IDENTITY AND ACCESS MANAGEMENT IN THE CLOUD FEDERATION ENVIRONMENTS

Thesis

Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

MANOJ V. THOMAS

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

DECEMBER, 2017

To my Lord!

DECLARATION

by the Ph.D. Research Scholar

I hereby *declare* that the Research Thesis entitled **IDENTITY AND ACCESS MANAGEMENT IN THE CLOUD FEDERATION ENVIRONMENTS** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a *bonafide report of the research work carried out by me*. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Manoj V. Thomas

Reg. No.: 121175 CS12F03

Department of Computer Science and Engineering

Place: NITK, Surathkal.

Date: December 06, 2017

CERTIFICATE

This is to *certify* that the Research Thesis entitled **IDENTITY AND ACCESS MANAGEMENT IN THE CLOUD FEDERATION ENVIRONMENTS** submitted by **MANOJ V. THOMAS** (Reg. No.: 121175 CS12F03), as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of **Doctor of Philosophy**.

Prof. K. Chandrasekaran
(Research Guide)

Chairman - DRPC

ACKNOWLEDGEMENTS

I am gratefully indebted to my guide, Prof. K. Chandrasekaran for including me into his research team, and for the unfailing support I received during my research. The years I spent with him were a real learning experience. Every meeting or discussion I had with him not only widened my horizon, but also inspired and motivated me to aim higher standards and to keep a positive attitude always in my life. Without his continuous support and patience, I would not have been able to prepare this thesis.

I take this opportunity to sincerely thank the members of the RPAC committee, Dr. Annappa B. and Dr. Raj Mohan B. for their insightful comments and suggestions throughout the research for improving my work.

I thank National Institute of Technology Karnataka, Surathkal for providing all the necessary facilities during this period. I would like to thank the faculty and the staff of the Department of Computer Science and Engineering for their continuous support and help. I also thank the fellow research scholars of this department for their help, motivation and valuable suggestions.

I extend my sincere gratitude to my parents, my wife Priya and my kids Eva and Daniel for their unconditional support and all the sacrifices they had for this cause. I also sincerely thank all my friends who helped me whenever I needed along the way.

Above all, I bow my head in gratitude before the Almighty for all the blessings and gifts I received in my life.

Place: NITK, Surathkal Manoj V. Thomas

Date: December 06, 2017

ABSTRACT

Cloud Federation is an emerging technology where Cloud Service Providers (CSPs) offering specialized services to customers collaborate in order to reap the real benefits of Cloud Computing. By collaboration, the member CSPs of the federation achieve better resource utilization and Quality of Service (QoS), thereby improving their business prospects. As there are different cloud services available in the cloud federation environment, if all the variety of services have their own authentication mechanisms, the various cloud users will have to log in and verify their credentials each and every time they use a different set of services from the cloud federation. This gives rise to the multiple credentials problem. In the cloud federation environment, the Single Sign-On (SSO) authentication mechanism can be used to verify the legitimate users without requiring them to get authenticated with each service provider separately. In this thesis, we discuss the design and implementation of SSO mechanism in the cloud federation scenario using the CloudSim toolkit. We have used the Fully Hashed Menezes-Qu-Vanstone (FHMQV) protocol for the key exchange and the Symmetric Key Encryption technique AES-256 for encrypting the identity tokens in the cloud federation environment. The analysis of the results shows that the proposed SSO approach reduces the average user response time considerably by solving the multiple credentials problem, besides providing the required security features.

When a CSP in the cloud federation runs out of resources, suitable partner needs to be identified for offloading the customer requests for resources, and this is a challenging task due to the lack of global coordination among them. The cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. In this work, we propose the design and implementation of an efficient partner selection mechanism in the cloud federation, using the Analytic Hierarchy Process (AHP) and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) methods, and also considering the trust values of various CSPs in the federation.

The AHP method is used to calculate the weights of the QoS parameters used in the TOPSIS method which is used to rank the various CSPs in the cloud federation according to the user requirements. Simulation results show the effectiveness of this approach in order to efficiently select the trustworthy partners in large scale federations to ensure the required QoS to the cloud consumers.

In this work, we also propose a trust-based framework for the management of dynamic QoS violations, when one CSP requests resources from another CSP in the federation. We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results shows that by calculating the local trust and the recommended trust values of the CSPs, the dynamic QoS violations can be effectively solved. Thus, the proposed approach improves the performance, responsiveness, efficiency, reputation and the profits of the CSPs in the federation.

In this thesis, we have also presented a trust-based approach for the management of dynamic break-glass access in the cloud federation environments. By using the multi-cloud based health care services, the quality of the health care given to patients can be improved, while reducing the overall health care cost. Thus, there should be an effective way to handle access requests to PHR data during emergency situations, when the patients' information is stored in a cloud federation environment. In this work, we are proposing a trust and risk-based framework for finding the legitimacy of the emergency access requests in the cloud federation environment. The proposed mechanism calculates the risk involved in the access request and takes a suitable access decision by calculating the trust value of the user. We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results shows that the proposed approach is efficient in dealing with the break-glass access requests in the cloud federation environment. Thus, the approach improves the performance, responsiveness and the efficiency of the healthcare services delivered by the CSPs in the federation environment.

Keywords: Cloud Federation; Single Sign-On; Partner Selection; QoS Violation; Break-Glass Access; Local Trust; Recommended Trust.

Contents

Abstract i

List of Figures viii

List of Abbreviations xi

1 Introduction **1**

1.1 Cloud Federation 2

1.2 Identity and Access Management (IAM) in the Cloud Federation 5

1.3 Partner Selection in the Cloud Federation 6

1.4 QoS Management in the Cloud Federation 7

1.5 Break-Glass Access Management in the Cloud Federation 8

1.6 Role of Trust in the Cloud Federation Environments 10

1.7 Scope and Objectives of the Thesis 11

1.7.1 Research Statement 13

1.7.2 Major Research Contributions 13

1.8 Organization of the Thesis 17

1.9 Summary 18

1.10 Topics Covered in Next Chapter 18

2 Literature Review **19**

2.1 Identity and Access Management (IAM) in the Cloud Federation 19

2.2 Resource Management in the Cloud Federation 23

2.3 QoS Management in the Cloud Federation 29

2.4 Break-Glass Access Management in the Cloud Federation 34

2.5 Summary 38

2.6 Topics Covered in Next Chapter 41

3 Single Sign-On (SSO) in Cloud Federation **43**

3.1 SSO Authentication in Cloud Federation 46

3.1.1 Registration of the User with the Identity Provider (IdP) 48

3.1.2 Registration of the User with the Cloud Service Provider (CSP) . 49

3.1.3 Requesting services from the CSP 51

3.2 Workflow Model of the Single Sign-On Approach 52

3.3 Experimental Results 53

3.3.1 Experimental Setup 54

3.3.2 Security of the Data transferred in the Federation 54

3.3.3 Results and Analysis 56

3.4 Pros and Cons of the Approach 58

3.5 Summary 58

3.6 Topics Covered in Next Chapter 59

4 Dynamic Partner Selection in the Cloud Federation Environment **61**

4.1 AHP and the TOPSIS methods 63

4.1.1 Analytic Hierarchy Process (AHP) 63

4.1.2 Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) 66

4.2 Dynamic Partner Selection in the Cloud Federation Environment 68

4.2.1 Overall Flow of the Proposed Approach 69

4.2.2 Rank Calculation in the Federation 70

4.2.3 Resource Allocation 70

4.2.4 Remote Resource Allocation 71

4.3 Experimental Results 73

4.3.1 Experimental Setup 73

4.3.2 Prerequisites for the Implementation 74

4.3.3 Results and Analysis 74

4.3.4 Pros and Cons of the Approach 81

4.4 Summary 82

4.5 Topics Covered in Next Chapter 82

5 Trust-Based Management of Dynamic QoS Violations in the Cloud Federation Environment 83

5.1 Access Control Framework 84

5.1.1 Authentication 85

5.1.2 Authorization 86

5.2 Proposed Approach for the Management of Dynamic QoS Violations in the Cloud Federation 86

5.2.1 Setting up the Cloud Federation Environment 87

5.2.2 User Authentication 88

5.2.3 Processing of the User Request 88

5.2.4 Rank Calculation of the CSP 88

5.2.5 SLA Verification 89

5.2.6 Local Trust Calculation 89

5.2.7 Recommended Trust Calculation 90

5.2.8 Total Trust Calculation 90

5.2.9 Resources Allocation 91

5.3 Local Trust Calculation of the CSP 91

5.3.1 Probability of Success 91

5.3.2 History of Interaction 92

5.3.3 Existing Trust 92

5.3.4 Degree of Association 92

5.3.5 QoS Value 93

5.3.6 Trust Decay Factor of the CSP 94

5.4 Recommended Trust Calculation of the CSP 96

5.4.1 Selection of Trusted CSPs 96

5.4.2 Recommended Trust Calculation 96

5.5 Workflow of the Proposed Approach 97

5.5.1 Local Resource Allocation 97

5.5.2 Remote Resource Allocation 98

5.6 Experimental Results 100

5.6.1 Experimental Setup 100

5.6.2 SSO in Cloud Federation . . . 101
5.6.3 Management of Dynamic QoS Violations . . . 101
5.6.4 Results and Analysis . . . 109
5.6.5 Pros and Cons of the Approach . . . 112
5.7 Summary . . . 113
5.8 Topics Covered in Next Chapter . . . 113
6 Trust and Risk-Based Break-Glass Access Management in the Cloud Federation 115
6.1 PHR Management in the Cloud Federation . . . 117
6.1.1 Risk Management in the Access Control . . . 118
6.2 Access Control Framework . . . 119
6.2.1 Authentication . . . 120
6.2.2 Authorization . . . 120
6.3 Proposed Approach for the Management of Break-Glass Access Requests in the Cloud Federation . . . 120
6.3.1 User Authentication . . . 120
6.3.2 Authorization . . . 121
6.3.3 Calculation of Risk and Trust . . . 122
6.3.4 Logging and Auditing . . . 122
6.3.5 Trust Update of the User . . . 123
6.4 Risk Calculation . . . 123
6.4.1 The impact factor (Factor-1) . . . 123
6.4.2 The sensitivity factor (Factor-2) . . . 124
6.4.3 The probability of malicious access factor (Factor-3) . . . 124
6.5 Local Trust Calculation of the User . . . 125
6.5.1 Probability of Success . . . 125
6.5.2 Degree of Association . . . 125
6.5.3 History of Interaction . . . 126
6.5.4 Existing Trust . . . 126
6.5.5 Access Level . . . 126
6.5.6 Access Right . . . 126

6.5.7 Permitted Factor 127
6.5.8 Genuine Factor 127
6.5.9 Trust Decay Factor of the User 127
6.6 Recommended Trust Calculation 128
6.6.1 Selection of Trusted CSPs 128
6.6.2 Trust Decay Factor of the CSP 130
6.6.3 Recommended Trust Calculation of the User 130
6.7 Workflow of the Proposed Approach 131
6.8 Experimental Results 133
6.8.1 Experimental Setup 133
6.8.2 Break-Glass Mechanism in the Cloud Federation Environment . . 133
6.8.3 Results and Analysis 138
6.9 Pros and Cons of the Approach 140
6.10 Summary 141
6.11 Topics Covered in Next Chapter 142
7 Conclusion & Future Directions **143**
Bibliography 150
Publications 162

List of Figures

1.1 Overview of the Cloud Federation 3

3.1 Overview of the Single Sign-On (SSO) in Cloud Federation 46
3.2 Overall flow of the SSO in the Cloud Federation 47
3.3 User Registration with the IdP in the Cloud Federation 48
3.4 User Registration with the CSP in the Cloud Federation 50
3.5 Processing the Resource Request in the Cloud Federation 51
3.6 Workflow Model of the Single Sign-On Approach 52
3.7 Execution Time with SSO in the Cloud Federation 57

4.1 Overall Flow of the Work 69
4.2 Rank Calculation in the Cloud Federation 70
4.3 Local Resource Allocation in the Cloud Federation 71
4.4 Remote Resource Allocation in the Cloud Federation 72
4.5 QoS offered by the CSPs in the Federation 75
4.6 Weight Table 76
4.7 Trust Table 77
4.8 Rank Table 78
4.9 Resource Allocation in the Federation 80

5.1 Overview of the Access Control Framework 85
5.2 Overall flow of the Management of Dynamic QoS Violations 87
5.3 Rank Calculation in the Cloud Federation 89
5.4 Local Resource Allocation in the Cloud Federation 98
5.5 Remote Resource Allocation in the Cloud Federation 99
5.6 User Request to CSP-1 102

5.7 QoS offered by the CSPs in the federation 102
5.8 Weight Table 103
5.9 Rank Table 103
5.10 Trust Table of CSP-1 104
5.11 Selection of CSPs in the Cloud Federation 105
5.12 Calculation of Local Trust 106
5.13 Trust Table of CSP-4 107
5.14 Trusted CSPs 108
5.15 Recommendation Table 108
5.16 Filtered Recommendation Table 109
5.17 Analysis of the Accepted Requests in the Federation 110
5.18 Analysis of the Service Decision Time 111

6.1 PHR Management in the Cloud Federation 118
6.2 Overview of the Access Control Framework with Break-Glass Mechanism 119
6.3 Proposed Approach for the Break-Glass Mechanism 121
6.4 Workflow of the proposed Approach for Break-Glass Access Management 132
6.5 Risk Calculation 134
6.6 Risk-Trust Table 134
6.7 Calculation of Local Trust of the User 135
6.8 CSP-Trust Table 136
6.9 Trusted CSPs 137
6.10 Recommended Trust Table 137
6.11 Access Decision 138
6.12 Analysis of the Accepted/Rejected Requests 139
6.13 Analysis of the Access Decision Time 141

List of Abbreviations

Acronym	Explanation
AES	Advanced Encryption Standard
AHP	Analytic Hierarchy Process
API	Application Programming Interface
CCFM	Cross-Cloud Federation Manager
CSC	Cloud Service Consumer
CSP	Cloud Service Provider
ED	Emergency Department
FHMQV	Fully Hashed Menezes-Qu-Vanstone
FIM	Federated Identity Management
IaaS	Infrastructure-as-a-Service
IdP	Identity Provider
MCDM	Multi-Criteria Decision Making
NIST	National Institute of Standards and Technology
PaaS	Platform-as-a-Service
PAS	Password Authentication Scheme
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHR	Personal Health Record
PII	Personally Identifiable Information

Acronym	Explanation
QoS	Quality of Service
SaaS	Software-as-a-Service
SAML	Security Assertion Markup Language
SLA	Service Level Agreement
SSO	Single Sign-On
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
VM	Virtual Machine

Chapter 1

Introduction

This chapter gives an introduction to the work carried out in this thesis. An overview of cloud federation, identity and access management in the cloud federation and the issue of partner selection in the cloud federation environment are presented in brief. Also, the importance of QoS and break-glass access management in the cloud federation environment is discussed. The scope, objectives and the major research contributions of the research work are discussed. And, the chapter ends with the overall organization of the thesis and a pointer to the topics discussed in the next chapter.

Cloud Computing is an emerging paradigm offering on-demand services over the internet. In this distributed computing model, a cloud service provider offers various services such as software, platform, processing power, storage etc. that can be accessed by the Cloud Service Consumers (CSCs) across the globe on a pay-as-you-go model (Agostinho et al. 2011). Currently, the cloud computing domain offers different cloud services such as the public, private and the hybrid clouds. The cloud computing environment provides service consumers various benefits such as the elasticity in using the resources, high availability, reduced maintenance costs and also the ability to pay as per their usage. These benefits have increased the acceptance of the cloud services over time. In this model, the cloud users can access the services offered by the providers, and they are free from various issues such as upgrades management, dealing with software licenses, hardware acquisition costs etc.

Cloud Computing has brought a whole new set of services such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) etc. which are made available to the users through the internet. The IaaS delivery model offers computing and storage resources as a service where the users pay only for the resources they

actually use. In IaaS service model, the providers offer their computing resources to the customers in the form of Virtual Machines (VMs). This allows the resources of the CSPs to be shared among the various cloud users. In PaaS service model, an application development platform along with a set of Application Programming Interfaces (APIs) are provided to the developers. In SaaS model, cloud users are allowed to access and use softwares or applications installed on the servers of Cloud Service Providers (CSPs) over the internet. Nowadays, cloud federation is an emerging technology to meet the highly dynamic resource requirements of the cloud consumers.

1.1 Cloud Federation

The widespread acceptance of cloud computing has contributed to the design and development of cloud federation (Buyya et al. 2010; Celesti et al. 2010a; Bernstein et al. 2011). Cloud Federation is an association of different Cloud Service Providers. In the standard cloud computing model, a client gets the required services from a single cloud service provider, and this approach has several challenges associated with it. In reality, the cloud service providers have finite amount of resources with them. Due to some reasons, if a CSP cannot handle the service requests initiated from the cloud customers, it may leave several customers who depend on that service provider, without access to the required resources and services. Also, depending on a single cloud service provider sometimes makes it difficult to ensure the adequate responsiveness and Quality of Service (QoS) to the clients. Also, a CSP cannot afford to lose an important customer because of the lack of available resources at the moment, and thereby not being able to cater to the needs of that customer. To overcome these limitations, CSPs got together as a federation. For many service providers, in order to meet the dynamic and unpredictable user requirements, cooperation with other service providers is an option. This cooperation can be utilized to access resources and services from other partners in the federation to deliver the required QoS to the customers. The CSPs in the federation can share the cloud infrastructure among them in order to have better resource utilization and improved QoS to the cloud consumers. Thus, this collaboration ensures the support in terms of information and resource sharing among the partners in the cloud federation environment, and the improved QoS in terms

of availability, reliability and response time of the services delivered by the various cloud service providers. Hence, the primary reasons for the formation of cloud federation are better resource utilization and the increased revenues for the CSPs, and the availability of reliable cloud services without vendor lock-in for the cloud consumers.

The overview of the cloud federation is shown in the figure 1.1. As described in the fig-

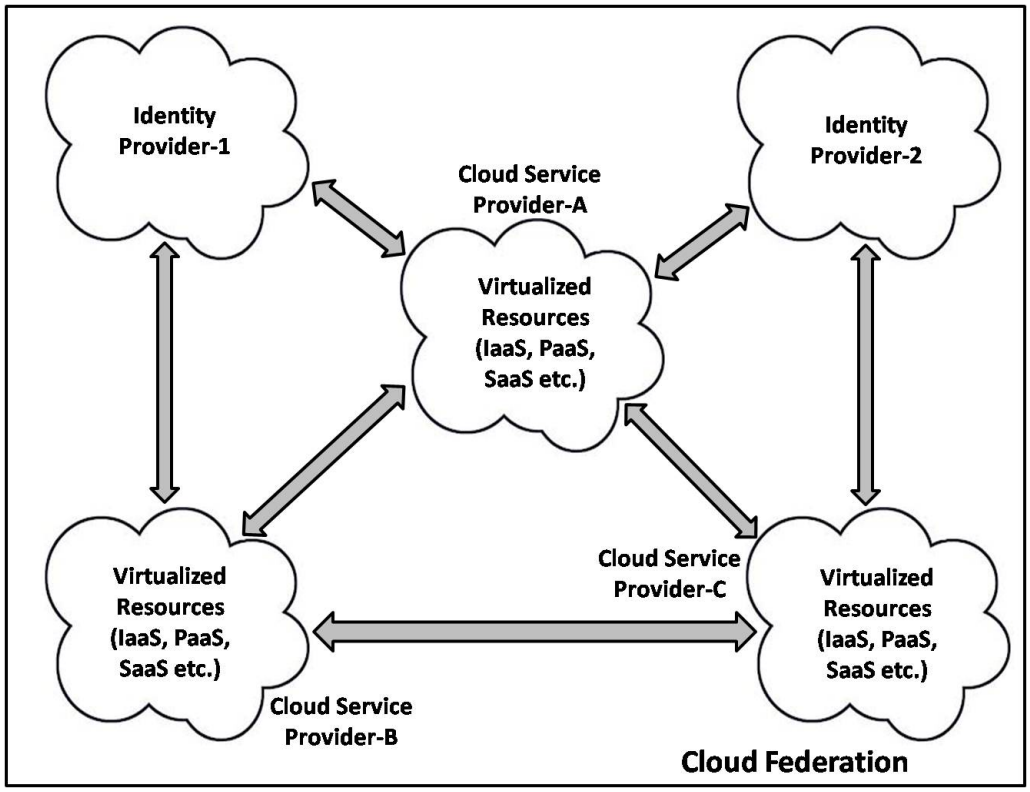

Figure 1.1 Overview of the Cloud Federation

ure, it is an association of cloud service providers and identity providers. The cloud service provider provides various services such as software services, platform services, infrastructure services etc. to the users on a pay-as-you-go model. The cloud service consumers are individual users or enterprises using the services of cloud service providers to meet their resource requirements. As shown in the figure, identity providers are also part of the cloud federation. Every cloud service provider is not an identity provider, and the Identity providers are those cloud service providers who offer identity management services to the cloud consumers. The cloud service providers rely on the identity providers for the identity management activities and since the identity management services are carried out by the identity providers, the cloud service providers can concentrate more on their core services.

In order to use the services of a particular CSP, the cloud service consumer has to get his identity verified by the identity provider trusted by that CSP in the federation. In the cloud federation shown in the figure, we have shown three cloud service providers (cloud service provider-A, cloud service provider-B and cloud service provider-C) and two identity providers (identity provider-1 and identity provider-2). The three cloud service providers have an agreement among them to share the resources. The cloud service provider-A trusts the identity provider-1 and identity provider-2 for the identity management activities of its customers. At the same time, the cloud service provider-B relies on the identity provider-1 and the cloud service provider-C depends on the identity provider-2 for the identity management services.

Thus, cloud federation helps a CSP to have a collection or pool of resources from different CSPs, and this aggregation of resources can take place at different service levels of the cloud computing stack such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) levels to have a pool of software, data, platform, compute, storage and network resources which could be utilized by the cloud users across the globe. Thus, cloud federation enables a CSP to have a single large pool of homogeneous or heterogeneous computing resources (for eg. Virtual Machines (VMs) which are federated from different CSPs). Cloud Federation at the SaaS level enables a CSP to combine different software services offered by other CSPs in the federation into a single service or application and deliver it to its cloud users. There are different types of cloud federation (Bermbach et al. 2013). It could be categorized into horizontal as well as vertical federations. Horizontal Federation refers to the aggregation of resources among the Cloud Service Providers at a particular service delivery model such as IaaS, PaaS or SaaS, whereas the Vertical Federation refers to the association of CSPs where one type of service delivery model of a CSP uses services from other CSPs at another service delivery model. For e.g., the PaaS service of a CSP uses IaaS services from other CSPs in the federation.

Hence, the motivating factors for the adoption of cloud federation paradigm are the enhanced collaboration between the various cloud service providers and the improved Quality of Service delivered to the cloud consumers. The collaboration enables the partners in the cloud federation environment to effectively share the resources and information among

them. The Quality of Service includes factors such as uptime, reliability, response time and the cost of services delivered by the various cloud service providers.

1.2 Identity and Access Management (IAM) in the Cloud Federation

In Cloud Computing or Services Computing, users access various resources or services after verification of their identities by the service provider. Access control deals with verifying the identity and access rights of users towards different resources in the system. Verification of the identity and the access privileges of the service consumers is utmost important in cloud computing or services computing, before they are allowed to access the various resources or services hosted by the service providers. The aim of an access control system is to protect the system resources against unauthorized or illegal access by the users. A secure and effective access control system facilitates resource sharing also. An effective access control system protects the confidentiality, integrity and availability of the resources in the system. An access control mechanism includes the processes of identification, authentication and the authorization. The identification process associates an identity with the users of the system and the authentication process verifies the identity of the users. The authorization process follows the authentication process and it determines the access rights of various users towards different resources in the system. Since different users have varying access rights associated with the resources in the system, effective access control mechanism is required to maintain the security of the resources.

In open service-oriented systems such as cloud computing, in many cases, the service providers and the service consumers are not known to each other beforehand. Since they do not have a pre-established trust value between them, the authentication of the users is to be carried out by the service providers in order to verify their identities and access privileges. Trust establishment between service consumers, service providers and identity providers also assumes very high importance in the current scenario. As the development of the internet is very fast, there are increasing demands for the cooperation of distributed, heterogeneous, and autonomous organizations, emphasizing the need for the development of an efficient access control model. In open distributed systems, secure authentication and authorization processes are required before access privileges are granted to the users.

In order to have a reliable and scalable architecture for the cloud federation which is in its inception stages, many issues still remain to be solved. For achieving secure and effective collaboration between heterogeneous cloud partners, issues related to the Federated Identity Management (FIM) need to be solved as a primary step, in order to maintain the confidentiality, integrity and the availability of the information or resources stored in the cloud federation environment. Even though various researchers have been working in the area of IAM in the cloud environments, scope still exists to develop an authentication mechanism in the cloud federation environment which is resistant against the possible attacks, and also involving multiple identity providers. Hence, in this work, we focus on implementing the Single Sign-On (SSO) authentication of cloud users in the cloud federation environment using the CloudSim toolkit. CloudSim is a generalized simulation framework which helps in modelling and simulating large cloud infrastructure on a single computing node (Calheiros et al. 2011). It has features which can be extended to model the cloud federation used for studying the cloudbursts. Even though various scheduling and provisioning policies can be designed using the CloudSim, as of now, the simulator lacks effective user authentication and authorization methods with it. Hence, we design and implement the authentication and authorization modules required for the proposed SSO approach, and incorporate them into the CloudSim toolkit. Thus, in this work, we discuss the design and implementation of the SSO mechanism in the cloud federation environment providing the services such as confidentiality, integrity and non-repudiation of the identity information.

1.3 Partner Selection in the Cloud Federation

When a cloud user makes a resource request that cannot be immediately met by a provider, the provider can do two things. One is to reject the user's request for service at that moment which may amount to the Service Level Agreement (SLA) violation, if the user has already established one with the CSP. Second one is, if the CSP is part of a cloud federation, it can transfer the resource request to other CSPs in the federation in order to avoid the possible SLA violation. In the latter case, cloud user can access the services offered by other partners in the federation without even being aware of that. In the federation, there can be many CSPs offering different types of services with different QoS features such as

availability, reliability, uptime, response time, cost etc. Also, a CSP in a cloud federation may not have equal trust values towards every other CSP in the federation, since the trust degree varies from CSP to CSP and also from time to time. Hence, the selection of a suitable CSP in the cloud federation, in order to avail the required resources for dealing with the service requests of cloud users is an important activity, if the user has an SLA already established with the CSP or even if the request is from a new user. The cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. Hence, there should be effective mechanisms for the partner selection in the cloud federation environment so that both the CSPs and the consumers are benefited. Hence, in this work, we are proposing a partner selection mechanism in the cloud federation environment that can be used by a CSP in the cloud federation to offload the user requests for resources, by considering the relevant QoS parameters and also the trust values of the CSPs.

