each other and workflow are explained in section 4.3.2.

4.3.1.1 MCC Initialization Protocol

The initialization protocol is the first step in MCC orchestration that is used to set up the system services, parameters and attributes required for multi-cloud collaboration. This protocol discusses how certificates are set up and MCC is initialised after receiving client request.

When the required services are booted in the model, and user request for multi-cloud access is received, authentication service in the Authenticator component establishes if it has the certificate for that user that can be used for authentication with foreign clouds. If the certificate does not exist in the cloud, Authenticator which is a RESTful web service submits a request on behalf of its cloud to Trusted Party (TP) for certificate generation. Trusted party is responsible to generate a certificate for cloud after receiving a request and cloud parameters and to use a function to map a certificate to client ID which is returned to the requesting cloud. The Authentication Service of cloud receives the certificate and stores it to be used for communication with foreign clouds.

Algorithm 1: System Initialization ()

```
1. BEGIN: Boot the required services to enable multi-cloud collaboration
2. while (the system is running)
3.      LC (Auth_service) -> Check (Cert)
4.      if Valid (Cert):
5.           goto 17
6.      else:
7.           Auth_service -> Send_request (Cert, ID) -> TP
8.      end if
9.      for i = 1 to n:
10.          LC -> Mapping_data(LC) -> TPi
11.          TPi -> Publish(Cert)
12.          TPi -> Send_generated_certificate (Cert) -> LC
13.          LC (Auth_service) -> Receive (Cert)
14.     end for
15.     LC (Auth_service) -> Check (Cert)
16.     if Valid (Cert):
17.          wait (Request)
18.     else:
19.          goto 7
20.     end if
21. END
```

4.3.1.2 MCC Authentication Protocol

The existing authentication solutions for multi-clouds have various limitation such as inability of standards (NIST, FISMA) to satisfy multi-clouds requirements, performance issues with SAML and scalability issues with centralised solutions. To address these issues, we propose an authentication protocol that describes how multi-cloud authentication is set up between participating clouds using MCC.

In a distributed multi-cloud environment, a large number of clouds are present with each cloud having tens of users, which makes credential management a big challenge. Moreover, in a dynamic collaboration setup between multi-clouds, each cloud might have different authentication mechanisms. This raises a need to develop a lightweight sign on (SSO) authentication mechanism by which any cloud user in the local cloud can authenticate itself with the foreign cloud, and access required resources. In MCC, for authentication orchestration we use a Trusted Party (TP) which acts as an identity provider on which a requesting user must hold a digital identity, based on which TP grants a digital certificate to that user that it can use to authenticate with the foreign cloud. Since the foreign cloud also trusts TP, the user is able to authenticate itself and access resources based on that certificate.

We assume that the local cloud's request is composed of two parts namely the certificate and the required cloud service. Initially, a certificate is sent by the local cloud (LC) to the foreign cloud (FC) for proving its identity. This certificate contains a set of attributes including the cloud identifier, digital signature, and validity period of the certificate. This message is encrypted by the public key of FC.

Foreign cloud initially checks the validity of the certificate sent by the local cloud. If the certificate is valid, FC then sends a response message to LC that it is authenticated. However, if the certificate is invalid, FC sends the message of failed authentication to LC and waits for a new certificate. This message is encrypted with the public key of LC. In case the message received from FC is that the authentication certificate was invalid, LC sends a message with its credentials to the Trusted Party (TP) to generate a new certificate. TP generates a new certificate and sends it to the LC which is sent from LC to FC.

FC checks the new certificate received from LC. If the certificate is invalid again, the authentication request is terminated. If authentication of LC is successful, both FC and LC exchange nonce messages to agree on a session key using Diffie-Hellman (DH) algorithm [124]. Since DH key exchange is performed after certificate exchange, it is called authenticated DH which is more secure compared to usual DH.

After cloud authentication, LC sends a message to FC containing client authorization details as well as resources required from FC. As FC receives details of cloud services which are to be accessed and required resources message, it locally computes if the tasks from LC have the authorization to access the required services. The corresponding FC computes the status of the IoT services associated with the request. The status is computed due to the fact that users on a local cloud can have the different status of privileges that can affect their level of access to resources. For example, only privileged users might have access to some expensive services and other users might not have access to them. The return result is one of the following possible statuses:

- Privileged
- Non-privileged

If the result returned is privileged users are granted access to services, otherwise they are not granted access.

Algorithm 2: Authentication and Authorization ()

```
1. BEGIN
2. Data: request: Communication request received by cloud controller
3. LC -> Send_request (authentication) -> FC (secured using SSL)
4. for j = 1 to n do:
5.         FC -> Verify (Cert, ID)
6.         if Verify (Cert, ID):
7.                 goto 17
8.         else:
9.                 FC -> Send_request (New_Cert) -> LC
10.        end if
11.        LC -> Send_request ((Cert),Profile) -> TP
12.        TP -> Send (Cert) ->CC [generates updated certificate for LC and sends to
           LC]
13.        LC -> Send_msg (Cert) -> FC
14.        if Not_valid (Cert):
15.                End
16.        else:
17.                FC -> Send_msg(n) -> LC
18.        end if
19.        FC -> Wait (response) -> LC
20.        if no_resp():
21.                End
22.        else:
23.                LC -> Send_msg(n+1) -> FC encrypted using LC- FC session key
           generated by DH
24.                FC -> Send_msg(request_authorization) -> LC encrypted using LC-
           FC session key
25.                LC-> Send_msg(Send_LCAuthorization_level) -> FC encrypted using
           LC-FC session key
26.                FC -> compute_local_level (LC)
27.                if compute_local_level = True:
28.                        FC->Authentication_local (LC,FC,+)
29.                else:
                           FC ->Authentication_local (LC,FC,-)
30.                end if
31.        end if
27. end for
28. END
```

4.3.1.3 SLA Negotiation

A key feature of MCC orchestration is the SLA co-ordinator components in MCC that receives user requirements and SLA's from the foreign cloud and negotiates a dynamic SLA between them. SLAs are key part of MCC to guarantee that users quality of service (QoS) requirements are satisfied. Various mechanisms exist to ensure service provisioning in multi-clouds with minimum SLA violations. The SLA co-ordinator in MCC is implemented as a middleware layer with the proposed system to facilitate communication between multi-clouds through adaptability and rapid response. SLA assurance is offered through negotiation, monitoring and enforcement stages by using lightweight mechanisms that are efficient and secure for multi-party collaborations across multi-clouds. SLA Negotiation involves agreeing on the service terms for SLA and QoS parameters, measuring metrics (service level objectives) and defining how the metrics will be measured. While service providers also check if they can provide requested service and perform basic risk evaluation in case. Moreover, the proposed threat model in chapter 3 can be used to highlight and determine possible threats, and service functionalities that attackers are most likely to target for taking precautionary measures to prevent such events.

The proposed model uses WS-Agreement [125] to express the functional security requirements and non-functional requirements requested by the client. The XML data structures are generated on the basis of WS-Agreement document, the service interface definition and its implementation. Therefore, QoS tags are associated with a new category to recognize security and other properties. To implement the negotiation, WS- Agreement Negotiation has been used. The protocols are implemented in the form of a REST based service and API.

4.3.1.4 SLA Enforcement

Once a user is authorized to access cloud resources, the next stage is the enforcement of security mechanisms by the provider. In this stage, mechanisms are implemented that can guarantee

SLA assurances. The proposed model focuses on the implementation of mechanisms for non-functional properties to ensure that service complies with the defined SLA policies. QoS parameters mentioned in SLAs are measured by maintaining current system configuration information and runtime information of parameters that are part of SLOs.

4.3.1.5 SLA Monitoring

Currently, no solutions exist to check for SLA compliance for user's support. Monitoring involves, i). verifying that SLAs are followed through infrastructure access, and ii). generating alert notification if the SLAs are violated to take corrective steps. In order to implement these functionalities, continuous monitoring techniques can be applied that employ IaaS monitoring techniques. The proposed model uses event driven modules to collect all generated events and performs required filtration operations before analysing them. Based on the captured events and their analysis, monitor informs the local cloud if the foreign cloud is compliant with the signed SLA or not. In case of SLA violations, user can enforce penalties on the provider.

4.3.2 MCC Components Integration

This section explains the workflow for the overall model and the implementation details of various components as shown in figure 4.2. A user request to connect to the foreign cloud is received by cloud controller that advertises request to connect to multiple foreign clouds and receives their responses. The client details, cloud responses and received SLAs are handled by cloud controller to choose the best offering in terms of quality of service.

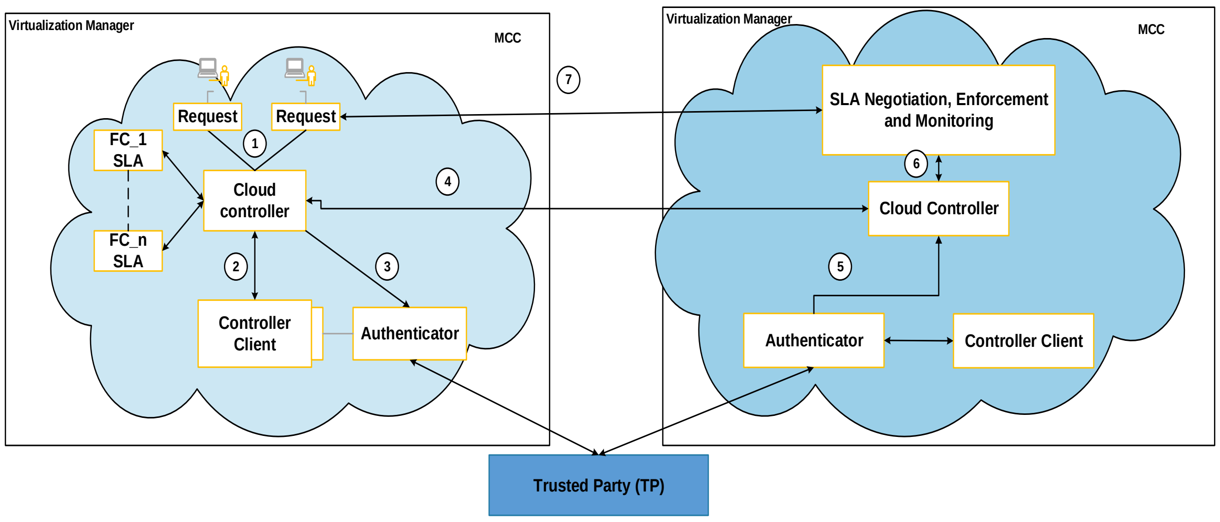


Figure 4-2 Workflow of the proposed model

4.3.2.1 Cloud Controller

This is the major component responsible to handle the multi-cloud communication and authentication. This component provides accelerated connection to market by combining customer requirements, checking available services and connecting with the foreign cloud. The controller is implemented as RESTful web service in cloud and its responsibilities are two folds. First in the local cloud when it wants to access a foreign cloud and second in a foreign cloud when a connection request is received.

When a user in local cloud needs to access service in a foreign cloud, it is the responsibility of a controller to establish a connection with the other foreign cloud. Before sending a message to the foreign cloud, it communicates with the local authenticator component to get the certificate. After sending authentication request, on behalf of local cloud it establishes the communication channel by sharing session keys.

In a foreign cloud, requests for communication from the local cloud are received by cloud controller. Cloud controller is then responsible to check whether, (a) the requested service is available in the foreign cloud, (b) the connecting local cloud is trustworthy, and (c) respond to the foreign clouds request.

4.3.2.2 Cloud Controller Client

This component is responsible to manage the local services and resources in a cloud. Once a cloud controller receives the communication request in a foreign cloud, it sends a message to controller client to check the availability of requested service. The controller client matches the request with available resources and returns the response about resources status to the controller. Similarly, when the local cloud wants to communicate with a foreign cloud, the controller client is responsible to check the details of a client who needs to access foreign cloud and pass them to the controller.

4.3.2.3 Authenticator

Authenticator component is responsible to manage the authentication of multi-clouds. Once the communication request from the local cloud reaches the foreign cloud, cloud Controller of foreign cloud connects with the authenticator to verify if the connecting (local) cloud is trusted or not. When the authenticator component receives the message containing the identity of local cloud and its digital certificate, it checks whether the certificate is valid and responds to controller component. Based on the response from the authenticator, cloud controller of foreign cloud responds to the cloud controller of the local cloud.

In a local cloud, when a collaboration request is to be sent to foreign cloud authenticator is responsible to contact trusted party (TP) which generates the certificate for the local cloud, signs it and returns it to the local cloud. Before sending a communication request to a foreign cloud, local cloud controller gets its certificate from the authenticator.

4.3.2.4 Trusted Party

Trusted party (TP) is the identity provider responsible to handle the authentication among multi-clouds. It has a list of trusted cloud providers, and before establishing session the connecting clouds communicate with it to acquire their certificate. After receiving a certificate request, it generates a certificate, signs it with its private key and returns it to the requesting cloud. Any cloud registered with a TP receiving a certificate signed with a private key of that particular TP considers it true.

It is the responsibility of the Authenticator component to ensure that the certificate obtained from TP is valid and to get a new certificate if the existing one is revoked or rejected by foreign cloud.

4.3.2.5 SLA Co-Ordinator

The usage of security mechanisms using SLA ensures that best services and provider are selected and secure collaboration between multi-clouds is setup. The foreign cloud provider handling user requests will be responsible to enforce mechanisms that protect the security properties given in multi-clouds. The next phase in SLA involves monitoring the service security properties in the foreign cloud and report the SLA compliance to the local cloud.

The key requirements from SLAs have been defined are:

- Receive request from the controller and associated SLAs to negotiate the best possible provider
- Review service availabilities from various providers
- Generate dynamic SLA's for the user and foreign cloud agreement
- Monitor service execution at the foreign clouds
- Prevent SLA violations by taking responsive action in case of SLA's violations by informing cloud provider

SLA-Coordinator is responsible to manage SLA's in the proposed model. It initially selects the suitable service for a client in local cloud based on his requirements using negotiation. Once a foreign cloud provider has been selected, security and QoS parameters are negotiated. The enforcement component is responsible to ensure that service execution in a foreign cloud is according to the QoS parameters agreed in the SLA. Moreover, the monitor is responsible to ensure that the service used by cloud provider complies with the SLA and in case there is a violation of SLA it reports that violation to the service provider.

4.4 Formal Verification of MCC Protocols

To establish collaboration across multi-clouds, we propose different further protocols in this section. These protocols represent the set of messages which will be transported across different entities to support authentication and collaboration. Cloud A in these protocols represents the local cloud and Cloud B represents the foreign cloud. Both these clouds have certain components that support the exchange of messages between them. The detailed design of these components responsible for multi-cloud communication is shown in section 2.

These protocols are verified using the formal logic called BAN (Burrows–Abadi–Needham). BAN logic is used to reason about beliefs, encryption, and protocols. BAN logic consists of three stages to analyse any protocol which are, (i) to express the initial assumptions, and goals as statement to translate them to symbolic notations, (ii) to verify the goal whether the goals are in fact reached, and (iii) to perform a group of rules for acquiring the authentication goal.

4.4.1 Definitions

The authentication protocols proposed in this chapter can be verified using the logic of authentication called BAN logic. Followings notations are used for formal definitions in protocols:

A, B, C	Three separate multi-clouds identified as A, B and C
ID_X	Identifier of X

S	Session key
SA	Session authority
Pri(X)	Private key of X
Pub(X)	Public key of X
SA (A,S)	Statement that A and S are session partners
$K_{(A,B)}$	Secret key generated by Pri(A) and Pub(B)
$B \mid\equiv X$	B believes statement X is true
$A \xleftrightarrow{K^{-}_{(A,B)}} B$	A still does not identify if B knows $K_{(A,B)}$
$A \xleftrightarrow{K^{+}_{(A,B)}} B$	B has sent confirmation A that it knows $K_{(A,B)}$
#M	Message M is fresh

4.4.2 BAN Rules

Rule 1. $A|\equiv (X, Y) \Longrightarrow A|\equiv X$ and $A|\equiv Y$,

A believes a set of statements if and only if A believes every individual statement, respectively.

Rule 2. $A|\equiv \#M \Longrightarrow A|\equiv \#(M, N)$ and $A|\equiv \#(N, M)$

If a part of the message is believed to be fresh than the whole message is considered fresh.

Rule 3. $A|\equiv B| \Longrightarrow X,\ A|\equiv B\ |\equiv X \Longrightarrow A|\equiv X$

If A believes that B has a control over statement X, and if A believes that B believes X, then A should believe X.

Rule 4. $A|\equiv \uparrow Pub\ (A),\ A|\equiv \uparrow Pub(B) \Longrightarrow A \overset{K^{-}_{(A,B)}}{\longleftrightarrow} B$

If A has B's public key and believes that the public keys of A and B are both good, A can believe that the secret is shared with no party other than B although unconfirmed.

Rule 5. $A|\equiv A \overset{K^{-}_{(A,B)}}{\longleftrightarrow} B$, A sees $[X]_{K(A,B)}$,

$A|\equiv \#X,\ \Longrightarrow\ A|\equiv A \overset{K^{+}_{(A,B)}}{\longleftrightarrow} B$

If A believes the secret $K^{-}_{(A,B)}$ is shared with no party other than B (but A does not know B knows) and X encrypted by the secret is fresh, then A can believe that the secret is confirmed by B.

Rule 6. A$|\equiv A \xleftrightarrow{K^{+}_{(A,B)}} B$, A sees X, A$|\equiv$ #X, $\Longrightarrow$ A$|\equiv B| \equiv$X

If A believes that the secret is shared with no party other than B and is confirmed, and X is fresh, then A can believe that B believes X.

Rule 7. A$|\equiv B \overset{K}{\leftrightarrow}$A , A sees $[X]_K \Longrightarrow$ A sees X,

If A knows the secret shared with B and see X encrypted by the secret, then A can see X.

4.4.3 Session Establishment Protocol

This protocol serves the purpose for a user (user A) of local cloud (cloud A) that wants to access services running in the foreign cloud (cloud B). Firstly, user A sends a message to its cloud controller service that it wants to access a service in cloud B. The controller service of cloud A sends a request to cloud B where it is received by the controller to establish a connection. Controller service of cloud B verifies the integrity of the message received from cloud A and authenticates user A using its unique ID and certificate. If the certificate is valid, cloud B responds to cloud A with authentication success message, which is followed by both clouds determining a shared secret key using Diffie Hellman key exchange algorithm. Using, this key session between multi clouds is established in which users from cloud A can communicate directly with cloud B. This protocol is shown in figure 4.3.