1.4 QoS Management in the Cloud Federation

Even though the cloud computing paradigm promises to offer infinite resources, in reality, the resources with each and every cloud service provider are finite. Sometimes, there could be requests from the cloud users for rapid increase in the usage of their computing, memory or network resources due to various reasons such as failure of a server or data centre or to meet the sudden request made by their own clients. In this case, when the cloud service provider runs out of resources, the service provider can get the required services from partners in the cloud federation. Normally, there will be Service Level Agreements (SLAs) between the partner CSPs in a cloud federation regarding the details of the services agreed among them. Due to the dynamic nature of customer requirements, sometimes a CSP in a federation may urgently need some resources (from other CSPs in the federation) to meet the customer requirements, as the requested resources are unavailable with the CSP at that time. Since the CSPs in the cloud federation operate by the Service Level Agreements among them, a CSP can get the services as per the QoS agreement in the SLA. Normally, the process of SLA renegotiation is carried out among the CSPs in order to modify the QoS

parameters of the services agreed among them. Now, if a request comes to a CSP from another CSP in the federation for some resources whose QoS features are not as per their prior agreement, how to dynamically deal with such a request in the federation without the time consuming SLA renegotiation at that time is an issue to be considered.

QoS/SLA violation in the cloud federation occurs when one CSP requires some service from another CSP whose QoS features differ from what has been agreed in the SLA between them. Suppose that there is an SLA agreed between CSP-A and CSP-B in the cloud federation. Also, assume that as per the SLA, CSP-B has agreed to give the service consisting of a maximum of n number of VMs of type '*small*' to CSP-A. Now, imagine that CSP-A makes a service request of m VMs ($m > n$). Also the type of the VMs requested is '*large*'. This is an example of the QoS/SLA violations between the CSPs. Even though this example is simple, we have considered this just to show the working of our approach. Hence, in order to make the best use of the federation, we need a dynamic management of these possible QoS violations among the partners in the federation so that the mutual benefits of the CSPs in the federation, in terms of reliability, reputation and the economic benefits are improved.

Although it has been discussed that efficient resource allocation and utilization requires a high degree of trust values (Vijayakumar and Banu 2008), to the best of our knowledge, the issue of how to solve the dynamic QoS violations in a cloud federation environment has not been addressed in a satisfactory manner. Considering the future scope of cloud federation and also the role of an effective trust management system in the domain, we are proposing our approach in this thesis.

1.5 Break-Glass Access Management in the Cloud Federation

Electronic Medical Records controlled and managed by the patients are known as Personal Health Records (PHRs). By utilizing the cloud based health care applications; the various users such as patients, doctors, nurses, other medical professionals etc. can access the medical data of the patients anytime, anywhere. Since the PHR data are treated as highly sensitive, proper access control mechanisms need to be enforced in dealing with access requests involving PHR data, in order to permit only the authorized users to access the

data. With the emergence of cloud and the cloud federation paradigms, the PHR service providers find it effective to shift their applications and storage to the cloud, in order to reduce the operational cost. By using the multi-cloud based health care services, the quality of the health care given to patients can be improved, while reducing the overall health care cost. In order to enhance the cloud provider's service capabilities, new technologies such as cloud mashups were introduced (Chandrasekhar et al. 2013). Cloud mashups in the health care domain combine different services from multiple cloud providers into a single service or application. This service composition helps the CSPs offer more efficient services and functionalities to clients at lower costs.

During emergency situations, availability of the healthcare data is more important than confidentiality, and hence relevant medical data should be made available to the concerned people irrespective of the employed access control model. The break-glass concept was introduced in (Joint 2004), and it is the way to extend a person's access rights in exceptional situations. Since the modelling of all emergency situations is difficult to achieve, it is possible that the personal information of patients are misused. In the health care domain, there is the possibility of some users trying to access the health data of patients beyond their access rights for making undue advantages.

Considering the importance of the health care management using the services from multiple cloud service providers, there should be an effective mechanism to deal with the break-glass requests from the PHR users in such a domain. Even though, the researchers have been working regarding the security of the PHR data, one of the issues which has not got a perfect solution is how to handle access requests to PHR data during emergency situations, when the patients' information is stored in a multi-cloud or cloud federation environment. Although researchers have been working in the protection of the health data of the patients, to the best of our knowledge, the issue of solving the break-glass access management, taking trust values of the users into consideration in a cloud federation environment has not been addressed in a satisfactory manner.

Considering the future scope of cloud federation in the health care domain and also the role of an effective trust management system in the domain, we are incorporating the trust management of cloud users to the proposed solution of break-glass access management in

the cloud federation environment. The 'emergency' in break-glass access is determined by the PHR user at that time, and the decision to permit the emergency request is taken considering the risk value of the access request, and also the trust value of the PHR user. Only trusted users are allowed to access the PHR data of patients. Every PHR user is permitted to make the access request, and the permission to access the PHR data is subject to the trustworthiness of the user requesting the access. Also, permitted break-glass access is logged and audited, and no two successive break-glass attempts are permitted without auditing. Here, the element of risk is when the trusted user misbehaves which can be detected during the auditing process of the break-glass access.

1.6 Role of Trust in the Cloud Federation Environments

The effective management of trust among the entities is significant for the service computing environment as it is for the activities involving human beings. Human beings trust others depending upon the environment or contexts, and these trust values change from time to time. According to Azzedin and Maheswaran (Azzedin and Maheswaran 2002), trust is defined as: "trust is the firm belief in the competence of an entity to act as expected such that the firm belief is not a fixed value associated with the entity, but rather it is subject to entity's behaviour and applies only within a specific context at a given time". Hence, trust is a dynamic aspect of an entity which varies from very trustworthy to very untrustworthy. The trust value of an entity is derived based on the previous experience with that entity in a specific context. Also, the trust value associated with a particular entity is not the same always as it varies from time to time. One entity can trust another entity in a system also based on the reputation of that particular entity. In this way, the reputation of an entity can be effectively used for building the trust (Vijayakumar et al. 2012; Jøsang et al. 2007). Azzedin and Maheswaran (Azzedin and Maheswaran 2002) define reputation of an entity as: "the expectation of its behaviour based on other entities' observations or information about the entity's past behaviour at a given time". For an entity, there can be either direct or indirect experience with another entity. Direct experience shows that the entities have had some direct interactions between them in the past, and also how one entity learns about the behaviour of the other entity using these interactions. Indirect experience

of an entity is developed based on the recommendations given by other trusted members in the community. Hence, while calculating the trust value of an entity, it considers the reputation of that entity also. Thus, reputation has direct effect on the trust of an entity. That means, a good trust value of an entity results in good reputation of that entity and vice versa (Habib et al. 2010).

In a multi-cloud or cloud federation environment, it requires the association among multiple clouds, and the effective establishment and management of trust among the various CSPs and also between cloud users and CSPs is of paramount importance (Abawajy 2011). In the multi-cloud environment, the partner clouds are independent and loosely coupled which makes the trust management a challenging one. In this environment, the trust management system should help in distinguishing various entities as trustable or not so that effective cooperation of the CSPs are ensured. Even though cloud federation offers various advantages, establishment of trust among the partners in the federation is a challenging issue (Govil et al. 2012). In order to make the best use of cloud federation in terms of resource management, there should be an efficient mechanism for the establishment and evaluation of the dynamic trust between the CSPs, and also between the users and the CSPs in the federation (Sánchez et al. 2012). Researchers have been working on various trust models in the cloud computing domain that evaluate the trust values of various CSPs (Li and Ping 2009; Ahmed and Xiang 2011). Majority of these trust models focused on evaluating and managing the trust between cloud users and the CSPs. Very few of the proposed trust models focuses on effective trust management in the cloud federation domain and hence, the present cloud federation scenario requires effective trust management approaches.

1.7 Scope and Objectives of the Thesis

One of the goals of this research is to develop an efficient Single Sign-On (SSO) mechanism in the cloud federation environment. In our work, we have considered multiple identity providers for dealing with the identity management functions. In our implementation, for securing the data in transit such as during the transfer of the identity tokens of the cloud users between CSPs and also between a CSP and an IdP, we have used the Symmet-

ric Key Encryption technique using Advanced Encryption Standard, AES-256. Also, we have used Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol for key exchange between the entities in the simulation. Hence, efficient and secure communication between the entities is ensured.

The research work also discusses an efficient partner selection approach in the cloud federation environment. In order to identify the suitable partner for offloading the user's request, the CSP ranks the partners by assigning suitable weights to the QoS parameters. The weights are assigned as per the users' requirements using the AHP method, and these weights are later used in the TOPSIS method to ranks the various CSPs in the federation. Also, in our work, the trust values of the various CSPs are considered while selecting a suitable partner in the federation for meeting the resource requirements.

Another goal of the research work is to develop a trust-based framework for dealing with the management of dynamic QoS violations in the cloud federation environment. The proposed method calculates the local and the recommended trust values of the CSPs in the federation for granting the resources in case of QoS violations. Various trust parameters are identified and calculated for determining the local and the recommended trust values of CSPs. Trust Decay Factor is used to deal with the change in trust values of CSPs over time. In this work, trusted CSPs are identified to calculate the recommended trust of a CSP in the federation environment. Depending on the total trust value of a CSP, the resource request from that CSP is either accepted or rejected, in case of QoS violations.

Finally, this research work also discusses a trust and risk-based approach for dealing with the emergency access requests of cloud users in a multi-cloud based health care environment. The proposed approach evaluates the risk involved in the emergency access request by considering various parameters. Then, the local and the recommended trust values of the requesting PHR user is calculated, and the decision is taken as to whether the break-glass access request can be permitted or not. In this work, relevant trust parameters are identified and calculated for determining the trust values of PHR users and the CSPs in the federation. Here also, trusted CSPs in the federation are identified for getting the feedback, and the decay factor is used to manage the change in the trust values over a time period.

The proposed approaches are implemented using the CloudSim simulation toolkit, and the performance is evaluated.

Thus, the major objectives of this research are as follows:

1. To propose and develop an effective authentication mechanism in the cloud federation environment when the cloud users simultaneously access multiple services from different CSPs.

2. To propose and develop an effective partner selection approach in the cloud federation environment for offloading the cloud users' requests for resources.

3. To propose and develop an efficient mechanism for the management of dynamic QoS violations while offloading the cloud users' resource requests in the cloud federation environment.

4. To propose and develop an effective approach for the management of dynamic break-glass access in the cloud federation environment.

1.7.1 Research Statement

"To design and develop an identity and access management mechanism in the cloud federation environment which incorporates effective solutions for the authentication of cloud users, partner selection in the cloud federation, management of dynamic QoS violations, and the management of dynamic break-glass access in the cloud federation environment."

1.7.2 Major Research Contributions

The brief description of the achievements of this research work is listed below.

1. Design and development of a Single Sign-On mechanism in the cloud federation

In this work, we have developed a SSO mechanism for authenticating the users in the cloud federation environment. The proposed approach involves multiple identity providers, and it is resistant against various attacks such as impersonation attacks, MITM attacks, replay attacks etc. The developed mechanism also ensures the secure communication between the entities such as CSPs, IdPs etc. in the cloud federation as we have used the Symmetric Key Encryption technique using Advanced Encryption Standard, AES-256. Also, we have

used Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol for secure key exchange between the entities in the simulation. The analysis of the results shows that the proposed approach reduces the average user response time considerably by improving the performance of the user authentication in the cloud federation environment.

Publications

1. Manoj V. Thomas and K. Chandrasekaran, *"Agent-based approach for identity and access management in the inter-cloud environments"*, International Journal of Trust Management in Computing and Communications, Inderscience Publishers, ISSN: 2048-8378, Volume 2, Issue 2, pp. 125-149, 2014. (DBLP Computer Science Bibliography, Google Scholar, INSPEC(IET), ProQuest indexed) `http://www.inderscienceonline.com/doi/abs/10.1504/IJTMCC.2014.064144`

2. Manoj V. Thomas, Anand Dhole, and K. Chandrasekaran, *"Single Sign-On in Cloud Federation using CloudSim"*, International Journal of Computer Network and Information Security (IJCNIS), ISSN: 2074-9104, Volume 7, Issue 6, pp. 50-58, 2015. (Google Scholar, DOAJ, INSPEC (IET), ProQuest indexed) `http://www.mecs-press.org/ijcnis/ijcnis-v7-n6/IJCNIS-V7-N6-6.pdf`

2. Design and development of a partner selection approach in the cloud federation.

In this research work, we have developed a partner selection approach in the cloud federation environment. The proposed approach helps a CSP to select the most suitable CSP in the federation for offloading the user's request, when it does not have enough resources. The AHP method helps the CSP to assign suitable weights to the relevant QoS parameters based on user requirements. These weights are used in the TOPSIS method to rank the various CSPs in the federation. In this work, the current trust values of the CSPs in the federation are also taken into account before selecting the partner CSP for meeting the resource requirements. The analysis of the results shows the efficiency of the proposed approach.

Publications

1. Manoj V. Thomas and K. Chandrasekaran, *"Agent-Based Cloud Broker Architecture for Distributed Access Control in the Inter-Cloud Environments"*, International Journal of Information Processing (IJIP), ISSN: 0973-8215, Volume 8, Issue 1, pp. 107-123, 2014. (Google Scholar, ICI indexed) https://www.ijipbangalore.org/abstracts_8(1)/p10.pdf

2. Manoj V. Thomas and K. Chandrasekaran, *"Dynamic Partner Selection in Cloud Federation for ensuring the Quality of Service for Cloud consumers*, International Journal of Modeling, Simulation, and Scientific Computing, World Scientific Publishers, ISSN: 1793-9623, Volume 8, Issue 3, 2017. (Scopus and ESCI indexed). http://www.worldscientific.com/doi/pdf/10.1142/S1793962317500362

3. Design and development of a trust-based framework for dealing with the dynamic QoS violations in the cloud federation.

In this work, we have developed a trust-based framework for the management of dynamic QoS violations in the cloud federation environment. The suitable partner is selected, and the SSO authentication is implemented using the approaches discussed above. The proposed method calculates the local and recommended trust values of the CSPs in the federation for granting the resources in case of QoS violations. Five trust parameters are identified for calculating the local trust value of a CSP. Trusted CSPs are identified by calculating their local trust values, and the recommended trust value of a CSP is calculated by aggregating the feedback from these trusted CSPs. Any outliers in the recommended trust values are filtered, and then, the total trust value of a CSP is calculated. Trust Decay Factor is designed and used to deal with the change in trust values of CSPs over time. Depending on the total trust value of a CSP, the resource request from that CSP is either accepted or rejected, in case of QoS violations in the cloud federation environment. The analysis of the results shows that the proposed approach improves the performance of the cloud federation environment.

Publications

1. Manoj V. Thomas and K. Chandrasekaran, *"Workflow Model for Distributed Access Control"*, in Proc. Third IEEE International Conference on Advances in Computing and Communications (ACC-2013), Cochin, India, pp. 363-366, IEEE, 2013. http://ieeexplore.ieee.org/document/6686409/

2. Manoj V. Thomas and K. Chandrasekaran, *"A Trust-Based Approach for Management of Dynamic QoS Violations in Cloud Federation Environments*, Open Journal of Cloud Computing (OJCC), ISSN: 2199-1987, Volume 2, Issue 2, pp. 21-43, 2015. (Google Scholar, OAI, Worldcat indexed) https://www.ronpub.com/OJCC_2015v2i2n03_Thomas.pdf

4. Design and development of a trust and risk-based framework for dealing with the dynamic break-glass access requests in the cloud federation.

The research work also discusses the trust and risk-based framework developed for calculating the legitimacy of the emergency access requests in a multi-cloud based health care environment. Three parameters have been identified to evaluate the risk involved in the access request made in the proposed approach. Also, eight parameters have been selected and evaluated to determine the local trust value of the requesting user. Trust decay factor is designed and used to manage the change in the trust values over time. Another five parameters are selected and evaluated to determine the trusted CSPs. Recommended trust of the requesting user is calculated by taking feedback from the trusted CSPs. Then, the total trust value of the requesting PHR user is calculated, and the decision is taken as to whether the break-glass access request can be permitted or not. After every break-glass access, the trust value of the user is to be updated. The analysis of the results shows the efficiency of the proposed approach.

Publications

1. Manoj V. Thomas and K. Chandrasekaran, *"An Access Control Model for Cloud Computing Environments"*, in Proc. Second IEEE International Conference on Advanced Computing, Networking and Security (ADCONS-2013), Surathkal, India, pp. 226-231, IEEE, 2013. http://ieeexplore.ieee.org/document/6714168/

2. Manoj V. Thomas and K. Chandrasekaran, *"Trust and Risk-Based Approach for the Management of Dynamic Break-Glass Access in the Cloud Federation Environments*, International Journal of Computer Science and Information Security, ISSN: 1947-5500, Volume 14, Issue 7, pp. 141-152, 2016. (Google Scholar, DOAJ, ESCI indexed) https://sites.google.com/site/ijcsis/vol-14-no-7-jul-2016

1.8 Organization of the Thesis

This thesis contains seven chapters: Chapter 1 gives an introduction into cloud federation domain, identity and access management, partner selection, QoS management and the break-glass access management in the cloud federation environment. It also presents a brief discussion on trust management in the cloud computing domain. The chapter also explains the motivation, scope, objectives and the major contributions of the research work.

Chapter 2 discusses the related work by reviewing the relevant works carried out by the researchers in the fields of identity management, partner selection, resource management and access control in the cloud domain in the context of the proposed approaches. The pros and cons of the related works are analyzed to identify the research gap.

Chapter 3 focuses on the proposed SSO mechanism in the cloud federation environment. Overall flow and the details of various steps of the SSO approach are given. Implementation details, experimental results and its analysis are presented. Pros and cons of the proposed approach are also analysed.

Chapter 4 presents the proposed approach for partner selection in the cloud federation environment. Details of the AHP and the TOPSIS methods are given. Weight Table and the Trust Table are discussed. Details of the rank calculation and the overall flow of the proposed partner selection approach are presented. Pros and cons of the proposed approach are also discussed. Analysis of the experimental results is provided.

Proposed approach for the management of dynamic QoS violations in the cloud federation environment is described in the Chapter 5. The proposed access control framework is described. The various functional components in the proposed approach are discussed. Details of the calculation of local and recommended trust values are presented. Workflow of the proposed approach and the pros and cons of the approach are discussed. Also, the analysis of the experimental results is presented.

Chapter 6 focuses on the proposed approach for the management of dynamic break-glass access in the cloud federation environment. The proposed access control framework is discussed. Details of the risk calculation of the access request and the calculation of the local trust and recommended trust values of the PHR users are presented. Trust update of the user is also given. Pros and cons of the proposed approach and the analysis of the experimental results are also presented. Finally, Chapter 7 summarizes the conclusions and provides directions for future research.

1.9 Summary

This chapter presents an introduction into the cloud federation domain, and the various issues such as identity and access management, partner selection, QoS management and the break-glass access management in the cloud federation environment. A brief introduction on trust management in the cloud computing domain is presented. Finally, the chapter explains the motivation, scope, objectives and the major contributions of the research work.

1.10 Topics Covered in Next Chapter

The next chapter provides the literature review in the cloud federation environment covering the identity and access management, resource management, QoS management and the break-glass access management issues in the domain.

Chapter 2

Literature Review

Cloud Federation is an emerging technology gaining acceptance among the cloud service providers and the cloud users alike. The research in this domain is in the nascent stage, and there are many issues to be solved before it can be efficiently deployed and used by the cloud users. In this chapter, relevant research works in the areas of identity and access management, resource management, QoS management and the break-glass access management in the cloud federation are analysed to understand the pros and cons of the approaches proposed by the various researchers. The research gap is identified which plays the pivotal role in the research work presented in this thesis.

2.1 Identity and Access Management (IAM) in the Cloud Federation

Researchers have been working in the area of identity and access management in cloud and federated cloud environments, and the relevant papers are discussed in this section. In (Gunjan et al. 2012), the issue of identity management in the cloud computing scenario is discussed. They explain the privacy issues associated with cloud computing. The paper also gives a review of the existing approaches in identity management in cloud computing. In this work, loss of user control, lack of trust between various entities and the multitenancy issues are considered as the major problems in the cloud computing model. Basics of the authentication process have been explained in (Goriawala 2013) discussing single factor authentication, two-factor authentication and multi-factor authentication based on three factors such as something you know (passwords, PINs etc.), something you have (tokens, smart cards etc.) and something you are (DNA, finger-prints etc.). It discusses the common authentication techniques such as password based authentication, biometric based authentication and also a combination of these techniques. National Institute of

Standards and Technology (NIST) (McCallister et al. 2010) have mentioned the thorough guidelines or recommendations for protecting the confidentiality of Personally Identifiable Information (PII).

Fugkeaw et al. (Fugkeaw et al. 2007) have presented an SSO model based on Multi-Agent System (MAS) and strong authentication based on PKI. Multi-Agent System is a technique in which a group of systems solves the problem by working together which would not have been solved, had the systems worked independently. However, this mechanism could be improved with respect to the access control during the transmission of messages. User authorization is also not considered as part of this work. Juntapremjitt et al. (Juntapremjitt et al. 2008) have extended their previous work in order to utilize the services of distributed Role Based Access Control (dRBAC) model. In (Zwattendorfer and Tauber 2012), the authors used the STORK (Secure Identity Across Borders Linked) framework for the interoperability between electronic IDs (eIDs) of various nations within Europe for SSO.

A security analysis of various Single Sign-On mechanisms in the distributed networks is presented in (Wang et al. 2013). They have identified various flaws such as impersonation of a user, impersonation of a service provider etc. during the authentication process. They have also proposed a secure scheme by using RSA-based signatures. (Mukhopadhyay and Argles 2011) provide a different approach for secure Single Sign-On mechanism in which they have used QR codes for achieving the Single Sign-On. Since many Single Sign-On approaches aren't secure against real-time attacks, this work presents an anti-phishing Single Sign-On solution which is resistant against active attacks and phishing attacks. Major assumption made in this paper is that all the users have smart phones with camera feature for the scanning of QR codes.

(Celesti et al. 2010c) have implemented a three-phase mechanism for cross-cloud Single Sign-On authentication. The three phases discussed in this paper are discovery, matchmaking and authentication. They have focused mainly on authentication in their work and tried to solve the issue by defining their own Security Assertion Markup Language (SAML) profiles. They also extended their work in (Celesti et al. 2011a) by developing a CLoud Enabled Virtual EnviRonment (CLEVER). Also, in this work they have focused only on a

single identity provider rather than having multiple identity providers. Chin-Chen Chang et al. (Chang and Lee 2012) present a secure Single Sign-On mechanism for computers which are distributed over the network where the devices in the network may be mobile. For access control, this paper uses unitary tokens based on secure hash functions, nonce values and public key encryption techniques. Recently, the authors in the network security domain have adopted MAS for network security. (Kumar and Cohen 2000) have proposed the MAS based network security for authentication and authorization. They have implemented an Adaptive Agent Architecture based on MAS. Chang et al. (Chang and Choi 2011) discuss the various issues for authentication and access control such as ensuring security of the user's data. Secure authentication for mobile users is discussed in (Kim and Hong 2012) using Consolidated Authentication Models (CAM).

Nowadays, many of the systems on the internet are still using the password-based authentication or Password Authentication Scheme (PAS). It is still the most accepted mechanism in many cases for distinguishing the legitimate users from the malicious or illegitimate ones. Therefore, lots of research is still going on in this area for providing secure and dynamic authentication schemes. (Chang and Chang 2004) have identified the flaws in the Quadratic Residue approach and Lin et al. (Lin et al. 2003) have tried to enhance the security of the Optimal Strong Authentication Password protocol which was earlier vulnerable to stolen-verifier attack. Our approach adopted in this thesis is able to resist this attack because we have used FHMQV which is resistant against impersonation and Man-in-the-Middle (MITM) attacks, even if session keys are leaked. (Leung et al. 2003) present various security flaws in the authentication scheme using smart cards such as the off-line password guessing attacks, impersonation attacks, the intruder-in-the-middle attacks and the denial-of-service attacks.

In (Stihler et al. 2012), the authors propose the architecture for Federated Identity Management in a scenario similar to the federated cloud environment. The work focuses on information or resource sharing across the cloud service models such as SaaS, PaaS and IaaS. The cloud federation is aimed more at vertical level, in which various SaaS providers and underlying PaaS/IaaS providers can collaborate or federate to form the heterogeneous cloud federation. This work does not focus on the federation at the horizontal level such

as between various Infrastructure-as-a-Service providers.

The work in (Bernstein and Vij 2010b) discusses the inter-cloud security considerations. In (Bernstein and Vij 2010a), the authors propose an authentication mechanism for inter-cloud environments using SAML profile over XMPP. The architecture discussed in this work is based on the internet scale. The work shown in (Yan et al. 2009) discusses a Federated Identity Management approach using Hierarchical Identity-Based Cryptography. This mechanism makes collaboration possible within a hybrid cloud, which is a combination of private and public clouds. The work focuses on the Private Key Generator (PKG) hierarchical model. This model assumes a root PKG for managing the entire hybrid cloud. The root PKG generates private keys for PKGs of the member clouds associated with the hybrid cloud. Before applying this model to the inter-cloud scenario, issues regarding the control of the root PKG should be solved.

A robust remote authentication scheme is presented in (Fan et al. 2005). But, its major drawbacks are higher computation and communication costs and also its inability to prevent the insider attacks. In order to overcome the problems of guessing attacks in the authentication schemes, (Lee et al. 2002) have proposed an authentication scheme which uses one-way hash functions. (Chen et al. 2011) propose an approach for secure and dynamic PAS scheme which solves the issue of authentication failures dynamically by using the theory of quadratic residue. But, this paper does not deal with secure communication between the entities which we have handled in our work.

In (Ren and Wu 2012), the authors proposed a dynamic approach for PAS. It uses One-Time Passwords (OTP) unlike traditional static passwords. This OTP is used for authentication along with the user's private identity information and also considering the current authenticating time. Major benefits of this work are its resistance to various real time attacks in the network such as the MITM, replay attacks etc. At the same time, the method adopted is still vulnerable to types of phishing attacks which aren't possible in our approach. Anastasi et al. (Anastasi et al. 2013) discuss the simulation of cloud federation environments using CloudSim toolkit. But, in this work, they have not discussed user authentication and authorization mechanisms. Calheiros et al. (Calheiros et al. 2011) describes the modelling and simulation of cloud federation environments using CloudSim

toolkit. This approach also lacks effective user authentication and authorization processes in their work. The summary of the literature review discussed is shown in the table 2.1.

Considering the various authentication mechanisms proposed by the researchers, it is seen that secure transfer of identity information of the cloud users among the CSPs and the IdPs is an issue to be considered. In the current cloud federation scenario, multiple identity providers need to be considered for dealing with the identity management service requests of cloud users. Also, an effective authentication mechanism is required to be developed which is secure against various possible attacks. CloudSim (Calheiros et al. 2011) is one of the popular tools used by the researchers for modelling, simulation and experimentation of cloud computing environments. But, as of now, authentication and authorization with Single Sign-On (SSO) has not been effectively addressed in the CloudSim toolkit. As security is absolutely necessary in today's world with every new technology, we propose our SSO authentication mechanism in the cloud federation environments. In our work, we have integrated the SSO authentication and the authorization processes in the cloud federation environment. The proposed approach achieves secure transfer of information among various entities in the cloud federation. Our approach also aids the research community in simulating cloud federation environments with the required security features.

2.2 Resource Management in the Cloud Federation

Many works that are related to the effective resource management in the cloud computing environments have been published, and the relevant papers are discussed in this section. (Goiri et al. 2012; Goiri et al. 2010) use the characteristics of various cloud providers in a federated cloud such as the resource utilization level of a CSP, pricing of the VMs, capital costs, operational costs etc. while handling the resource requests of cloud customers. Depending on the above characteristics, upon receiving a resource request, a CSP takes the suitable decision considering various options such as allocation within itself, outsourcing to other federated clouds, insourcing from other federated clouds, turning on/off various nodes in the data centres etc. But in this work, the major focus is on the cost aspect of the resource management, rather than selecting a provider that best matches the QoS parame-

Table 2.1 Summary of Literature Review-IAM

Sl. No.	Authors	Pros	Cons
1	(Fugkeaw et al. 2007)	SSO model in multi-application environment.	Not in cloud federation and secure transfer of identity information is not considered. User Authorization is not discussed.
2	(Wang et al. 2013)	Security Analysis of SSO mechanisms	Not in cloud federation. Vulnerable to various attacks.
3	(Celesti et al. 2010c; Celesti et al. 2011a)	Considered cloud federation and implemented Cross Cloud Federation Manager.	Single identity provider is considered.
4	(Chang and Lee 2012)	Achieves SSO in a network of computers using security tokens.	Not in cloud federation
5	(Chen et al. 2011)	Handles dynamic authentication failures.	Does not deal with secure key exchange during authentication.
6	(Ren and Wu 2012)	Password Authentication Approach using OTP scheme.	Vulnerable to phishing attacks.
7	(Anastasi et al. 2013)	Modelling cloud federation environments	Does not discuss user authentication and authorization processes.
8	(Calheiros et al. 2011)	Modelling and simulation of federated cloud environments	Lacks effective user authentication and authorization mechanisms.

ters of the service request and the trust level specified.

(Truong-Huu and Tham 2014) discuss the two relevant issues of the current cloud market such as finding the optimal prices for cloud resources to attract a pool of potential users, and deciding whether to cooperate with their competitors to gain higher revenues upon receiving their own users' resource requests. They propose a game-theoretic approach to address these challenges. The proposed cooperation algorithm ensures that the partners involved can improve their final revenues. But in this work, they focus only on the revenue aspect of the cloud business model. Other QoS parameters and the trust aspects are not discussed.

(Agostinho et al. 2011) propose the Bio-inspired approach using genetic algorithm for virtual machine allocation in the federated cloud environment. In this work, the selection of data centres in the cloud federation is done using the shortest path algorithm considering the link cost in the domain, and the selection of servers in a data centre is done using the genetic algorithm. But, in this work, they consider only the link capacity between the partners in the federation, and an analysis of various relevant QoS parameters of the CSPs is not done.

(Wu and Khoury 2012) provide a QoS-based research component composition architecture for research collaboration using distance based evolutionary algorithm. The aim of the algorithm is to compose and optimize research components according to multi-QoS attributes. Trust parameters of the CSPs are not considered in this work. (Mihailescu and Teo 2010) present the idea of dynamic resource pricing in the federated clouds. Here, depending on the current market condition, the dynamic pricing scheme is able to balance the number of resource requests and the amount of available resources. They have implemented a scheme for distributed auctions where multiple auctioneers can allocate different types of resources at the same time. But, in this work, they have considered only the cost factor of the resources.

(Fiorese et al. 2013) discuss a multi-criteria approach to select the appropriate service providers in a large scale multi-provider environment. This approach uses the Analytic Hierarchy Process (AHP) method in a Peer-to-Peer Service Overlay Network composed of several service providers, in order to select the most suitable one to deliver the requested

service. Here, the AHP method is used for quantitative analysis of the service providers by using multiple parameters. But, our work focuses on partner selection in the cloud federation environment for offloading the customer requests. In our approach, we have considered the QoS parameters of the CSPs, trust values of the service providers, weightage of the QoS factors as per the user requests and also the ranking of the CSPs. A game-theory based distributed resource management mechanism for data intensive IaaS cloud providers in a federation environment is proposed by (Hassan and Huh 2011). But, in this approach, the authors have considered only the cost factors. They have not considered whether the CSP in the federation is trustworthy or not.

(Hassan et al. 2009) propose a multi-objective optimization model for partner selection in a market-oriented dynamic collaboration (DC) platform of cloud service providers minimizing the conflicts among the providers during their negotiation. The price and QoS of the services offered by the various CSPs and the success of any previous association among the CSPs in the past are also considered. For this multi-objective optimization, they developed multi-objective genetic algorithm. In this case, they have considered the group of CSPs satisfying the users' request, and the group bid of different CSPs is treated as a single bid. Our work focuses on how each CSP selects the best CSP in the federation in order to meet the user's resource requirements.

(Xin and Datta 2010) discuss how trust can promote collaboration among the service providers considering both direct trust and indirect trust among them. In their partner selection approach, the effective way of considering the QoS parameters of the CSPs is not discussed. Here, before selecting a Guest Service Provider (GSP), the Master Service Provider (MSP) considers the direct trust, indirect trust and also the customers' reviews of the GSP. MSP selects the trustworthy CSPs from the cloud computing market, and they form a collaborative group to fulfil the new service requirements of the customers. A hybrid algorithm for selecting the partner in the federation is proposed by (Song et al. 2009b) in which they use Artificial Neural Network for gathering and analyzing the information about previous tasks between the CSPs, and then multi-objective genetic algorithm is used to solve the cloud partner selection problem. But, in this case also, groups of CSPs are considered for meeting the resource requirements of a single user request. The proposed

approach is not implemented in the federation context. This approach considered only the price and the reliability values of a CSP for a given service. Other QoS parameters and trust values of the CSPs are not considered while forming the collaboration in this work. In our work, we are selecting the most suitable CSP by ranking the CSPs in the federation considering the weights of QoS attributes.

(Song et al. 2009a) have presented an auction-based cloud market model for trading services. They have modified the auction policy of Combinatorial Auctions (CA)-based market model that allows the bidders to make groups, and submit their bids for a set of services to the auctioneer as a single bid. This method tries to minimize the total service prices for the consumers while maintaining their QoS requirements. The problem with this approach is that the group formation among the cloud providers is an NP-hard problem. Also, in this work, the context is that of a cloud market, not that of cloud federation. Our context is a cloud federation where one CSP doesn't want to lose its customers on account of the lack of available resources right at the moment, and it contacts the best CSPs in the federation to offload the customer's request and hence ensures that the required QoS is delivered to the customers.

(Kertész et al. 2012) discuss the integrated federated management and monitoring approach for the autonomous service provisioning in the federated clouds. The users submit the service requests to the brokering component called the Generic Meta Broker Service (GMBS). The service provider's information and health metrics are stored in a Global Service Registry (GSR). The GMBS matches the service request with the information stored in the GSR and selects the suitable cloud broker. In this approach, every CSP has a broker for dealing with the users' requests. Our approach also uses a concept similar to GSR. We also maintain a table in our implementation, containing the information about the QoS offered by various CSPs in the federation which is then used for ranking the CSPs depending on the QoS requirements of the cloud users. In addition to that, our approach of partner selection uses the trust values of the CSPs which is vital in the operation of the cloud federation, and the proposed mechanism is comparatively simpler and more efficient.

(Le et al. 2012) present a Gossip-based Hybrid Multi-attributes Overlay (GHMO) for resource discovery in federated clouds. The Gossip Protocol is used to update the neigh-

bours regarding the status change of a CSP. The Structured Protocol is used for identifying the similarities among the CSPs and thereby forming the groups. When resource request is forwarded to a CSP, based on common parameters, the groups of CSPs are formed for dealing with the request. (Chen et al. 2013) proposed a game-theoretic approach to solve the service selection problem in the cloud environment. The proposed solution is based on game theory and for each provider; they merge the consumer's game with the provider's game. Based on the responses arising from the interaction between the client and other CSPs, a CSP can select the suitable cloud provider as the business partner. But, our approach uses a simpler and effective way of selecting a partner in the cloud federation in order to meet the QoS requirements of cloud users.

(Siebenhaar et al. 2011) discuss an approach for collaborative service provisioning. In this work, the partner selection approach for the collaboration considered only two QoS parameters such as price and the availability of the service offered by the CSPs. The context is a cloud market place, and not the cloud federation. (Abawajy 2009; Abawajy 2011) discusses a trust-based partner selection approach in the inter-cloud environment. In these works, when the users contact a CSP, and if the CSP does not have enough resources to meet the user requirements, the CSP selects other trustworthy CSPs for availing the resources. Here, only the trust and reputation values of a CSP are considered, and other QoS parameters are not considered while selecting a suitable partner in the inter-cloud environment.

(Wenge et al. 2012) discuss a selection process that can be used by cloud providers for collaboration with other CSPs. The authors highlight that a collaborative cloud must be selected only after considering various factors such as the technical aspects, pricing, QoS, mutual benefits and security services offered by the CSPs. This work also discusses the security issues to be considered in multi-cloud environments while selecting the collaborative partners. But, in this work, implementation details are not given. (Kanwal et al. 2014) present the trust evaluation model used by a CSP to evaluate the trustworthiness of other CSPs while participating in the federation. Evaluation of the trust is based on the feedback collected from the registered users and also by analysing the SLAs of the CSPs for Quality of Protection (QoP) attributes. But, QoS parameters are not considered in this work.

(Ghosh et al. 2015) discuss a trust evaluation framework used by the cloud customers to select trustworthy CSPs. The proposed framework acts as an intermediary between the cloud customers and the CSPs, and it helps a customer to select an appropriate service provider to ensure the guaranteed service quality. This work combines the trustworthiness and the competence of the CSPs to estimate the risk of interaction with a particular CSP. Trust value of a CSP is calculated by considering the direct trust and the indirect trust, and the competence of a CSP is assessed based on transparency in the SLA guarantees. But, this work does not consider the QoS features of the CSPs and is not implemented in the context of cloud federation. (Haresh et al. 2011) propose an agent-based resource allocation mechanism in the federated cloud environment. In their work, based on the requirements of the user agent, the broker agent performs the negotiation with the resource provider agent. But, they have considered only the costs of service, and no other QoS parameters are considered. Trust values of the CSPs are also not considered in this work.

The summary of the literature review discussed is given in the tables 2.2 and 2.3. Considering the various approaches proposed by the researchers, it is seen that the cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. In the current cloud federation scenario, the process of partner selection should consider the various QoS parameters and the trust values of other CSPs for the effective selection of partners for offloading the customer requests. Thus, there should be an effective mechanism for the partner selection in the cloud federation so that both the CSPs and the consumers are benefited. Since it is clear that there is no single efficient solution for partner selection in the cloud federation environment meeting the needs of the CSPs and CSCs, we propose our mechanism in this work.

2.3 QoS Management in the Cloud Federation

In this section, we discuss the relevant research activities in the area of resource management in the cloud federation environment. Here, the focus is on how the various approaches

Table 2.2 Summary of Literature Review-Resource Management

Sl. No.	Authors	Pros	Cons
1	(Goiri et al. 2012; Goiri et al. 2010)	CSP takes the federation decision when it receives the resource request.	Only resource utilization level and the cost factors of the CSPs are considered.
2	(Truong-Huu and Tham 2014)	Decision regarding cooperation with other service providers in the cloud market.	Only revenue aspect of the cloud business model is considered.
3	(Agostinho et al. 2011)	VM allocation in the cloud federation environment.	Only the link capacity and the bandwidth cost are considered.
4	(Fiorese et al. 2013)	AHP method is used to select the service provider in a multi-provider environment.	Only negative QoS attributes such as cost, distance, delay etc. are considered.
5	(Hassan and Huh 2011)	Resource management mechanism for IaaS cloud providers in a federation environment.	Only the cost factor considered.
6	(Hassan et al. 2009)	Partner selection in a market-oriented platform of CSPs. Cost and past collaboration history are considered.	Group bid is treated as single bid. Other QoS parameters are not considered.

Table 2.3 Summary of Literature Review-Resource Management Cont'd.

Sl. No.	Authors	Pros	Cons
7	(Xin and Datta 2010)	Direct and indirect trust values are considered for collaboration.	Other QoS parameters are not considered.
8	(Song et al. 2009b)	Hybrid algorithm for selecting the partner in the federation. Cost and past collaboration details are considered.	Groups of CSPs (group bid) are considered for meeting the customers' resource requests.
9	(Song et al. 2009a)	Auction-based cloud market model.	The context is that of a cloud market and group bid is treated as single bid.
10	(Kertész et al. 2012)	Selects the CSP based on availability, computational and data transfer capabilities.	Cost and trust parameters of the CSPs are not considered.
11	(Siebenhaar et al. 2011)	Approach for collaborative service provisioning in a cloud market place.	Considered only price and availability of the services.
12	(Abawajy 2009; Abawajy 2011)	Trust based partner selection approach.	Only trust values of the CSPs are considred. Other QoS parameters are not considered.
13	(Ghosh et al. 2015)	Trust evaluation framework for selecting the trustworthy CSPs.	QoS parameters other than trust are not considered. Not implemented in cloud federation.

proposed by the researchers achieve effective QoS management in the cloud federation. Specifically, it analyzes the works of researchers to examine the degree of management of QoS violations among the CSPs in the federated cloud environment. (Goiri et al. 2010; Goiri et al. 2012) use the characteristics of cloud providers in a federated cloud such as the resource utilization level, pricing of the VMs, capital costs, operational costs etc. while handling the resource requests of the cloud customers. Depending on the above characteristics, upon receiving a resource request, a CSP takes the decision regarding allocation within itself, outsourcing to other federated clouds, insourcing from other federated clouds, turning on/off various nodes in the data centres etc. But in this work, the major focus is on the cost aspect of the resource management, and the management of QoS violations is not discussed.

(Wu and Khoury 2012) provide a QoS-based research component composition architecture for research collaboration using distance based evolutionary algorithm. The aim of the algorithm is to compose and optimize research components according to multi-QoS attributes. A game theory based distributed resource management mechanism for data intensive IaaS cloud providers in a federation environment is proposed by (Hassan and Huh 2011). These works also lack the management of dynamic QoS violations among the CSPs.

(Hassan et al. 2009) propose a multi-objective optimization model for partner selection in a market-oriented dynamic collaboration (DC) platform of cloud service providers. The price and QoS of the service offered by the various CSPs, and also the success of any previous association among them in the past are considered. In this case, they have considered the group of CSPs satisfying the users' request, and the group bid of different CSPs is treated as a single bid. But, our work focuses on how each CSP in the federation handles the dynamic QoS violations so that the best possible service is offered to the CSPs without requiring the SLA renegotiation at that time.

(Kertész et al. 2012) discuss the integrated federated management and monitoring approach for the autonomous service provisioning in the federated clouds. The users submit the service requests to the brokering component called the Generic Meta Broker Service (GMBS). The service provider's information and health metrics are stored in a Global Ser-

vice Registry (GSR). The GMBS matches the service request with the information stored in the GSR and selects the suitable cloud broker. In this approach, management of QoS violations is not included.

(Chen et al. 2013) proposed a game-theoretic approach to solve the service selection problem in the cloud environment. The proposed solution is based on game-theory and for each provider; they merge the consumer's game with the provider's game. Based on the responses arising from the interaction among the client and other CSPs, a CSP can select the suitable cloud provider as the business partner. In (Stihler et al. 2012), the authors propose the architecture for Federated Identity Management in a scenario similar to the federated cloud environment. The work focuses on sharing of information or resources across all the three cloud service models such as SaaS, PaaS and IaaS. In these research works also, management of QoS violations is not discussed.

The works carried out by Celesti et al. in (Celesti et al. 2010a; Celesti et al. 2010b; Celesti et al. 2010c) and Tusa et al. in (Tusa et al. 2011) present a heterogeneous horizontal cloud federation model for CLoud-Enabled Virtual EnviRonment (CLEVER). These works use the concept of a middleware component called the Cross-Cloud Federation Manager (CCFM) that could be integrated into the Cloud Manager component of the cloud service provider. The CCFM consists of three subcomponents, called the Discovery Agent, Match-Making Agent and Authentication Agent, and they are responsible for performing the required functions for the cloud federation. In these works also, the authors do not discuss the management QoS violations in the cloud federation scenario.

Bernstein et al. presented a blueprint for the design of Inter-cloud in (Bernstein et al. 2009; Bernstein et al. 2011) and (Bernstein and Vij 2011). This blueprint is designed considering the interoperability factor among the various cloud service providers and is focused at the internet scale. In this work also, dynamic QoS violations are not discussed.

The Cloud Scheduler project (Armstrong et al. 2010) focuses on developing a model for resource provisioning and sharing among the various participating clouds. In this work, the authors concentrate more on the scheduling of applications among the partners in the federation, and QoS violations among the CSPs is not addressed.

The summary of the literature review discussed is given in the table 2.4. Based on the

literature review, it is seen that the issue of resource management in the cloud federation environments lacks effective solutions to meet the requirements of the present day cloud federation environments, which emphasizes the need for further research in this domain. Considering the various approaches proposed by the researchers, it is seen that, in order to make the best use of the federation, we need a dynamic management of the possible QoS violations among the partners in the federation so that the mutual benefits of the CSPs in terms of reliability, reputation and the economic benefits are improved. Hence, the proposed mechanism effectively incorporates the trust management in the cloud federation environment to deal with the management of dynamic QoS violations, and thereby improving the reliability and efficiency of the CSPs in the cloud federation.

2.4 Break-Glass Access Management in the Cloud Federation

Multi-cloud based health care is an emerging area and in this section, we discuss the relevant research activities in the area of cloud-based PHR management with a focus on effective break-glass access control in the domain.

In (Thummavet and Vasupongayya 2013), an approach for emergency access management of PHR information is discussed. In this scheme, the PHR data of the patients are classified into different categories with varying sensitivity levels. This work uses the predefined emergency staff concept. Based on the policy enforced by the PHR owner, the emergency staff can access the various categories of PHR information. In this work, threshold cryptosystem is used in which a trusted group of people selected by the PHR owner grants the access rights to emergency staff when the PHR owner is unable to grant the permission during emergencies. But, the various issues such as how to decide the members in the emergency team, and also how to make sure that the trusted group of people are online when the break-glass access is requested are not discussed in this paper. Also, is every member in the emergency staff trusted equally by the PHR owners for granting access to the PHR data during emergencies, is another issue to be solved. In our proposed work, every member of the medical staff associated with the multi-cloud based health care solution can request the emergency (break glass) access to the patients' data. Eligibility of the

Table 2.4 Summary of Literature Review-QoS Management in Cloud Federation

Sl. No.	Authors	Pros	Cons
1	(Goiri et al. 2010; Goiri et al. 2012)	Depending on the resource utilization level, pricing of VMs, costs of resources, suitable federation decision is taken by the CSPs.	Only the cost aspect of resource management is considered. Management of QoS violations is not considered.
2	(Wu and Khoury 2012)	QoS based research component composition architecture is discussed.	QoS violations is not discussed.
3	(Hassan and Huh 2011; Chen et al. 2013)	Game-theory based distributed resource management mechanism for IaaS CSPs.	Management of dynamic QoS violations among the CSPs is not addressed.
4	(Hassan et al. 2009)	A method for partner selection in a dynamic collaboration platform of CSPs.	Group bid is treated as single bid considering the price and QoS of individual service. QoS violations is not discussed.
5	(Kertész et al. 2012)	Broker-Based autonomous resource provisioning in the federated cloud environment.	Management of QoS violations is not considered.
6	(Stihler et al. 2012)	Sharing of cloud resources across the three service models such as SaaS, PaaS and IaaS.	QoS violations is not addressed.
7	(Celesti et al. 2010a; Celesti et al. 2010b; Celesti et al. 2010c; Tusa et al. 2011)	Agent-Based cloud federation model for resource management.	Management of dynamic QoS violations among the CSPs in the federation is not discussed.
8	(Bernstein et al. 2009; Bernstein et al. 2011; Bernstein and Vij 2011)	Blue print of intercloud considering the interoperability among the various CSPs.	QoS violations is not addressed.
9	(Armstrong et al. 2010)	Scheduling of applications among the partners in the federation.	Management of QoS violations is not discussed.

access request is dynamically determined by calculating the risk value of the emergency access, and also the trust value of the user requesting the emergency access.

In (Huda et al. 2009), the authors proposed a PHR system that uses health smart cards for authentication of users. This work also uses predefined emergency staff concept. The authors in (Sun et al. 2011) propose the HCPP (Healthcare system for Patient Privacy) in which backup mechanisms are used for dealing with emergency situations. Here, the PHR owner decides what is the emergency information to be shared with. This work also uses predefined emergency staff concept. But, during emergency situations, practically it can be somebody else other than the predefined medical staff, who would be attending to a patient. Hence, if we restrict the break glass access to a few selected people, it may not always be useful.

In a cloud-based PHR management system, since a patient or PHR owner loses physical control of his sensitive medical data stored in the cloud, the PHR data are encrypted by the owners before they are stored in the cloud servers. There are many research works dealing with the PHR management in the cloud environment using various encryption techniques such as Attribute-Based Encryption (ABE) (Brucker et al. 2010; Li et al. 2013; Goyal et al. 2006) and its variants such as CP-ABE (Bethencourt et al. 2007; Li et al. 2013; Foreman 2006), KP-ABE (Goyal et al. 2006) etc. But, the effective management of emergency access requests by the PHR users is to be incorporated with them in order to meet the real-life emergency situations. The authors in (Singh and Vipra Gupta 2013) also discuss the use of emergency department (ED) for dealing with the break-glass access requests in the cloud environment. However, the location of the ED is not specified in this work. In their work, no risk-calculation of the access request is considered. Also, the trust values of the CSPs and the PHR users are not considered by them. In this work, the ED has to verify an emergency situation before the emergency access keys are given to a specific PHR user. But, how to verify the emergency situation is not discussed in this work. In our work, we are proposing an approach for verifying the legitimacy of emergency access requests in the cloud federation environment. We are not using the 'emergency staff' concept; instead, all members of the medical staff associated with the PHR service provider are allowed the break glass access to the patients' data after the associated risk and trust

calculations.

In (Künzi et al. 2009), the authors discuss the emergency access to PHRs in a distributed environment using Digital Rights Management (DRM) techniques. But, how to decide the legitimacy of the break glass access request by a user is not discussed. In (Li et al. 2013), the authors proposed a patient-centric framework for the management of health care data stored in the cloud. They used ABE for encrypting the PHR data of patients and the concept of emergency department (ED) is used to provide the break-glass access. A medical staff who wants to access the patient's health data in an emergency situation, has to contact the ED which will give the required key after proper authentication of the user and also by verifying the emergency situations. After the emergency access, the patient also revokes the emergency key through the ED. In this case, the patient has to recover from the emergency situation to compute a re-key for ensuring the security of the PHR system. The re-key is then submitted to the ED for future break-glass access. This work also uses the concept of emergency staff whose access requests need to be verified by the ED. Again, how to verify the legitimacy of the access requests during emergencies is not included in this work.

In (Benaloh et al. 2009), the authors discuss the PHR management system in which the PHR files are organized in a hierarchical manner to make the key distribution efficient. However, the issue of break glass access management is not included in this work. In (Dong et al. 2008), the authors propose a proxy encryption based access control mechanism in a multi-user environment in which every access operation involves a proxy server. But, this approach does not offer a fine-grained access control and a proper break-glass mechanism. In (Li et al. 2010) also, the work uses ED for handling the break-glass access requests for patients' health data. Here also, how the emergency department verifies the emergency situations is not discussed. In (Tong et al. 2013), they have used the threshold-based decryption scheme for emergency access. Here, the decryption keys are shared among a set of trusted persons and they are combined to get the required decryption key. In this work, the genuineness of the access request should be verified by a trusted person before he gives his share of key and also, he should not be able to later deny his contribution of the key. But, how the genuineness of the access request will be practically verified

is not specified.

In (Sun et al. 2011), the authors discuss a PHR management system wherein the personal health data encrypted by the patients can be decrypted when legitimate access requests from the medical staff are made. Here also, how to verify the legitimacy of the access requests is not discussed. In (Ermakova and Fabian 2013), the authors discuss an architecture for sharing health data in a multi-cloud environment. In this architecture, the health data of the patients are distributed as fragments across different cloud service providers, and it uses the concept of secret sharing to assemble the required health data upon request. However, in this multi-cloud architecture, break glass access is not implemented.

The summary of the literature review discussed is given in the tables 2.5 and 2.6. Based on the various approaches proposed by the researchers, it is seen that, how to identify the legitimate access request is an issue to be solved in the multi-cloud based healthcare domain. Considering the importance of the health care management using services from multiple cloud service providers, there should be an effective mechanism to deal with the emergency access requests or break-glass access requests from the PHR users in such a domain. Thus, the issue of PHR management in the inter-cloud environments lacks effective solutions to meet the requirements of the present day cloud federation environments emphasizing the need for further research in this domain. Hence, in this work, we are proposing a method that determines the legitimacy of the break-glass access requests by calculating the risk involved in the access requests, and also by calculating the trust values of the user requesting the PHR data of the patients during emergencies.

2.5 Summary

In this chapter, the related literature in the cloud federation domain is surveyed. The literature survey reviews the related works on identity management, resource management, QoS and break-glass access management in the cloud federation scenario. The pros and

Table 2.5 Summary of Literature Review-Break-Glass (BG) Access Management

Sl. No.	Authors	Pros	Cons
1	(Thummavet and Vasupongayya 2013)	PHR data are classified into different categories. Threshold cryptosystem is used.	Uses emergency staff concept. How to make sure that trusted people are online is an issue. The degree of trustworthiness of the PHR owner towards members in the emergency staff is another issue.
2	(Huda et al. 2009)	Uses health smart cards for the authentication of users.	Uses predefined emergency staff concept.
3	(Sun et al. 2011)	Backup mechanisms are used for dealing with the emergency situations.	Uses predefined emergency staff concept.
4	(Brucker et al. 2010)	Stores the PHR data of patients encrypted using the ABE.	Effective break-glass mechanism is not incorporated into this work.
5	(Bethencourt et al. 2007; Foreman 2006)	Stores the PHR data of patients encrypted using the CP-ABE.	Dynamic break-glass mechanism is not discussed in these works.
6	(Goyal et al. 2006)	Stores the PHR data of patients encrypted using the KP-ABE.	Break-Glass access management needs to be effectively incorporated into the work.
7	(Singh and Vipra Gupta 2013)	Uses ED for dealing with the break-glass access in the cloud environment.	Location of ED is not specified. Risk calculation is not done. How to verify the emergency situation before the keys are granted is not discussed.
8	(Künzi et al. 2009)	Handles the emergency access to PHRs in a distributed environment using DRM techniques.	How to decide the legitimacy of the break-glass access request by a user is not specified.

Table 2.6 Summary of Literature Review-BG Access Management-Cont'd.