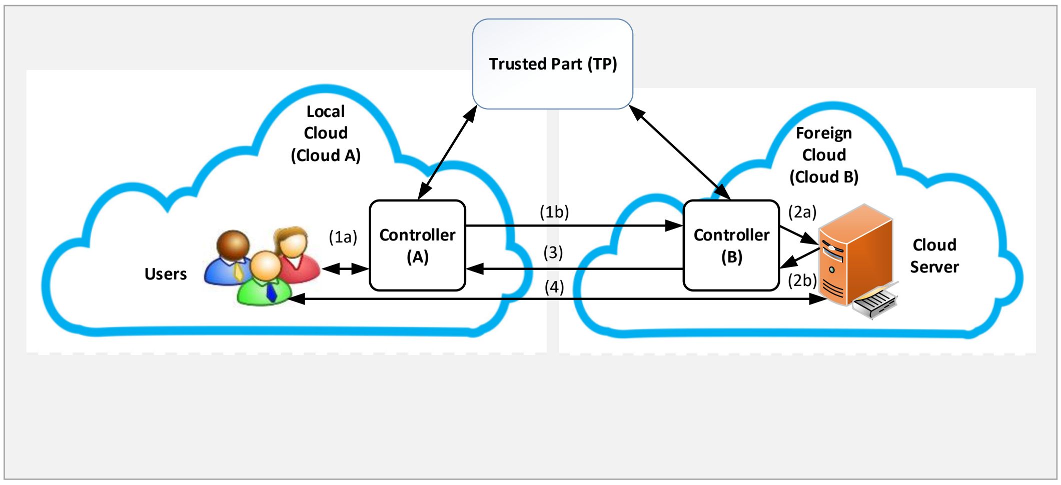


Figure 4-3 Session Establishment Protocol

Details of messages in the figure 4.3:

(1a) Local cloud (cloud A) user request to access service in foreign cloud (cloud B)

(1b) Controller of local cloud (cloud A) sends a request to controller service in cloud B to access resources

(2a) Cloud B verifies credentials of cloud A user making request to access resources

(2b) Verify credentials of cloud A user and send response to controller

(3) Forward authentication response to controller of cloud A. There are two possibilities in this case.

a. In case of successful authentication: Controllers of cloud A and B agree a session key using Diffie Hellman Key Exchange process and establish communication
b. In case of unsuccessful authentication: Cloud A controller is notified and it can make request by acquiring new certificate from Trusted Party or close communication.

(4) User is authenticated to access resources in foreign cloud

4.4.3.1 Verification

1. A$\rightarrow$B: [Request, $ID_{A,}$ $Cert_A$, N]

2. B$\rightarrow$A: [Verify, ID_B, ID_A, N_0]

3. A$\rightarrow$B: [Confirm, SP(A,S), ID_{SA}, ID_B, ID_A, $N_1]_{K(A,B)}$

4. B$\rightarrow$A: [Reply, ID_A, ID_B, $N_2]_{K(B,A)}$

The messages 1 and 2 are protected by SSL while 3,4 are encrypted using secret keys using Diffie Hellman key exchange algorithm.

The goals of the protocols can be described as follows:

(1) Verifying identity of A who wants to establish a session, and accept it as a session partner which can be described as: $B|\equiv \xrightarrow{Pub(A)} A$

(2) Building confirmed secret key to be shared between A and B which can be described as: $A \mid\equiv B \mid \equiv K_{(B,A)}$

The assumptions of the protocol can be described as follows:

$A|\equiv\uparrow Pub(B), B| \equiv\uparrow Pub(A)$ which means that A and B believe the keys of others are secure

$B|\equiv\uparrow Pub(B),\ A| \equiv\uparrow Pub(A)$ which means that A and B believe their private keys are secure

$B|\equiv \#N,\ B| \equiv \#N_1,\ B| \equiv \#N_2$ which means that entities believe the Nonce N_1 and N_2 are fresh.

Theorem 1: The goals of the protocol for cloud authentication are satisfied under the assumption of this protocol.

Proof: It is needed to deduce that: $B \equiv \xrightarrow{Pub(A)} A$ and

$A \mid\equiv B \mid \equiv K_{(B,A)}$ from the assumptions of the protocol.

From the following assumptions: $B\mid \equiv\uparrow \mathrm{Pub(B)}$ and $B\mid \equiv\uparrow \mathrm{Pub(A)}$ ……… (a)

First goal is achieved in the following way:

- From equation (a) and message 1 it can be derived from that, $B \mid\equiv [\mathrm{Request}, \mathrm{ID}_{A}, \mathrm{Cert}_{A}]$
- Similarly from the assumptions: $B \mid\equiv \#N_1$
- It follows that, $B \mid\equiv \#[\mathrm{Request}, \mathrm{ID}_{A}, \mathrm{Cert}_{A}]$ by

  Rule 2

- $B \mid\equiv \# [\mathrm{Cert}_{A}]$ by Rule 2

The **second goal** is achieved in this way:

- Equation (a) follows that $B \mid\equiv B \xleftrightarrow{K^{-}_{(B,A)}} A$ by

  Rule 4 …..(i)

- Similarly, from the assumptions: $B \mid\equiv \#N_1$
- It follows that $B \mid\equiv \#[\mathrm{ID_A},\ \mathrm{ID_B}, \mathrm{SP(A,S)}, \mathrm{N_1}]$ by Rule 2 …….(ii)
- By using the equations (i), (ii) it can be deduced that

  $B \mid\equiv B \xleftrightarrow{K^{+}_{(B,\ A)}} A$ by Rule 5 ⋯⋯(iii)

- Using Rule 6 and eq. (iii) it can be deduced that:

  $A \mid\equiv B \mid \equiv K_{(B,A)}$

4.4.4 Session leaving Protocol

This protocol is defined for the case when a user in local cloud (cloud A) wants to leave the session with foreign cloud (cloud B) and end its communication with the foreign cloud. Firstly, the user A from cloud A (which is using cloud B services) will send a request to its controller service that it wants to leave the session. When controller service of cloud B receives request from cloud A to close the session, it verifies the integrity of cloud A who sent the message and closes the session between multi-clouds.

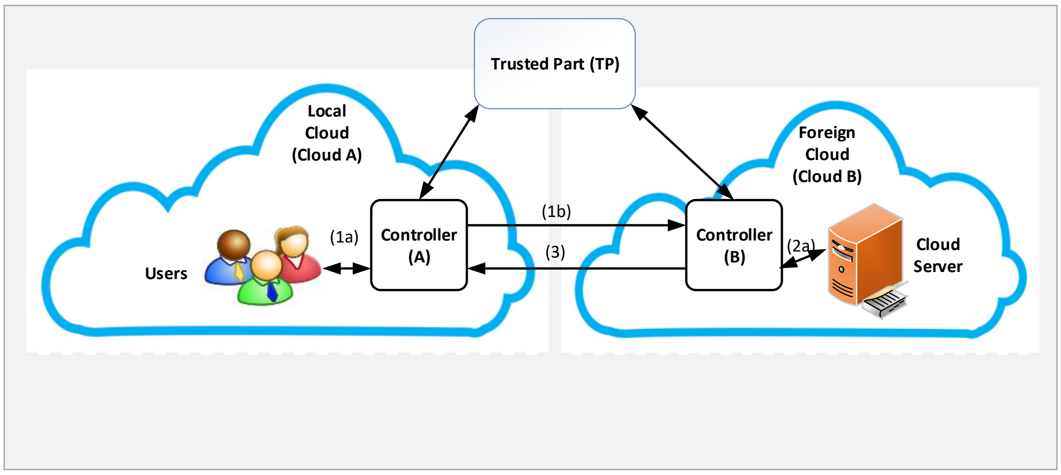


Figure 4-4 Session Leaving Protocol

Details of messages in this process shown in figure 4.4 are as follows:

(1) Local cloud (cloud A) user request to close session with resources in foreign cloud (cloud B)

(2) Controller of local cloud (cloud A) sends a request to controller service in cloud B to close existing session of that user

(3) Accept request to close a particular user’s session in Cloud B

(4) Foreign cloud (cloud B) closes existing session, removes the allocated resources and its controller sends the confirmation of closing session to the controller of cloud A.

4.4.4.1 Verification

1. B->A: $[ID_B, N]_{K(B,A)}$

2. A->B: $[\text{Request}, \text{Exit}(A,S), ID_A, ID_B, N_1]_{K(A,B)}$

3. B->A: $[ID_B, ID_A, \text{Rm}(S), N_2]K_{(B,A)}$

The goals of the above protocol can be described as:

(1) Request to remove A from session S which can be labelled as: $B| \equiv \text{Exit}(A, S)$

(2) Destroying session key and notifying A that session has been removed. This can be described as: $A \mid \equiv B \mid \equiv \text{Rm}(S)$

The assumptions of the protocol can be described as follows:

$A|\equiv\uparrow \text{Pub}(B), B| \equiv\uparrow \text{Pub}(A)$ which means that A and B believe the keys of others are secure

$B|\equiv\uparrow \text{Pub}(B),\ A| \equiv\uparrow \text{Pub}(A)$ which means that A and B believe their private keys are secure

$A|\equiv \#N_1,\ B| \equiv \#N_1,\ A| \equiv \#N_2$ which means that entities believe the Nonce N_1 and N_2 are fresh.

$A \mid \equiv B| \equiv \text{Exit}(A, S)$ which means that A and B believe they can control leaving the session

The first goal is achieved in the following way:

- From message 1 and Rule 7, $B |\equiv B \xleftrightarrow{K^{+}_{(A,\ B)}} A$ ---(i)
- By Rule 2, $A |\equiv \#[\text{Request}, \text{Exit}(A,S), ID_A, ID_B, N_1]_{K(A,B)}$ …… (ii)
- From (i) and (ii), $A \mid \equiv B| \equiv [\text{Request}, \text{Exit}(A,S), ID_A, ID_B, N_1]_{K(A,B)}$ ….. (iii)
- From Rule 1 and (iii), it can derived that:

  $A \mid \equiv B| \equiv \text{Exit}(A,S)$ …… (iv)

- Using Rule 3 and eq. (iv) it can be derived that:

$B| \equiv$ Exit(A,S)

The second goal is achieved in the following way:

- From the following assumptions: $A| \equiv\uparrow \text{Pub}(B)$ and $A| \equiv\uparrow \text{Pub}(A)$ ……… (b)
- $A |\equiv A \xleftrightarrow{K^{-}_{(A,B)}} B$ by Rule 4 …..(iv)
- From eq (iv) and Rule 7, $A |\equiv [ID_B, ID_A, Rm(S), N_2]$ ………… (v)

  By the assumption $A| \equiv \#N_2$ and eq (v),

- By Rule 2, $A |\equiv \# [ID_B, ID_A, Rm(S), N_2]$ ………… (vi)
- From equations (iv), (v), (vi) and Rule 6 it can be derived that,

  $A | \equiv B| \equiv [ID_B, ID_A, Rm(S), N_2]$ …….. (vii)

- By Rule 1, eq (vii) leads to $A | \equiv B| \equiv Rm(S)$

4.5 Business Case

MCC can enable participants in the ecosystem to takes various roles depending on the situation. For example, a user can turn into a provider whenever it has spare resources by offering them on the market. MCC can be used to enable users of a cloud platform to access services in another cloud. There are many other business cases of this model that can help to improve the business supply chain. Consider a case in which an organization named E-Packagers is using cloud resources and services on a cloud service provider. The company has set up a cloud and they require its resources during peak times between 9 am to 5 pm on working days and usage of these resources and their services on weekends is close to none. In this scenario, E-Packagers cloud resources and services will be idle and their usage will be really low. However, using MCC the company can further lease its services to be used by users from other clouds who can directly contact E-Packagers and use their services for a certain time without their cloud provider interaction. This can help the company to generate additional revenues and users to access services with lesser conditions in less time.

4.6 Experimentation

To examine the feasibility of the proposed design empirically, it was implemented in two different clouds. The purpose of these experiments is to assess the, (i) scalability of the proposed system and (ii) runtime overheads of the system during collaboration between multi-clouds. The performance of authentication protocols is compared with the standard authentication protocols. Furthermore, we discuss developing MCC cloud controller as a modular service by splitting its functionality into message queues and networking service that can avoid performance bottlenecks and enable MCC to handle large number of requests coming from clients during collaboration.

The prototype was tested on two different cloud infrastructures. One of the cloud infrastructures was an OpenStack cloud based in University of Derby. This setup consists of six server machines. Each machine has 12 cores with two 6-core Intel Xeon processors running at 2.4 GHz with 32 GB RAM and 2 TB storage capacity. The cloud nodes on which the experiments were performed had 4 VCPUs running at 2.4 GHz each, 8 GB RAM, and data storage of 100 GB per node. The second cloud was also based on OpenStack and it was set up on machine with Intel Core(TM) i7-4790 processors running at 3.6 GHz with 16 GB RAM and 1 TB storage capacity. The cloud nodes on this machine had 4 VCPUs, 8 GB RAM and 100 GB storage.

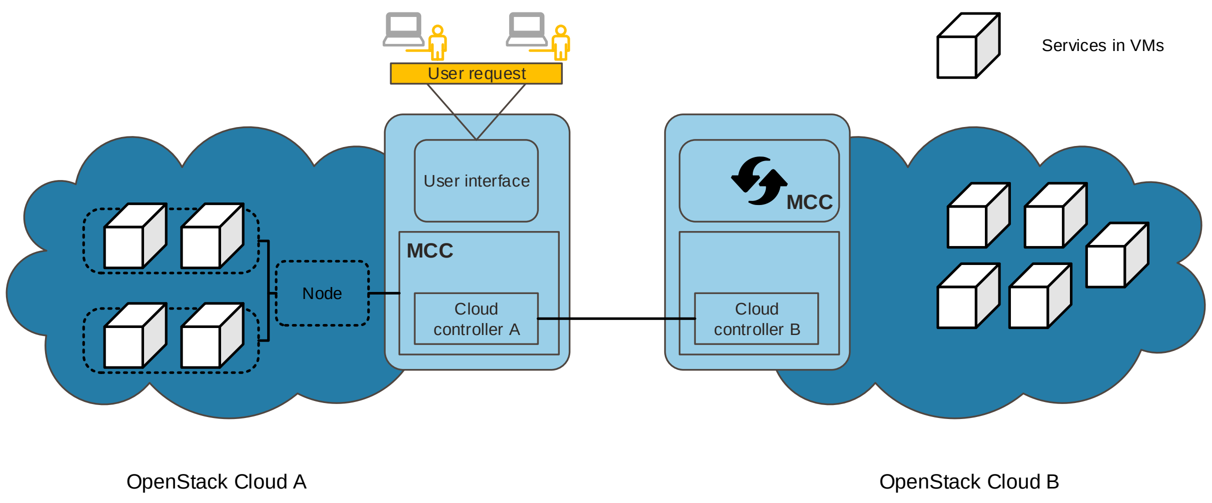


Figure 4-5 Experimental Setup

Both the Cloud Controller and Cloud Authenticator are employed as web services which help in avoiding tightly bound security. While WS-Agreement was used to implement the SLA components. To enable the interaction among components in prototype according to the proposed system, cloud controller of local cloud submits requests for resources to other foreign clouds. When the foreign cloud controller initializes and receives a request for available services from a local cloud, it shares exchange information about available services and their characteristics.

In the experiments, to check the scalability of the system initially a large number of service requests were created in local cloud so that they can be connected to multiple instances in foreign cloud. To start the communication, cloud controller from a local cloud invokes the cloud controller in foreign cloud. This is then followed by various operations in foreign cloud including checking the availability of the required services, verifying if the local cloud user that wants to connect is authorized and SLA negotiation to agree the functional and non-functional requirements of services that need to be satisfied. After performing authentication and communication among multiple instances, a large number of users from local cloud were able to request for multi-cloud collaboration and access service instances in the foreign cloud, and those instances were generated according to negotiated SLA parameters.

To evaluate the overhead caused by protocol, the time taken by different operations was calculated. The time taken by different instances during authentication with instances in the foreign cloud using the proposed system is shown respectively in figure 4.6.

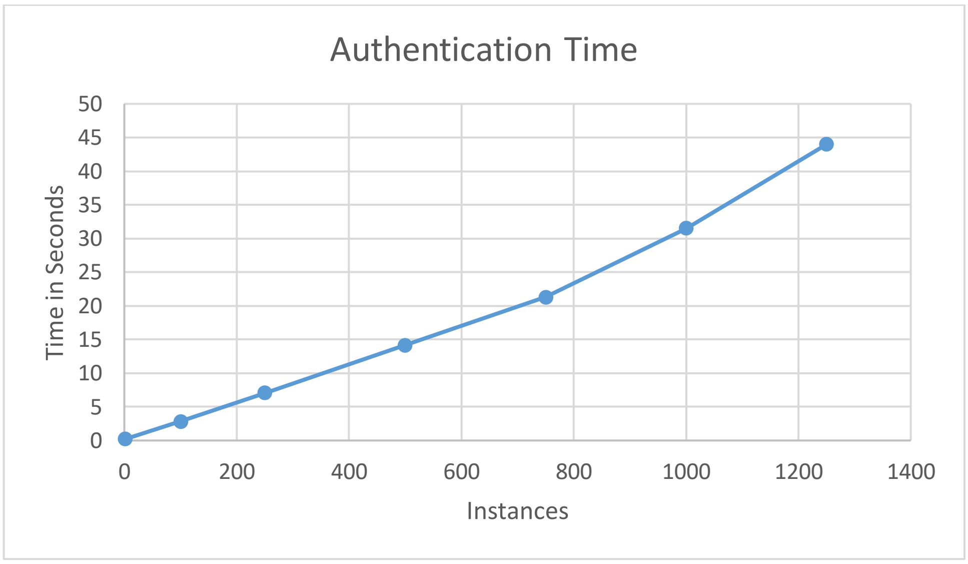


Figure 4-6 Authentication time for various instances

To assess the effectiveness of our proposed prototype, we compared the results with other commonly used authentication protocols like SAML and Kerberos. Figure 4.7 shows that proposed authentication protocol is very efficient and scalable compared to other protocols. The proposed authentication protocol has better performance than traditional protocols like SAML and Kerberos as it is designed specifically for heterogeneous multi-cloud scenarios. Kerberos is a centralized protocol and distributes tickets to all communicating parties which increases its processing time. Although SAML is a distributed authentication protocol, it does not support heterogeneous client attributes, and when used in a secure way (in conjunction with SSL) it takes longer than proposed protocol to perform authentication of multiple clients. Thus the purposed authentication solutions takes less time to handle large number of requests compared to SAML and Kerberos leading to lower costs.

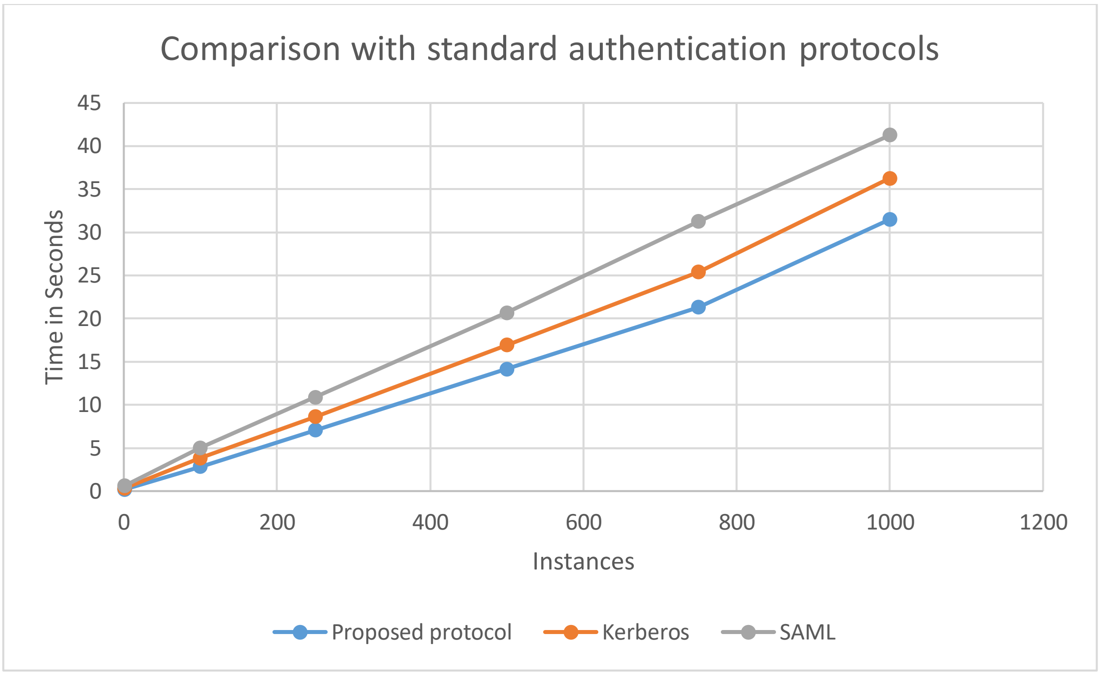


Figure 4-7 Performance comparison of proposed authentication scheme in multi-cloud scenario

To further improve the scalability during orchestration of MCC, its key component cloud controller is scaled and distributed to decouple its logical functionalities. Deploying cloud controller on a single VM can raise many challenges during large scale collaboration such as single point of failure and performance bottlenecks. Therefore, we distribute cloud controller to modular components based on their functionality. A message queue is used to manage incoming client requests and a separate task queue for facilitating collaboration among cloud controllers in multi-clouds. The message queue consumes requests from clients, and a messaging service forwards authentication requests and collaboration messages to foreign clouds. Therefore, the cloud controller can start processing users requests as soon as they arrive by having multiple instances. A large number of user tasks were created to check the performance of controller component in local cloud to set up collaboration across multiple clouds. Figure 4.8 shows the performance of cloud controller after many tasks are executed in foreign cloud which shows that controller is scalable and can handle large requests without significant overhead.