Sl. No.	Authors	Pros	Cons
9	(Li et al. 2013)	Uses ABE and ED.	The patient has to recompute the emergency key. Uses the concept of emergency staff whose access requests need to be verified by the ED. How to verify the legitimacy of the access request is an issue to be addressed.
10	(Benaloh et al. 2009)	PHR files are organized in a hierarchical manner to make the key distribution efficient.	The issue of break-glass access management is not included in this work.
11	(Dong et al. 2008)	Uses proxy encryption based access control mechanism in a multi-user environment.	Does not offer fine-grained access control and a proper break-glass mechanism.
12	(Li et al. 2010)	Uses ED for handling the break-glass access requests for patients' data.	How the ED verifies the emergency situation is not discussed.
13	(Tong et al. 2013)	Threshold-Based decryption scheme for emergency access.	How the trusted persons verify the genuineness of the access request before giving the keys is not discussed.
14	(Ermakova and Fabian 2013)	An architecture for sharing health data in a multi-cloud environment. Uses the concept of secret sharing to assemble the health data.	Management of break-glass access is not implemented.

cons of the relevant approaches are discussed and the research gap is identified. Our research work proposes a Single Sign-On (SSO) authentication mechanism for the identity management in the cloud federation domain in chapter 3. Also, the proposed approaches for the partner selection and the dynamic management of QoS violations are discussed in chapters 4 and 5 respectively. Finally, an approach for determining the legitimacy of the break-glass access request in the cloud federation scenario is discussed in chapter 6. The details of the experiments and the analysis of the results are also given.

2.6 Topics Covered in Next Chapter

The next chapter discusses the Single Sign-On approach proposed in the cloud federation environment. It shows how authentication of cloud users can be effectively carried out in the cloud federation environment. The proposed mechanism is implemented using the CloudSim toolkit, and the analysis of the obtained results also is provided.

Chapter 3

Single Sign-On (SSO) in Cloud Federation

This chapter discusses the proposed approach of Single Sign-On authentication in the cloud federation environment. The various processes involved and the workflow model of the proposed approach are presented. Experimental set up for the research work is discussed and the obtained results are analysed to draw the inferences. The pros and cons of the proposed authentication mechanism are analysed. The summary and the pointer to the topics discussed in the next chapter are also given towards the end of the chapter.

Even though the cloud computing paradigm promises to offer infinite resources, in reality, the resources with each and every cloud service provider are finite. Sometimes, there could be requests from cloud users for rapid increase in the usage of their computing, memory or network resources due to various reasons such as failure of a server or data centre, or to meet the sudden request made by their own clients. In this case, when the cloud service provider runs out of resources, the service provider can get the required services from partners in the cloud federation, if the CSP is a part of a federation. This scenario underlines the urgent requirement for the proper identity management in the cloud federation environment. As there are different services available in the cloud environment, if all the variety of services have their own authentication mechanisms, then the various cloud users will have to log in and verify their credentials each and every time they use a different set of services, even though the services are from the same cloud service provider. This gives rise to the multiple credentials problem. To overcome this problem, we make use of the Single Sign-On (SSO) authentication mechanism in the cloud federation environment.

Federated identity management approach (like Single Sign-On authentication) is required for the current cloud federation scenario. The users in a cloud federation don't need

to use separate credentials for each cloud service provider or service they subscribe to; instead, they can have the identity tokens issued by the Identity Provider (IdP). The cloud users can then submit the security tokens issued by the Identity Provider to the service providers in the cloud federation. This approach is both efficient and secure, and relieves the users of the multiple credentials problem when accessing services from multiple CSPs.

Single Sign-On (SSO) is a mechanism used for authentication in which a service consumer is required to be authenticated only once while accessing various services from multiple service providers (Chang and Lee 2012), or when accessing multiple services from the same service provider. The process of SSO involves the association among the following entities: Cloud Service Consumer (CSC), Relying Party or Cloud Service Provider (CSP) and the Identity Provider (IdP). The CSP and the IdP have mutual trust established between them. That is, IdP offers identity management functions to the CSP. Before accessing the services from the CSP, the cloud service consumer has to get authenticated as a valid user from the IdP. Since the CSP and IdP are part of the association and they have mutual trust with each other, the user is allowed to access the services from the CSP after successful authentication by the IdP. The consumer trusts the CSP. That is why; he uses the services of the CSP. The CSP trusts the consumer upon providing a valid identity token from the IdP trusted by the CSP. The CSP publishes the list of IdPs it trusts. Hence, the customer can use the services of the CSP, provided he trusts the IdPs selected by the CSP. Hence, by accessing the services of a CSP, the customer indirectly trusts the IdPs also. The IdP provides identity management functions to a CSP. IdPs identify the customers through valid credentials and upon producing valid credentials; it trusts the customer, otherwise not. Thus, the identity federation supports Single Sign-On as the users are able to access multiple services from the same or different CSPs using the identity tokens issued by the Identity Provider. Because of this association, the service providers can concentrate more on their core services, since the identity management operations are taken care of by the Identity Providers.

Thus, in the Single Sign-On (SSO) mechanism in cloud federation, a user needs to verify his credentials and get authenticated himself only once during an active session of accessing cloud services. The cloud users are benefited in such a way that they will be able

to seamlessly access all the related services that are offered by a single CSP or multiple CSPs in the federation without the need for providing the identity credentials again and again while accessing different services. It also helps in increasing the productivity of the users as well as the developers by reducing the number of times a user must login, and also by reducing the number of credentials a user has to remember.

The researchers have been working in the area of identity and access management (IAM) in the cloud federation environment. Considering the various authentication mechanisms proposed by the researchers (details are given in the section 2.1), it is seen that secure transfer of identity information of the cloud users among the CSPs and the IdPs is an issue to be considered. In the current cloud federation scenario, multiple identity providers need to be considered for dealing with the identity management service requests of cloud users. Also, an effective authentication mechanism is required to be developed which is secure against various possible attacks. Hence, we discuss the proposed mechanism for SSO authentication in the cloud federation environment in this chapter. We have considered the security aspects of the data transferred between various entities in the cloud federation by using Advanced Encryption Standard, AES-256 algorithm (Pub 2001) and the Fully Hashed Menezes-Qu-Vanstone (FHMQV) protocol (Sarr et al. 2010). AES-256 algorithm is used for encrypting the data transferred between the entities and FHMQV protocol is used for securely sharing the keys between them. In this work, we have also considered multiple identity providers for the identity management services.

The overall view of the SSO in cloud federation is shown in the figure 3.1. As shown in the figure, the cloud users access the services from multiple CSPs after their identities are verified by the identity provider. In this context, every CSP and consumer does not trust every IdP. Every CSP will list out its trusted IdPs based on its past experience or reputation of various IdPs. Every consumer also does not trust every IdP. If due to previous transaction history, a customer has lost its trust with an IdP, and if this IdP is the only trusted IdP of a CSP, that customer cannot use the services offered by that CSP. Hence, mapping from CSP to IdP is many-to-many. Several important factors come into play when forming a cloud federation, such as SLA negotiation, trust establishment between the cloud providers etc. Since our work focuses on the SSO approach in an existing federation, these topics are out

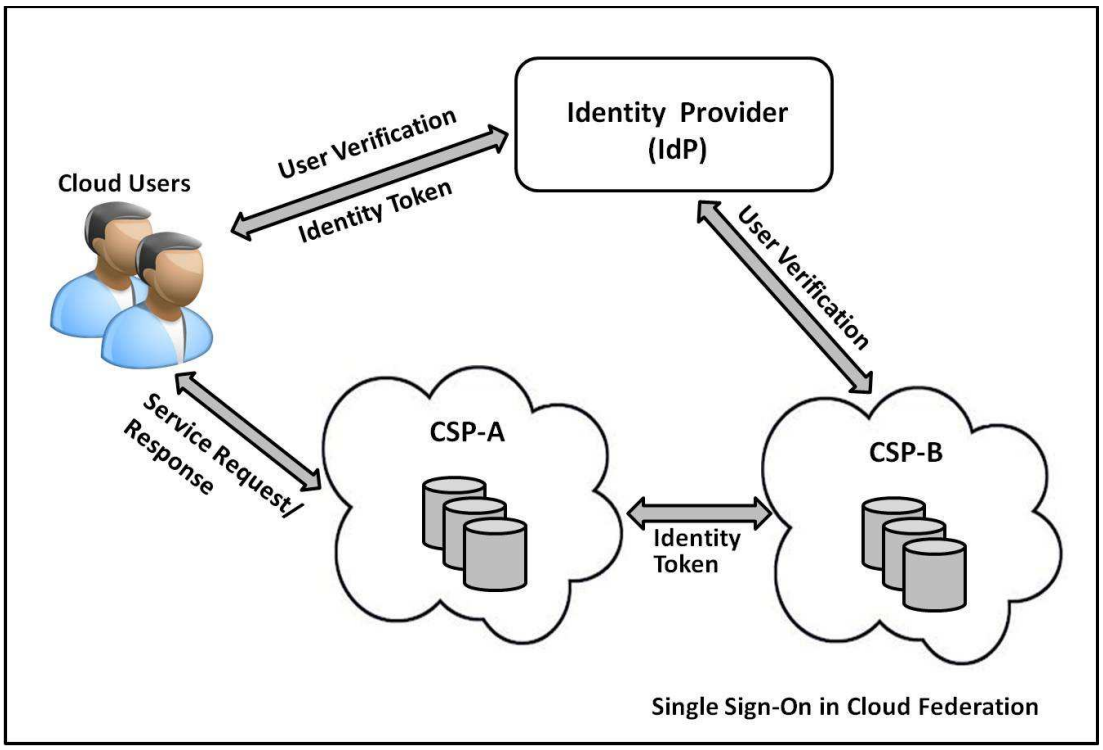

Figure 3.1 Overview of the Single Sign-On (SSO) in Cloud Federation

of the scope of this chapter. In this work, our scope of the federation is limited to the IaaS level and we deal with the resource requests for virtual machines (VMs) in order to explain the SSO approach adopted.

3.1 SSO Authentication in Cloud Federation

Single Sign-On (SSO) approaches can be categorized based on different parameters such as *how* and *where* the mechanism is used, the type of credentials used for authentication etc. (Radha and Reddy 2012). For example, the parameter "where" defines the domains in which the SSO is applied to access the services such as the intranet, extranet or the internet. The parameter "how" defines whether the architecture of the Single Sign-On mechanism is simple or complex. In the complex architecture, it deals with multiple identity providers for supporting the identity of the users. In this case, each user can have multiple sets of credentials. The "credentials" used for authentication can be of different types such as "tokens" or "certificates". In our work, we have implemented the SSO mechanism where multiple identity providers are used for the management and administration of user identities, and also the users can have accounts with different identity providers at the same

time. In our simulation, the credential used for SSO mechanism in the cloud federation scenario is the "token".

The overall flow of the Single Sign-On approach implemented in our work is shown in the figure 3.2. In the figure, in order to access the cloud services in the federation,

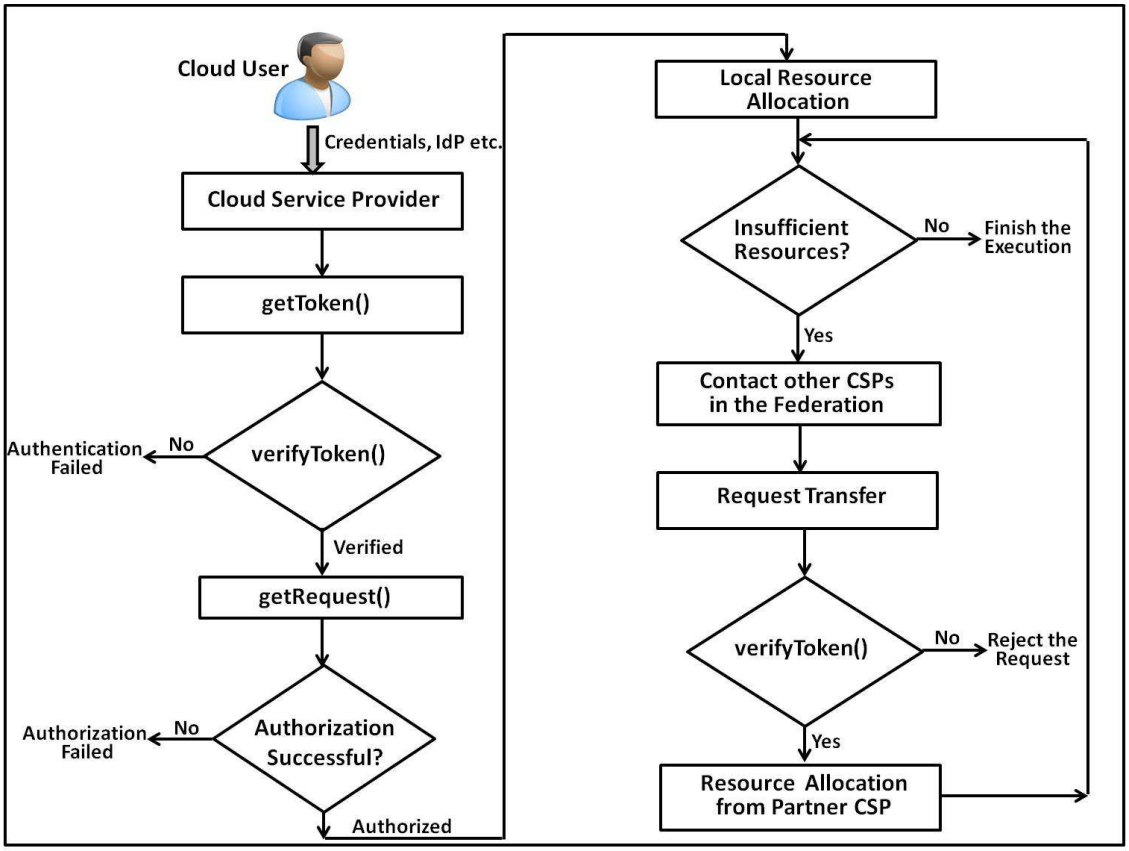

Figure 3.2 Overall flow of the SSO in the Cloud Federation

the cloud user submits the credentials and also the details of the Identity Provider (IdP) supported by the CSP. The CSP verifies the identity token of the user by contacting the IdP mentioned (provided that this IdP is trusted by the CSP considered). Upon successful authentication, the user request is processed to verify the access request of the user. If the verification of the authorization is successful, the local resources are allocated to the user. If the local resources are insufficient to meet the client's request submitted to a CSP, it contacts other CSPs in the federation for the allocation of the required resources. Upon receiving the resource request along with the corresponding identity token, the other CSPs in the federation verify the identity of the user by contacting the corresponding IdP. In this case, the user does not need to enter the identity credentials each time he gets resources

from the cloud partners in the federation. The identity credentials are submitted only once to the first CSP alone while making the access request. Thus, the proposed model has the following main components: 1) Cloud User 2) Cloud Service Provider (CSP) and 3) Identity Provider (IdP). In this proposed approach, whenever a cloud user wants to use the services from any of the cloud service providers in the federation, he requires the following three processes.

3.1.1 Registration of the User with the Identity Provider (IdP)

In order to use the services from a CSP in the federation, the cloud user has to register with any one of the supported identity providers for getting the identity token associated with him. This token is further used for authentication and authorization in the Single Sign-On (SSO) module. The flow diagram of this process is shown in the figure 3.3. As

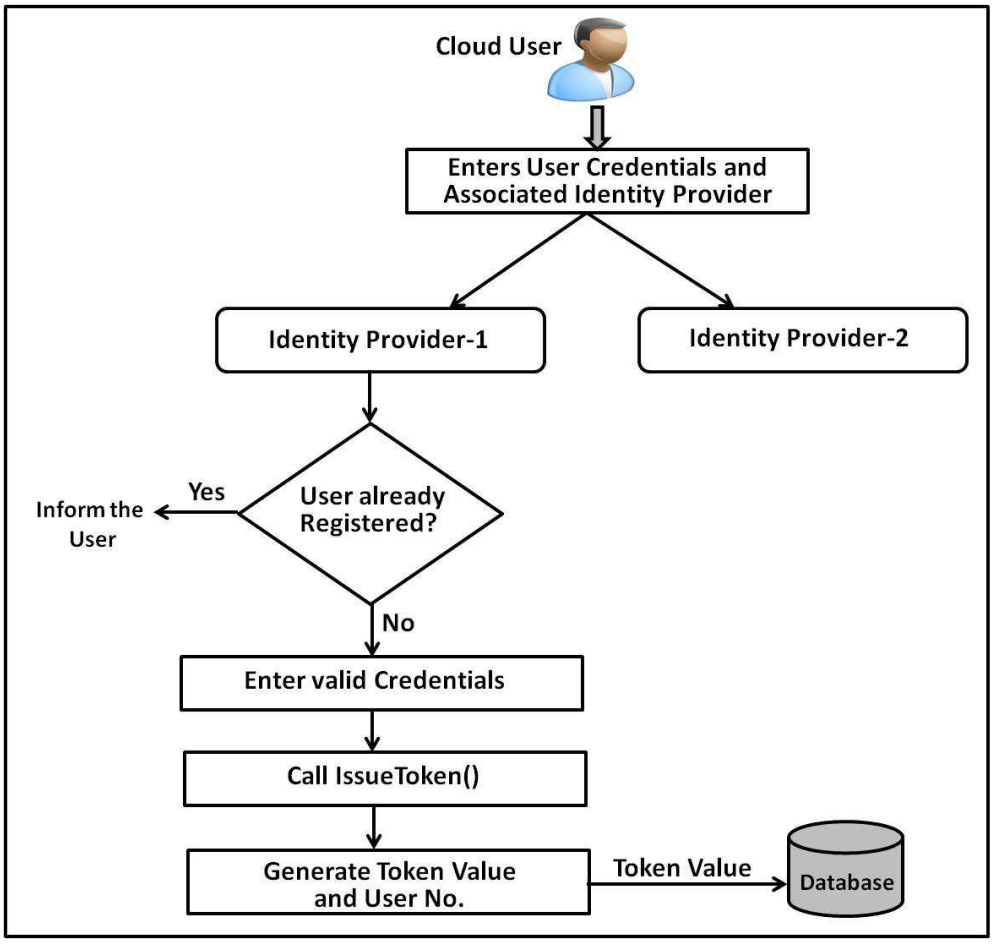

Figure 3.3 User Registration with the IdP in the Cloud Federation

shown in the figure, in this process, the user provides his credentials such as the preferred

user name and password to the selected identity provider. The user's registration request is then redirected to that particular identity provider. On receipt of this request, the identity provider checks for duplicate and invalid entries such as user name, credentials etc. Upon verification, if the identity provider finds that the particular user is new and the credentials are valid, it calls the issueToken() method and generates a token for that user, and also stores the generated token in the database for subsequent verification. In this case, the user registers with any one of the IdPs trusted by the CSP (published by that CSP). Thus, the association between a user and an IdP is maintained by the IdP. The information of IdP is provided by the user, and the CSP contacts the IdP for verifying the user.

3.1.2 Registration of the User with the Cloud Service Provider (CSP)

For availing the services from any CSP in the federation, a user has to register with the corresponding cloud service provider. This step is equivalent to negotiating an SLA with the CSP regarding the various QoS attributes of the services requested. In our simulated experiment, the users negotiate the number and type of virtual machines, and also the access rights associated with the VMs. The flow diagram of this process is shown in the figure 3.4. It is assumed that user is aware of the primary CSP where he has an SLA agreed with, and also the corresponding IdP(s). The federation architecture is transparent to the user as he is unaware of other CSPs in the federation. The user does not sign an SLA with any other CSPs in the federation, other than the primary CSP. The federation is formed among the CSPs who trust each other, and the CSPs in the federation have SLA among them for sharing the resources. If a CSP in the federation is found to behave maliciously, the CSP can be terminated with federation services. As shown in the figure, whenever the user requests for some services from a cloud service provider, it must be verified that the user is registered with any one of the identity providers trusted by that CSP such that the particular identity provider provides the identity management services to the cloud service provider. On successful verification, the user enters the resource request details such as the number of virtual machines, type of virtual machines, access rights associated etc. Now, the user access request is verified with the UserRights table of the concerned CSP to ensure the validity of the entry, and then the details are entered into the database. Thus, in order to register with a CSP, firstly, the user should have registered with an IdP trusted by that CSP.

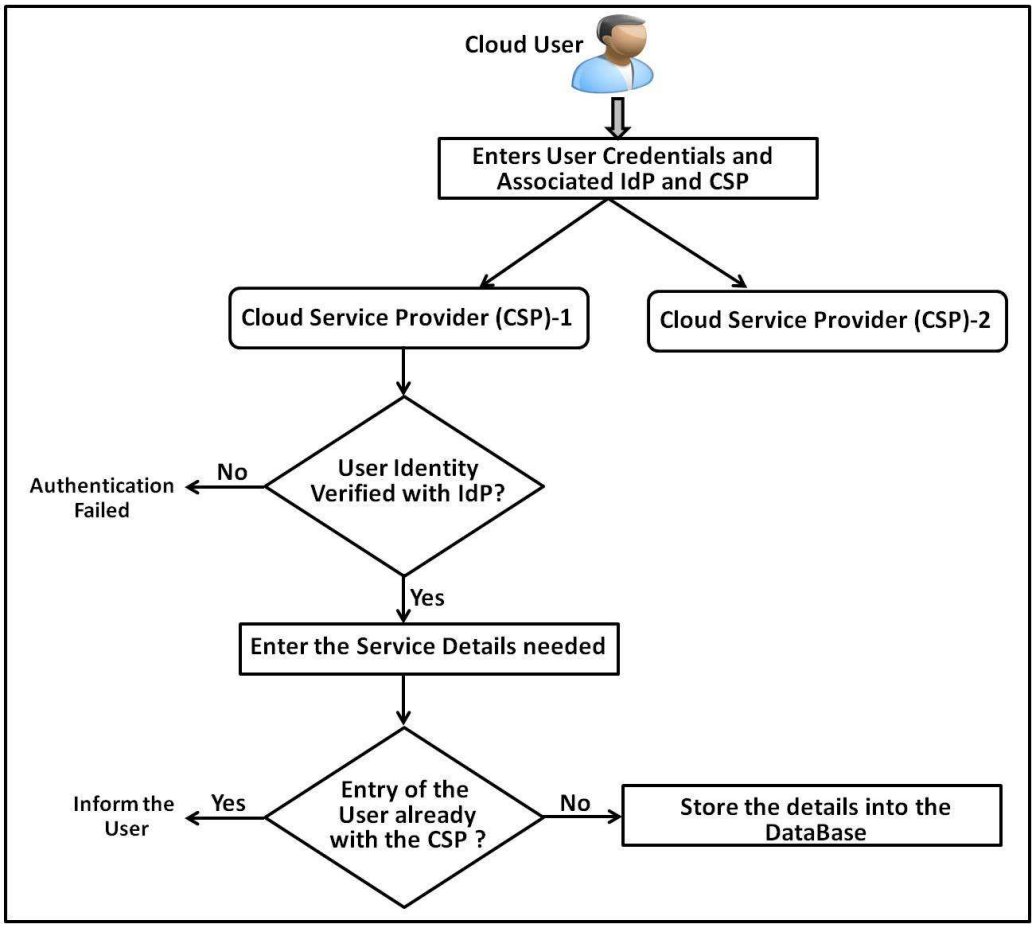

Figure 3.4 User Registration with the CSP in the Cloud Federation

The table UserRights, stored by a CSP has the fields such as UserRights No, User_ID, Number of VMs, Type of VMs, AccessRights. Here, the field User_ID is obtained from the token returned by the IdP.

Suppose that User-1 registers with name "A" on CSP-1. Another user, User-2 can register with the same name "A" on another CSP, CSP-2, provided he had registered with the name "A" on a different IdP trusted by the CSP-2. This is not a security problem, because IdP is different and hence the User_ID and token values are different. In this case, the identity of the user can be verified by contacting the corresponding IdP. The CSP ascertains if any user is using a registered name by contacting the IdP with the credentials supplied by the user. Also, it is possible for User-1 to register with different names to different CSPs, as it is possible that same user can have different names with different IdPs just like users have multiple email ids and phone numbers. In this case, each CSP verifies the user by contacting the IdP specified.

3.1.3 Requesting services from the CSP

After the above mentioned two steps, the user can submit his request to the cloud service provider for availing the services. The flow diagram of the processing of the user's request is shown in the figure 3.5. As shown in the figure, while making the access request for

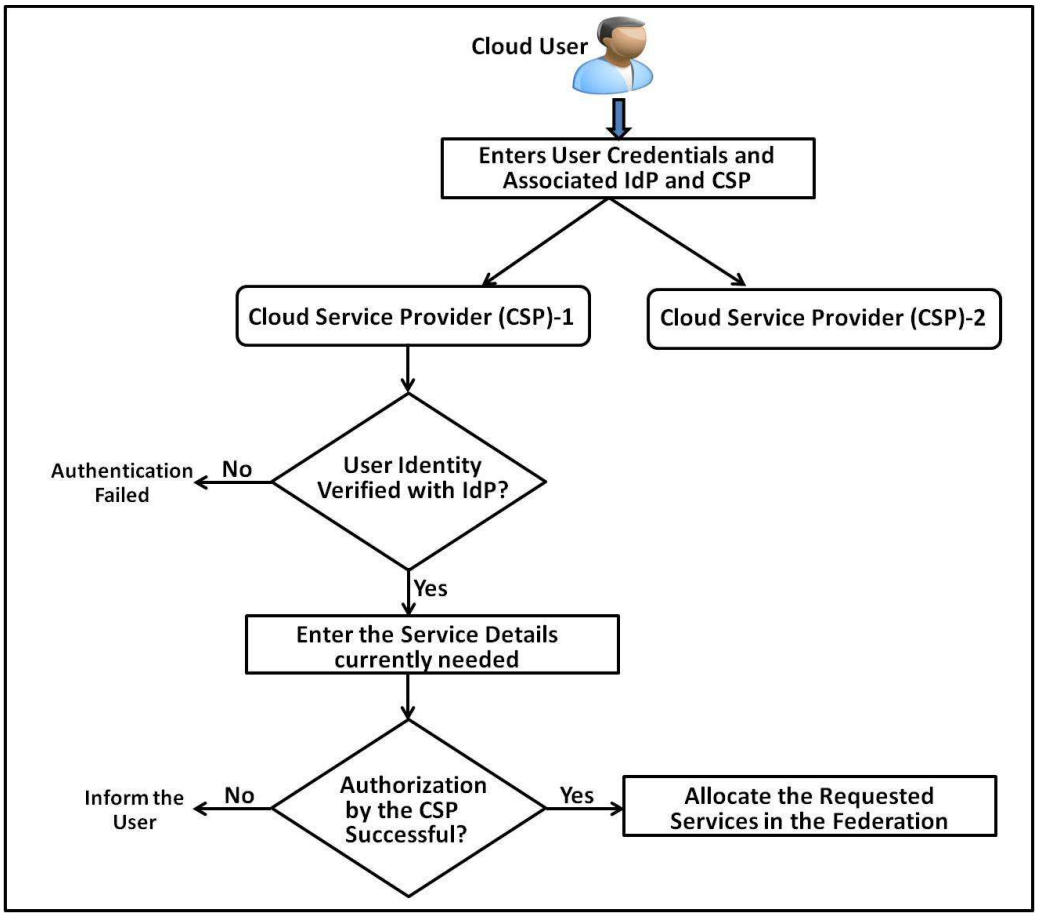

Figure 3.5 Processing the Resource Request in the Cloud Federation

the allocation of virtual machines, the user submits the details of the identity credentials and the specific identity provider to the selected cloud service provider. The identity of the user is verified by the CSP by contacting the corresponding identity provider. This step is necessary to provide the authentication of the user. Upon successful authentication, the user's request for virtual machines with the specified access rights are validated using the CSP's database, and depending upon the result of the validation, the user's access request is either accepted or rejected. After the successful authentication and authorization, execution of the user's request starts. In case the CSP does not have enough resources

to meet the resource request of the cloud user, the CSP requests resources from other CSPs in the federation and satisfies the user's request as shown in the figure 3.2. In our implementation, in this case, once the user is authenticated at a particular CSP, the access token from the CSP is transferred to other CSPs in the federation for accessing the cloud resources in the federation. That means, the same user is not required to submit the identity credentials again and again for accessing services from various CSPs in the federation.

3.2 Workflow Model of the Single Sign-On Approach

The sequence of steps involved in the workflow of the implementation of SSO in cloud federation is shown in the figure 3.6. In this figure, for simplicity, we have shown only two

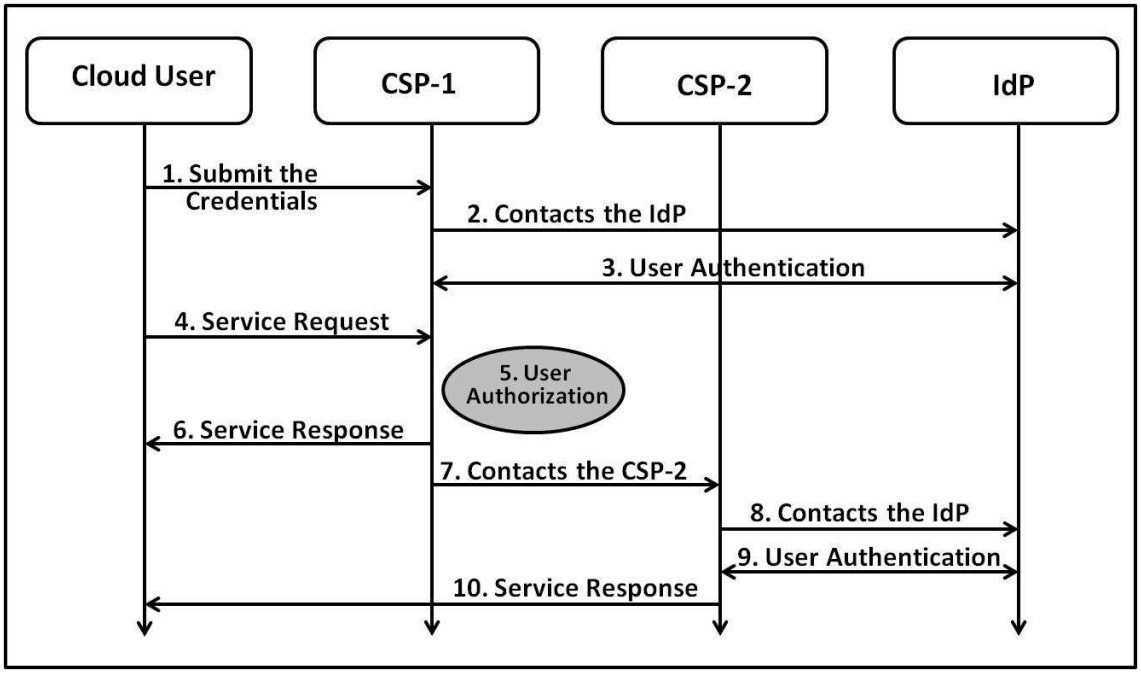

Figure 3.6 Workflow Model of the Single Sign-On Approach

CSPs in the cloud federation. As shown in the figure, the various steps involved are:

1. The Cloud User wants to access the service hosted by the CSP-1, and submits the identity credentials to the CSP-1.

2. The CSP-1 contacts the associated Identity Provider (IdP) for the authentication of the user.

3. The CSP-1 gets the result of user verification from the IdP.

4. The user submits the service request for accessing the resources from the CSP-1.
5. User authorization is performed by the CSP-1 to decide whether to accept or reject the request.
6. The local resources (if available at the CSP-1) are allocated to the user.
7. The CSP-1 contacts the other CSP(s) in the federation (CSP-2), if the local resources are not sufficient to satisfy the user's request.
8. The CSP-2 contacts the IdP for the authentication of the user.
9. The CSP-2 gets the result of user verification from the IdP.
10. The CSP-2 allocates the resources (if available) to the user after authorization.

In this work, a secure SSO is ensured by CSP-1 to another CSP in the federation. An identity token is passed from CSP-1 to CSP-2 which can be used by CSP-2 to verify the identity of the specified user by contacting the IdP. The token contains various fields such as User_ID, user name, password, token value, IdP, Datacentre (original CSP) etc. CSP-2 authenticates the cloud user by contacting the IdP. CSP-2 and IdP have mutual trust between them. In our work, it is assumed that CSP-2 trusts the IdP. If the IdP is not trusted by CSP-2, and the user has identity with that IdP, then CSP-2 will not give resources to the user. If the IdP behaves maliciously, then the purpose of federation is lost. In that case, the CSP will lose the trust with the IdP, and stop federating with that IdP in the future. In this work, we have used multiple IdPs for the identity management functions of the users in the federation. Also, the proposed method ensures the security of the data transferred between various entities such as cloud users, cloud service providers and identity providers by using AES-256 and FHMQV protocols, and making the method resistant against various attacks such as man-in-the-middle attack, replay attack, impersonation attack etc.

3.3 Experimental Results

The primary objective of this experiment is to design and implement the Single Sign-On authentication and the authorization modules, and also to verify that they are working correctly in the cloud federation environment simulated using the CloudSim toolkit. Our test

scenario consists of a number of cloud service providers and multiple identity providers. The cloud users can have multiple accounts with different identity providers, and the cloud service providers may use the services of one or more identity providers.

3.3.1 Experimental Setup

We have carried out the simulation experiments on a system with Intel (R) Core (TM) i7-3770, CPU 3.40 GHz, 8.00 GB RAM and 32-bit Operating System (Ubuntu 14.04). Softwares used for the implementation include CloudSim-3.0.3, Eclipse IDE version 3.8, MySQL Workbench Community (GPL) for Linux/Unix version 6.0.8 and Java version 1.7.0_55.

For implementing the SSO module and the authorization module, we have used the Java programming language. We have used MySQL database to store the data related to the various users of the CSPs. Each cloud service provider (CSP) has a table for storing the user's access rights which shows the access rights associated with a particular registered user, and this information is used during the authorization phase of dealing with the access request. Each Identity Provider (IdP) stores the user related data and the user credentials in the corresponding Identity Provider table. We also have a table in the database showing the mapping of which CSP uses the services of which identity provider.

3.3.2 Security of the Data transferred in the Federation

In our implementation, for securing the data in transit such as during the transfer of the identity tokens of the cloud users between CSPs in the federation and also between a CSP and an IdP, we have used the Symmetric Key Encryption technique using Advanced Encryption Standard, AES-256. Also, we have used Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol for key exchange between the entities in the simulation. AES is a protocol mentioned in the set of standard protocols for security by the National Institute of Standards and Technology (NIST) (Pub 2001) and the FHMQV protocol has its root in Diffie-Hellman (DH) protocol. The FHMQV protocol (Sarr et al. 2010) defines the Full Exponential Challenge Response (FXCR) and Full Dual exponential Challenge Response (FDCR) schemes which preserve the performance of the (H)MQV protocol, in addition to providing resistance against various attacks such as the impersonation attack,

man-in-the-middle attack, replay attack etc.

FHMQV Key Exchange Protocol

Alice and Bob want to share the secret key which is used for encryption and decryption of the messages exchanged between them. Let P and g be prime numbers, and H is a hash function. The major steps of this key sharing protocol are discussed below.

(I) (i) Alice selects a prime number, x and calculates $X = g^x \bmod P$ and sends that to Bob.

(ii) Bob selects a prime number, y and calculates $Y = g^y \bmod P$ and sends that to Alice.

(II) On receipt of Y, Alice does the following:

(i) Selects a challenge, a (a prime number), and calculates $A = g^a \bmod P$. Now this is sent to Bob.

(ii) Calculates $e = H(Y, X)$ and sends this to Bob.

(III) On receipt of X, Bob does the following:

(i) Selects a challenge, b (a prime number), and calculates $B = g^b \bmod P$. Now this is sent to Alice.

(ii) Calculates $d = H(X, Y)$ and sends this to Alice.

(IV) After step II, on receipt of the values, Bob does the following:

(i) Calculates $S_B = y + eb \bmod P$

(ii) Calculates $\sigma_B = (XA^d)^{S_B}$

$= (XA^d)^{(y+eb)} \bmod P$

$= (g^x (g^a)^d)^{(y+eb)} \bmod P$

$= (g^{(x+ad)})^{(y+eb)} \bmod P$

$= g^{(x+ad)(y+eb)} \bmod P$

(iii) Calculates $K = H(\sigma_B, X, Y)$

(V) After step III, on receipt of the values, Alice does the following:

(i) Calculates $S_A = x + da \bmod P$

(ii) Calculates $\sigma_A = (YB^e)^{S_A}$

$= (YB^e)^{(x+da)} \bmod P$

$= (g^y (g^b)^e)^{(x+da)} \bmod P$

$= (g^{(y+be)})^{(x+da)} \bmod P$

$= g^{(x+ad)(y+eb)} \bmod P$

(iii) Calculates $K = H(\sigma_A, X, Y)$

(VI) The shared key is K.

In our work, if the SLA of user-1 can only be partially met by CSP-2, the access request of user-1 is not forwarded to CSP-2. CSP-1 has the details of QoS offered by other CSPs in the federation, and it selects the best CSP in the federation that meets the SLA fully. There is no partial meeting of the SLA by CSP-2 which amounts to SLA violation for the user. That is not acceptable for user-1, and hence, CSP-1 selects the best CSP in the federation that meets the SLA of user-1 and CSP-1 completely. In any case, malicious CSP is not allowed to be part of the federation in future.

3.3.3 Results and Analysis

In order to test the proposed approach, we have implemented the cloud federation scenario with 25 CSPs with each CSP having two heterogeneous hosts associated with it. The user makes the resource request to any CSP in the federation, and the resource request is made in such a way that the resource requirements of the user cannot be met by a single CSP alone. In the simulated federation environment, if a CSP cannot handle the access request with its own available resources, the access request is transferred to other CSPs in the federation as we have already discussed.

The figure 3.7 shows the number of SSO involved and the corresponding execution time associated with the user requests in the simulation. In the figure, we have shown the maximum number of SSO in the federation associated with a single user request as 20. From the figure, it is seen that the average execution time taken for 20 SSO operations

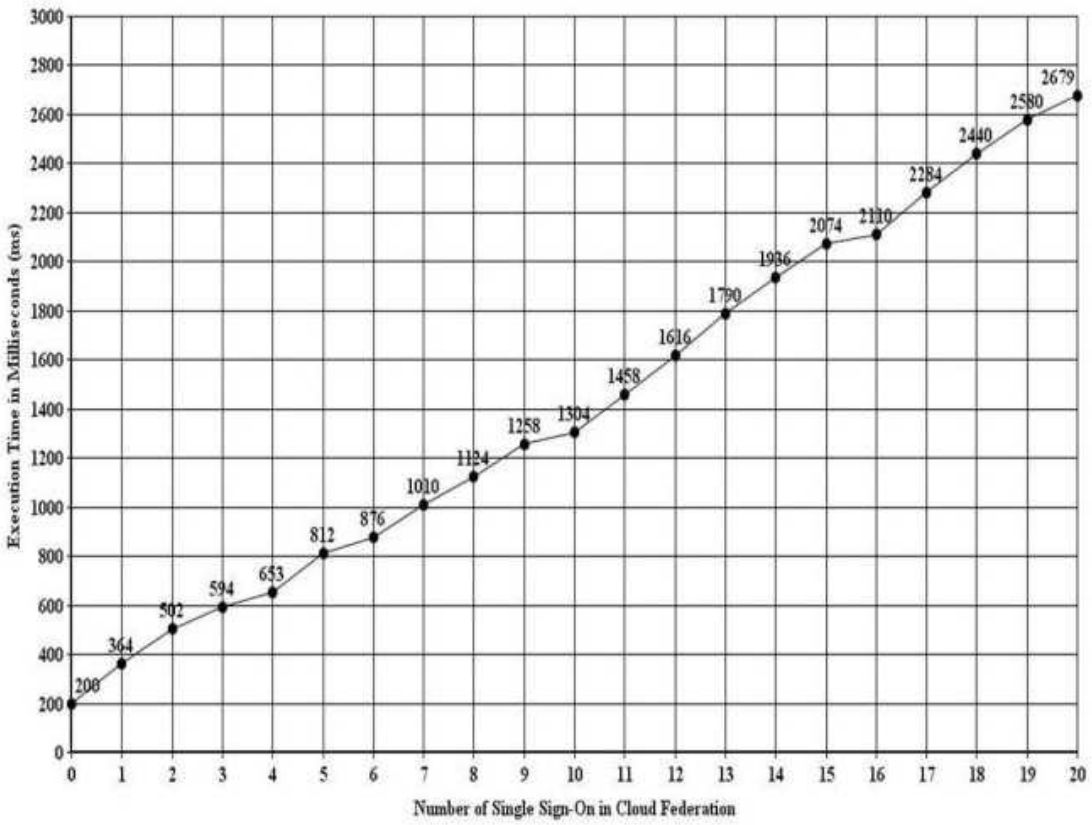

Figure 3.7 Execution Time with SSO in the Cloud Federation

associated with the user request, involving 20 CSPs in the cloud federation is 2679 milliseconds as observed in our simulation. Without SSO, as the number of times a user's request is transferred from one service provider to another increases, the number of logins he needs to perform also increases, thus increasing the response time of the services delivered to the user. If we assume that, on an average, two seconds are required by a cloud user to enter the user name and the password at a CSP, then as the number of CSPs in the federation increases, the number of logins needed also increases and hence, the total time needed for user verification will be much higher than the time taken for the corresponding authentication using the SSO approach. Hence, this shows that the proposed SSO approach reduces the average user response time considerably, besides providing the required security features. Also, by using the Single Sign-On authentication mechanism, it reduces the load of the cloud users and developers in dealing with multiple credentials while accessing services from various CSPs in the federation.

The proposed approach of SSO was implemented in a cloud federation environment

involving multiple identity providers. The security aspects of the data transferred between the various entities are taken care of. There are few similar works available showing the SSO in cloud federation involving multiple identity providers. Also, important to note that the SSO authentication was implemented in the cloud federation environment simulated using the CloudSim toolkit. The developed modules of authentication and authorization can be incorporated into the CloudSim toolkit for further research using the tool.

3.4 Pros and Cons of the Approach

The major advantage of the proposed Single Sign-On authentication mechanism is that it solves the multiple credentials problem effectively as a user does not need to log in each and every time he uses a different set of services from the cloud federation environment. The proposed approach ensures the required level of security in the cloud federation, and the method is resistant against various attacks such as man-in-the-middle attack, impersonation attack, replay attack etc. The service providers can concentrate more on their core services as the identity management functions are take care of by the identity provider. The approach also requires that the cloud user should have an identity registered with the identity provider, and the identity provider should be trusted by the cloud service provider as the service provider depends on the identity provider for the verification of the identity of the cloud user.

3.5 Summary

Authentication of cloud users is an important activity to preserve the confidentiality, integrity and availability of the information stored in the cloud. An efficient authentication mechanism is required to be implemented in the cloud federation environment for improving the quality of service delivered to the cloud users. In this chapter, we have implemented the Single Sign-On authentication mechanism in the cloud federation environment using the CloudSim toolkit, considering multiple cloud service providers and identity providers. We have also considered the security aspects of the data transferred between the various entities during the SSO mechanism in the cloud federation. We have used symmetric key encryption technique, AES-256 for encrypting the data before they are transferred among various entities in the federation. Also, we have used FHMQV key sharing protocol for

securely exchanging the encryption keys between any two entities. The simulation results show that the SSO approach is highly beneficial while accessing multiple services from CSPs in the cloud federation, as it reduces the execution time of the user request for resources in the cloud federation.

3.6 Topics Covered in Next Chapter

The next chapter discusses the proposed partner selection approach in the cloud federation environment. It shows how the partner CSP can be selected for resource allocation when a CSP does not have enough resources to meet the user's requirements. The proposed approach is implemented using the CloudSim toolkit, and the analysis of the results is also given.

Chapter 4

Dynamic Partner Selection in the Cloud Federation Environment

This chapter discusses the proposed approach for the dynamic partner selection in the cloud federation environment. Overall flow of the approach is presented and the details of the Analytic Hierarchy Process (AHP) and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) methods are discussed. Experimental set up of the research work is presented and the analysis of the obtained results is carried out. Pros and cons of the proposed approach is discussed. And, the chapter ends with its summary and the pointer to the topics discussed in the next chapter.

Generally, when a cloud service provider runs short of its resources at a time, it might cause the following effects (Celesti et al. 2011b).

(i) The CSP may not be able to meet the Service Level Agreements (SLAs) already established with various cloud service users regarding the quantity and quality of the services.

(ii) The CSP may not be able to accept further service requests from the cloud users across the globe.

(iii) The CSP may not be able to accept the SLA modification requests from the existing users in order to enhance the amount and the associated characteristics of the services.

Thus, in cloud computing, when a CSP does not have enough resources, the CSP may not be able to satisfy the cloud users' requests for resources, at least for a period of time,

until some of its resources are released. This can have a significant adverse economic impact especially for small and medium clouds. Such a problem can be solved by a CSP being part of a Cloud Federation or the Inter-Cloud, as each cloud service provider in the federation is able to transparently enlarge and effectively use its own resource capabilities. In the federation, there can be many CSPs offering different types of services with different QoS features such as availability, reliability, uptime, response time, cost etc. Also, a CSP in a cloud federation may not have equal trust values towards every other CSP in the federation, since the trust degree varies from CSP to CSP and also from time to time. Hence, the selection of a suitable CSP in the cloud federation, in order to avail the required resources for dealing with the service requests of cloud users is an important activity. When a CSP runs out of resources in the cloud federation, in order to offload the customer requests for resources to other CSP(s), identifying a suitable partner is a challenging task due to the lack of global coordination among them.

The researchers have been working in the area of resource management in the cloud federation environment. Considering the various approaches proposed by the researchers (details are given in the section 2.2), it is seen that the cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. In the current cloud federation scenario, the process of partner selection should consider the various QoS parameters and the trust values of other CSPs for the effective selection of partners for offloading the customer requests. Thus, there should be an effective mechanism for the partner selection in the cloud federation so that both the CSPs and the consumers are benefited. Hence, we discuss the proposed mechanism for the partner selection in the cloud federation environment in this chapter.

In this chapter, we propose the design and implementation of an efficient partner selection mechanism in the cloud federation environment. The major contributions of this work are:

(i) Design of a mechanism to rank the CSPs in the federation using the AHP and the TOPSIS methods. The QoS requirements of the users are given suitable weights using the AHP method. The QoS values of the CSPs in the federation and the user requirements are used

for ranking the CSPs in the federation using the TOPSIS method.
(ii) Incorporation of trust values of the CSPs in the federation into the partner selection approach in order to filter out untrustworthy CSPs.
(iii) Implementation of the proposed approach using the CloudSim toolkit.
(iv) Discussion of the results obtained highlighting the advantages and the disadvantages of the proposed approach.

To the best of our knowledge, this is the first work that effectively ranks the various CSPs in the cloud federation using the TOPSIS method considering the weights of the QoS parameters determined using the AHP method. Also, the final selection of the CSP is done only if the trust value of the selected CSP is above the trust threshold. The proposed mechanism can be used by a CSP in the cloud federation to rank other CSPs whenever that CSP runs out of resources with its own cloud. In this work, our scope of the federation is limited to the IaaS level and we deal with the resource requests for VMs in order to show the partner selection approach adopted in this work. Simulation results show the effectiveness of this approach in order to efficiently select the trustworthy partners in large scale federations to ensure the required QoS to the cloud consumers.

4.1 AHP and the TOPSIS methods

In this section, we discuss the AHP and the TOPSIS methods to show how we have incorporated them in our process of partner selection in the cloud federation.

4.1.1 Analytic Hierarchy Process (AHP)

AHP is one of the most popular Multi-Criteria Decision Making (MCDM) methods that was originally proposed by Prof. Thomas L. Saaty (Triantaphyllou and Mann 1995). This method is used to assign the required weights to the various QoS parameters as per the user service request. These weights are later used in the TOPSIS method to rank the CSPs in the federation. In this method, ratio scales for assigning weights are derived from paired comparisons of QoS attributes. Paired comparisons are used to decide the relative importance of each attribute considered. In order to apply the AHP method, it is required to identify the QoS parameters corresponding to a user request. Here, we have identified five QoS parameters such as uptime, reliability, VM cost, Bandwidth cost and response time.

For a particular user request, pair-wise comparison of the QoS attributes means the user preference for each pair of QoS attributes. That is, each QoS attribute is compared with every other QoS attribute for its relative importance. For e,g., between two QoS attributes such as uptime and VM cost, the value of 2 for the pair-wise comparison means that the user considers uptime twice as important as the VM cost. Partner selection is done when the primary CSP does not have enough resources to meet the user requirements. Hence, based on the user preferences, a suitable partner is identified for offloading the customer request. Relative importance assigned to each QoS attribute is static for a particular user request as it does not change over time. In this approach, the decision-maker has to express his opinion about the value of one single pairwise comparison at a time. It helps to assign proper weights to each of the QoS attributes considered, and the assigned weights are finally represented in the form of a priority vector. This method is easy to use and scalable. Also, the approach is not data intensive and can be easily scaled to accommodate multiple decision making problems due to its hierarchical structure (Zhang et al. 2013).

In our case, we have considered the QoS parameters Uptime, Reliability, VM Cost, Response time and Bandwidth (BW) Cost. Pairwise comparisons of the various QoS parameters are made with the grades ranging from 1 to 9. That is, if QoS parameter 'a' is more important than QoS parameter 'b' and is rated at 2, then 'b' must be less important than 'a' and is graded at 1/2. In our simulation, for a particular user request, pairwise comparisons are carried out for all the QoS parameters to be considered, and the matrix is completed as per the user requirements and it is shown as table 4.1. Thus, table 4.1 is constructed by comparing each QoS attribute with every other QoS attribute for their relative importance. When the QoS attribute 'a' is compared with QoS attribute 'b', a value of 3/2 means that their relative importance is 3:2, and the relative importance of 'b' and 'a' is 2:3. When a QoS attribute is compared with itself, value of 1 is given for their relative preference.

Now, from the table 4.1, the Comparison Matrix or Priority Matrix of size 5X5 for the

Table 4.1 Pair-wise Comparison of QoS Attributes.

QoS Parameter	Uptime	Reliability	VM Cost	Response Time	BW Cost
Uptime	1	1	2	3/2	3/2
Reliability	1	1	4/3	5/4	5/4
VM Cost	1/2	3/4	1	4/5	1
Response Time	2/3	4/5	5/4	1	4/3
BW Cost	2/3	4/5	1	3/4	1

chosen five QoS parameters is formed as shown below:

$$\begin{bmatrix} 1 & 1 & 2 & 1.5 & 1.5 \\ 1 & 1 & 1.33 & 1.25 & 1.25 \\ .5 & .75 & 1 & .8 & 1 \\ .67 & .8 & 1.25 & 1 & 1.33 \\ .67 & .8 & 1 & .75 & 1 \end{bmatrix}$$

In order to find the weights to be assigned to the QoS parameters, the Priority Vector X is calculated from the above matrix by first normalizing the column entries, by dividing each entry in each column by the sum of the entries in the respective column. After this step, the matrix obtained in our case is as shown below:

$$\begin{bmatrix} .2604 & .2299 & .304 & .283 & .2467 \\ .2604 & .2299 & .2021 & .2358 & .2056 \\ .1302 & .1724 & .152 & .1509 & .1645 \\ .1745 & .1839 & .19 & .1887 & .2188 \\ .1745 & .1839 & .152 & .1415 & .1645 \end{bmatrix}$$

Now, take the average of each row of the above matrix and the Weight Vector (Priority

Vector) X is obtained as shown below:

$$\begin{bmatrix} .265 \\ .227 \\ .154 \\ .191 \\ .163 \end{bmatrix}$$

Hence, in this case, the final weights to be assigned to each of the attributes Uptime, Reliability, VM Cost, Response time and Bandwidth Cost as per the user requirements are .265, .227, .154, .191 and .163 respectively. These weight values are used in the TOPSIS method for ranking the partners in the cloud federation.

4.1.2 Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

The TOPSIS method is used for multi-criteria decision analysis (Yoon and Hwang 1995; Yoon 1987; Hwang et al. 1993). This method helps in effectively ranking the CSPs in the federation depending on the QoS requirements of the cloud users. In this method, two alternative solutions such as the 'Positive Ideal Solution' and the 'Negative Ideal Solution' are hypothesized. The Positive Ideal Solution is the one which has the best values for all the attributes considered, and the Negative Ideal Solution is the one which has the worst values for all the attributes. TOPSIS method helps to choose an alternative from different options that has the shortest geometric distance from the Positive Ideal Solution and the longest geometric distance from the Negative Ideal Solution. Considering the QoS attributes of various CSPs in the federation as shown in the figure 4.5, and also the User Request discussed in this work, the values of various attributes in the Positive Ideal Solution are 0.0535, 0.0460, 0.0062, 0.0208, 0.0076. Also, the values of various attributes in the Negative Ideal Solution are 0.0508, 0.0439, 0.0470, 0.0620, 0.0685. The geometric distance between each alternative and the positive and negative ideal solutions is calculated. Then, the relative closeness of each attribute to the Positive Ideal Solution is calculated for selecting the best CSP as shown in the section 4.1.2.

In this method, a set of alternatives are compared by assigning weights for each attribute in the alternatives. The scores for attributes are then normalized and the geometric distance between each alternative and the ideal alternative which has the best score in each of the selected attributes is calculated. The TOPSIS method assumes that the attributes are monotonically increasing or decreasing. The advantages of this approach include other than being a simple process, it is comparatively easier to use and program, and in our case, the number of involved steps remains the same irrespective of the number of QoS attributes considered.

In our case, we have simulated a cloud federation environment consisting of 25 CSPs (i. e., the number of alternatives, m=25). Also, we have considered five QoS attributes such as Uptime, Reliability, VM Cost, Response Time and Bandwidth Cost (i. e., the number of attributes, n=5). We also have the attribute values for each alternative. Let x_{ij} be the value of $Alternative_i$ with respect to $Attribute_j$. Hence, we have the matrix $X = (x_{ij})_{mXn}$. Let P be the set of positive attributes (more value of the attribute indicates better result) and P' be the set of negative attributes (less value of the attribute indicates better result). In our implementation, P={Uptime, Reliability} and P'={VM Cost, Response Time, Bandwidth Cost}. The various steps of the TOPSIS method are explained as follows.

Step 1

From the matrix X discussed above, construct the Normalized Decision Matrix, R=$(r_{ij})_{mXn}$. That is, r_{ij}=$\frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$ where $i = 1,2,...,m$ and $j = 1,2,...,n$.