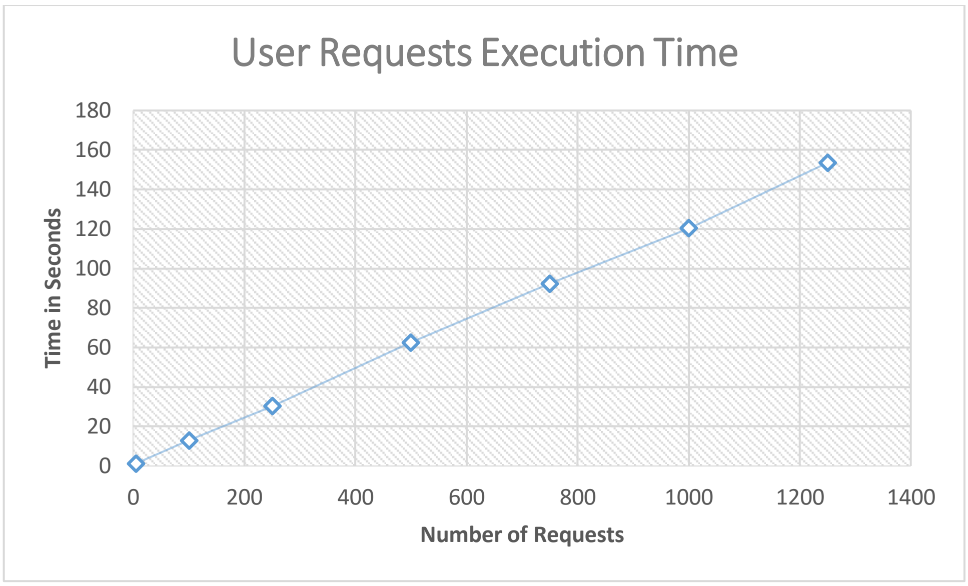


Figure 4-8 User request execution time with scaled controller component

4.6 Summary

To summarise, cloud has been widely adopted to deliver services, but it also introduces challenges due to the limitations of resources and services in a single cloud. This chapter provides an approach to establish secure collaboration across multi-clouds to access services running in the foreign cloud. A novel authentication scheme is presented by which communicating clouds can authenticate each other dynamically and SLA approach is used to ensure service execution in the foreign cloud is according to the agreed SLA parameters between the user and the provider. Various protocols are proposed for multi-cloud collaboration and verified formally. Moreover, we also present a detailed system design to implement these protocols. The experiments are performed on two cloud systems based on OpenStack and the results show that our protocols only result in a limited overhead.

Chapter 5 A Model for Orchestrating Efficient Service Selection and Dynamic Service Access in Multi-Clouds

5.1 Introduction

As discussed in the previous chapter, service orchestration issues in multi-clouds along with many security concerns are a major barrier in their large-scale adoption and application. Consider a scenario in which a cloud (local cloud) user is accessing service located in another cloud (foreign cloud). That cloud user would have no mechanism to verify that the service being used is trustworthy and neither do they have insights on what is happening with their data being handled by services. In order to trust the cloud services, users depend on their assurances given by the cloud provider. Cloud providers give very limited evidence or accountability to users which offers them the ability to hide some behavior of the service.

Another key challenge in multi-cloud collaboration is to develop solutions for multi-cloud that can enable the users to select most suitable service in foreign cloud according to their requirements as well as to ensure that services in the foreign cloud are compliant with the service level agreement (SLA) between user and cloud provider. In order to address these challenges, we extend the proposed Multi-cloud Collaboration (MCC) model in this chapter to facilitate multi-cloud collaboration and provide guarantees to the user that the software or service (such as IoT service) running on a foreign cloud node is secure and the agreed service level agreements (SLAs) are not being violated.

We provide a service selection algorithm to select the best service from multiple cloud providers that best match user quality of service requirements (QoS). In the next stage, service level agreements (SLAs) are used to ensure security and handle service execution in the foreign cloud. The usage of SLA mechanisms ensures that QoS parameters including the functional (CPU, RAM, memory etc.) and non-functional requirements (bandwidth, latency, availability, reliability etc.) of users for a particular service are negotiated and secure collaboration between multi-clouds is setup. The multi-cloud handling user requests will be responsible to enforce mechanisms that fulfil the QoS requirements agreed in the SLA. While the monitoring phase in SLA involves monitoring the service execution in the foreign cloud to check its compliance with the SLA and report it back to the user. Experiments indicate that the proposed approach supports collaboration among a large number of services across multi-clouds and incurs a minor overhead.

The major contributions of this chapter are the following:

- The proposed Multi-Cloud Collaboration (MCC) presented in the previous chapter is extended with additional functionalities to provide users with secure dynamic collaboration and access to services in multi-clouds. The protocols to support these functionalities and additional model components responsible for multi-cloud collaboration are presented.
- A service selection protocol is proposed to select a foreign cloud provider that best matches user requirements. MCC achieves high accuracy by providing distance correlation weighting mechanism among a large number of services QoS parameters in a decentralized environment.

- Mechanisms to set up service level agreements (SLAs) for multi-cloud collaboration have been presented. The various stages in setting up SLA include negotiation, enforcement and monitoring. They help in negotiating QoS parameters for services between the user and foreign cloud provider, enforce a mechanism to comply with the agreed SLAs, monitor client usage of service in the foreign cloud and report back any violation of SLA.
- Dynamic and lightweight authentication protocol to set up single sign on (SSO) presented in the previous chapter is integrated within the system. Experiments have been carried out to check the efficiency of the proposed service service scheduling algorithm as well as the effectiveness of proposed SLA mechanisms in negotiation, enforcement and monitoring.

5.2 Extension of Multi-Cloud Collaboration (MCC) Model

Based on heterogeneous requirements of multi-clouds and services, we propose extensions to MCC that can enable dynamic collaboration between users and services in multi-clouds. The architecture of the proposed model involves various components that have been implemented in each participating cloud to achieve the secure multi cloud authentication. The additional components involved in system design are a scheduler to select suitable service meeting user specifications and SLA coordinator for managing SLA negotiation, enforcement and authorization. Figure 5.1 displays various components in a local cloud to communicate and collaborate with foreign clouds. All the system components serve different functions which are described below.

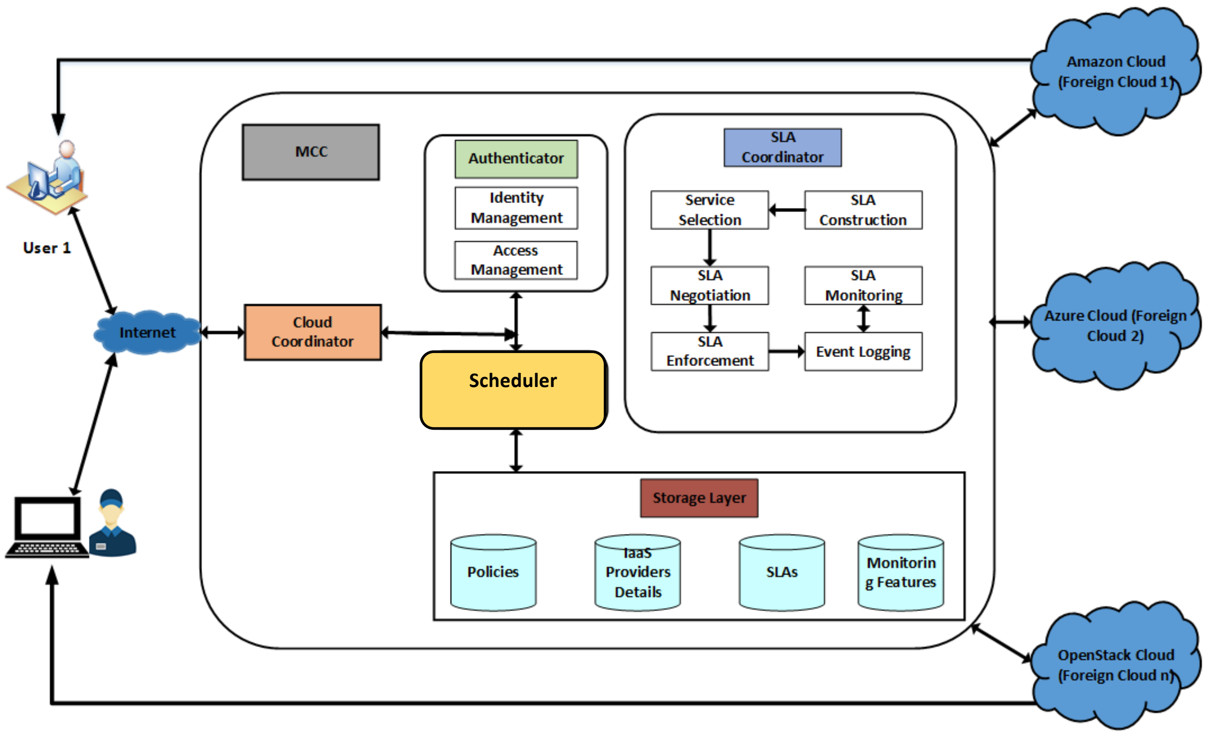


Figure 5-1 The extended Multi-Cloud Collaboration model

5.2.1 Initialization and Authentication

The initialization protocol is the first step that is used to set up the system services, parameters and attributes required for multi-cloud collaboration. It is responsible to boot the required services such as certificate generation, passing the client request to access multi-cloud service and launching services for the proposed model. The initialization protocol is similar to the one described earlier in chapter 4.

The authentication protocol from chapter 4 is used in this model that can provide dynamic authentication between multi-clouds without initial agreement. The authentication protocol uses a single sign on (SSO) authentication mechanism by which any cloud user in the local cloud can authenticate itself with the foreign cloud, and access required resources. If the user is successfully authenticated, it is granted access to resources otherwise foreign cloud makes a request for a new certificate to handle authentication request.

5.2.2 Service Scheduling in MCC

Cloud workloads are usually submitted in the form of jobs and may encompass one to several numbers of tasks. In a traditional cloud model, tasks from customers are received by cloud providers, who will then schedule and execute them in the server resources. Schedulers up on receiving the jobs, locate appropriate resources and place the jobs within virtual machines or containers deployed in the physical servers [126]. Despite the orchestration of a monumental number of server resources in a single datacentre, issues still exist owing to the increasing and diversified demands of customers, particularly for the cases of scientific and mission critical applications. Lack of resources in one common problem prevailing among small-scale service providers that directly lead to compromising service quality and affect their reputation. The traditional concept of a stand-alone datacentre applies mutual exclusion policies during resource constraints, whereby tasks or jobs with fewer priorities are pulled out of execution in order to avail uninterrupted services for priority jobs. Or the providers have to kill, fail or evict the less priority tasks during resource constraints, evicting low priority tasks temporarily pauses their execution to accommodate the execution of priority tasks. The development of scheduler for multi-cloud infrastructures nullifies lack of resources for processing jobs, by the way of promising anytime resource availability, namely providing users with the illusion of infinite pool of resources those can be availed through a collaborative multi-cloud infrastructure [127].

Multi-cloud collaboration paradigm can provide users uninterrupted access to services in foreign clouds. As the cloud services states can change dynamically during runtime, the collaboration between users in multi-clouds can make service automatic detection and access more complicated. Cloud customers can have varying requirements and in multi-cloud scenarios, best services that can meet their required quality of service (QoS) needs to be selected from various providers. To this end, a collaborative scheduling model with integrated security mechanisms to ensure secure transmission of tasks for services access among different participating clouds is an integral requirement in any multi-cloud service infrastructure.

5.2.2.1 MCC scheduling architecture:

A de-centralised architecture is adopted for the meta-scheduling scheme among the participating clouds. To select the most suitable services, the first scheduling goal is the efficient discovery according to the characteristics of services advertised by foreign clouds. The root meta-scheduler in the local cloud initially communicates with the local cloud controller to get list of available resources in foreign clouds. Service discovery for dynamic multi-cloud collaboration could be hard due to requirements such as satisfying QoS, service functionalities and other metrics. There might be cases when a single service would be able to satisfy all user requirements and the service that matches most requirements might need to be selected. This leads to partial match-making where the service that matches most required QoS criteria will be selected. In this section, we propose an efficient and dynamic service scheduling algorithm for the selection of cloud services in multi-cloud scenarios based on partial or closest matching of service QoS attributes.

The proposed scheduling algorithm has two essential characteristics.

- Firstly, the proposed algorithm supports service selection among all services of multi clouds in a dynamic decentralized environment with high accuracy.
- Secondly, different QoS requirements of services can be supported. In case, there is no exact match of user QoS requirements with available services, services matching the most requirements are selected using partial matching. The algorithm can handle a large number of services by using distance co-relation weighting mechanism and it can support various services QoS requirements such as response time, availability, reliability, cost, energy, throughput, latency and best practises.

5.2.2.2 Request description

The process of service selection in MCC across multi-clouds starts with the user request being passed to service scheduling module which includes the required service and desired QoS. The local cloud controller receives a response from various foreign clouds that can deliver required services, and it communicates with the service scheduling module to select the required service. Such as a user SLA might include high throughput compared to cost saving while it might be opposite for another user.

Here we represent various denotations for request types:

- R_Q represents a set of user functional QoS requirements, $R_Q = \{q_1, q_2, q_3, \ldots., q_n\}$, where $n \varepsilon N$
- C_K represents a set of available foreign clouds offering services with required functionality, $C_K = \{c_1, c_2, \ldots\ldots\ldots c_k\}$ where $k \varepsilon N$
- S is a subset of C_K i.e. available services across multi-clouds with similar functionality, $S = \{s_1, s_2, s_3 \ldots s_m\}$, where $m \varepsilon N$
- Each service S has Q_S property matrices, $Q_S = \{Q_{S1}, Q_{S2}, Q_{S3} \ldots Q_{Si}\}$, where $Q_{Si} = \{q_{i1}, q_{i2}, q_{i3} \ldots. q_{ij}\}$, $i, j \varepsilon N$. Q_{Si} represents quality matrices for service i.

5.2.2.3 QoS analysis

Once the QoS requirements have been gathered, the module collects all possible service offers and their associated QoS parameters. These are used to construct and rank an accuracy matrix for the calculation of offers.

In an ideal scenario, user QoS requirements R_Q must be similar to the QoS parameters mentioned in $Q_{Si.}$ In other words, an ideal service for user request can be represented as,

$$\mathbf{R_Q \leftarrow Q_{Si}}$$

However, in a real case scenario that user requirements R_Q and the number of quality matrices Q_{Si} will be different. Therefore, R_Q is taken as a baseline and quality matrices could be arranged in the following way:

- If the quality service matrix Q_{Si} lacks in the user $R_{Q,}$ it is removed and R_Q is assigned 0

To construct an accuracy matrix, n consumers request R_Q which are identified along with m available services that can satisfy user requirements, and an m*n matrix is constructed which is called R. The columns in the matrix represent QoS parameters R_Q while each available service is represented in a row for the selection process.

Requirement matrix, R can be defined as:

$$\begin{array}{c|c} & \begin{array}{cccc} \mathbf{R_{Q1}} & \mathbf{R_{Q2}} & \cdots & \mathbf{R_{Qn}} \end{array} \\ \begin{array}{c} \mathbf{S_1} \\ \mathbf{S_2} \\ .. \\ \mathbf{S_m} \end{array} & \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{pmatrix} \end{array}$$

A service not satisfying the mandatory QoS requirements R_Q is removed from the selection process.

5.2.2.4 Priority computation of services across multi-clouds

To perform priority computation of services, QoS based ranking is incorporated in MCC. The local cloud can choose the most optimum foreign cloud to send the requests for providing users access to services across multiple foreign clouds. This prioritises the foreign cloud selection depending on the customised requirements of individual requests, which could be energy-efficiency, enhancing performance, boosting revenue, quick turnarounds etc.

The priority calculation of the services in different multi-clouds is computed using accuracy matrix A, which is dependent on the tendency of QoS parameters. This tendency describes how the numeric value of a user QoS parameters changes for a service to be observed as better. It indicates whether high or low values of a QoS parameter are preferred in an ideal case. For example, an ideal service will require high availability and throughput while its response time and latency should be low. The scheduling of tasks according to QoS parameters in various clouds depicted in the figure 5.2 below.

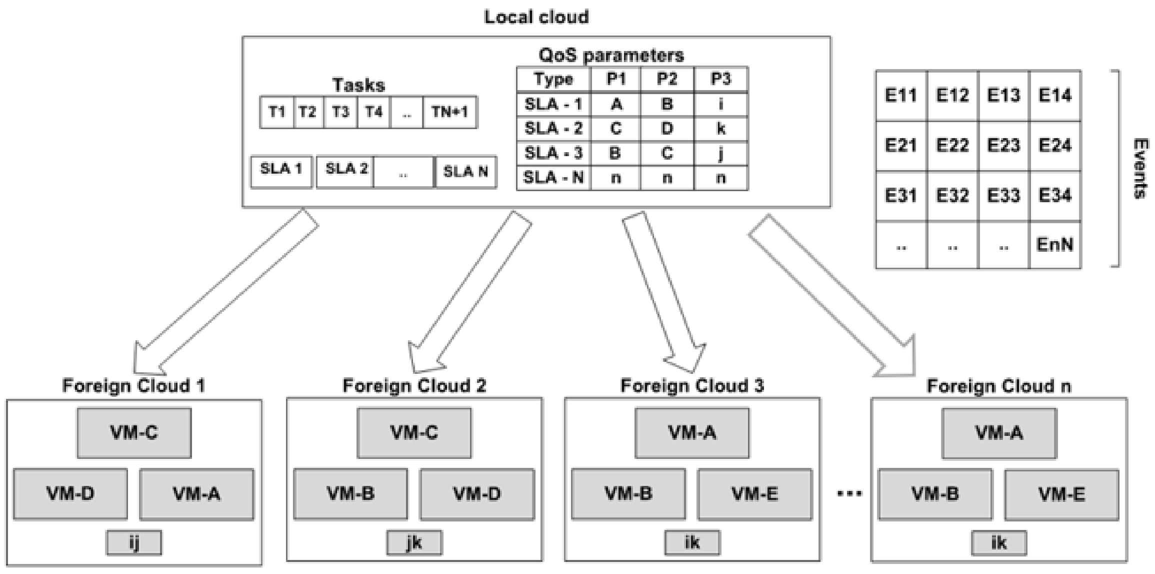


Figure 5-2 QoS driven scheduling in multiple foreign clouds

Using the user described QoS range and offered service QoS, elements of the accuracy matrix are calculated using case dependent formulae as mentioned in the equations below:

For values with high tendency:

$\frac{Q_{ij}}{Q_1}$ when $Q_{ij} < Q_1$

$\frac{Q_{ij} - Q_1}{Q_h - Q_1} + \alpha$

$\frac{Q_{ij}}{Q_{max}} + \beta$ when $Q_{ij} > Q_h$

For values with low tendency:

$\frac{Q_h}{Q_{ij}}$ when $Q_{ij} > Q_h$

$\frac{Q_h - Q_{ij}}{Q_h - Q_1} + \alpha$ when $Q_1 \leq Q_{ij} \leq Q_h$

$\frac{Q_{min}}{Q_{ij}} + \beta$ when $Q_{ij} < Q_1$

In the above equations, Q_{ij} is the value of ith QoS property of jth service, Q_1 is the lower limit of user requirements for an attribute, Q_h is the highest limit of user requirements for an attribute. Q_{max} and Q_{min} are respectively the maximum and minimum values of a QoS property being offered by a service. α and β belong to $\{1, 2, 3, \ldots\}$ where $\alpha < \beta$. The results from the above equations are normalized in the range [0, 1].