Step 2

From the matrix R discussed in the Step 1, calculate the Weighted Normalized Decision Matrix, W=$(v_{ij})_{mXn}$. Assume that we have weight w_j assigned for each $Attribute_j$ where $j = 1,2,...,n$. In this work, we have done the weight calculation of various QoS attributes using the AHP method as explained in the previous section. Hence, the matrix W is calculated by multiplying each column of the Normalized Decision Matrix, R by its corresponding assigned weight, w_j. That is, v_{ij}= r_{ij}*w_j where $i = 1,2,...,m$ and $j = 1,2,...,n$.

Step 3

Determine the Positive Ideal Solution and the Negative Ideal Solution. The Positive Ideal Solution is calculated as A^*=$\{v_1^*, ..., v_j^*, ..., v_n^*\}$, where v_j^*=$\{Max\ (v_{ij})$, if $Attribute_j \in P$; $Min\ (v_{ij})$, if $Attribute_j \in P'$, for $i = 1, 2, ..., m\}$. Similarly, the Negative Ideal Solution is calculated as A'=$\{v'_1, ..., v'_j, ..., v'_n\}$, where v'_j=$\{Min\ (v_{ij})$, if $Attribute_j \in P$; $Max\ (v_{ij})$, if $Attribute_j \in P'$, for $i = 1, 2, ..., m\}$

Step 4

Calculate the geometric distance (L-2 distance) between each alternative and the positive and negative ideal solutions. Calculate the separation measures for each alternative. The distance of the $Alternative_i$, where $i = 1, 2, ..., m$, from the Positive Ideal Solution is calculated as:

(S_i^*)=$\sqrt{\sum_{j=1}^{n}(v_j^* - v_{ij})^2}$

Similarly, the distance of the $Alternative_i$, where $i = 1, 2, ..., m$, from the Negative Ideal Solution is calculated as:

(S'_i)=$\sqrt{\sum_{j=1}^{n}(v'_j - v_{ij})^2}$

Step 5

Calculate the relative closeness, C_i^* of each of the alternative to the Positive Ideal Solution. C_i^*= S'_i / (S_i^* + S'_i), $0 < C_i^* < 1$. Now, rank the alternatives in the decreasing order of the value of C_i^*, with the highest value indicating the topmost rank.

4.2 Dynamic Partner Selection in the Cloud Federation Environment

In this section, we discuss our approach of partner selection in the cloud federation scenario. We discuss the overall flow of the resource allocation process highlighting the rank calculation, local resource allocation and the remote resource allocation processes in the cloud federation environment.

4.2.1 Overall Flow of the Proposed Approach

The figure 4.1 shows the overall steps carried out, when the user request needs to be forwarded to a partner in the cloud federation for resource allocation. As shown in the figure, during the simulation, the CloudSim tool (Calheiros et al. 2011) is initialized and the QoS

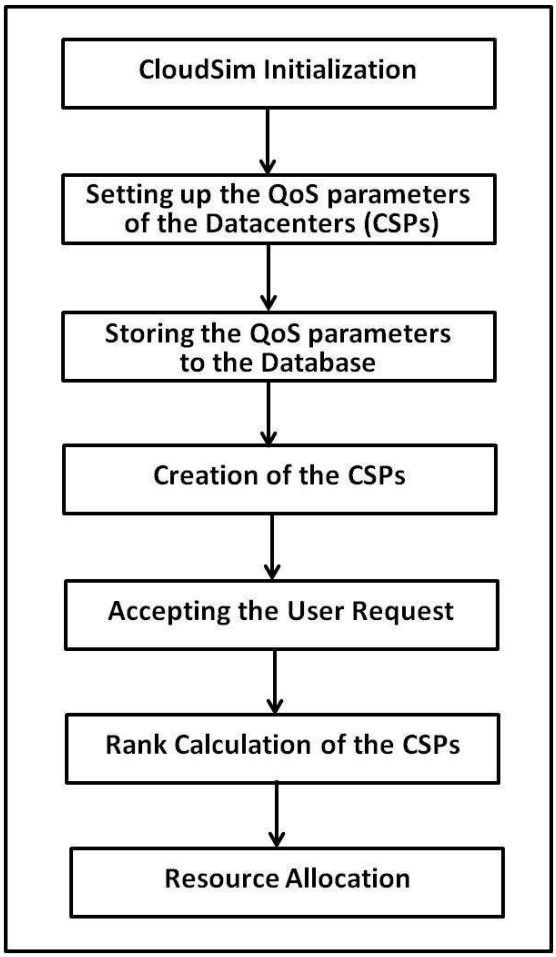

Figure 4.1 Overall Flow of the Work

parameters of the cloud service providers considered in the simulation are set up. We have created 25 CSPs in our simulation to show our approach of partner selection in the cloud federation. In our simulation, a CSP in the federation has the details of the QoS offered by other CSPs in the federation such as the Uptime, Reliability, VM Cost, Response Time, Bandwidth Cost, Instance-Type etc. The CSP stores these details into the database for later access when dealing with the resource requests of the cloud customers. The required number of CSPs with the necessary features are created in the federation. When a cloud user makes a resource request to a CSP, the CSP performs the Rank Calculation of other

CSPs in the federation, depending upon the QoS requirements of the user, and the required resources are allocated to the user from the federation using the information from the Rank Table generated.

4.2.2 Rank Calculation in the Federation

The figure 4.2 shows the various steps involved in the process of ranking the CSPs in the federation depending on the user requirements. When a user request is processed by

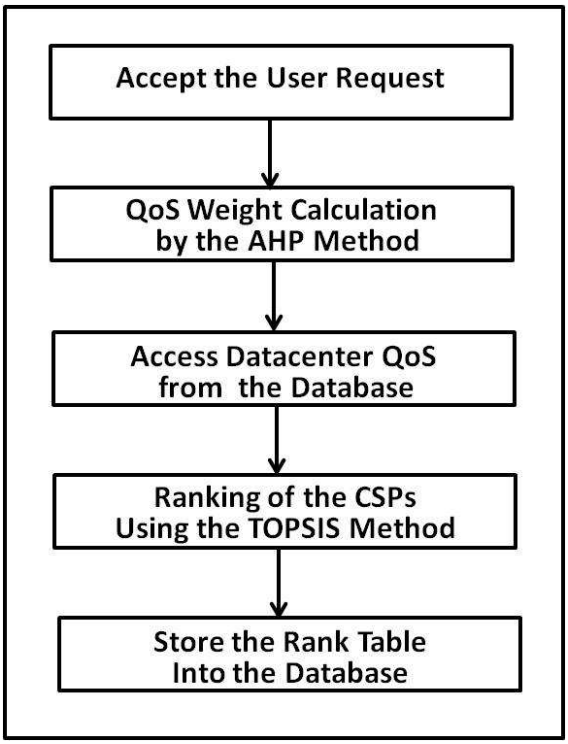

Figure 4.2 Rank Calculation in the Cloud Federation

the CSP, the QoS parameters associated with the user request are given suitable weights using the AHP method as explained before. Then, the QoS features of the cloud service providers in the federation are accessed from the database and the ranking is performed using the TOPSIS method as detailed in the previous section. The calculated rank values are stored into the database for further reference by the CSP.

4.2.3 Resource Allocation

The figure 4.3 shows the Local Resource Allocation process in our implementation. In our simulation, we have considered the IaaS level of resource management. The Broker class

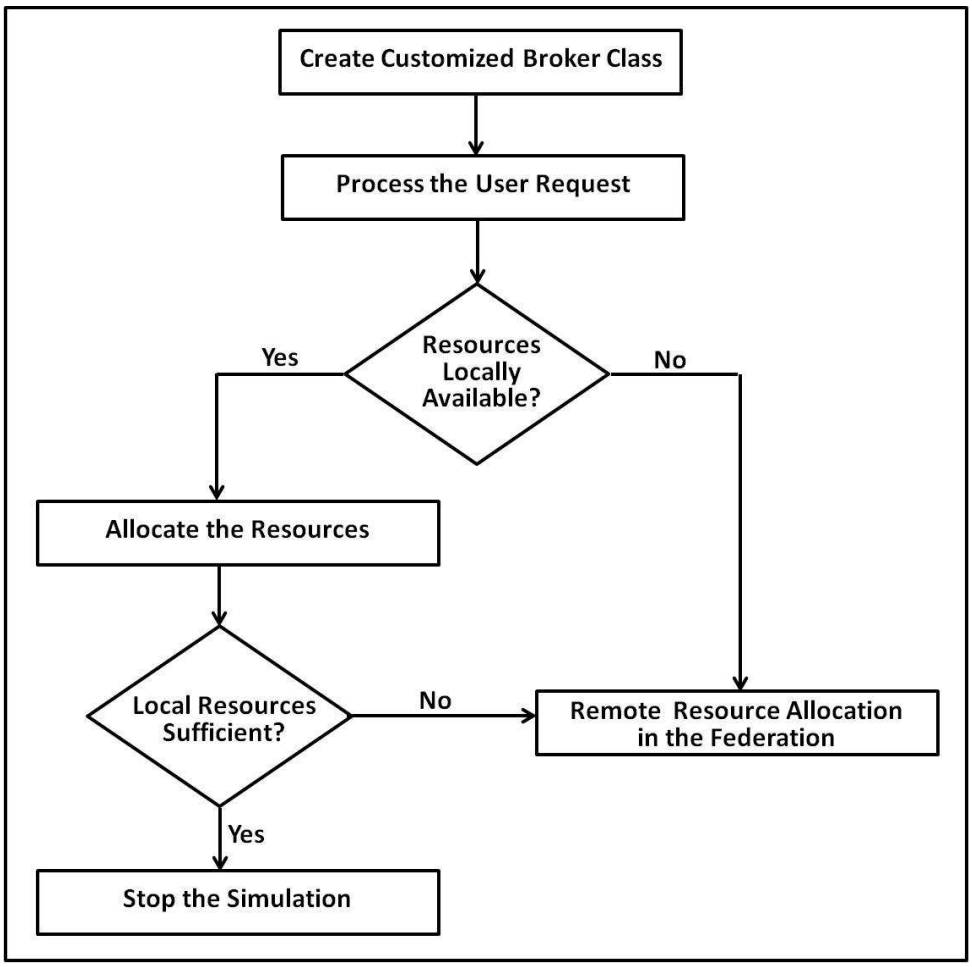

Figure 4.3 Local Resource Allocation in the Cloud Federation

of the CloudSim (Calheiros et al. 2011) is extended to deal with the resource allocation process. Upon receiving a resource request from a cloud user, the CSP checks if the requested resources matching the QoS requirements of the user are locally available with the CSP. If the required resources are available at the moment, it initiates the VM allocation process locally at that CSP, otherwise, if the local resources are not sufficient to meet the client requirements, the Remote Resource Allocation process in the federation is initiated.

4.2.4 Remote Resource Allocation

The figure 4.4 shows the remote allocation of resources in the partner CSPs of the federation, when the local resources are not sufficient to meet the current user requirements. We have assumed that there are SLAs established between the CSPs in the federation to share VMs among them. We have created VMs in different CSPs depending upon the user

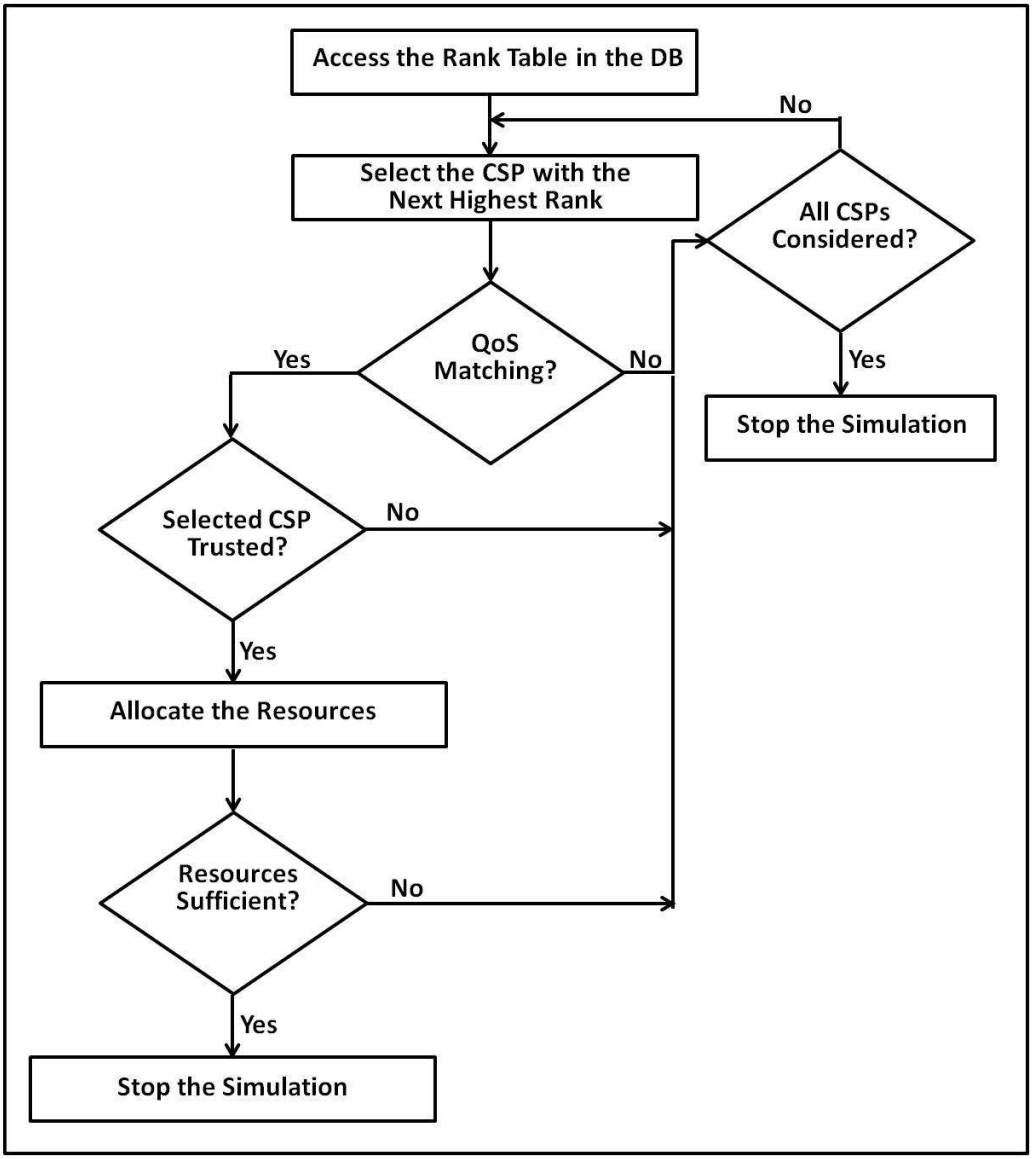

Figure 4.4 Remote Resource Allocation in the Cloud Federation

requirements, and also considering the QoS offered by the partners and their trust values in the federation. When a CSP finds that the resource request from a user cannot be met locally, it uses the Rank Table stored locally to identify the CSP(s) in the federation to ask for resources. From the Rank Table, it selects the CSP having the best rank to check if the QoS requirements of the user are matching with that offered by the selected CSP. If it finds that the QoS details are matching, then the selected CSP's current trust value is checked to verify if the current trust value is above the threshold value set by the CSP. If the CSP finds that the selected CSP in the federation is trustworthy, it requests the available resources from that CSP and allocate them to the requesting user. Now, if the QoS of the selected

CSP is not exactly matching with the user's requirements or the trust value of the selected CSP is less than the trust threshold, or even if the resource requirements of the user cannot be met with this selected CSP alone, then the Rank Table is accessed again to select the CSP with the next highest rank, until either sufficient resources are allocated or all the CSPs in the Rank Table are considered.

4.3 Experimental Results

The primary objective of this experiment is to implement the proposed partner selection approach in the cloud federation environment, and also to verify that the different modules are working correctly in the cloud federation environment simulated using the CloudSim toolkit. Our test scenario consists of different CSPs having their own QoS attributes.

4.3.1 Experimental Setup

We have carried out the simulation experiments on a system with Intel (R) Core (TM) i7-3770, CPU 3.40 GHz, 8.00 GB RAM and 32-bit Operating System (Ubuntu 14.04). Softwares used for our implementation include CloudSim-3.0.3, Eclipse IDE version 3.8, MySQL Workbench Community (GPL) for Linux/Unix version 6.0.8 and Java version 1.7.0_55.

For implementing the Partner Selection module, we have used the Java programming language. We have used MySQL database to store the QoS values of all the CSPs in the Federation which is used for ranking the CSPs. Also, for effectively handling the user's request for resources, each cloud service provider (CSP) has tables for storing the weight values of all the QoS parameters of the user's request and the calculated rank values of the CSPs in the federation.

In our work, we have considered that user requests are coming to CSP-1 in the federation. In our experiment, we have configured that CSP-1 has a capacity of executing 4 VMs, and the current user request is for 15 VMs, as shown in the figure 4.5. The QoS requirements of this request are: Number of VMs=15, Uptime=99.91%, Reliability=99.95%, VM Cost=0.415$, Response Time=6 ms, Bandwidth Cost=0.005$ and the Instance-Type=large. We have considered this request for showing the working of our algorithm. At any time, whenever a CSP receives a resource request whose requirements

cannot be met by that CSP alone, our proposed algorithm is executed to select the suitable partner for availing the resources.

4.3.2 Prerequisites for the Implementation

The prerequisites for the full implementation of the model are explained in this section. The CSP that wants to offload the customer request to some other CSPs in the federation should have the details of the relevant QoS attributes of the current services offered by other CSPs in the federation in order to select the suitable CSP. Also, the CSP should have a mechanism to calculate the trust values of other CSPs in the federation to select the trustworthy CSP to offload the customer requests.

In the real cloud federation, there should be SLAs signed between any two CSPs in the federation for sharing the resources. The quality and quantity of the resources a CSP can get from another CSP in the federation depends on the SLA between the two CSPs in the real cloud federation. In our work, we have assumed that the CSP selected will share the currently available resources to the requesting CSP at any time. Also, the interoperability issues need to be addressed in the real cloud federation scenario, when the user gets the resources from multiple CSPs in the federation.

4.3.3 Results and Analysis

In our experiment, a cloud federation scenario consisting of 25 CSPs was simulated. Also, two heterogeneous hosts were associated with each CSP in the federation. The resource request is made by a user to any of the CSPs in such a way that the requested resources cannot be supplied by that single CSP alone. In the simulated federation environment, at any time, when a CSP cannot deal with the resource requirements of a client due to resource constraints, a suitable partner is selected using the proposed approach mentioned in the previous section, and the access request is transferred to that CSP in the federation. The screen shots showing the various stages of the implementation are discussed in this section.

i. QoS offered by the CSPs in the Federation

As each CSP may be catering to some consumers, available QoS parameters may be affected. For example, response time may increase if the server is busy processing user requests. We have considered QoS parameters such as uptime, reliability, VM cost, BW cost and response time in our work. It is assumed that every CSP stores the QoS values offered by other CSPs in the federation. If there is a change in the value of any of the QoS parameters, that should be updated in the database of all the other CSPs in the federation. It is the responsibility of the CSP to offer the agreed QoS value to other CSPs in the federation until it officially changes the QoS value through SLA-renegotiation. Otherwise, it amounts to SLA violation and affects the trust and reputation of that CSP in the federation.

The CSP that wants to offload the customer request to some other CSPs in the federation should have the details of the relevant QoS attributes of the current services offered by other CSPs in the federation in order to select the suitable CSP. In our case, the figure 4.5 shows the details of the QoS values offered by the various CSPs in the Federation, as

CSP-ID	Uptime(%)	Reliability(%)	VM Cost($)	Response-Time(ms)	BW Cost($)	Instance-Type
1	99.938	99.975	0.382	5.147	0.003	large
2	97.416	96.681	0.410	9.444	0.007	large
3	99.955	99.980	0.302	5.576	0.002	large
4	99.004	97.092	0.307	6.638	0.009	large
5	99.930	99.977	0.274	4.830	0.003	large
6	99.453	96.854	0.374	9.874	0.007	large
7	99.987	99.979	0.303	4.884	0.003	large
8	98.904	95.879	0.515	9.305	0.001	large
9	99.953	99.988	0.339	3.720	0.001	large
10	99.336	97.518	0.240	5.024	0.004	medium
11	99.980	99.999	0.312	3.314	0.001	large
12	98.617	97.421	0.116	6.566	0.005	medium
13	99.940	99.994	0.397	5.248	0.001	large
14	96.287	95.615	0.068	7.894	0.003	medium
15	99.933	99.995	0.252	4.573	0.002	large
16	95.003	98.178	0.377	8.865	0.009	large
17	99.922	99.965	0.358	4.403	0.005	large
18	98.333	96.658	0.500	6.671	0.006	large
19	99.957	99.992	0.387	3.807	0.002	large
20	96.447	98.312	0.194	5.986	0.001	medium
21	99.998	99.963	0.278	4.323	0.003	large
22	99.956	99.976	0.292	5.380	0.003	large
23	99.958	99.999	0.397	4.185	0.004	large
24	99.950	99.975	0.347	5.424	0.001	large
25	99.991	99.976	0.352	3.559	0.003	large

QoS Parameters of User Request

No of VMs Requested=15 Instance-Type=large Uptime=99.91% Reliability=99.95%

Response-Time=6ms VM Cost=0.415$ BW Cost=0.005$

Figure 4.5 QoS offered by the CSPs in the Federation

stored by the CSP-1 to which the user has made the resource request. The various steps

of the TOPSIS method are applied on this data to identify the suitable CSP as CSP-1 does not have enough resources to meet the user's request and hence it wants to offload the customer request to some other suitable CSPs in the federation. As mentioned earlier, we have used 25 CSPs in our simulation and the various QoS features of the CSPs such as Uptime, Reliability, VM Cost, Response Time, Bandwidth Cost, Instance-Type etc. are stored in the database as shown in the figure. The figure also shows the QoS requirements of the current request made by the user as Uptime=99.91%, Reliability=99.95%, VM Cost=0.415$, Response Time=6 ms, Bandwidth Cost=0.005$ and the Instance-Type=large.

ii. Weight Table

The figure 4.6 presents the Weight Table which shows the weights for the different QoS parameters as calculated using the AHP method as per the user requirements. The weights are assigned to different parameters in such a way that the sum of the weights of all the parameters is one. These weight values of the QoS parameters of the user request are used in the TOPSIS method to rank the various CSPs in the federation.

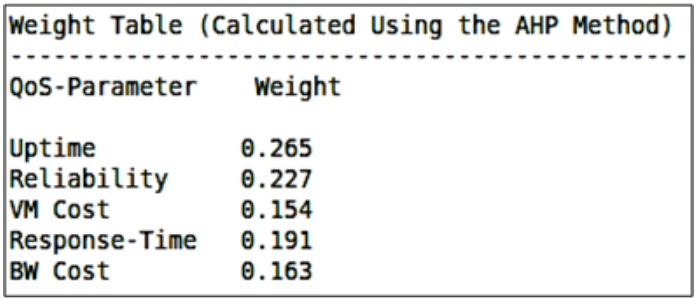

Figure 4.6 Weight Table

iii. Trust Table

The figure 4.7 shows the Trust Table maintained by the CSP-1 to which the user has made the resource request. This table shows the trust value of every other CSP in the federation as calculated by the CSP-1. Every CSP in the federation is assigned a trust value between 0 and 1 which shows how trustworthy that particular CSP is to the CSP-1. The trust value of a CSP is dependent on its past interaction history and/or the recommendation from other trusted CSPs in the federation.

Trust Table of CSP-1

CSP-ID	Trust Value
2	0.68321
3	0.96452
4	0.55475
5	0.70182
6	0.6858
7	0.84941
8	0.63114
9	0.52765
10	0.52275
11	0.88377
12	0.9991
13	0.83252
14	0.89613
15	0.9046
16	0.64566
17	0.68292
18	0.64186
19	0.81725
20	0.62034
21	0.51714
22	0.72369
23	0.51452
24	0.81076
25	0.89136

Figure 4.7 Trust Table

At any time, the local trust value of a CSP is calculated using five parameters such as probability of success, history of interaction, degree of association, existing trust and QoS values. Details of this calculation are given in the next chapter (chapter 4). Trust builds slowly and loses drastically. Initially, existing trust of a CSP will be zero. If a CSP had no interaction with a particular CSP, only degree of association may have some value. Later on, whenever a new resource request is arrived at a CSP from a user, the trust values of other CSPs are calculated, and the trust table will be updated accordingly. In this context, it is the trust between CSPs. The trust threshold to be selected depends on the trust model and its associated parameters. It is difficult to identify a generic value of trust as ideal trust threshold in all cases, as the trust threshold varies from CSP to CSP and also from federation to federation. On a scale of 0-1, a CSP may start with a threshold value of 0.5, and depending on the feedback, it can dynamically increase or decrease the trust threshold

value to include more trusted partners. Finding an optimal trust threshold in a particular context takes some time as it evolves over time. Trust is a context-sensitive, subjective and asymmetric parameter. Since the trust threshold is not a static one, adaptive dynamic trust threshold should be adopted. In our prototype simulation, trust threshold selected is 0.75.

iv. Rank Table

The figure 4.8 shows the Rank Table generated by the CSP-1 to which the user has made the resource request. The ranking of various CSPs in the federation is done using the TOPSIS method, and this table shows the relative preference of CSP-1 for the selection of partners in the federation when dealing with the current resource request. In our simulated

Rank Table of CSP-1 (Calculated using the TOPSIS Method)

CSP-ID	Calculated Value	Rank
20	0.7733	1
11	0.7694	2
15	0.7587	3
9	0.7488	4
3	0.7089	5
24	0.7071	6
5	0.6991	7
21	0.6945	8
7	0.6937	9
25	0.6844	10
19	0.6791	11
22	0.6775	12
13	0.6722	13
14	0.6567	14
10	0.6517	15
23	0.5899	16
12	0.5715	17
17	0.5639	18
8	0.5237	19
18	0.3540	20
4	0.3028	21
6	0.2579	22
2	0.1905	23
16	0.1562	24

Figure 4.8 Rank Table

cloud federation environment of 25 CSPs, this Rank Table shows the ranking of 24 CSPs by the CSP-1, and this table is used for the partner selection when the CSP-1 does not have

enough resources to meet the user's requirements.

v. Resource Allocation in the Cloud Federation

The screen shot explaining the resource allocation in the federation is shown in the figure 4.9. As mentioned earlier, we have used 25 CSPs to show our approach of partner selection in the federation. In our simulation, the user has made the request for 15 VMs with specified QoS requirements as shown in the figure 4.5. In the figure 4.9, it shows the local allocation of four VMs, and then the remaining requested resources (11 VMs) are obtained from the partners in the federation using the Rank Table and the Trust Table. When the CSP-1 identifies that it cannot meet the resource requirements of the user with its own available resources alone (in our case, 4 VMs), it accesses the Rank Table (figure 4.8) maintained locally, and identifies that the CSP-20 is having the highest rank. Then, after verification, the CSP-1 finds that the QoS requirements of the user do not exactly match with that of the CSP-20. Hence, the CSP-1 to which the user has made the request, selects the CSP with the next highest rank from the Rank Table for offloading the customer request. That is, it selects CSP-11. Now, the CSP-1 verifies that the QoS features of the selected CSP (CSP-11) and the requirements of the user match. Also, by accessing the Trust Table (figure 4.7), the CSP-1 verifies that the trust value of the CSP-11 (0.88377) is above the Trust Threshold (0.75 in our case). Hence, the 4 VMs (in accordance with the current availability) are allocated from the CSP-11.