α and β are used to differentiate between loose range, preferred range and tight range. The preferred range for any service is between Q_1 and Q_h. If a value falls in this range, α is added to normalize the value so that results are in range $(\alpha, \alpha + 1)$. The values in the loose range (between Q_{min} and Q_1 for high tendency parameters, and between Q_h and Q_{max} for low tendency parameters) are normalized between 0 and 1. While the values in the tight range (between Q_h and Q_{max} for high tendency parameters, and between Q_{min} and Q_1 for low tendency parameters) are normalized by adding β so that results are in the range $(\beta, \beta + 1)$. Therefore, for all the values in accuracy matrix lie between $(0, \beta + 1)$ which helps in consistency. Moreover, $\beta > \alpha$ which always guarantees that higher range always has a higher value in accuracy matrix than other two ranges.

The results are used to calculate the accuracy matrix A. It shows how precisely each service matches the user requirements. After constructing the accuracy matrix, the rank of each service can be calculated in the following way:

$$\mathbf{R_i} = \sum_{k=1}^{m} \sum_{j=1}^{n} \boldsymbol{A_{ij}} * \boldsymbol{W_j} * \boldsymbol{x_{ijk}}$$

In the above equation, R_i represents the rank of the service i, A_{ij} represents the accuracy value of the jth QoS property of service i, and W_j represents the weight of the jth QoS property. x_{ijk} = 1 if the user is assigned access to service S_i in the foreign cloud C_k otherwise it is 0. Once the tasks are processed in the foreign cloud, the outputs are sent to the local cloud via respective root meta-schedulers and controllers.

5.2.2.5 Benefits of the Proposed Model

The de-centralised scheme of scheduling proposed in this chapter facilitates the foreign cloud to take a measurable control of the tasks received from the local cloud. This helps the scheduler in the foreign cloud to act as a master to coordinate with its respective local resources to place and allocate the tasks for processing. In this distributed architecture of scheduling, schedulers are responsible for resource management of their responsible server clusters in the datacentres. If certain resources fail, then their respective clusters can be replaced by other resources until repairing or replacing the failed resource.

Algorithm: QoS Driven Task Scheduling Algorithm for Multi-Clouds

```
1. BEGIN
2. Data: Input: <Client functional and non-functional QoS requirements (CR)>,
                <List of services (LS)>
3. Cloud LF= {1, 2,…., k}; // Total list of available foreign clouds
4. Service LS = {1,2,…, n}; // Total list of available services across foreign clouds
5. <Service,CR> ServiceContenderList (SCL) = NULL //List of services satisfying requirements
across various foreign clouds
6. Cloud C=NULL // Single cloud instance
5. Service S=NULL // Single service instance
6. CR Q=NULL //Single QoS requirement
7. <Service, CR> O=NULL;
8. For each C in LF:
9.      For each S in LS do:
9.           if (Satisfy(S ,CR)) //Add to SCL all appropriate services matching user requirements
10.          SCL.add (S,CR)
11.     end if
12. end for
13. For each O in SCL do:
14.     for each Q in O.CR do:
15.          Normalize (AccuracyMatrix (Max(Q),Min(Q) )) //Generate accuracy matrix
16.     end for
17. end for
18. For each O in SCL:
19.     Score = Calculate_Score(O.Service ) // Calculate score of each service
20. end for
21. SCL.sort(Score) // Rank all services in SCL
22. Return SCL
23. END
```

5.2.3 SLA Negotiation

The SLA coordinator receives user requirements and SLA's from the foreign cloud and negotiates a dynamic SLA between them. These SLAs exist within the customer domain that wants to access foreign cloud resources. From the client viewpoint, SLAs define the mechanisms to securely access services while the SLAs are utilized by cloud administrators to manage the mechanisms to offer cloud services. SLA-coordinator negotiates the SLAs on behalf of the user if there is the full match of QoS requirements in the stated SLAs. However, as described earlier there might be a partial match after which user can have the ability to negotiate SLAs itself. Therefore, SLA coordinator component in MCC offers added features to customers such as negotiating an SLA or switching to a new provider in a multi-cloud scenario if selected provider and user cannot agree on an SLA.

As discussed earlier, scheduling component checks the service specs like base service, features, cost and recommends them to user. SLA Negotiation involves agreeing on the service terms for SLA and QoS parameters, measuring metrics (service level objectives) and defining how the metrics will be measured. While service providers also check if they can provide requested service and perform basic risk evaluation in case. As provider reputation is on a stake if it fails to provide the service agreed in SLA.

Integrating the security parameters within SLAs is a novel problem and a very limited research has been done in this area. For the case of secure multi-cloud collaboration, we propose a service level objective (SLO) called service identity which can help customers to negotiate the SLAs for secure service execution on the foreign cloud.

5.2.3.1 Service Identity

Service identity as an important property to maintain strong service security and compliance in a foreign cloud. A set j services F_j deployed on a single cloud platform with functional properties $Func_i$ and non-functional properties $NFunc_i$ can be defined as:

$$\mathbf{F_i = \{Func_i, NFunc_i\} \quad 1 \leq i \leq j}$$

During service execution in a foreign cloud, both functional and non-functional properties of service instances being used by users must be maintained. Functional properties of instances that could be violated include a change in the code or implementation of service to make it do certain other activities affecting the original behavior of service. While a few non-functional issues can include service taking more processing time, charging more cost than agreed or remaining unavailable during required times.

If F is the original service deployed by the service provider in cloud after agreeing SLAs and F' is the instance of that service running in cloud that is being used by client, the service identity can be satisfied only if $F{=}F'$ holds true for that particular instance of F running in the cloud during the entire lifecycle of F from deployment to decommissioning. The service identity can be described by the following equation:

$$F \equiv F' \quad \ldots \text{(a)}$$

In order for functional properties of a service instance F' to hold, its functional properties must be same as original instance. While the case for non-functional properties is more complex as the service states can change dynamically during runtime. Moreover, each user will have different QoS requirements from a service. As an example, users X and Y using different instances F' of same service F can have varying availability, and cost requirements. Therefore, we define a threshold value for non-functional parameters of a service instance that it must maintain to ensure service identity.

The non-functional parameters of a service agreed in the SLA can be defined as tuple:

$$\mathbf{NFunc = \{Min_i, Max_i, W_i\}\ 0 \leq i \leq l}$$

i is the QoS parameter, Min and Max show the accepted boundaries or threshold values for that parameter, and W denotes the weight assigned to a particular parameter by user which shows the importance given to that parameter by user. If i is not defined, it is given value of 0.5 that shows medium importance to that parameter. In order for non-functional properties to hold true in an instance, the following condition must be satisfied at all times:

$$Min_i \leq NFunc_i \leq Max_i \ldots \text{(b)}$$

To comply with functional requirements such as security different techniques can be agreed in the SLA which can ensure that functional behaviour of service instances F' will not change. For example, to maintain the service identity trusted platform module (TPM) mechanism could be used. The functional property of a service could be defined as:

$$F\text{-}F' = \emptyset \ldots \text{(c)}$$

If both equations (b) and (c) hold than equation (a) will hold. However, in case if service security is compromised than the equation will become $F' \supset F$ meaning that service identity does not hold.

Meanwhile, various authors have proposed definitions of other functional and non-functional metrics (SLOs) for services that can be agreed between customer and provider during SLA negotiation. These parameters include request latency, availability, accessibility, service throughput, completion time, and mean times to repair and failure, energy cost and financial cost.

The proposed system uses WSDL to express the functional security requirements and non-functional requirements. The XML data structures are generated on the basis of the WSDL document, the service interface definition and its implementation. Therefore, QoS tags are associated with a new category to recognize security and other properties. The protocols for SLA management are implemented in the form of a REST based service and API.

5.2.4 SLA Enforcement

Once a user is authorized to access cloud resources, the next stage is the enforcement of security mechanisms by the provider. In this stage, mechanisms are implemented that can guarantee SLA assurances. The enforcement of agreed SLA is done in two stages. First stage involves implementing the software modules that can be activated for the acquisition of resources for enforcing security policies and second stages involve dynamic reconfiguration of the resources after a security alert is generated.

This research focuses on the implementation of mechanisms for non-functional properties of services to ensure that service complies with the defined SLA policy. The enforcement of policies for SLA enforcement is done by foreign cloud in its infrastructure by acquiring enough resources for service execution and employing the required mechanisms. QoS parameters mentioned in SLAs are measured by maintaining current system configuration information and runtime information of parameters that are part of SLOs (measurable metrics). Depending on the client requirements some or all SLA parameters could be measured, and SLOs such as request latency or service throughput could be measured by retrieving resource metrics.

Development of mechanisms for maintaining functional property is not in the scope of this thesis. We discuss various mechanisms that exist in the literature that could be deployed for secure service execution such as trusted computing. Trusted computing is paradigm used to enforce trustworthy behaviour of computing platforms. It is based on using a hardware crypto-processor module named Trusted Platform Module (TPM) [128]. This feature can be used to run services on only those cloud nodes whose fingerprints are trusted [109]. Various mechanisms for cloud computing based on TPM have been proposed that are used for the security of services, data and other resources. Excalibur [109] is a system that can be used to design trusted computing services for the cloud. It uses policy sealed data (data encrypted according to customer policy) that can only be unsealed (decrypted) by nodes whose configuration match the node policy. Excalibur uses Attribute Based Encryption to bind policies and attributes to node configurations. A mechanism that uses a hardened hypervisor to attest that the image of the VM running on a cloud node is the same as the one uploaded originally by the service provider and initiated by cloud was proposed by Santos N. et al. [129]. It confines the execution of VM to secure nodes inside the cloud and guarantees that even the system admin with root privileges cannot tamper with the VM memory. Some other recommendations provided by NIST for hardening the hypervisor include maintaining proper isolation, separating the duties of administrative functions and restricting administrator access to security checks [130].

5.2.5 SLA Monitoring

Currently, no solutions exist to check for SLA compliance for user support. However, researchers have recommended using the monitoring mechanisms to check for SLA compliance on the cloud provider which involves two basic activities. The first one is the verification that SLAs are respected via access to underlying infrastructure that is inaccessible to users. Once SLAs are measured they are compared to the thresholds agreed in the SLA. While other is the generation of alerts if SLAs are broken to inform enforcement layer to activate countermeasures that can protect services.

Monitoring could either be performed by the client from data received from the cloud provider or by cloud provider at the infrastructure level which is the focus of this chapter. The input to monitoring component provided by a cloud provider is the formal requirements to be monitored in a formal language such as XML. The monitoring component than derives the pattern of events that could occur during service execution and imply SLA violation. In the proposed system uses event driven modules to collect all generated events and performs required filtration operations before analysing them. The description of event captors and monitor is used to monitor the SLA parameter.

The analysis is performed based on captured events to check if any generated events show an SLA violation. If a security violation is reported by the monitoring component, it logs the event and estimates the current status of service. Monitor also reports to the user if the foreign cloud is compliant with the signed SLA or not. In the case of SLA violations, the user can enforce penalties on the provider.

5.3 MCC Design

This section explains the workflow for the overall model and the implementation details of various components. A user request to connect to the foreign cloud for accessing a service is received by cloud controller that advertises request to connect to multiple foreign clouds and receives their responses. Scheduler selects the best provider based on user QoS requirements. The authenticator is responsible for authentication while the SLA coordinator is responsible for SLA management. The details of components such as controller, authenticator and trusted party have been described earlier in section 4.3.2. The workflow of the proposed system is shown in figure 5.3.

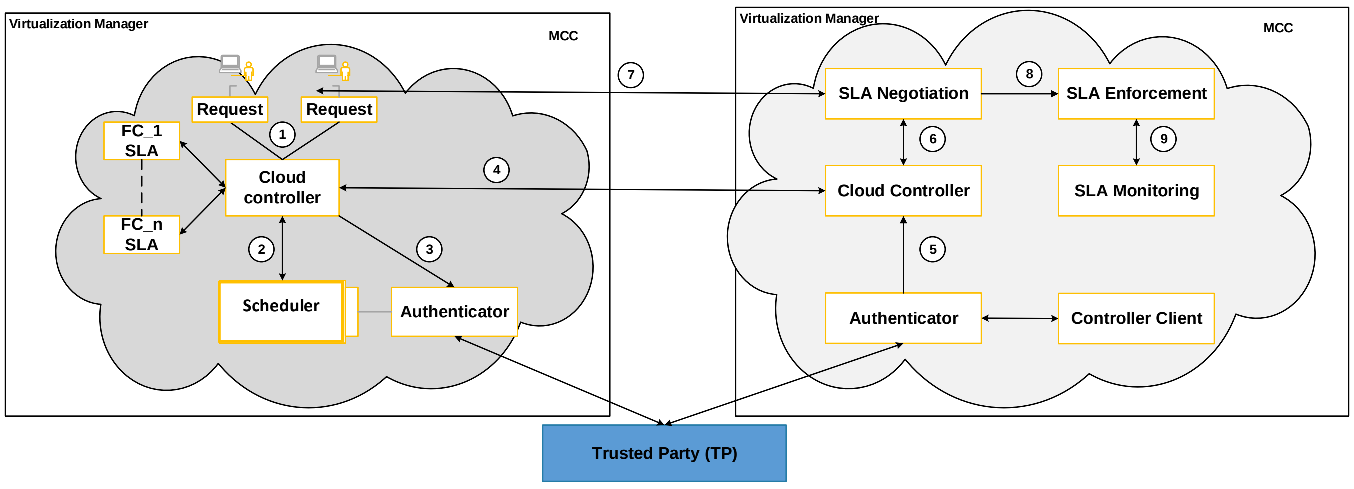


Figure 5-3 Workflow of the extended MCC model

This section explains the workflow for the overall system and the implementation details of various components. The overall process for cloud selection, negotiation and SLA compliance is managed is managed in the following three stages:

1. A user request to connect to the foreign cloud is received by cloud controller. This request has a list of required service QoS parameters that enable local cloud to find an appropriate foreign cloud that can meet client requirements.
2. Local cloud advertises this connection request to multiple foreign clouds and receives their responses that are passed to the match-making module. These responses contain the various service offers from foreign clouds in the form of advertised SLAs and local cloud controller stores them in its repository.
3. The scheduling component in local cloud controller finds the best matching service meeting the user requirements. In case of non-matching or partial matching, the results are returned to the user with recommended service that can agree or disagree to use the recommended service, and cancel the connection request or wait for service availability.
4. The scheduling component performs the service selection and recommends it to the authenticator component that performs authentication between the user and foreign

cloud. The user is authenticated by foreign and authorization is also performed. To perform authentication we have developed a lightweight SSO solution.

5. After authentication, SLA negotiation on the behalf of the user. Once access is granted, SLA enforcement is done in the foreign cloud to provide service that complies with QoS requirements of user agreed in the SLA.
6. SLA monitoring is carried out in the foreign cloud to ensure that all QoS parameters as well as the functional and non-functional requirements of user are satisfied as per the SLA. In case of violation, monitoring component records it and informs the user.

5.4 Experiments and Results

5.4.1 Evaluation

To examine the feasibility of the proposed design empirically, it was implemented on two different clouds mentioned in experimental setup. Web applications are used to assess use case scenarios as they represent the most common type of application domain in cloud today along with other transactional applications.

The first goal of the evaluation is to determine the performance of the system with an increased number of services requests, to calculate the overall increase in user request processing time and determine the impact on various operations in the model as the user's requests increase. While the second goal of the experiments is to check the efficiency of proposed model MCC by measuring events through monitoring that are generated during collaboration between multi-clouds. Various measurement intervals are used to measure changing behaviour of an application during its execution. This helps us in determining optimal measurement interval for checking SLA enforcement by SLA coordinator component in MCC.

Another goal of the evaluation is to determine the number of violations that occur for a pre-defined set of SLA metrics. To detect the violations in a system, SLA thresholds are setup based on historical data and resource consumption data of that particular kind of applications. SLA violation during service execution can occur due to unpredicted resource usage by a service or due to the fact the SLAs might have been agreed on service usage and provider might assign VM hosting that service to another user that can reduce resource availability. This can help the provider to estimate a possible number of violations for an SLA metric and take precautionary measures so that these violations can be reduced. We calculate SLA violations for different measurement intervals and determine the best intervals to measure SLA violations. The costing function is proposed to calculate the cost of missing SLA violations if higher measurement intervals are used. Moreover, to show the scalability of MCC scheduling module experiments are carried out to show it can schedule job with high accuracy and minimal performance overhead.

5.4.2 Experimental Setup

The prototype was tested on two different cloud infrastructures. One of the cloud infrastructures was an OpenStack cloud based in University of Derby. This setup consists of six server machines. Each machine has 12 cores with two 6-core Intel Xeon processors running at 2.4 GHz with 32 GB RAM and 2 TB storage capacity. The cloud nodes on which the experiments were performed had 4 VCPUs running at 2.4 GHz each, 8 GB RAM, and data storage of 100 GB per node. The second cloud was also based on Amazon AWS. The cloud nodes on this machine had 4 VCPUs, 8 GB RAM and 100 GB storage.

Both the Cloud Controller and Cloud Authenticator are employed as web services which help in avoiding tightly bound security. While WS-Agreement was used to implement the SLA components. To enable the interaction among components in prototype according to the proposed system, cloud controller of local cloud submits requests for resources to other foreign clouds. When the foreign cloud controller initializes and receives a request for available services from a local cloud, it shares exchange information about available services and their characteristics. The web application to which user requires access in the foreign cloud is a spring boot application developed using REST services and hosted on a Tomcat server. And the database setup for this application is a MySQL database developed using Java Persistence API (JPA) and accessed through RDS instances.