Since 7 more VMs need to be allocated to the user in order to meet his requirements, as shown in the figure 4.9, the CSP-1 selects the CSP with the next highest rank from the Rank Table for offloading the customer request. Now, CSP-15 is selected and after verification, it is found that the QoS requirements of the user and that offered by the CSP-15 match, and also the trust value of the CSP-15 (0.9046) is above the Trust Threshold (0.75). Hence, the 4 VMs (in accordance with the current availability) are allocated from the CSP-15.

Now, 3 more VMs need to be allocated to meet the resource requirements of the user. The process continues for the next selected CSP (CSP-9) as shown in the figure 4.9, and the CSP-1 finds that for the CSP-9, even though the QoS requirements of the user match with that of the CSP-9, the CSP-1's trust value for CSP-9 (0.52765) is found to be less than the current Trust Threshold. Hence, for allocating the remaining VMs (3 VMs), the

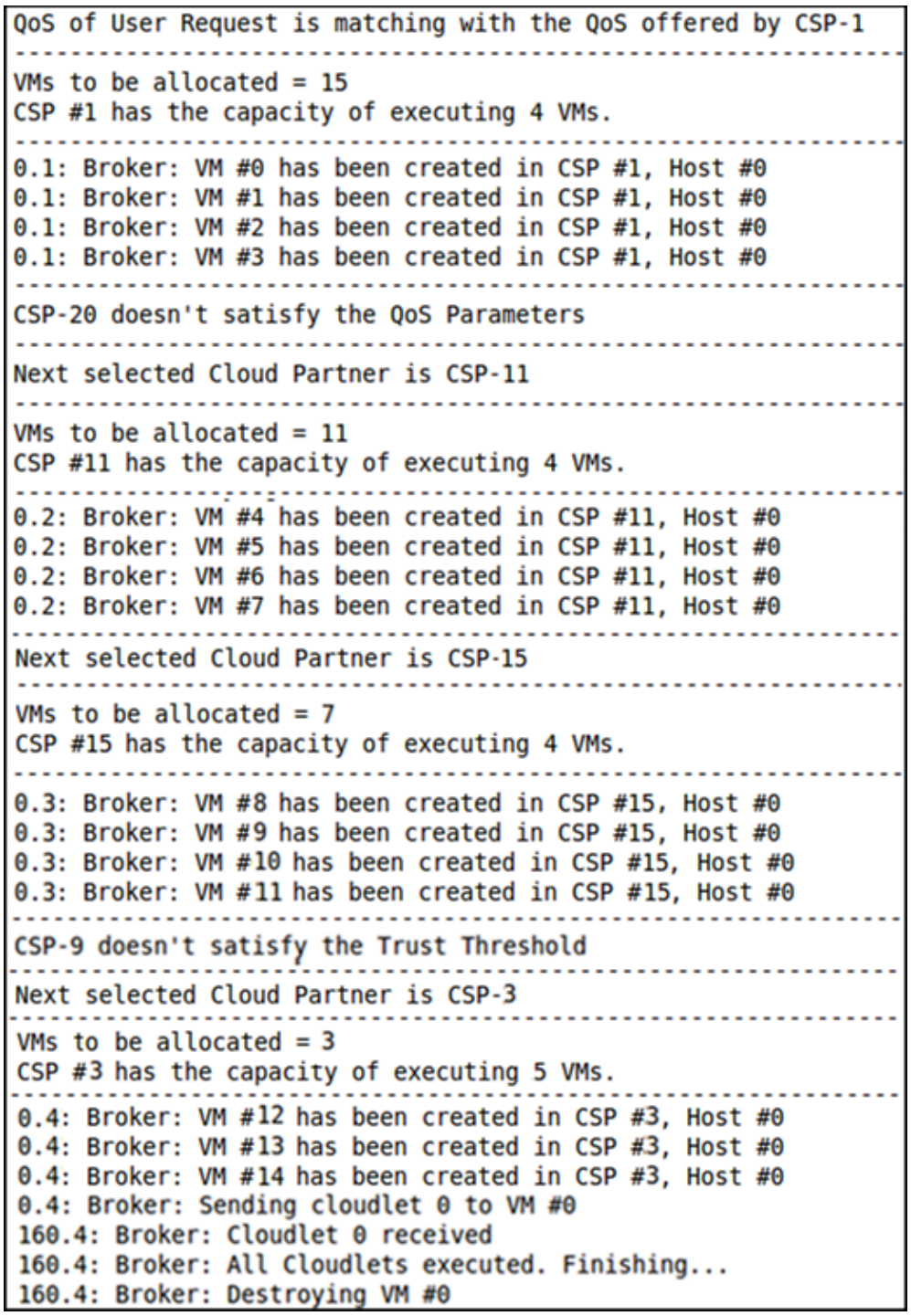

Figure 4.9 Resource Allocation in the Federation

Rank Table is accessed again to select the next best CSP for dealing with the user's request. Hence, CSP-3 is selected in our simulation. For this CSP, its QoS features and the user's requirements match, and also the trust value of the CSP-3 (0.96452) is above the Trust Threshold maintained by the CSP-1. Hence, as shown in the figure 4.9, the remaining

required resources (3 VMs) are requested from this CSP and allocated to the user, and finally the process ends as the requested resources are allocated from the trusted partners in the federation.

Hence, by considering the Rank Table and the Trust Table, the CSP-1 is able to identify the suitable partners in the cloud federation for offloading the customer request so that the QoS requirements specified by the customer is satisfied. Even though the number of VMs requested by the user, the number of QoS parameters considered etc. are reasonably simple enough to understand the work done, the approach is easily scalable to include more CSPs with larger resource capacities and more associated QoS parameters.

4.3.4 Pros and Cons of the Approach

The major advantage of the proposed approach for partner selection is the fact that it helps the CSPs to optimize the search for partners. Instead of searching all the CSPs in the federation to select the suitable one, now the search can be done based on the Rank Table generated. This is especially useful when there are so many CSPs in the federation with different cost and other QoS parameters. Another advantage of the proposed approach is the ability to specify suitable weights to the QoS attributes corresponding to the user requests. Different users will have non-similar priorities, and they can be specified in the proposed approach while selecting the suitable partners in the cloud federation for offloading the users' requests to them. Also, the proposed approach considers the trust values of the CSPs before selecting the suitable CSP from the federation for resource allocation, and his helps to avoid non-trustworthy or malicious CSPs from possible collaboration.

The disadvantage includes if some CSPs in the federation have QoS values according to the calculated weightage, and even if any of the QoS parameters is not exactly matching with what is required by the cloud customer, then that CSP also gets ranked. But, this is not a major disadvantage considering the comparisons required otherwise in which all the QoS attributes of all CSPs in the federation need to be verified to ensure that the user's requirements are met while selecting a suitable partner. In that case, a CSP can consider the CSP with the next highest rank from the Rank Table, for offloading the user's request as shown in the figure 4.9. The CSP that wants to offload the customer request to some other CSPs should have the details of the relevant QoS attributes of the services offered by

other CSPs in the federation for selecting the suitable CSP.

Thus, the proposed method considers QoS attributes and trust values of the CSPs in the federation to select the best CSP for offloading the user's resource request. The user's preferences can be specified as the weights given to the QoS attributes using the AHP method, and the CSPs in the federation are ranked using the TOPSIS method. The selected CSP is used for resource allocation to the cloud users.

4.4 Summary

Cloud Federation is a promising paradigm to meet the highly dynamic resource requirements of the cloud customers. Effective partner selection in the cloud federation is an important task to be performed to maintain the QoS required by the cloud customers. Also, trust is an important issue in the operation of the cloud federation. In this chapter, we have proposed the approach for partner selection in the cloud federation using the AHP and the TOPSIS methods, and also considering the trust values of the CSPs in the federation. Simulation results show the effectiveness of this approach while selecting the trustworthy partners in large scale federations in order to achieve the required QoS.

4.5 Topics Covered in Next Chapter

The next chapter discusses the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. It shows how the efficiency of the cloud federation can be improved by calculating the local and the recommended trust values of CSPs in the federation for dynamically managing the QoS violations among the partners. The proposed approach is implemented using the CloudSim toolkit, and the analysis of the results is carried out.

Chapter 5

Trust-Based Management of Dynamic QoS Violations in the Cloud Federation Environment

This chapter discusses the proposed trust-based approach for the management of dynamic QoS violations in the cloud federation environment. Overall flow of the proposed approach and the details of the local and recommended trust calculation of the CSP are presented. Experimental set up and the analysis of the obtained results are given. Pros and cons of the proposed mechanism is discussed, and finally the chapter ends with its summary and the topics covered in the next chapter.

In the cloud federation environment, whenever a CSP runs out of resources, it can get the resources from other partner CSPs in the federation. Normally, there will be Service Level Agreements (SLAs) between the partners in the cloud federation to share the resources. Due to the dynamic nature of customer requirements, sometimes a CSP in a federation may urgently need some resources from other CSPs in the federation to meet its customer requirements as the requested resources are unavailable with the CSP at that time. Since the CSPs in the cloud federation operate by the Service Level Agreements among them, a CSP can get the services from other CSPs as per the QoS agreement in the SLA among the CSPs. Normally, the process of SLA renegotiation is carried out among the CSPs in order to modify the QoS parameters of the services agreed among them. Now, if a request comes to a CSP from another CSP in the federation for some resources whose QoS features are not as per their prior agreement, how to dynamically deal with such a request in the federation without the time consuming SLA renegotiation at that time is an issue to be considered.

The researchers have been working in the area of resource management in the cloud federation environment. Considering the various approaches proposed by the researchers (details are given in the section 2.3), it is seen that, in order to make the best use of the federation, we need a dynamic management of the possible QoS violations among the partners in the federation so that the mutual benefits of the CSPs in terms of reliability, reputation and the economic benefits are improved. Hence, we discuss the proposed mechanism for the management of dynamic QoS violations in the cloud federation environment in this chapter.

In this chapter, we propose a trust-based mechanism to deal with the QoS or SLA violations among the CSPs in the federation so that without the SLA renegotiation, a CSP can get the required services from other CSPs in the federation, even though the QoS of the service requested is not exactly as per the SLA agreed between the CSPs at that time. In this work, we have implemented the partner selection process when one CSP does not have enough resources, using the AHP and the TOPSIS methods, and also considering the trust values of various CSPs in the federation. We have also implemented the SSO authentication in the cloud federation using the FHMQV protocol and AES-256 algorithm. Trust value of a CSP is decided by calculating the local and recommended trust values of the CSP in the federation, and based on the final trust value of a CSP, the access request for resources from that CSP is either accepted or rejected. As far as we know, this is the first work that employs the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results highlighting the advantages and the disadvantages of the proposed approach is also given. Thus, the proposed approach is used to dynamically manage the QoS violations among the partners in the federation and also, it enables the CSPs to improve their profits and the reputation in the cloud federation environment.

5.1 Access Control Framework

The overview of the access control framework dealing with the dynamic QoS violations in the cloud federation as implemented in our work is shown in the figure 5.1. In our implementation, in order to meet the resource requirements of a user, when the local resources

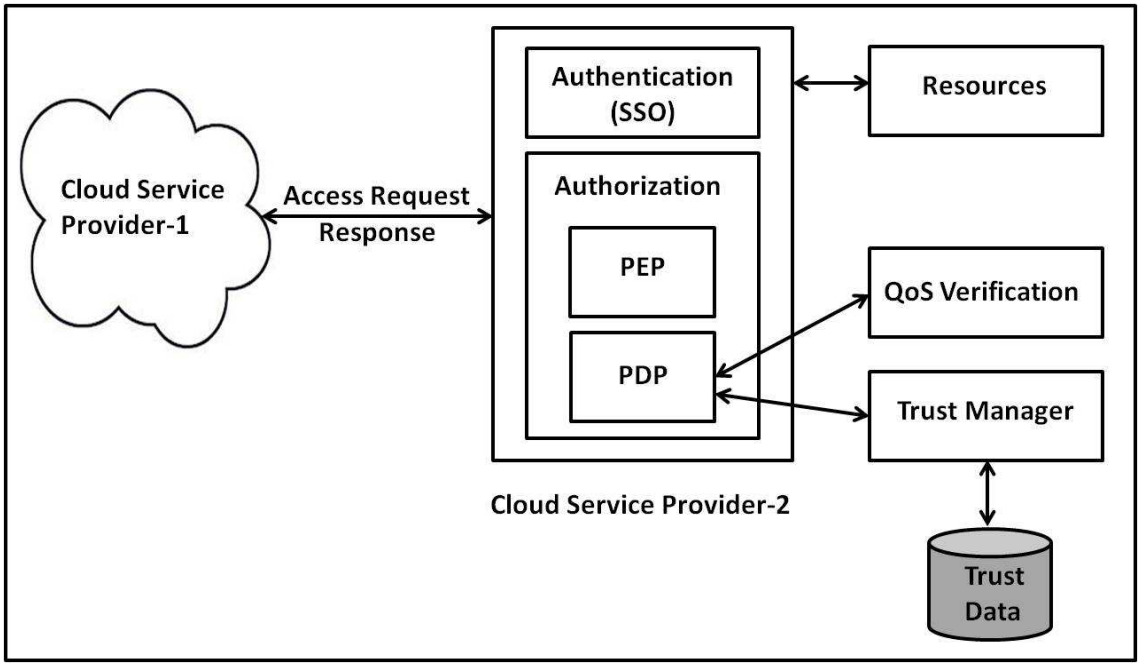

Figure 5.1 Overview of the Access Control Framework

are unavailable, a CSP (CSP-1) selects the most suitable CSP in the federation using the AHP and the TOPSIS methods. Now with the selected CSP, if there is a QoS violation when it gets the access request from the CSP-1, our proposed approach comes into action. Hence, the access control framework of the selected CSP takes the access decision as to whether the access request should be permitted or not considering the trust and reputation of the requesting CSP (CSP-1).

Thus, the various components in this framework as shown in the figure 5.1 are:

5.1.1 Authentication

This module deals with the authentication of the requesting user. In our work, we have implemented Single Sign-On (SSO) mechanism for the authentication of requesting users at different CSPs in the federation. In our implementation, an access request of the user is passed from one CSP to another, in case a CSP runs out of resources at a particular time. Every user needs to be authenticated before availing services from the CSPs in the federation. In this work, we have implemented the SSO approach in the cloud federation environment using AES-256 algorithm (Pub 2001) and the FHMQV protocol (Sarr et al. 2010).

5.1.2 Authorization

When a CSP gets service request from other CSPs in the federation, it takes the access decision dynamically considering various factors. Hence, this module verifies the access rights of the requesting CSP. This module of a CSP has two components, PEP (Policy Enforcement Point) and PDP (Policy Decision Point).

(i) PEP-The PEP contacts the PDP for access decision and implements the access decision taken by the PDP.

(ii) PDP-Whenever a CSP receives a service request from another CSP, this component verifies the request and takes a decision as to whether the request should be permitted or not. As shown in the figure 5.1, this component contacts the QoS-Verification module for verifying the QoS terms of the agreed service with the requesting CSP. In case a CSP requests some services whose QoS features do not exactly match with that mentioned in the SLA, the PDP contacts the Trust Manager module for calculating the trust value of the requesting CSP. Trust Manager calculates the local trust value by accessing the trust data stored locally, and the recommended trust value (reputation) by contacting other trusted CSPs in the federation. If the final trust value of the requesting CSP is above the trust threshold, the resource request from the requesting CSP (CSP-1) is accepted, otherwise rejected.

5.2 Proposed Approach for the Management of Dynamic QoS Violations in the Cloud Federation

QoS violations are detected by comparing the details of the resource request of a CSP to another CSP with the SLA agreed between them. QoS/SLA violation in the cloud federation occurs when one CSP requires some service from another CSP whose QoS features differ from what has been agreed in the SLA between them. Suppose that there is an SLA agreed between CSP-A and CSP-B in the cloud federation. Also, assume that as per the SLA, CSP-B has agreed to give the service consisting of a maximum of n number of VMs of type '*small*' to CSP-A. Now, imagine that CSP-A makes a service request of m VMs ($m > n$). Also the type of the VMs requested is '*large*'. This is an example of the QoS/SLA violations between the CSPs. Even though this example is simple, we have

considered this just to show the working of our approach. Such dynamic QoS violations in the cloud federation environment are handled by the proposed trust based mechanism. The various functional components in the proposed approach for dealing with the dynamic QoS violations are shown in the figure 5.2. They are discussed in the following sections.

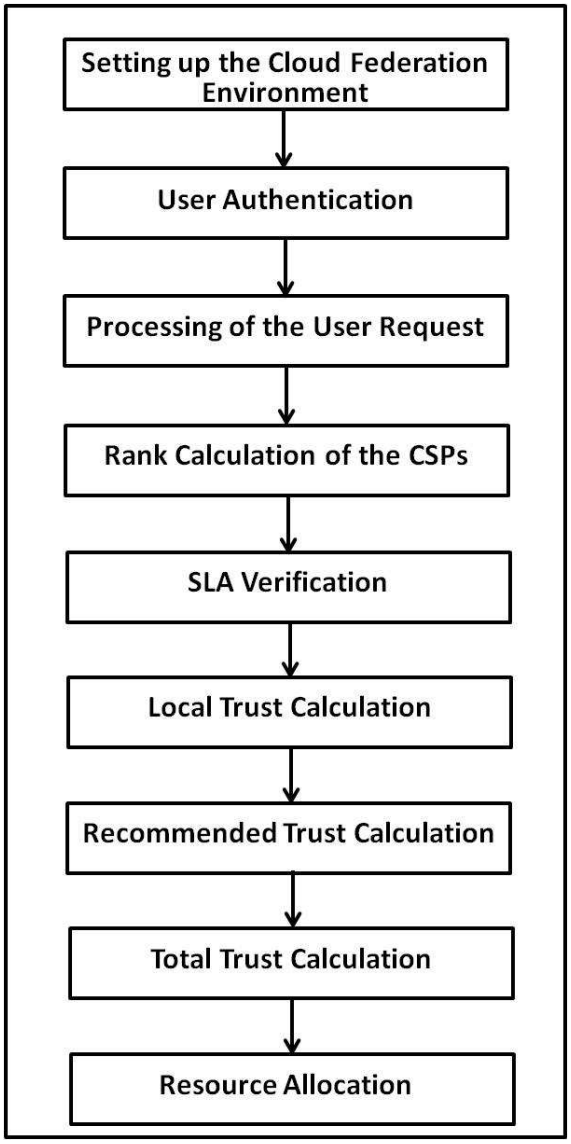

Figure 5.2 Overall flow of the Management of Dynamic QoS Violations

5.2.1 Setting up the Cloud Federation Environment

The required cloud federation environment needs to be set up for implementing and testing the proposed approach. We have set up a cloud federation environment of 25 CSPs using the CloudSim toolkit (Calheiros et al. 2011) to implement the proposed mechanism.

5.2.2 User Authentication

In the cloud federation environment, a user requesting the service needs to be authenticated. When one CSP does not have enough resources, it can transfer the access request of the user to other CSPs in the federation. In order to avail the requested services from that CSP, the user needs to be authenticated there also. We have implemented the SSO authentication as explained in the chapter 3 to facilitate that. In this case, the user need not enter the identity credentials again and again, but only once at the first CSP of the federation.

5.2.3 Processing of the User Request

The user request is analyzed to verify the details of the requested service such as number of VMs, type of VMs etc.

5.2.4 Rank Calculation of the CSP

In our implementation, when a CSP does not have enough resources to meet the requirements of the user, it ranks the various CSPs in the federation so that the best CSP can be selected for transferring the user's request. In the federation, there can be many CSPs offering different types of services with different QoS features such as availability, reliability, uptime, response time, cost etc. Also, a CSP in a cloud federation may not have equal trust values towards every other CSP in the federation at a time. Hence, for any CSP in the cloud federation, the selection of suitable CSP(s) for availing the required resources is an important activity in order to increase its business value. The cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. We have used AHP (Triantaphyllou and Mann 1995) and the TOPSIS (Yoon and Hwang 1995; Hwang et al. 1993) methods for the rank calculation of the CSPs in the federation. The various steps in the process of Rank Calculation are shown in the figure 5.3. When a user request is processed by the CSP, the QoS parameters associated with the user request are given suitable weights using the AHP method, and these weights are used in the TOPSIS method to rank the various CSPs in the cloud federation according to the user requirements. The calculated rank values are stored into the database for further reference by the CSP.

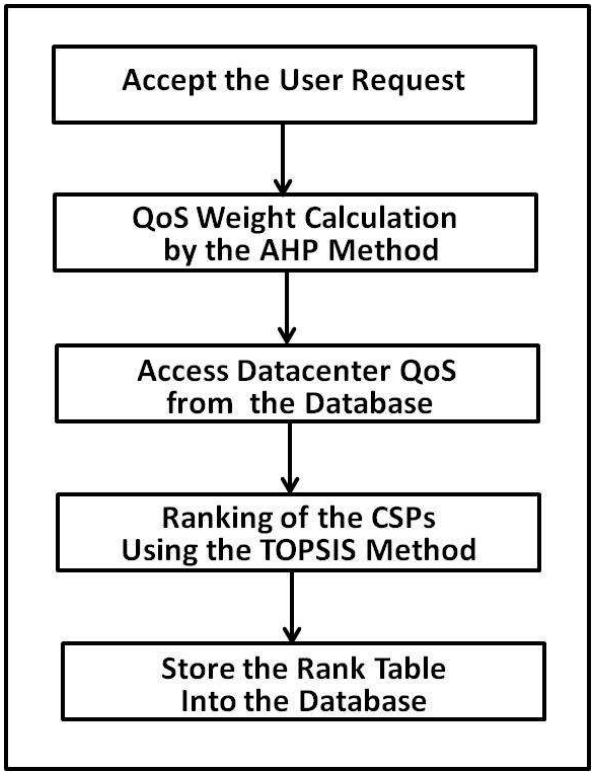

Figure 5.3 Rank Calculation in the Cloud Federation

5.2.5 SLA Verification

In our work, when a CSP runs out of resources, it selects a suitable partner CSP from the federation using the process of Rank Calculation (as explained in the previous section) to transfer the user's resource request. Now, when the selected CSP gets the resource request from a CSP, it verifies the SLA or QoS agreements with the requesting CSP. If the QoS features of the current request match with that present in the SLA, the request is accepted by the CSP and the available resources are given to the requesting CSP. If there is a violation of the QoS agreed between them, then the proposed trust-based mechanism is used to deal with the resource request as explained in the following sections.

5.2.6 Local Trust Calculation

In our proposed approach, whenever a CSP gets a service request from another CSP, if there is an SLA or QoS violation, as a matter of mutually improving the economic benefits and the reputation of the CSPs in the federation, the CSP receiving the request calculates the trust value of the requesting CSP locally. If the local trust calculated is above the

predefined trust threshold, the service request is accepted, otherwise, the CSP calculates the recommended trust of the requesting CSP. The details of the local trust calculation of the requesting CSP are given in the section 5.3.

5.2.7 Recommended Trust Calculation

This module calculates the recommended trust value of the requesting CSP. The trusted CSPs are selected by calculating the trust values of various CSPs in the federation. Then, the feedback regarding the requesting CSP is collected from those trusted CSPs to calculate the recommended trust of the requesting CSP. The details of this process are shown in the section 5.4.

5.2.8 Total Trust Calculation

In our implementation, in order to take a decision on whether to accept or reject the access request made by a CSP, its local trust and recommended trust values are calculated and finally, the total trust of the requesting CSP is calculated as:

Total Trust of the CSP=(Local Trust + Recommended Trust)/2. Now, depending on the total trust value, suitable access decision is taken. Thus, if the total trust value is above the trust threshold, access request is permitted and if that is below the trust threshold, access request is rejected. If the trust is less than the trust threshold, stop federating with that CSP as the CSP cannot compromise on the security of its customers' data.

The trust threshold to be selected depends on the trust model and its associated parameters. It is difficult to identify a generic value of trust as ideal trust threshold in all cases, as the trust threshold varies from CSP to CSP and also from federation to federation. On a scale of 0-1, a CSP may start with a threshold value of 0.5, and depending on the feedback, it can dynamically increase or decrease the trust threshold value to include more trusted partners. Finding an optimal trust threshold in a particular context takes some time as it evolves over time. Trust is a context-sensitive, subjective and asymmetric parameter. Since the trust threshold is not a static one, adaptive dynamic trust threshold should be adopted. The trust threshold selected in our work is 0.6.

When a CSP requests resources from another CSP in the federation, the following situations occur, and they are handled by the CSP as explained.

(a) When the trust is low and SLA not violated, the proposed mechanism is not required and the access request may be rejected.

(b) When the trust is low and SLA violated, the access request is rejected as the trust is less than the trust threshold.

(c) When the trust is medium and SLA not violated, the proposed mechanism is not required and the access request may be rejected.

(d) When trust is medium and SLA violated, the access request is rejected as the medium trust is less than the trust threshold.

(e) When the trust is high and SLA not violated, the proposed mechanism is not required and the access request is permitted.

(f) When the trust is high and SLA violated, the access request is permitted as the trust is greater than the trust threshold.

5.2.9 Resources Allocation

Depending on the total trust value of the requesting CSP, the PDP of the CSP as shown in the figure 5.1, takes a decision to permit the access request if the trust value of the requesting CSP is above the trust threshold maintained in the system. The access request of the CSP is rejected if the final trust value is less than the trust threshold.

Local Trust Calculation and the Recommended Trust Calculation of the requesting CSP are discussed in details in the following sections.

5.3 Local Trust Calculation of the CSP

The calculation of the local trust of the requesting CSP by another CSP in the federation involves the following parameters.

5.3.1 Probability of Success

The Probability of Success of the requesting CSP with any other CSP in the federation shows how many transactions with that CSP were successful in the past. This can be

calculated as: Probability of Success=(x/y), where x is the total number of successful transactions and y is the number of total transactions initiated with that CSP.

5.3.2 History of Interaction

This shows the lead of the number of successful transactions over the number of unsuccessful transactions with a particular CSP. It is calculated as: History of Interaction=$(x-a)/y$, where x is the total number of successful transactions, a is the total number of unsuccessful transactions and y is the number of total transactions by a specific CSP with another CSP in the federation.

5.3.3 Existing Trust

This shows the existing trust value of a CSP before the current trust value is calculated. As this factor indicates, a CSP with a higher existing trust value is expected to have a more positive impact on the calculation of its current trust value than a CSP having a lower existing trust value, or a CSP joined the federation recently.

5.3.4 Degree of Association

For calculating the local trust value of a CSP in the federation, the total period of association of the CSP with the federation is taken into account, by considering the date and time of joining of the CSP with the cloud federation. Based on the date and time of joining the federation, the Degree of Association is given a value x for the CSP, where $x \in [0, 1]$. In our simulation, the parameter x takes values 0, .1, .2, .4, .6, .8 and 1 corresponding to seven ranges of the time period such as less than one month, between 1 month and 2 months, 2 months and 4 months, 4 months and 6 months, 6 months and 9 months, 9 months and 12 months (1 year) and greater than 12 months respectively.