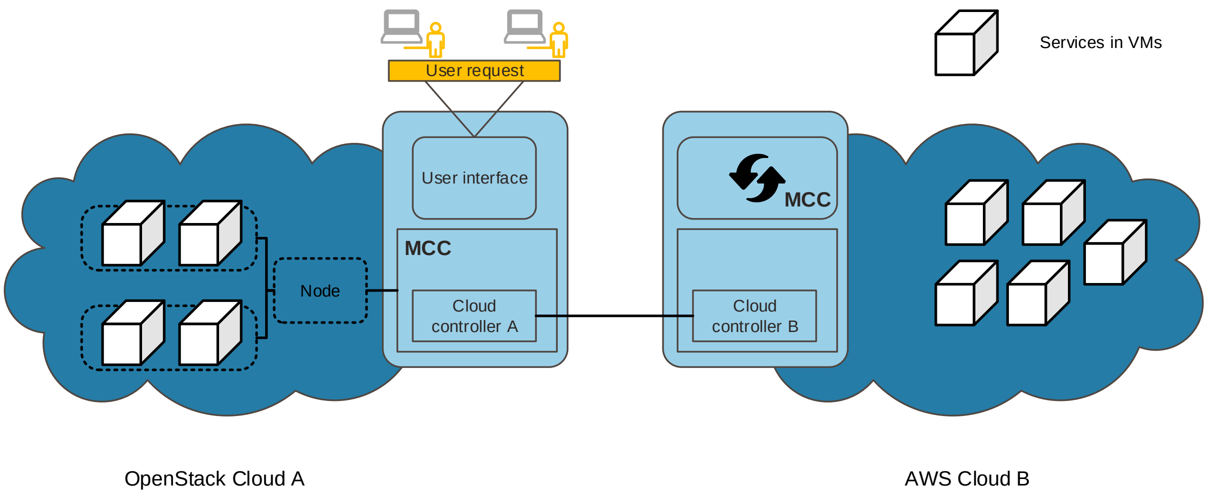


Figure 5-4 Experimental Setup between OpenStack and AWS

5.4.3 Experiments

As mentioned earlier, web applications are used to assess the model as they represent a common class of applications. Initially, the setup is performed by using user requests, authentication and authorization in MCC similar to experiments in the previous chapter.

We defined six SLA parameters for the web application as the quality of service requirements that must be guaranteed during system execution. These QoS requirements were divided into functional requirements and non-functional requirements which are described in table 1. In these experiments, SLA formalization is not addressed, and the focus is on specifying SLA requirements which are commonly defined and used for managing user services. These functional and non-functional requirements for web applications have also been determined using their historical data that determines the resources required for that service (functional requirements) and also the QoS parameters (non-functional) that are most critical for these type of services to achieve maximum performance.

The evaluation scenario is presented in figure 5.4. The web application is a spring boot application built with REST services and deployed on a Tomcat server. The virtual machines are deployed on the foreign cloud that can execute the service with its required resources to serve clients request. The allocated VM executes web application according to the functional and non-functional requirements which specified using the WSDL and launches the application.

The measure the effectiveness of SLA-coordinator events generated during service execution are continuously monitored to detect SLA violation. The low-level metrics are constantly been checked to verify if there is any violation. To perform monitoring, pre-defined set of rules and values expressed in WSDL format are compared with the monitored resources used during web application execution. To monitor functional parameters the common application metrics such as CPU, storage and memory usage are calculated while for non-functional parameters low-level metrics are calculated using a monitoring agent which captures runtime events and compares them with the SLA parameters threshold to detect a violation. During application execution custom events can be generated that have event-specific information, and they are transported locally using API calls and remotely through Java Messaging Service (JMS). Events are generated by event generators which in this case is the Java code executed in the VM and that writes them to the respective event stream. The collected events are then filtered, aggregated and processed using event co-relation to generate metric values that are stored in the database. We pre-defined a set of events that can be compared with these runtime events and used for metric correlation to detect SLA violations.

The information regarding time taken by each process is also measured, and user and foreign cloud provider SLAs are sent to the foreign cloud provider. Once user is authenticated and makes a request to access resource virtual machines are configured and services are assigned to those VMs that run it according to agreed SLAs.

5.4.4 Results

The web application is a spring boot application using REST API that is designed to perform simulated activities of a business (shopping) hosted on a Tomcat server. The database implemented is MySQL developed using Java Persistence API (JPA) that stores the application client and products details. To manage the MySQL database RDS instances are used in the foreign cloud. The workload on web application included multiple sessions to handle different transactions, database requests for data access, update and deletion while maintaining accuracy and security. The SLA parameters defined for this service and agreed between user and provider during SLA negotiation are divided between functional and non-functional requirements. The non-functional requirements include accessibility, throughput and availability while the functional requirements are CPU usage, memory usage and storage. Any utilization of resources not between these thresholds is regarded as an SLA violation.

During web application execution, monitoring is performed and several measurement intervals are set to detect the violations in a particular time period. The measurement intervals are chosen to monitor the changing behaviour of an application during its execution as shown in table 5.1.

Measurement Interval	**25 sec**	**50 sec**	**100 sec**	**150 sec**	**200 sec**	**250 sec**	**300 sec**
Events Measured	1619	1333	1070	797	538	259	52

Table 5-1 Measurement intervals and number of events measured in them

A total number of violations detected for each measurement interval for the number of QoS metric agreed in SLA for a defined time interval. For each parameter, the number of detected violations in a time interval are shown in figures 5.5 to 5.8. While the total number of recorded events and violations in each stage are shown in figure 5.9. These results show that our system can monitor events during service execution and find any SLA violations. The table 5.2 shows the total number of SLA violations and the total number of events captured.

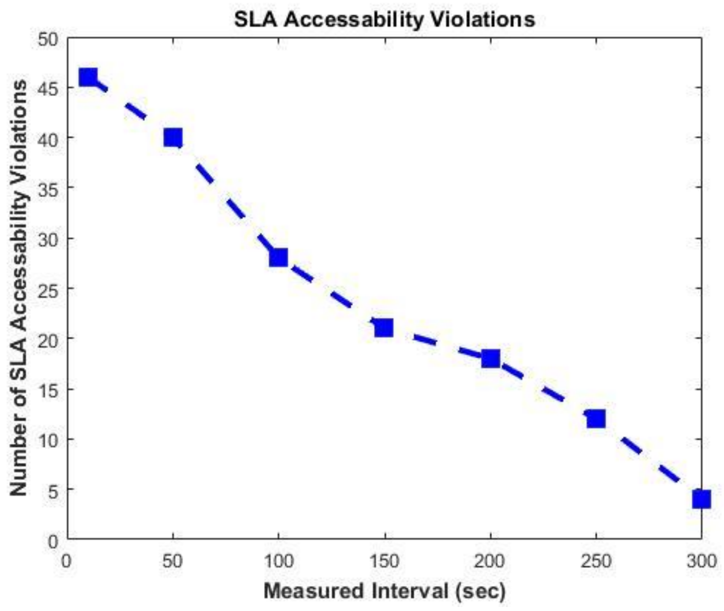


Figure 5-5 SLA Violations of Accessibility at various time intervals

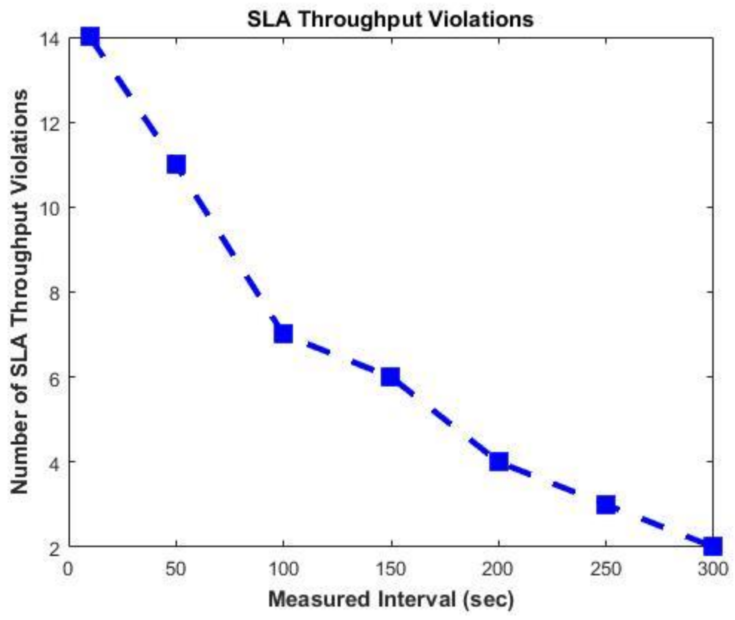


Figure 5-6 SLA Violations of Throughput at various time intervals

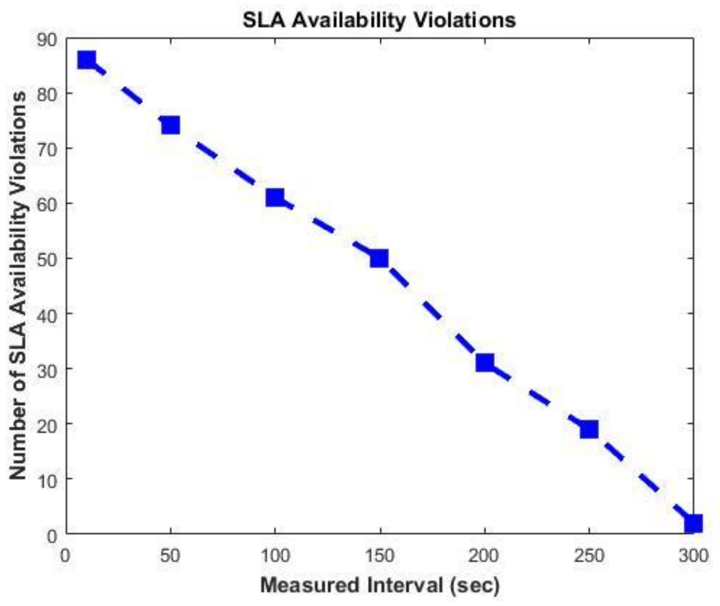


Figure 5-7 SLA Violations of Availability at various time intervals

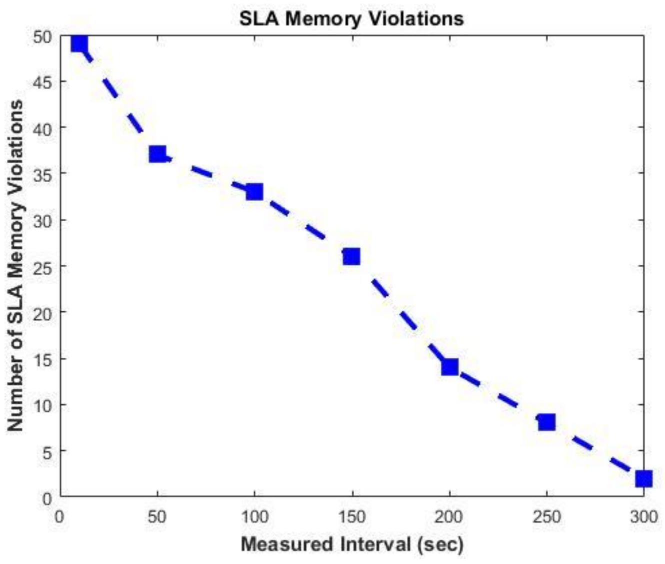


Figure 5-8 SLA Violations of Memory at various time intervals

Measurement Interval	**Events Measured**
25 sec	Total Events: 1619
	Number of Violations: 181
50 sec	Total Events: 1333
	Number of Violations: 167
100 sec	Total Events: 1070
	Number of Violations: 130
150 sec	Total Events: 797
	Number of Violations: 103
200 sec	Total Events: 538
	Number of Violations: 62
250 sec	Total Events: 259
	Number of Violations: 41
300 sec	Total Events: 52
	Number of Violations: 8

Table 5-2 Number of events and SLA violations measured for a time interval

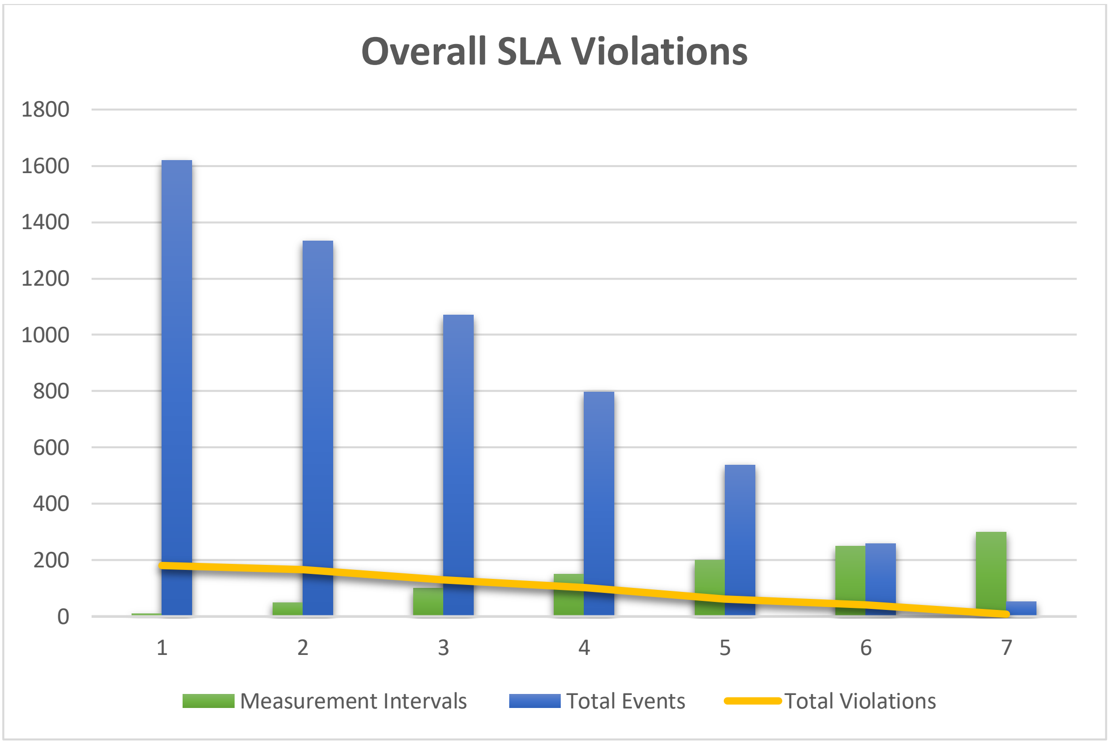


Figure 5-9 The number of events measured in a time interval and recorded SLA violations

Along with the number of violations, the number of undetected SLA violations are also determined. As each measurement interval is supposed to capture all SLA violations, undetected violations for an interval are calculated by finding the difference between total SLA violations for an interval and actually captured violations in that interval. This leads to determine the cost of SLA violations that can be recorded for measurement intervals. The cost analysis is helpful to determine the best measurement interval for SLA monitoring in MCC that can capture violations at acceptable performance overhead.

To measure the number of undetected violations we compare the number of measured violations with violations measured in the reference interval. As reference interval of 25 seconds is used, violations of each other interval are compared with it to detect the number of undetected violations. Using a lower measurement interval than 25 seconds can increase the measured events, however it massively increases the system overhead as shown in figure 5.11. The lowest measurement interval used in figure 5.11 is 5 seconds that shows maximum system overhead. Therefore, considering cost and system overhead, 25 seconds interval is chosen as the minimum and optimum measurement interval.

Using the cost function defined in [131], we can also determine the cost of SLA measurements. The cost function is defined as:

$$\textbf{Cost (C)} = \boldsymbol{\alpha} * \textbf{Cm} + \sum_{\boldsymbol{\beta} \in \{\textbf{memory},\textbf{storage},\textbf{availability}..\}} \mu\,(\boldsymbol{\beta}) * \textbf{Cv}$$

In the above equation, α denotes the total number of SLA measurements, and Cm is the cost of these measurements. μ (β) denotes the number of undetected SLA violations and Cv represents the cost of missed SLA violations. This function helps us to analyse the results and determine the measurements cost of application mentioned above. The factors Cm and Cv in the above equations represent cost values which are agreed for each particular type of application. As a web application is being used in our case, SLA violations could have a huge impact on performance such as slowing it down so missing an SLA violation has a higher cost. On the other hand, measurements to detect SLA violations only focus on server monitoring and incur lower overhead so they have comparatively lower cost. Therefore, we derive the cost values using cost functions and assign values of $0.10 and $0.20 respectively for measuring an SLA violation and missing a violation. These values are not standard values and are determined using the cost function methods described in the literature [132].

As mentioned earlier, the reference interval used in 25 seconds and the assumption is that all violations are measured in this interval. Therefore, in 25 seconds interval, there is no cost of missing an SLA violation. By using the cost function, the results are given in the figures 5.10 amd 5.11 below.

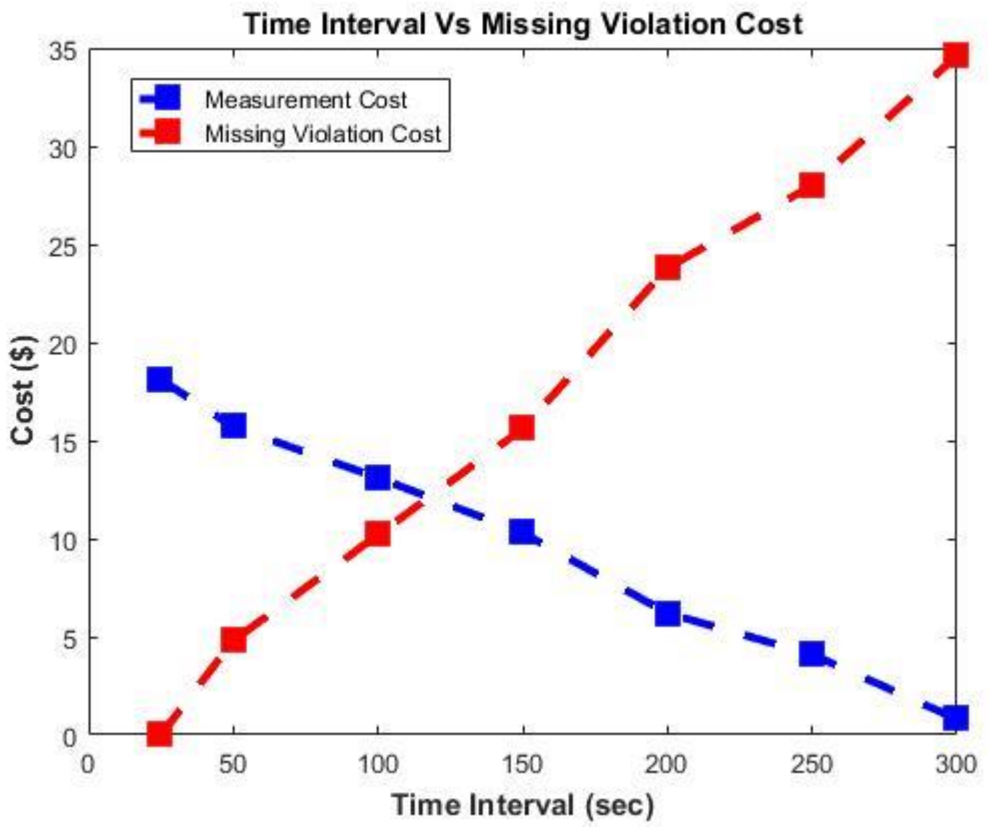


Figure 5-10 Time interval in seconds vs Cost of missing violations in $

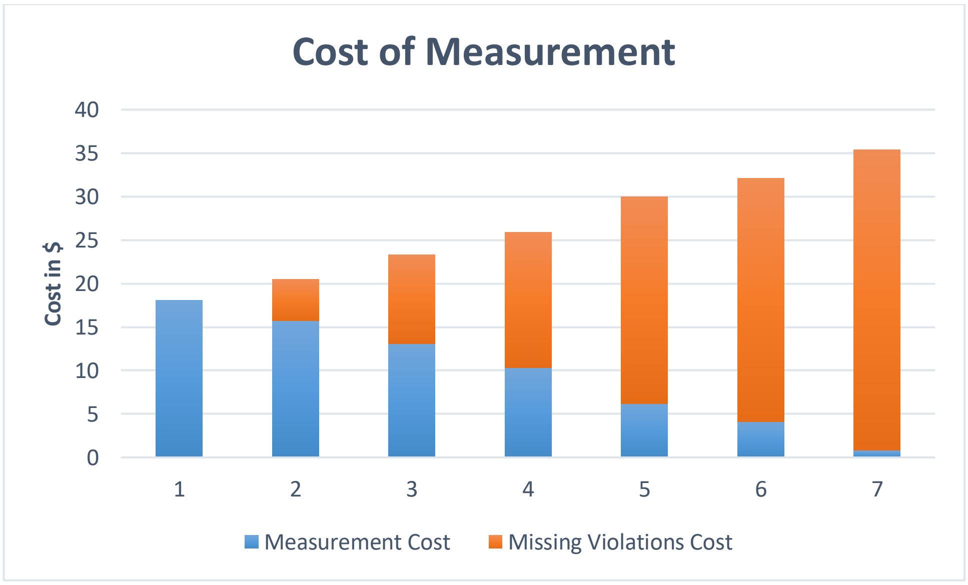


Figure 5-11 Cost of measurements at various time intervals

The results show that the cost of measurement increases as the measurement intervals are increased. This is due to the fact the for higher measurement intervals the number of missed SLA violations is very high. Therefore, based on the results 25 seconds measurement interval could be considered as the best interval showing the web applications are sensitive and require continuous monitoring to ensure that SLA requirements are satisfied. Missing the SLA violations will lead to a higher cost to the provider.

To determine the performance overhead caused by the purposed system, we calculated the time taken by different stages which is used in determining time taken by various measurement stages. The various stages for which overhead was measured are mapping the agreed QoS parameters in SLA with the metrics to be monitored, monitoring application execution in a virtual machine, and processing the monitored data to log events and report violations. These measurements indicate the performance intensive operations and are useful in determining acceptable level of measurement intervals in terms of performance overhead. As shown in the figure 5.12 below, the overhead is maximum for least measurement interval and gradually decreases with the increase in the measurement interval.

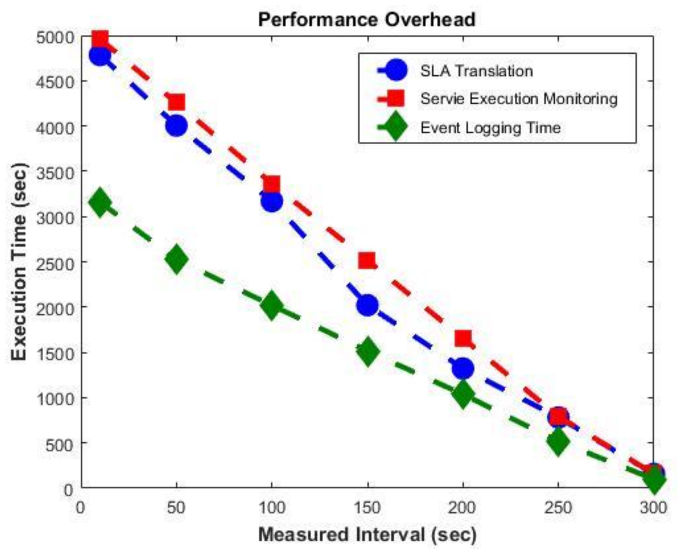


Figure 5-12 Performance overhead of various operations

To check the scalability and efficiency of the service selection algorithm, a large number of user service selection requests (tasks) were created , and the algorithm was successfully able to select the service with the highest match of QoS properties using the scheduling algorithm. Moreover, using the scheduling algorithm the selected foreign cloud was used to forward user requests to access a particular service matching QoS properties. Results shown in figure 5.13 elaborate that the scheduling algorithm is efficient and scalable. Moreover, the scheduling algorithm has been compared with traditional scheduling algorithm FCFS and our alogrithm showed 22 percent more efficiency on avergae in processing requests compared to FCFS.

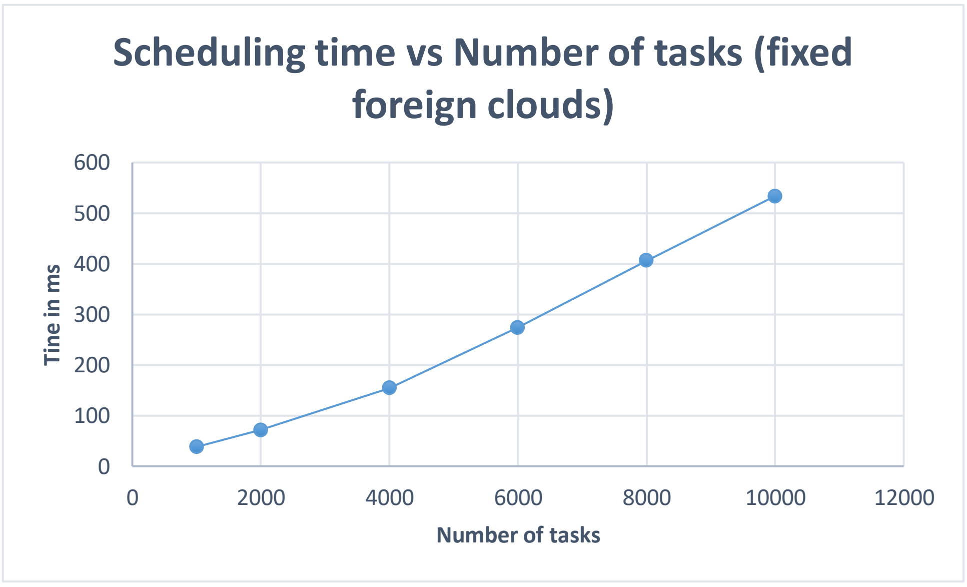
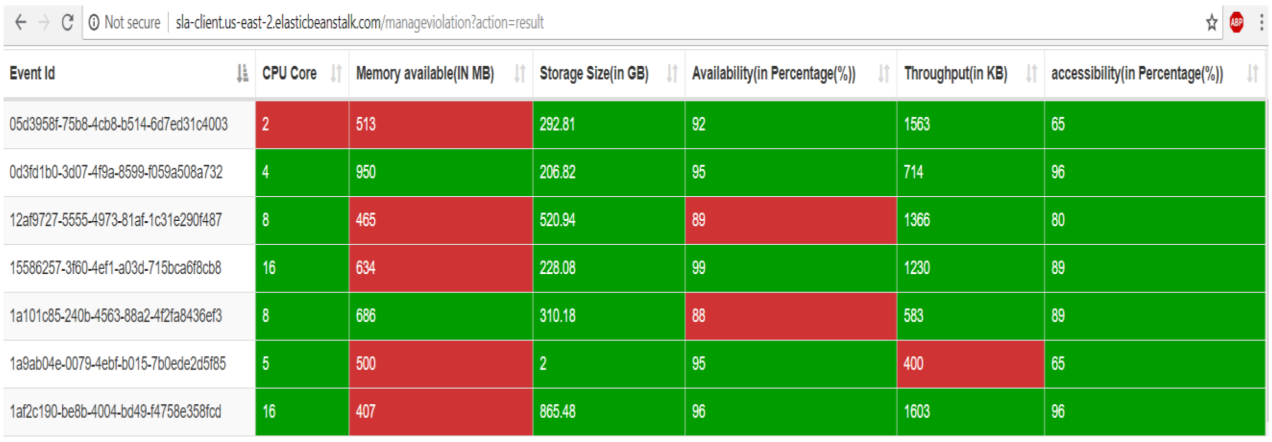


Figure 5-13 Scheduling time versus number of tasks


sla-client.us-east-2.elasticbeanstalk.com/manageviolation?action=result

Event Id	CPU Core	Memory available(IN MB)	Storage Size(in GB)	Availability(in Percentage(%))	Throughput(in KB)	accessibility(in Percentage(%))
05d3958f-75b8-4cb8-b514-6d7ed31c4003	🟥 2	🟥 513	🟩 292.81	🟩 92	🟩 1563	🟩 65
0d3fd1b0-3d07-4f9a-8599-f059a508a732	🟩 4	🟩 950	🟩 206.82	🟩 95	🟩 714	🟩 96
12af9727-5555-4973-81af-1c31e290f487	🟩 8	🟥 465	🟩 520.94	🟥 89	🟩 1366	🟩 80
15586257-3f60-4ef1-a03d-715bca6f8cb8	🟩 16	🟥 634	🟩 228.08	🟩 99	🟩 1230	🟩 89
1a101c85-240b-4563-88a2-4f2fa8436ef3	🟩 8	🟩 686	🟩 310.18	🟥 88	🟩 583	🟩 89
1a9ab04e-0079-4ebf-b015-7b0ede2d5f85	🟩 5	🟥 500	🟩 2	🟩 95	🟥 400	🟩 65
1af2c190-be8b-4004-bd49-f4758e358fcd	🟩 16	🟥 407	🟩 865.48	🟩 96	🟩 1603	🟩 96

Figure 5-14 UI of client side showing SLA parameters compliance in foreign cloud (Red color shows SLA violations while green shows SLA compliance)

To measure the performance SLA co-ordinator and effectiveness of monitoring we did experiments to measure the accuracy monitoring component during service execution in the foreign cloud. A basic user interface (UI) was created on the client side to report any SLA violations of the SLA metrics. Figure 5.14 shows the client UI after accessing a few services in the foreign cloud. The boxes in red are SLA violations that were captured while green boxes indicate the SLA parameters that were successfully implemented and followed.

5.5 Limitations

To determine the valid threshold of an SLA parameter for an application we used historical data. Our experiments are based on web applications for which we defined thresholds for parameters such as availability, throughput and accessibility. However, these thresholds might change for various kinds of applications and for providing user access to certain kinds of applications in a foreign cloud using MCC, their historical data will need to be determined for having valid thresholds. Moreover, to determine the cost of measurements and defining the best measurement interval we focused on web applications as they represent a popular class of applications in cloud. The measurement interval of 25 sec was concluded as the best interval with the least measurement cost and the highest number of detected SLA violations, however, the most suitable time interval might change for other applications.

5.6 Summary

To summarise, multi clouds offer a promising solution to efficiently deliver services but their adoption also raises challenges due to lack of supporting frameworks. This chapter provides an extension of the proposed model to establish secure collaboration across multi-clouds to access services running in the foreign cloud. Service scheduling technique is proposed to select the best service matching user requirements among multiple foreign clouds, and SLA approach is used to ensure service execution in the foreign cloud is according to the agreed SLA parameters between user and the provider. Moreover, we also present a detailed system design to implement these protocols and model. The experiments are performed on two cloud systems based on OpenStack and Amazon AWS and the results show that our protocols only result in a limited overhead. Furthermore, our results indicate that smaller measurement intervals lead to higher accuracy and low cost in monitoring service execution.

Chapter 6 Conclusion and Future Work

6.1 Overview

This chapter comprises of two parts. In the first part, we present the conclusions derived from our research. We describe our research goals, major contributions and the results achieved during this research. In the later part of this chapter, the future directions of our research work are discussed.

6.2 Major Contributions

Cloud has been widely adopted to deliver services, but it also introduces challenges due to the limitations of resources and services in a single cloud. Multi-clouds offer a promising solution to efficiently deliver services, but their adoption also raises challenges due to lack of supporting frameworks. A comprehensive literature review was carried out in the area of cloud computing and multi-clouds to critically review the existing methodologies in multi-cloud collaborations. We analysed state-of-the-art research mechanisms for multi-cloud environments and identified research gaps in multi-cloud collaboration, which reflects the first objective of this thesis. This literature review directed the research in two core areas: firstly, developing security mechanisms that can ensure the dynamic collaboration between multi-clouds depending on user requirements even if there is no pre-arranged agreement between those clouds to communicate, secondly, to secure the communication that has been established between multi-clouds starting from authentication, authorization, and service selection followed by its execution in foreign cloud.

In the second phase of this research threat modelling of services in cloud computing was carried out. In this thesis, the threat modelling approach has been presented to determine the threats for cloud services. Along with the important threats to services, this research also presents the methodologies to exploit those threats. Different possible attackers who can exploit specific threats have also been identified. The threat model presents a holistic picture of services security in the cloud through structured analysis in which security requirements, threats and attacks on cloud services correspond to each other. Threat modelling of services can be generalized to determine the threats for specific service interfaces user is accessing so that possible security mechanisms for those interfaces can be implemented. Moreover, this model can be used to determine potential threats in relation to a foreign cloud architecture in which user will be accessing services.

The third objective has been achieved across chapter 4. Here, the novel multi-cloud collaboration (MCC) model is proposed that provides an approach to establish secure collaboration across multi-clouds to access services running in the foreign cloud. An authentication scheme to establish dynamic authentication between communicating clouds is presented based on a single sign on (SSO) authentication mechanism by which any cloud user in the local cloud can authenticate itself with foreign cloud and access required resources. It offers an end to end cryptographic solution for multi-clouds and satisfies the pre-requisites of multi-cloud collaboration such as having no prior trust provisioning. Various secure collaboration protocols are proposed for MCC and formally verified using the BAN logic. Moreover, we also present the detailed model design to implement these protocols. The experiments are performed on two cloud systems based on OpenStack and the results shows that our protocols only result in a limited overhead compared to traditional security protocols such as SAML and Kerberos. The implementation of key MCC components such as controller into various modular components allows to handle higher request loads and reduces failures risks.

The fourth objective of our research is achieved across chapter 5. It presents an extended version of the model MCC detailed in the previous chapter and provides additional features such as service scheduling and service level agreement (SLA). Service scheduling is proposed to select the best service matching user quality of service (QoS) requirements among a dynamic decentralized multi-clouds environment comprising multiple foreign clouds with high accuracy. By using the distance co-relation weighting mechanism our model can support various services QoS requirements such as response time, availability, reliability, throughput and latency. The SLA approach is used to ensure service execution in a foreign cloud is according to the agreed SLA (including both functional and non-functional) parameters between the user and the provider. The experiments were performed on two cloud systems based on OpenStack and Amazon AWS and various measurement intervals are used to measure changing behavior of an application during its execution. We calculated SLA violations for different measurement intervals and determine the best intervals to measure SLA violations, and considering cost and system overhead the 25 seconds interval is chosen as the minimum and optimum measurement interval. This is due to the fact the for higher measurement intervals the number of missed SLA violations is very high. Moreover, the scalability and accuracy of the scheduling algorithm was examined and it was successfully able to select the service with the highest match of QoS properties using accuracy matrix.

The proposed model is an architectural solution that can be deployed by the organization on top of their existing business models. Therefore, we present various use cases of integrating the proposed system with big data applications like IoT and E-Healthcare along with presenting the model for using proposed model (MCC) for meta-scheduling in multi-clouds.

6.3 Future Work

In the proposed work, chapter 3 proposes a threat model for the identification of security threats to different service functionalities in the presence of a specific attacker. Based on the threat model, "***automated reasoning techniques***" can be applied on the model to determine possible threats for a service interface and determine the best security mechanisms to secure that service in the event of those possible threats. Using semantic analysis and automated reasoning can help increase portability and interoperability when multiple parties such as cloud users, providers and administrators are involved. Although having additional security features can impact the performance of the system, implementing necessary mechanisms to protect services and data from most critical threats can significantly decrease the impact of an attack on a service.

In chapters 4 and 5, the MCC model has been proposed for providing the users with dynamic and secure access to services running in the foreign cloud. In the experiments section of chapter 4, it is discussed that controller needs to be implemented as a modular component and a messaging service forwards authentication requests and collaboration messages to foreign clouds. In order to manage large number of requests in MCC, "***load balancing***" techniques for cloud can be applied to make decisions based on incoming load and manage efficient networking such as task migration. Moreover, integrating load balancing with scheduling module can help to ensure required QoS metrics are maintained by efficient allocation of tasks to foreign clouds by "network aware and workflow specific scheduling and load balancing".

Along with load balancing, *“**fault tolerance**”* techniques can be applied to modular MCC architecture to catch a fault at runtime (which may be due to hardware, software faults, or malicious attackers) and deal with it without affecting system performance. For a multi-cloud collaboration and orchestration, fault detection can be implemented at local cloud in controller (hardware, software) and scheduler (task) level. Other faults such as third party authenticator failure or network congestion can also be added. Moreover, fault prevention and recovery can be implemented at scheduler (task) level or user level in which exceptions can be defined based on SLAs or QoS requirements to validate proper execution.

The MCC model gives users the ability to access particular services in foreign clouds which is a significant advantage to cloud users. However, currently very limited pricing models have been proposed for multi clouds that can provide an accurate estimate to users about the cost of using services dynamically. *“**Pricing models**”* for multiple clouds need to incorporate rates for leasing, QoS delivery, SLA management, security and management costs. Developing the pricing models for dynamic collaboration and integrating with the proposed model can benefit organizations in moving towards multi-cloud adoption and maximize the revenues. The pricing models can be incorporated with scheduling techniques to chose an optimal cloud with reasonable price.

In chapter 5, we discuss that to ensure the functional security properties of services in foreign cloud various architectures have been proposed. Although hardware techniques such as Trusted Platform Module (TPM) can be employed to ensure secure service execution, they have considerable performance issues. Therefore, “***enhanced cryptograph-based software mechanisms***” can be developed to secure functional properties of services and applications while they are being executed in the foreign cloud.

Appendix A: Use Cases of Multi-Cloud Collaboration (MCC) Model

A.1 Overview

The proposed model is based on an architectural solution that can be used to setup multi-cloud collaboration between any clouds irrespective of their underlying implementation. In this chapter, we present the cases of applying the proposed model to real-world scenarios like meta-scheduling in multi-clouds, e-health and Internet of Things (IoT). We also describe the benefits of applying the proposed model to these applications.

A.2 Orchestrating Dynamic and Secure Access of IoT Services across Multi-Clouds Using MCC

The Internet of Things (IoT) devices have complex requirements but their limitations in terms of storage, network, computing, data analytics, scalability and big data management require them to be used it with a technology like cloud computing. IoT backend with cloud computing can present new ways to offer services that are massively scalable, can be dynamically configured, and delivered on demand with large-scale infrastructure resources. However, a single cloud infrastructure might be unable to deal with the increasing demand of cloud services in which hundreds of users might be accessing cloud resources, leading to a big data problem and the need for efficient frameworks to handle large amount of user requests for IoT services.

These challenges require new functional elements and provisioning schemes. To this end, we propose the usage of proposed system for multi-cloud collaboration with IoT technologies. This can optimise the user requirements by allowing them to choose best IoT services from many services hosted in various cloud platforms, and provide them with more infrastructure and platform resources to meet their requirements. We present a novel model for dynamic and secure IoT services access across multi-clouds using cloud on-demand model.

A.2.1 Motivation for IoT Services Access across Multi-Clouds

The Internet of Things (IoT) paradigm has revolutionized the IT industry by bringing together technologies such as Radio Frequency Identification (RFID), Wireless Sensor and Actor Networks (WSANs) and ubiquitous computing domains. Internet of Things (IoT) connects billions of devices over the Internet. The heterogeneous IoT objects are provided with sensing and actuation capabilities, that enable them to capture information from physical objects and send it as data streams [134]. Moreover, IoT objects directly co-operate with physical and virtual resources over the internet to deliver data and functionalities to end users and applications.

IoT has played a critical role in advancing human lives by bringing applications with usage in real world. From user's perspective, IoT plays a critical role in application scenarios such as smart homes, healthcare, vehicular networks, and enhanced learning. While from business view point, the major applications of IoT are in the areas of logistics, transportation, agriculture, retail and smart cities. It is predicted that the growth of global IoT services market will be at a compound annual growth rate (CAGR) of 24 percent until 2021 [135]. As the number of IoT devices increases and they generate large volumes of big data, it brings forwards the challenges related to data collection, analysis, management, and storage.

Cloud computing has been proposed as a solution that can potentially solve the problem of managing big data in IoT [136]. In particular, the multi-cloud architecture can provide a solution to handling IoT big data as well as the variety and proliferation of its services offers. In terms of IoT, the services to be shared between multi-clouds can include SaaS, PaaS or IaaS service while the clients using these services can be other clouds, organizations or a single user.

A.2.2 Using MCC to Access IoT Services

In this section, we propose an architecture called MC-IoT to provide users with an ability to dynamically access IoT services across multi-clouds. In an IoT based multi-cloud architecture, hundreds of cloud users might be using thousands of IoT services across multi-clouds. MC-IoT has been designed with the goal to enhance secure multi-cloud collaboration and provide an advanced development model that can reduce a company's time-to-market.

The multi-cloud communication scenario that provides access of IoT services to users across multi-clouds is shown in figure A.1.

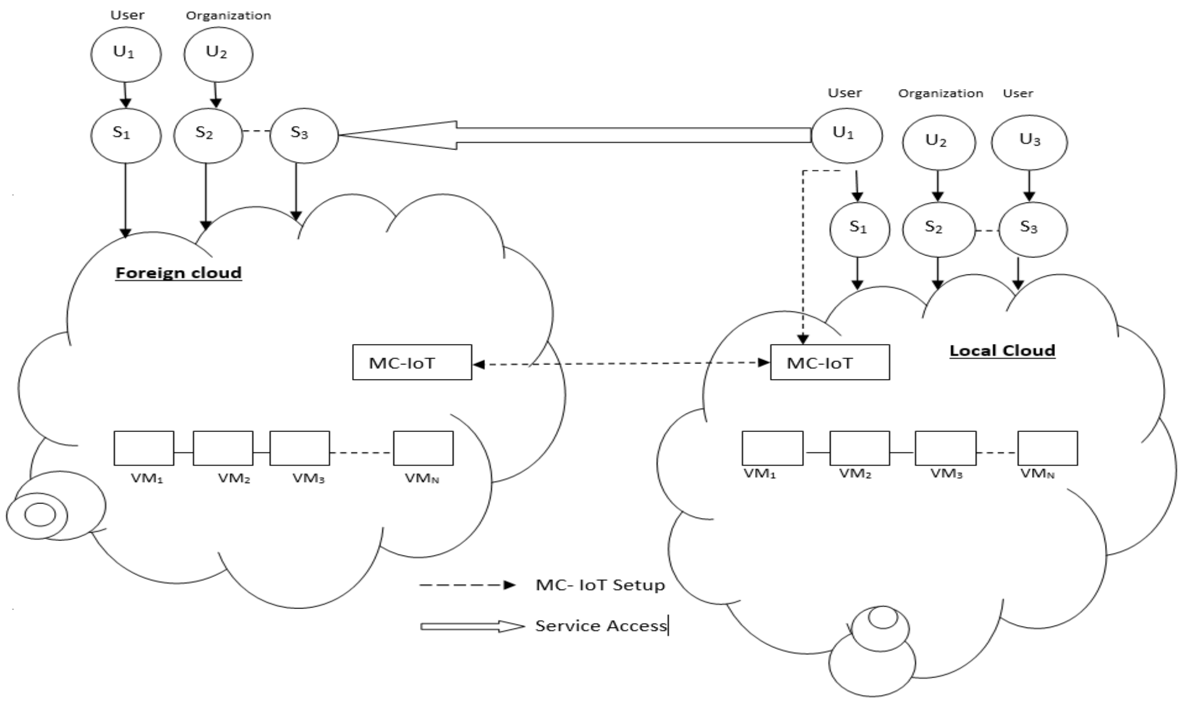


Figure A-1 Multi-cloud collaboration scenario where user U1 made a request to MC-IoT in local cloud to access service S3 in foreign cloud. The multi-cloud collaboration is setup using MC-IoT after which U1 can directly access service S3.

Based on heterogeneous requirements of multi-clouds and IoT services, MC-IoT can enable dynamic collaboration between users and services in multi-clouds. The architecture of MC-IoT shown in figure A.1 is based on MCC that has various components implemented in each participating cloud to achieve the secure multi cloud authentication, service selection and service level agreement (SLA) management discussed in section 4.3.2.

A.3 Orchestrating Dynamic and Secure Access of E-HealthCare Applications across Multi-Clouds Using MCC

Ubiquitous healthcare using sensor-based e-Healthcare systems has been considered as a future of modern healthcare. The healthcare systems include collaborative sharing of e-Health data and services which requires new functional elements and provisioning schemes that can be met by multi-clouds. However, lack of systems for e-Healthcare usage in multi-clouds can reduce the adoption of such systems at large scale. We present a case for using multi-clouds to process a large amount of sensor data for e-Healthcare systems and provide a novel model that can be used to enable secure data sharing and e-Health services access across multi-clouds.

A.3.1 Motivation for E-Health Application Access across Multi-Clouds

Healthcare solutions enable the delivery of healthcare services any required time, however its deployment also raises several challenges. The population of people aged over 65 is expected to increase in developed countries in the next 20 years, and it will also bring more healthcare challenges. Due to rise in healthcare cost, more sophisticated procedures such as e-Healthcare are required. Sensor based e-Healthcare systems can monitor patient’s health remotely and doctor can view patients health using e-Health applications without the need of patients visiting doctor.

The Sensors have seen a widespread adoption in medical sector due to increase in healthcare costs, and need for healthcare systems in medical diagnosis. Their usage in medical industry varies from simple applications like (body temperature measurement) to complex applications. Some other types of sensors used in healthcare are pressure, flow, image and bio sensors. Sensors play a critical role to increase safety and improve the life quality as they are cost-effective, accurate, reliable, smart, smaller in size and consume much less energy. Due to these characteristics sensors have gained an important place in e-Healthcare applications. The sensors market in healthcare is estimated to reach 1.9 billion USD by 2022 [137].

The use of electronic and communication devices in healthcare is referred to as e-Healthcare. Healthcare sensors including implantable, self-powered, bio-sensors, micro-electromechanical silicon and nano-sensors can potentially bring huge benefits to e-Healthcare industry in the coming years. Some of the benefits offered to patients include remote monitoring of patients with chronic illness, helping in treatment of diseases, and monitoring of health statistics by patients themselves can help them to steps to improve their health. With the significant advantages offered by using sensor data in healthcare, the challenge arises with storing huge amount of data generated by sensors. Moreover, e-Healthcare requires data processing, storage and analytics that can be potentially be used by collaborative healthcare entities and applications.

Recently, multi-clouds have been proposed as a solution that can potentially solve the problem of managing big data in healthcare among collaborative entities [96]. Despite the benefits offered by multi-cloud to e-Healthcare sector, there are several key challenges associated with the adoption of multi-clouds, particularly healthcare system lack the solutions and frameworks that can facilitate its usage. Many healthcare organizations are still hesitant to use cloud due to security issues.

A.3.2 Proposed Model to Access E-HealthCare Applications across Multi-Clouds

In a cloud hosted healthcare system, cloud acts as a data centre for healthcare centre and it is responsible to host all applications, services and data. The user perspective in this case remains the same, as they are concerned with traditional health services including e-Health services and underlying resources are transparent to them. The proposed architecture for integrating healthcare including e-Heath services and sensor-based data with multi-clouds is shown in figure. As shown in the figure 1, users including patients and healthcare workers will only need to get authenticated by their local cloud and the proposed system will enable them to use services in foreign clouds according to requirement. The proposed system design can revolutionize the healthcare by providing key benefits such as ability to use multiple e-Health services on various platforms, scale computing resources such as storage according to requirements and share collaborative data with health care workers from other clouds.

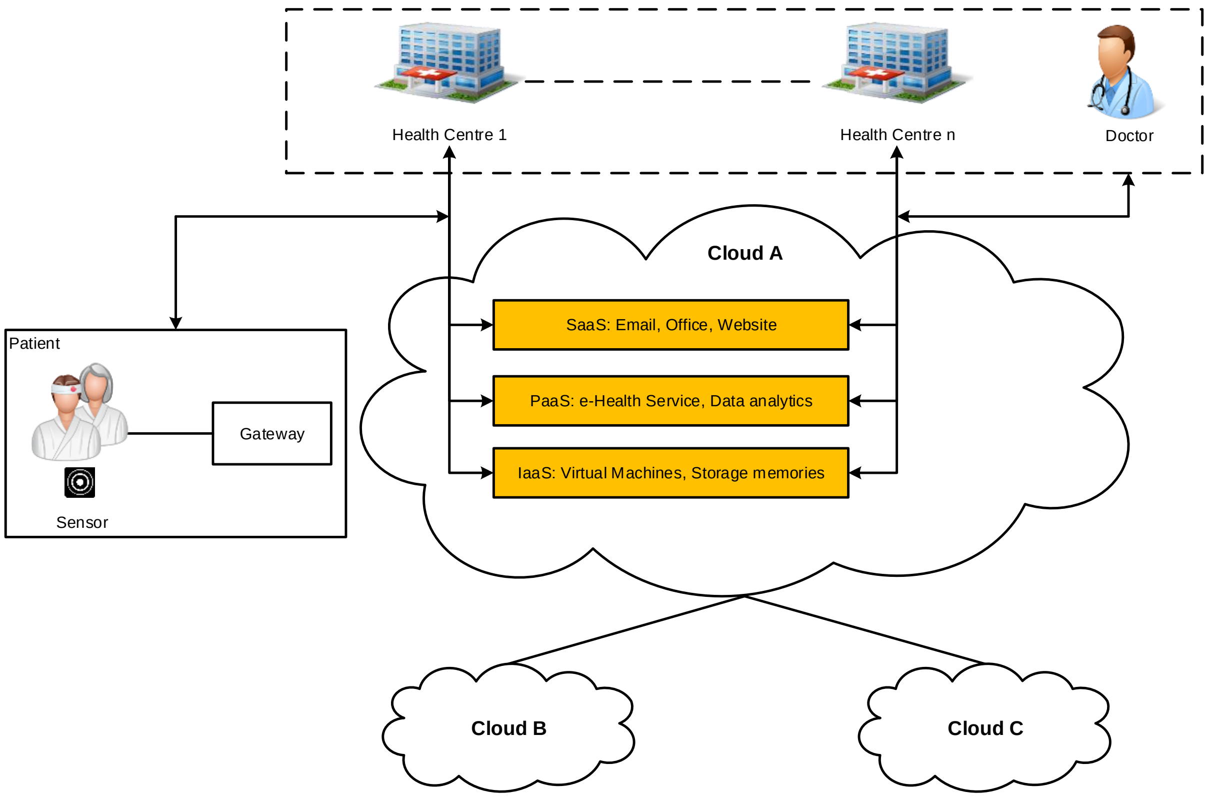

Health Centre 1
Health Centre n
Doctor
Cloud A
SaaS: Email, Office, Website
PaaS: e-Health Service, Data analytics
IaaS: Virtual Machines, Storage memories
Patient
Gateway
Sensor
Cloud B
Cloud C

Figure A-2 Proposed e-Healthcare system using multi-clouds

In order to establish multi-cloud collaboration, a new system has been presented. The architecture of the proposed system involves various components that have been implemented in each participating cloud to achieve the secure multi-cloud authentication and collaboration as shown in figure A.2. These components involved in system design are cloud controller, cloud manager and authenticator, and how these components handle the communication requests coming from foreign clouds is discussed in section 4.3.2.

A.3.3 Benefits of the Proposed Model

Multi-cloud system can provide a service based and application-oriented infrastructure that can be suitable for e-Healthcare systems due to many reasons including the following:

- Healthcare workers need inter-organizational and collaborative data sharing
- Some e-Health services need a specific platform to run

- Healthcare workers might need to use a e-Health service being run on remote platform only for a limited period that will be economically inefficient to be purchased for long time
- Sensors generate a large amount of medical data
- Number of patient’s records being managed is very large
- Performing data analytics on large datasets of healthcare needs more resources than traditional infrastructure.

Based on heterogeneous requirements of multi-cloud and e-Healthcare services, the proposed can be integrated with healthcare systems resulting in a novel system that can enable dynamic collaboration between e-Health services in multi-clouds.

Appendix B: Key Definitions

Term	Definition
Cloud	Cloud computing is a technology that offers various services ranging from infrastructure to storage, computation, software and application over the internet. The aim of cloud computing is to better utilize the distributed resources; remotely placed together, to grab maximum throughput and to combat large-scale computational problems.
Multi-cloud	A multi-cloud environment is dependent on multiple clouds, so a user can be reliant on cloud services from AWS, Microsoft, or OpenStack which are communicating. Multi-clouds are considered as a single heterogeneous architecture offering on-demand services from multiple cloud providers
Distributed System	A distributed system is a collection of computers working together that share states and operate concurrently to act as a single coherent network serving a user.

Local cloud	In this thesis, we refer to a cloud making a connection request to an external cloud as a local cloud. Essentially, during collaboration it is a cloud in which user making a request to access external resources is located.
Foreign cloud	Foreign cloud is referred to as a cloud to which the user makes a connection request. In multi-cloud collaboration, foreign cloud has resources to which the user making a request needs access.
Orchestration	Orchestration is a term referred to automated arrangements, services, management of complex systems and coordination among multiple computing systems.

References

1. Paraiso, F., N. Haderer, P. Merle, R. Rouvoy, and L. Seinturier. *A federated multi-cloud PaaS infrastructure*. in *2012 IEEE Fifth International Conference on Cloud Computing*. 2012. IEEE.
2. 451 Research, https://451research.com/images/Marketing/press_releases/Pre_ReInvent_2018_press_release_final_11_22.pdf, Last accessed: 15/08/2018
3. 2015, U.c.a.t.f., http://www.outsourcery.co.uk/media/1377/cif-outsourcery-white-paper-16.pdf, Last accessed: 18/08/2018
4. Grozev, N. and R. Buyya, *Inter-Cloud architectures and application brokering: taxonomy and survey.* Software: Practice and Experience, 2014. **44**(3): p. 369-390.
5. Takabi, H., J.B. Joshi, and G.-J. Ahn, *Security and privacy challenges in cloud computing environments.* IEEE Security & Privacy, 2010. **8**(6): p. 24-31.
6. Shen, Z., S. Subbiah, X. Gu, and J. Wilkes. *Cloudscale: elastic resource scaling for multi-tenant cloud systems*. in *Proceedings of the 2nd ACM Symposium on Cloud Computing*. 2011. ACM.
7. Moreno-Vozmediano, R., R.S. Montero, and I.M. Llorente, *Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures.* Computer, 2012. **45**(12): p. 65-72.
8. Gritzalis, S., D. Spinellis, and P. Georgiadis, *Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification.* Computer Communications, 1999. **22**(8): p. 697-709.
9. Reese, G., *Cloud application architectures: building applications and infrastructure in the cloud*. 2009: " O'Reilly Media, Inc.".
10. Amazon Web Services,

What is Cloud Computing?, http://aws.amazon.com/what-is-cloud-computing/, Last accessed: 20/03/2018

11. Gurav, U. and R. Shaikh. *Virtualization: a key feature of cloud computing*. in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. 2010. ACM.
12. MK, M.M. and H. Sanjay, *Scalability for Cloud*, in *Critical Research on Scalability and Security Issues in Virtual Cloud Environments*. 2018, IGI Global. p. 1-18.
13. Botta, A., W. De Donato, V. Persico, and A. Pescapé, *Integration of cloud computing and internet of things: a survey*. Future generation computer systems, 2016. **56**: p. 684-700.
14. Mohammed, B., M. Kiran, K.M. Maiyama, M.M. Kamala, and I.U. Awan, *Failover strategy for fault tolerance in cloud computing environment*. Software: Practice and Experience, 2017. **47**(9): p. 1243-1274.
15. Abdelmaboud, A., D.N. Jawawi, I. Ghani, A. Elsafi, and B. Kitchenham, *Quality of service approaches in cloud computing: A systematic mapping study*. Journal of Systems and Software, 2015. **101**: p. 159-179.
16. Goyal, S., *Public vs private vs hybrid vs community-cloud computing: a critical review*. International Journal of Computer Network and Information Security, 2014. **6**(3): p. 20.
17. Puthal, D., B. Sahoo, S. Mishra, and S. Swain. *Cloud computing features, issues, and challenges: a big picture*. in *2015 International Conference on Computational Intelligence and Networks*. 2015. IEEE.
18. Rimal, B.P., E. Choi, and I. Lumb. *A taxonomy and survey of cloud computing systems*. in *2009 Fifth International Joint Conference on INC, IMS and IDC*. 2009. Ieee.
19. Amazon Web Services, http://aws.amazon.com/, Last accessed: 26/10/2018
20. GoGrid, GoGrid, http://www.gogrid.com/, Last accessed: 06/05/2018
21. Rackspace, Rackspace Cloud, http://www.rackspacecloud.com/, Last accessed: 04/06/2018
22. Computing, O.C., http://oracle.com/us/technologies/cloud, Last accessed: 08/06/2018
23. GigaSpaces, http://www.gigaspaces.com/, Last accessed: 08/6/2018
24. RightScale, http://www.rightscale.com/, Last accessed: 09/06/2018
25. Chicago, U.o., Nimbus is Cloud Computing for Science, http://www.nimbusproject.org/, Last accessed: 09/06/2018
26. Google App Engine, https://cloud.google.com/appengine/, Last accessed: 08/06/2018
27. Microsoft Azure, https://azure.microsoft.com/en-us/, Last accessed: 11/10/2018

28. Oracle Cloud Platform, https://cloud.oracle.com/home, Last accessed: 09/06/2018

29. Höfer, C. and G. Karagiannis, *Cloud computing services: taxonomy and comparison.* Journal of Internet Services and Applications, 2011. **2**(2): p. 81-94.

30. Hecht, T., P. Smith, and M. Schöller. *Critical services in the cloud: Understanding security and resilience risks*. in *2014 6th International Workshop on Reliable Networks Design and Modeling (RNDM)*. 2014. IEEE.

31. Whaiduzzaman, M., M.N. Haque, M. Rejaul Karim Chowdhury, and A. Gani, *A study on strategic provisioning of cloud computing services.* The Scientific World Journal, 2014. **2014**.

32. Choudhary, V. *Software as a service: Implications for investment in software development*. in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*. 2007. IEEE.

33. Subashini, S. and V. Kavitha, *A survey on security issues in service delivery models of cloud computing.* Journal of network and computer applications, 2011. **34**(1): p. 1-11.

34. Hindy, H., D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, *A taxonomy and survey of intrusion detection system design techniques, network threats and datasets.* arXiv preprint arXiv:1806.03517, 2018.

35. Grobauer, B., T. Walloschek, and E. Stocker, *Understanding cloud computing vulnerabilities.* IEEE Security & privacy, 2010. **9**(2): p. 50-57.

36. Mather, T., S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. 2009: " O'Reilly Media, Inc.".

37. Keene, C., *The Keene View on Cloud Computing*, 2009, Accessed.

38. Xu, K., X. Zhang, M. Song, and J. Song. *Mobile mashup: Architecture, challenges and suggestions*. in *2009 International Conference on Management and Service Science*. 2009. IEEE.

39. Hashizume, K., D.G. Rosado, E. Fernández-Medina, and E.B. Fernandez, *An analysis of security issues for cloud computing.* Journal of internet services and applications, 2013. **4**(1): p. 5.

40. Rittinghouse, J.W. and J.F. Ransome, *Cloud computing: implementation, management, and security*. 2016: CRC press.

41. Chandramouli, R. and P. Mell, *State of security readiness.* XRDS: Crossroads, The ACM Magazine for Students, 2010. **16**(3): p. 23-25.

42. Dahbur, K., B. Mohammad, and A.B. Tarakji. *A survey of risks, threats and vulnerabilities in cloud computing*. in *Proceedings of the 2011 International conference on intelligent semantic Web-services and applications*. 2011. ACM.

43. Jaeger, T. and J. Schiffman, *Outlook: Cloudy with a chance of security challenges and improvements.* IEEE Security & Privacy, 2010. **8**(1): p. 77-80.

44. Jasti, A., P. Shah, R. Nagaraj, and R. Pendse. *Security in multi-tenancy cloud*. in *44th Annual 2010 IEEE International Carnahan Conference on Security Technology*. 2010. IEEE.

45. Garfinkel, T. and M. Rosenblum. *When Virtual Is Harder than Real: Security Challenges in Virtual Machine Based Computing Environments*. in *HotOS*. 2005.

46. Reuben, J.S., *A survey on virtual machine security.* Helsinki University of Technology, 2007. **2**(36).

47. Owens, D., *Securing elasticity in the cloud.* Communications of the ACM, 2010. **53**(6): p. 46-51.

48. Ertaul, L., S. Singhal, and G. Saldamli. *Security Challenges in Cloud Computing*. in *Security and Management*. 2010.

49. Varghese, B. and R. Buyya, *Next generation cloud computing: New trends and research directions.* Future Generation Computer Systems, 2018. **79**: p. 849-861.

50. Rani, B.K., B.P. Rani, and A.V. Babu, *Cloud computing and inter-clouds–types, topologies and research issues.* Procedia Computer Science, 2015. **50**: p. 24-29.

51. Petcu, D., G. Macariu, S. Panica, and C. Crăciun, *Portable cloud applications—from theory to practice.* Future Generation Computer Systems, 2013. **29**(6): p. 1417-1430.

52. Wu, Z. and H.V. Madhyastha, *Understanding the latency benefits of multi-cloud webservice deployments.* ACM SIGCOMM Computer Communication Review, 2013. **43**(2): p. 13-20.

53. Buyya, R., R. Ranjan, and R.N. Calheiros. *Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services*. in *International Conference on Algorithms and Architectures for Parallel Processing*. 2010. Springer.

54. Team, A., Summary of the amazon ec2 and amazon RDS service disruption in the us east region., https://aws.amazon.com/message/65648/, Last accessed: 03/06/2017

55. Xu, X. and X. Zhao. *A framework for privacy-aware computing on hybrid clouds with mixed-sensitivity data*. in *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International*

Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. 2015. IEEE.

56. Toosi, A.N., R.O. Sinnott, and R. Buyya, *Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka.* Future Generation Computer Systems, 2018. **79**: p. 765-775.

57. Rochwerger, B., C. Vázquez, D. Breitgand, D. Hadas, M. Villari, P. Massonet, E. Levy, A. Galis, I.M. Llorente, and R.S. Montero, *An architecture for federated cloud computing.* Cloud Computing, 2010: p. 391-411.

58. Sotomayor, B., R.S. Montero, I.M. Llorente, and I. Foster, *Virtual infrastructure management in private and hybrid clouds.* IEEE Internet computing, 2009. **13**(5): p. 14-22.

59. Catteddu, D. and G. Hogben, *Cloud computing Risk Assessment: Benefits, risks and recommendations for information security.* ENISA report. Online: http://www. enisa. europa. eu/act/rm/files/deliverables/cloud-computing-riskassessment/at_download/fullReport, 2009.

60. OPTIMIS, Cloud Legal Guidelines, OPTIMIS FP7 project deliverable no. D7.2.1.1., http://www.optimis-project.eu/sites/default/files/D7.2.1.1~OPTIMIS~Cloud~Legal~Guidelines.pdf, Last accessed: 07/06/2017

61. Rochwerger, B., D. Breitgand, E. Levy, A. Galis, K. Nagin, I.M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, and J. Caceres, *The reservoir model and architecture for open federated cloud computing.* IBM Journal of Research and Development, 2009. **53**(4): p. 4: 1-4: 11.

62. Carlini, E., M. Coppola, P. Dazzi, L. Ricci, and G. Righetti. *Cloud federations in contrail*. in *European Conference on Parallel Processing*. 2011. Springer.

63. Jofre, J., C. Velayos, G. Landi, M. Giertych, A.C. Hume, G. Francis, and A.V. Oton, *Federation of the BonFIRE multi-cloud infrastructure with networking facilities.* Computer Networks, 2014. **61**: p. 184-196.

64. Petcu, D., B. Di Martino, S. Venticinque, M. Rak, T. Máhr, G.E. Lopez, F. Brito, R. Cossu, M. Stopar, and S. Šperka, *Experiences in building a mOSAIC of clouds.* Journal of Cloud Computing: Advances, Systems and Applications, 2013. **2**(1): p. 12.

65. Buyya, R., C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.* Future Generation computer systems, 2009. **25**(6): p. 599-616.

66. Yang, X., B. Nasser, M. Surridge, and S. Middleton, *A business-oriented cloud federation model for real-time applications.* Future Generation Computer Systems, 2012. **28**(8): p. 1158-1167.

67. Zeginis, D., F. D'andria, S. Bocconi, J.G. Cruz, O.C. Martin, P. Gouvas, G. Ledakis, and K.A. Tarabanis, *A user-centric multi-PaaS application management solution for hybrid multi-Cloud scenarios.* Scalable Computing: Practice and Experience, 2013. **14**(1): p. 17-32.

68. Mazur, L., J. Xie, S.D. Daniel, and C.J. Saretto, *Authenticating using cloud authentication*, 2013, Google Patents.

69. Krutz, R.L. and R.D. Vines, *Cloud security: A comprehensive guide to secure cloud computing*. 2010: Wiley Publishing.

70. Kandukuri, B.R. and A. Rakshit. *Cloud security issues*. in *2009 IEEE International Conference on Services Computing*. 2009. IEEE.

71. Nacer, H., N. Djebari, H. Slimani, and D. Aissani, *A distributed authentication model for composite Web services.* Computers & Security, 2017. **70**: p. 144-178.

72. Noureddine, M. and R. Bashroush, *An authentication model towards cloud federation in the enterprise.* Journal of Systems and Software, 2013. **86**(9): p. 2269-2275.

73. Celesti, A., F. Tusa, M. Villari, and A. Puliafito. *Three-phase cross-cloud federation model: The cloud sso authentication*. in *2010 Second International Conference on Advances in Future Internet*. 2010. IEEE.

74. Polzonetti, A. and A. Bettacchi. *Cloud services implementing security: a case study*. in *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*. 2017. ACM.

75. Demchenko, Y., C. Ngo, C. de Laat, and C. Lee. *Federated access control in heterogeneous intercloud environment: Basic models and architecture patterns*. in *2014 IEEE International Conference on Cloud Engineering*. 2014. IEEE.

76. Celesti, A., F. Tusa, M. Villari, and A. Puliafito, *Federation establishment between CLEVER clouds through a SAML SSO authentication profile.* Int. J. on Adv. in Internet Tech, 2011. **4**(1).

77. Celesti, A., F. Tusa, M. Villari, and A. Puliafito. *Security and cloud computing: Intercloud identity management infrastructure*. in *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*. 2010. IEEE.

78. Li, B., J. Li, L. Liu, and C. Zhou, *Toward a flexible and fine-grained access control framework for infrastructure as a service clouds.* Security and Communication Networks, 2016. **9**(15): p. 2730-2743.

79. Calero, J.M.A., N. Edwards, J. Kirschnick, L. Wilcock, and M. Wray, *Toward a multi-tenancy authorization system for cloud services.* IEEE Security & Privacy, 2010. **8**(6): p. 48-55.

80. Migault, D., M.A. Simplicio, B.M. Barros, M. Pourzandi, T.R. Almeida, E.R. Andrade, and T.C. Carvalho. *A framework for enabling security services collaboration across multiple domains*. in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017. IEEE.

81. Almorsy, M., J. Grundy, and A.S. Ibrahim. *Collaboration-based cloud computing security management framework*. in *2011 IEEE 4th International Conference on Cloud Computing*. 2011. IEEE.

82. Gouglidis, A., I. Mavridis, and V.C. Hu, *Security policy verification for multi-domains in cloud systems.* International Journal of Information Security, 2014. **13**(2): p. 97-111.

83. Paladi, N., A. Michalas, and H.-V. Dang. *Towards secure cloud orchestration for multi-cloud deployments*. in *Proceedings of the 5th Workshop on CrossCloud Infrastructures & Platforms*. 2018. ACM.

84. Shao, J., H. Wei, Q. Wang, and H. Mei. *A runtime model based monitoring approach for cloud*. in *2010 IEEE 3rd International Conference on Cloud Computing*. 2010. IEEE.

85. Alhamazani, K., R. Ranjan, K. Mitra, P.P. Jayaraman, Z. Huang, L. Wang, and F. Rabhi. *Clams: Cross-layer multi-cloud application monitoring-as-a-service framework*. in *2014 IEEE International Conference on Services Computing*. 2014. IEEE.

86. Casola, V., A. De Benedictis, and M. Rak. *Security monitoring in the cloud: an SLA-based approach*. in *2015 10th International Conference on Availability, Reliability and Security*. 2015. IEEE.

87. Rebollo, O., D. Mellado, and E. Fernandez-Medina, *Isgcloud: a security governance framework for cloud computing.* The Computer Journal, 2014. **58**(10): p. 2233-2254.

88. Lee, C.-Y., K.M. Kavi, R.A. Paul, and M. Gomathisankaran. *Ontology of secure service level agreement*. in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*. 2015. IEEE.

89. Faniyi, F., R. Bahsoon, and G. Theodoropoulos, *A dynamic data-driven simulation approach for preventing service level agreement violations in cloud federation.* Procedia Computer Science, 2012. **9**: p. 1167-1176.

90. Rak, M., N. Suri, J. Luna, D. Petcu, V. Casola, and U. Villano. *Security as a service using an SLA-based approach via SPECS*. in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*. 2013. IEEE.

91. Sotiriadis, S., N. Bessis, F. Xhafa, and N. Antonopoulos. *From meta-computing to interoperable infrastructures: A review of meta-schedulers for HPC, grid and cloud*. in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*. 2012. IEEE.

92. Jakóbik, A., D. Grzonka, and F. Palmieri, *Non-deterministic security driven meta scheduler for distributed cloud organizations.* Simulation Modelling Practice and Theory, 2017. **76**: p. 67-81.

93. Subramani, V., R. Kettimuthu, S. Srinivasan, and S. Sadayappan. *Distributed job scheduling on computational grids using multiple simultaneous requests*. in *Proceedings 11th IEEE International Symposium on High Performance Distributed Computing*. 2002. IEEE.

94. Abdullah, M. and M. Othman, *Cost-based multi-QoS job scheduling using divisible load theory in cloud computing.* Procedia computer science, 2013. **18**: p. 928-935.

95. Tordsson, J., R.S. Montero, R. Moreno-Vozmediano, and I.M. Llorente, *Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers.* Future generation computer systems, 2012. **28**(2): p. 358-367.

96. Fabian, B., T. Ermakova, and P. Junghanns, *Collaborative and secure sharing of healthcare data in multi-clouds.* Information Systems, 2015. **48**: p. 132-150.

97. Sultan, N., *Making use of cloud computing for healthcare provision: Opportunities and challenges.* International Journal of Information Management, 2014. **34**(2): p. 177-184.

98. Wu, H., Q. Wang, and K. Wolter. *Mobile healthcare systems with multi-cloud offloading*. in *2013 IEEE 14th International Conference on Mobile Data Management*. 2013. IEEE.

99. Hung, K., Y.-T. Zhang, and B. Tai. *Wearable medical devices for tele-home healthcare*. in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2004. IEEE.

100. Myagmar, S., A.J. Lee, and W. Yurcik. *Threat modeling as a basis for security requirements*. in *Symposium on requirements engineering for information security (SREIS)*. 2005. Citeseer.

101. Lockhart, H., S. Andersen, J. Bohren, Y. Sverdlov, M. Hondo, H. Maruyama, A. Nadalin, N. Nagaratnam, T. Boubez, and K. Morrison, *Web services federation language (WS-federation) version 1.1*. International Business Machines Corp., URL: http://www-128. ibm. com/developerworks/library/specification/ws-fed/(17.08. 2007), 2006.

102. Alliance, C.S., *Top threats to cloud computing v1. 0*. White Paper, 2010. **23**.

103. Zhang, Y., A. Juels, M.K. Reiter, and T. Ristenpart. *Cross-VM side channels and their use to extract private keys*. in *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012. ACM.

104. Maniatis, P., D. Akhawe, K.R. Fall, E. Shi, and D. Song. *Do You Know Where Your Data Are? Secure Data Capsules for Deployable Data Protection*. in *HotOS*. 2011.

105. Ristenpart, T., E. Tromer, H. Shacham, and S. Savage. *Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds*. in *Proceedings of the 16th ACM conference on Computer and communications security*. 2009. ACM.

106. Brown, A. and J.S. Chase. *Trusted platform-as-a-service: a foundation for trustworthy cloud-hosted applications*. in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. 2011. ACM.

107. Du, J., W. Wei, X. Gu, and T. Yu. *RunTest: assuring integrity of dataflow processing in cloud computing infrastructures*. in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. 2010. ACM.

108. Santos, N., K.P. Gummadi, and R. Rodrigues, *Towards Trusted Cloud Computing*. HotCloud, 2009. **9**(9): p. 3.

109. Santos, N., R. Rodrigues, K.P. Gummadi, and S. Saroiu. *Policy-sealed data: A new abstraction for building trusted cloud services*. in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. 2012.

110. Fernandes, D.A., L.F. Soares, J.V. Gomes, M.M. Freire, and P.R. Inácio, *Security issues in cloud environments: a survey*. International Journal of Information Security, 2014. **13**(2): p. 113-170.

111. Chow, R., P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina. *Controlling data in the cloud: outsourcing computation without outsourcing control*.

in *Proceedings of the 2009 ACM workshop on Cloud computing security*. 2009. ACM.

112. Lau, B., S. Chung, C. Song, Y. Jang, W. Lee, and A. Boldyreva. *Mimesis aegis: A mimicry privacy shield–a system's approach to data privacy on public cloud*. in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 2014.

113. Kazim, M., R. Masood, M.A. Shibli, and A.G. Abbasi. *Security aspects of virtualization in cloud computing*. in *IFIP International Conference on Computer Information Systems and Industrial Management*. 2013. Springer.

114. Pappas, V., V. Kemerlis, A. Zavou, M. Polychronakis, and A.D. Keromytis, *CloudFence: Enabling users to audit the use of their cloud-resident data.* 2012.

115. Kazim, M. and S.Y. Zhu, *A survey on top security threats in cloud computing.* 2015.

116. Chen, D. and H. Zhao. *Data security and privacy protection issues in cloud computing*. in *2012 International Conference on Computer Science and Electronics Engineering*. 2012. IEEE.

117. Mannan, M., B.H. Kim, A. Ganjali, and D. Lie. *Unicorn: two-factor attestation for data security*. in *Proceedings of the 18th ACM conference on Computer and communications security*. 2011. ACM.

118. Ali, M., S.U. Khan, and A.V. Vasilakos, *Security in cloud computing: Opportunities and challenges.* Information sciences, 2015. **305**: p. 357-383.

119. Bakshi, A. and Y.B. Dujodwala. *Securing cloud from ddos attacks using intrusion detection system in virtual machine*. in *2010 Second International Conference on Communication Software and Networks*. 2010. IEEE.

120. Ferry, N., A. Rossini, F. Chauvel, B. Morin, and A. Solberg. *Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems*. in *2013 IEEE Sixth International Conference on cloud computing*. 2013. IEEE.

121. Nessett, D.M., *A critique of the burrows, abadi and needham logic.* ACM SIGOPS Operating Systems Review, 1990. **24**(2): p. 35-38.

122. OpenStack, OpenStack, https://www.openstack.org/, Last accessed: 14/08/2018

123. SAML, SAML, https://developers.onelogin.com/saml, Last accessed: 26/08/2018

124. Kerberos, Kerberos, http://web.mit.edu/kerberos/, Last accessed: 16/07/2018

125. Stantchev, V. and C. Schröpfer. *Negotiating and enforcing qos and slas in grid and cloud computing*. in *International Conference on Grid and Pervasive Computing*. 2009. Springer.

126. Zhang, F., J. Cao, K. Li, S.U. Khan, and K. Hwang, *Multi-objective scheduling of many tasks in cloud platforms.* Future Generation Computer Systems, 2014. **37**: p. 309-320.

127. Panneerselvam, J., L. Liu, and N. Antonopoulos, *An approach to optimise resource provision with energy-awareness in datacentres by combating task heterogeneity.* IEEE Transactions on Emerging Topics in Computing, 2018.

128. Awad, A., S. Kadry, B. Lee, and S. Zhang. *Property based attestation for a secure cloud monitoring system*. in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. 2014. IEEE.

129. Bouchenak, S., G. Chockler, H. Chockler, G. Gheorghe, N. Santos, and A. Shraer, *Verifying cloud services: present and future.* ACM SIGOPS operating systems review, 2013. **47**(2): p. 6-19.

130. Kazim, M. and S.Y. Zhu, *Virtualization security in cloud computing*, in *Guide to Security Assurance for Cloud Computing*. 2015, Springer. p. 51-63.

131. Emeakaroha, V.C., M.A. Netto, R.N. Calheiros, I. Brandic, R. Buyya, and C.A. De Rose, *Towards autonomic detection of SLA violations in Cloud infrastructures.* Future Generation Computer Systems, 2012. **28**(7): p. 1017-1029.

132. Celesti, A., M. Fazio, M. Giacobbe, A. Puliafito, and M. Villari. *Characterizing cloud federation in IoT*. in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 2016. IEEE.

133. Afshari, A., M. Mojahed, and R.M. Yusuff, *Simple additive weighting approach to personnel selection problem.* International Journal of Innovation, Management and Technology, 2010. **1**(5): p. 511.

134. Gubbi, J., R. Buyya, S. Marusic, and M. Palaniswami, *Internet of Things (IoT): A vision, architectural elements, and future directions.* Future generation computer systems, 2013. **29**(7): p. 1645-1660.

135. 2017 Roundup Of Internet Of Things Forecasts, https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-ofinternet-of-things-forecasts/#3a4c69321480, Last accessed: 14/08/2018

136. Botta, A., W. De Donato, V. Persico, and A. Pescapé. *On the integration of cloud computing and internet of things*. in *2014 International Conference on Future Internet of Things and Cloud*. 2014. IEEE.

137. Market For IoT Healthcare Sensors Heading To $1.9 Billion, https://www.sensorsmag.com/components/market-for-iot-healthcare-sensors-heading-to-1-9-billion, Last accessed: 03/06/2018