Thus, degree of association of a CSP considers how long that CSP has been associating with the federation. In our work, the local trust calculation of a CSP involves five parameters such as probability of success, history of interaction, degree of association, existing trust and QoS values. If the CSP behaves maliciously with another CSP during its association with the federation, the other parameters such as probability of success, history of interaction and QoS values will reflect that malicious activity, and hence the total local

trust value of that CSP will be reduced.

5.3.5 QoS Value

While calculating the local trust value of the requesting CSP in the federation, the QoS parameters are considered separately to distinguish one CSP from another in the federation. In our work, the QoS parameters considered are availability, reliability, confidentiality, integrity and response time. It shows the details of the previous interaction with that CSP in the past. Hence, this calculation involves the following factors:

Availability Factor: Availability Factor is calculated as (p/z), where p is the total number of times the service from the requesting CSP was available when requested and, z is the total number of service requests made to that CSP.

Reliability Factor: Reliability Factor is calculated as (q/p), where q is the total number of times the service was reliable and, p is the total number of times the service was available from that CSP. Reliability factor is evaluated based on the customer feedback. This factor takes into account how consistent the service quality such as bandwidth or any agreed QoS parameter has been as experienced by the customers.

Confidentiality Factor: Confidentiality Factor is calculated as (r/p), where r is the total number of times the confidentiality was intact with the service from the requesting CSP and, p is the total number of times service was available from that CSP. Confidentiality factor is evaluated based on the customer feedbacks. If the customer reports that confidentiality of his data was compromised after using the services from a CSP, the CSP will treat this as violation of confidentiality.

Integrity Factor: Integrity Factor is calculated as (s/p), where s is the total number of times the integrity was intact with the service from the requesting CSP and, p is the total number of times service was available from that CSP. If the cloud user uses the service of a CSP to store his personal data, and if it is found that data are modified maliciously while stored on the CSP, the integrity is violated.

Response Time Factor: Response Time Factor is calculated as (t/p), where t is the total number of times the response time was within the promised limit and, p is the total number of times service was available from that CSP.

Also, if any of the confidentiality, integrity and response time feature is violated, relia-

bility of the service is affected.

Hence, the final QoS Value of the requesting CSP in the federation is calculated as:

QoS Value=(Availability Factor + Reliability Factor + Confidentiality Factor + Integrity Factor + Response Time Factor)/5

Thus, the Local Trust Value of the CSP is calculated as:

Trust Value= (Probability of Success + History of Interaction + Degree of Association + Existing Trust + QoS Value) /5

QoS values are evaluated while calculating the local trust value of a CSP by another CSP. Generally, there are many CSPs in the cloud federation each with its own business priorities. Also, the trust value of a CSP calculated by any other CSP is subjective. That means, same CSP may be trusted differently by another two CSPs in the cloud federation. While calculating the QoS values, we have considered five factors such as availability, reliability, confidentiality, integrity and response time factors. In real time implementation, the weightage given for each parameter may be different from one CSP to another depending on their business objectives. Also, it depends on the type of service a CSP offers to its customers. For example, a CSP may require some application with very high response time, another CSP might require a service with a high degree of confidentiality, and a third one might require a service with a high degree of availability etc. Hence, by considering these factors, the various parameters can be given suitable weights by a CSP in the cloud federation environment. In our prototype simulation, just to show the working of the proposed approach, we have given equal weights to all the parameters. In real time cloud federation environment, it will vary from one CSP to another.

5.3.6 Trust Decay Factor of the CSP

In the cloud federation domain, the trust value of a CSP is considered to be dynamic and the calculated trust value decays over time. Hence, we have considered the Trust Decay Factor in our implementation, while calculating the local trust value of the requesting CSP in the federation. In our implementation, this decay factor is selected depending on when the requesting CSP had the last transaction with any other CSP in the federation. The decay factor is adjusted in such a way that the trust value gets decremented more when the date and time of the last transaction of a CSP with the requesting CSP becomes older. In

our implementation, this decay factor is represented as $1/x$, where $x \in [1, 2]$, depending on the date and time of the last transaction.

Hence, the Final Local Trust Value of the requesting CSP in the federation is calculated as:

Local Trust Value=Trust Value X Trust Decay Factor

We have selected the decay factor as $1/x$ to show the variation in the trust value of a CSP, where x depends on the time elapsed since the last transaction of the requesting CSP with any other CSP in the federation. In our prototype simulation, the parameter x takes values 1.1, 1.2, 1.4, 1.6, 1.8 and 2 corresponding to six ranges of the elapsed time since the last transaction, such as less than one month, 1-3 month(s), 3-6 months, 6-9 months, 9-12 months and greater than one year respectively. In real time implementation, the parameter x is also CSP-specific. Practically, different CSPs can use different values for x for the same time period. It also depends on how long the cloud federation has been in existence, and also how long the requesting CSP has been a member of this federation. Accordingly, a CSP in the federation can decide the value of x.

In our work, if no transaction has taken place between two CSPs, the trust decay factor selected is 1/2 as it can be taken as the case "greater than one year" since the last transaction. As per our trust model, the minimum trust value that a CSP can have about another CSP is when no transaction has taken place between them in the past. In that case, only the 'degree of association' may have non zero value for the local trust calculation, as the other factors are zero. This factor also takes the value 'zero', if the CSP has joined the federation within the last one month. In that case, the local trust value becomes 'zero' as per our design. This situation is not avoided. But, the CSP can use the recommended trust to start transacting with this CSP, and the trust value is boosted again through successful transactions between the CSPs, and this is the only way to improve the trust between any two CSPs. If the two CSPs had some successful transactions between them in the past, then they may have a non-zero trust value between them. If local trust is insufficient, the CSP will calculate the recommended trust at any time, but this will not affect the local trust value. The local trust value will change only through successful personal experience with any other CSP. Also, local trust value is calculated using the database stored locally.

Trust decay factor is used when the local trust value calculated using the database is to be used in a particular context. At any time, existing trust changes to the calculated local trust value, and not to the product of local trust value and the trust decay factor. The trust decay factor shows when the CSPs had the last transaction, and also using the concept 'trust decays over time', the local trust value is modified to a practically usable value by multiplying it with the trust decay factor.

5.4 Recommended Trust Calculation of the CSP

In our proposed approach, the recommended trust calculation of the requesting CSP involves the following steps.

5.4.1 Selection of Trusted CSPs

In order to calculate the recommended trust of the requesting CSP, the trusted CSPs in the federation are identified. When a CSP gets a resource request from another CSP, the CSP calculates the trust values of other CSPs in the federation to identify the trusted CSPs, and from this trusted CSPs, the feedback of the requesting CSP is collected. In order to select the trusted CSPs, the CSP calculates the trust values of other CSPs in the federation considering the parameters such as Probability of Success, History of Interaction, Existing Trust, Degree of Association and QoS Values, and these parameters are calculated as explained in the section 5.4. Those CSPs with trust values greater than a specific threshold are selected into the list of trusted CSPs.

5.4.2 Recommended Trust Calculation

After selecting the trusted CSPs in the federation, the CSP contacts the CSPs in the list of trusted CSPs regarding the feedback of the requesting CSP. The CSP contacts those CSPs (m out of n CSPs, where m is the number of trusted CSPs, n is the total number of CSPs in the federation and $m <= n$) and each of the m CSPs calculates its current trust value of the CSP specified, and communicates that trust value to the CSP that asked for it. The CSP then aggregates the trust values collected from the trusted CSPs to calculate the final recommended trust of the requesting CSP in the federation, and decides to grant or reject the resource request from that CSP, even if there is a QoS/SLA violation at that time.

After calculating the final recommended trust value, the CSP calculate the total trust value of the requesting CSP as:
Total Trust Value=(Local Trust + Recommended Trust)/2, and based on this total trust value of the requesting CSP, it takes a proper decision regarding the service request.

In our simulation, total trust value of a CSP is calculated as the average of the local trust and the recommended trust values. Local trust value is based on own experience of working with a particular CSP, and the recommended trust value is based on the feedback from other trusted CSPs. In our work, in order to calculate the recommended trust of a CSP, initially the trusted CSPs are selected. Here also, for selecting the trusted CSPs, the trust-threshold used is CSP-specific. Generally, it can be reasonably high (0.85 in our case). Then, the feedback is collected from these CSPs. Also, any outlier in the feedback is eliminated using the algorithm proposed in (Azzedin and Ridha 2010). Hence, this recommended trust value also assumes importance in the calculation of the total trust value. That is the reason why we have given equal weightage to local trust and the recommended trust. Again, in the real time cloud federation implementation, a CSP can use different weights such as 0.6 for the local trust value and 0.4 for the recommended trust value. In our prototype simulation, just to show the working of the proposed mechanism, we have used equal weights (0.5) for both the local trust and recommended trust values.

5.5 Workflow of the Proposed Approach

The workflow of the proposed approach for the management of dynamic QoS violations in the cloud federation environment is discussed in this section.

5.5.1 Local Resource Allocation

The figure 5.4 shows the Local Resource Allocation process in our implementation. In our simulation, we have considered the IaaS level of resource management. The Broker class of the CloudSim (Calheiros et al. 2011) is extended to deal with the resource allocation process. Upon receiving a resource request from an authenticated cloud user, the CSP checks if the requested resources matching the QoS requirements of the user are locally available with the CSP. If the required resources are available at the moment, it initiates the VM allocation locally at that CSP, otherwise, if the local resources are not sufficient to

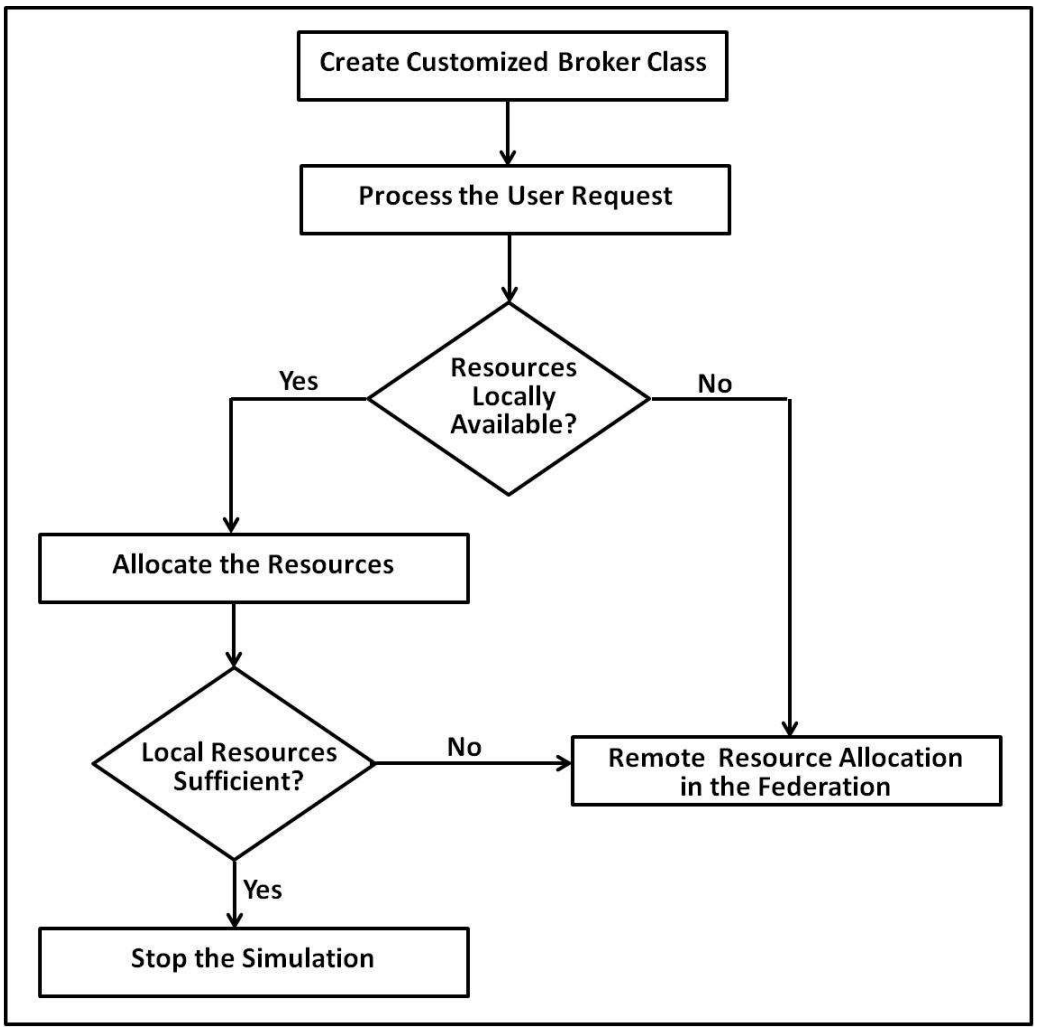

Figure 5.4 Local Resource Allocation in the Cloud Federation

meet the client's requirements, the Remote Resource Allocation process in the federation is initiated.

5.5.2 Remote Resource Allocation

The figure 5.5 shows the remote allocation of resources in the partner CSPs of the federation, when the local resources are not sufficient to meet the current user requirements. We have assumed that there are SLAs established among the CSPs in the federation to share VMs among them. When a CSP finds that the resource request from a user cannot be met locally, it uses the Rank Table stored locally to identify the CSP(s) in the federation to ask for resources. From the Rank Table, it selects the CSP having the best rank to check if the QoS requirements of the user are matching with that of the selected CSP. If it finds that the

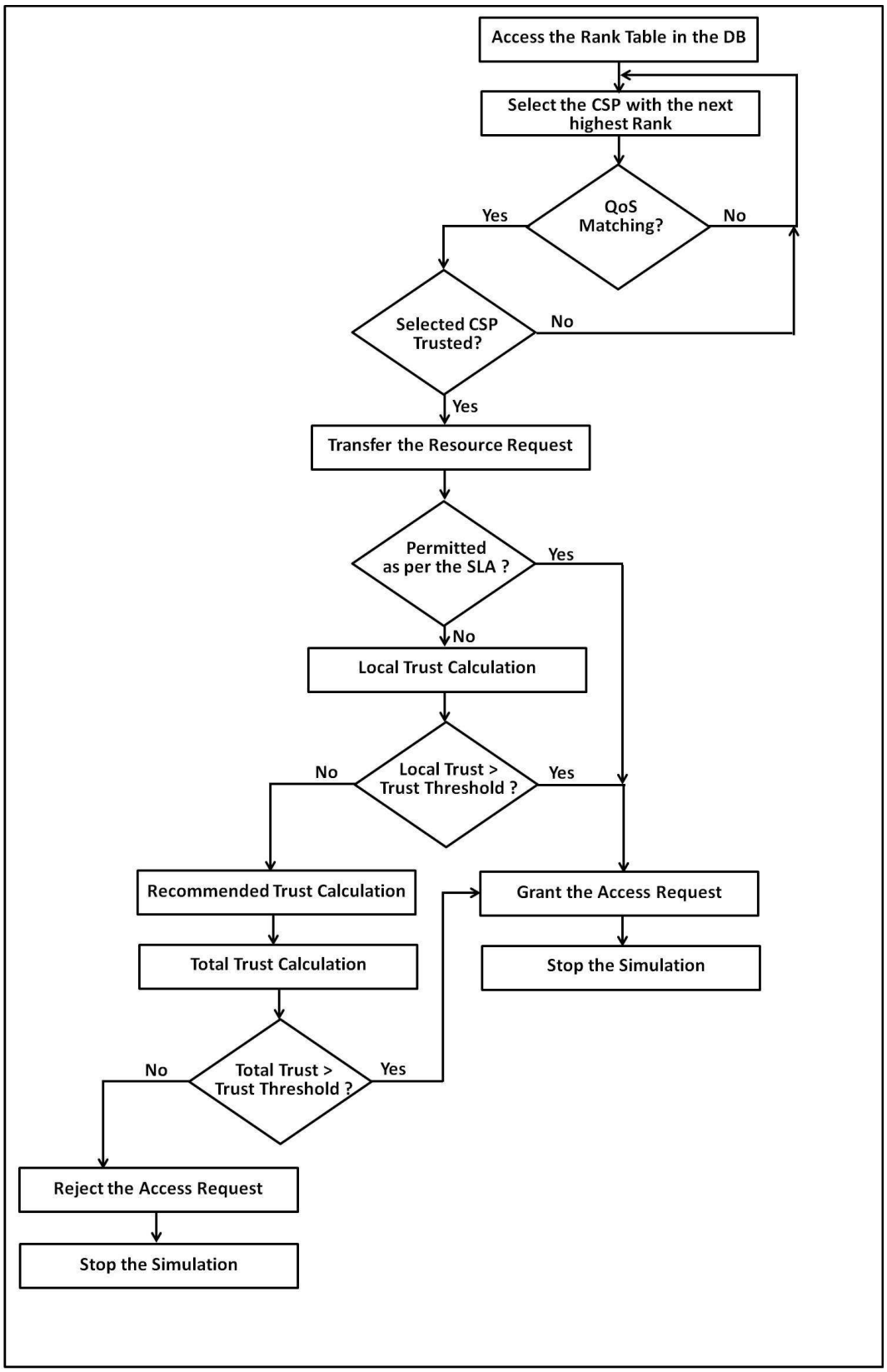

Figure 5.5 Remote Resource Allocation in the Cloud Federation

QoS details are matching, then the selected CSP's current trust value is checked to verify if the current trust value is above the threshold value set by the CSP. If the CSP finds that the selected CSP in the federation is trustworthy, it requests resources from the CSP by transferring the resource request to that CSP.

When a CSP gets the resource request from another CSP in the federation, the CSP verifies the SLA with the requesting CSP to ensure that the QoS requirements of the resource request is agreed by the SLA. If the QoS of the requested service is permitted by the SLA, the service is granted as per the available resources. If there is an SLA or QoS violation, then the local trust calculation of the requesting CSP is performed. If the calculated local trust value of the requesting CSP is greater than the threshold value, then the access request is permitted. If the local trust value of the CSP is less than the trust threshold, then the recommended trust of the requesting CSP is calculated as explained in the previous section. Then, the CSP calculates the total trust value of the CSP considering both the local and the recommended trust values. If the total trust value of the requesting CSP is greater than the threshold value, then the access request is permitted, in spite of the QoS violations. If the total trust value of the CSP is less than the trust threshold, the resource request of the CSP is rejected.

5.6 Experimental Results

The objective of this experiment is to implement and test the proposed approach for dynamically managing the QoS violations in the cloud federation environment. Our test scenario consists of 25 CSPs in the cloud federation.

5.6.1 Experimental Setup

We have carried out the simulation experiments on a system with Intel (R) Core (TM) i7-3770, CPU 3.40 GHz, 8.00 GB RAM and 32-bit Operating System (Ubuntu 14.04). Softwares used for the implementation include CloudSim-3.0.3, Eclipse IDE version 3.8, MySQL Workbench Community (GPL) for Linux/Unix version 6.0.8 and Java version 1.7.0_55.

5.6.2 SSO in Cloud Federation

In our implementation, for securing the data in transit such as during the transfer of the identity tokens of the cloud users between CSPs and also between a CSP and an IdP, we have used the Symmetric Key Encryption technique using AES-256. Also, we have used Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol (Sarr et al. 2010) for key exchange among the entities in the simulation. AES is a protocol mentioned in the set of standard protocols for security by the National Institute of Standards and Technology (NIST) (Pub 2001) and the FHMQV protocol has its root in Diffie-Hellman (DH) protocol. The FHMQV protocol defines the Full Exponential Challenge Response (FXCR) and Full Dual exponential Challenge Response (FDCR) schemes which preserve the performance of the (H)MQV protocol, in addition to providing resistance against various attacks such as the Impersonation Attack, Man-in-the-Middle Attack, replay attack etc.

5.6.3 Management of Dynamic QoS Violations

In this work, we have shown the entire sequence of events that leads to the QoS/SLA violation between two CSPs in the federation. We start with a user request to CSP-1, then the selection of CSPs in the federation and finally, the violation of QoS between two CSPs are discussed. Tables shown in figure 5.7, 5.8, 5.9 and 5.10 are similar in structure to the figures 4.5, 4.6, 4.8 and 4.7 respectively, but data are not the same. The context of cloud federation discussed here is different from what is discussed in chapter 4 as shown in figure 5.7 (compare this with figure 4.5). The details of rank table generation are not given in this chapter as it was discussed in chapter 4. Here, the rank table (figure 5.9) and the trust table (figure 5.10) used in this context is different from what is used in chapter 4.

In order to show the working of the proposed approach, the use case shown in the figure 5.6 is considered. We assume that CSP-1 has to give the service of 50 VMs to a particular user due to some business reasons. As per the current resource availability of CSP-1, it has the capacity of offering only 6 VMs to the user as shown in the figure. Now, the CSP-1 can get the required resources form other CSPs in the federation. Hence, CSP-1 uses the QoS values offered by the CSPs in the federation to rank the CSPs.

The figure 5.7 shows the QoS values offered by the various CSPs in the federation, as

```
------------------------------------------------------------------
QoS of User Request is matching with the QoS offered by CSP-1
Number of VMs Requested = 50
CSP-1 has a capacity of executing 6 VMs
------------------------------------------------------------------
0.1: Broker: VM #0 has been created in CSP-2, Host #0
0.1: Broker: VM #1 has been created in CSP-2, Host #0
0.1: Broker: VM #2 has been created in CSP-2, Host #0
0.1: Broker: VM #3 has been created in CSP-2, Host #1
0.1: Broker: VM #4 has been created in CSP-2, Host #0
0.1: Broker: VM #5 has been created in CSP-2, Host #1
------------------------------------------------------------------
```

Figure 5.6 User Request to CSP-1

```
------------------------------------------------------------------------------------------------------------------
QoS offered by the CSPs in the Federation
------------------------------------------------------------------------------------------------------------------
CSP_ID  Uptime(%)       Reliability(%)  VM Cost($)      Response Time(ms)       BW Cost($)      Instance Type
1       99.95           99.97           0.395                   6               0.003           large
2       99.988          99.95           0.161                   6               0.003           large
3       99.968          99.953          0.381                   2               0.003           large
4       99.935          99.962          0.084                   3               0.003           large
5       99.988          99.964          0.402                   3               0.002           large
6       99.959          99.954          0.126                   4               0.002           large
7       99.963          99.958          0.222                   6               0.001           large
8       99.939          99.971          0.332                   7               0.004           large
9       99.918          99.975          0.253                   2               0.002           large
10      99.995          99.99           0.308                   7               0.006           large
11      99.958          99.956          0.242                   3               0.005           large
12      99.945          99.971          0.228                   7               0.003           large
13      99.981          99.976          0.067                   7               0.005           large
14      99.911          99.987          0.15                    3               0.004           large
15      99.924          99.983          0.117                   5               0.003           large
16      99.912          99.98           0.248                   5               0.002           large
17      99.948          99.973          0.227                   3               0.003           large
18      99.952          99.967          0.319                   5               0.004           large
19      99.999          99.962          0.239                   2               0.002           large
20      99.944          99.975          0.337                   3               0.002           large
21      99.943          99.972          0.207                   5               0.003           large
22      99.987          99.957          0.196                   5               0.003           large
23      99.959          99.977          0.272                   2               0.003           large
24      99.97           99.952          0.204                   4               0.003           large
25      99.999          99.992          0.254                   5               0.003           large
------------------------------------------------------------------------------------------------------------------
```

Figure 5.7 QoS offered by the CSPs in the federation

stored by the CSP-1 to which the user has made the resource request. As mentioned earlier, we have used 25 CSPs in our simulation and the various QoS features of the CSPs such as Uptime, Reliability, VM Cost, Response Time, Bandwidth Cost, Instance-Type etc. are stored in the database as shown in the figure. Assume that the QoS requirements of the current request made by the user are Uptime=99.91%, Reliability=99.95%, VM Cost=0.415$, Response Time=6 ms, Bandwidth Cost=0.005$ and the Instance-Type=large.

The CSP-1 now uses the AHP and the TOPSIS methods to rank the 24 CSPs in the federation. The figure 5.8 shows the Weight Table as calculated using the AHP method (Tri-

Weight Table (Calculated Using the AHP Method)

QoS-Parameter	Weight
Uptime	0.265
Reliability	0.227
VM Cost	0.154
Response-Time	0.191
BW Cost	0.163

Figure 5.8 Weight Table

antaphyllou and Mann 1995) which includes the weights for the different QoS parameters depending upon the user requirements. The weights are assigned to different parameters in such a way that the sum of weights of all the parameters is one. These weight values of the QoS parameters of the user request are used in the TOPSIS method (Yoon and Hwang 1995; Hwang et al. 1993) to rank the various CSPs in the federation.

The figure 5.9 shows the Rank Table generated by the CSP-1 to which the user has

Rank Table of CSP-1 (Calculated using the TOPSIS Method)

CSP_ID	Calculated Value	Rank
19	0.701	1
4	0.696	2
9	0.693	3
6	0.686	4
23	0.646	5
17	0.641	6
14	0.629	7
20	0.624	8
24	0.62	9
15	0.618	10
7	0.608	11
16	0.604	12
3	0.597	13
5	0.595	14
22	0.592	15
21	0.588	16
2	0.573	17
25	0.572	18
11	0.56	19
12	0.527	20
18	0.519	21
13	0.517	22
8	0.474	23
10	0.437	24

Figure 5.9 Rank Table

made the resource request. The ranking of various CSPs in the federation is done using the TOPSIS method, and this table shows the relative preference of CSP-1 for the selection of partners in the federation, when dealing with the current resource request. In our simulated cloud federation environment of 25 CSPs, this Rank Table shows the ranking of 24 CSPs by the CSP-1, and this table is used for the partner selection when the CSP-1 does not have enough resources to meet the user's requirements.

The figure 5.10 shows the Trust Table maintained by the CSP-1 to which the user has made the resource request. This table shows the local trust value of every other CSP in

Trust Table of CSP-1

CSP_ID	Trust Value
2	0.519
3	0.544
4	0.684
5	0.663
6	0.572
7	0.806
8	0.783
9	0.452
10	0.48
11	0.571
12	0.588
13	0.599
14	0.483
15	0.455
16	0.845
17	0.828
18	0.501
19	0.534
20	0.635
21	0.513
22	0.58
23	0.577
24	0.504
25	0.839

Figure 5.10 Trust Table of CSP-1

the federation as calculated by the CSP-1. The trust values of the CSPs are calculated considering the parameters Probability of Success, History of Interaction, Existing Trust, Degree of Association and QoS Values as explained in the section 5.3. Every CSP in the

```
--------------------------------------------------------------------------------
Selection of CSP from Rank Table
--------------------------------------------------------------------------------
CSP_ID          Status          Reason
19              Not Selected    Trust Value less than Trust Threshold(0.65)
4               Selected        QoS and Trust Parameters(0.65) matched
--------------------------------------------------------------------------------
CSP selected from the Federation for meeting the user requirements is CSP-4
The Request for 44 VMs is transferred to the CSP-4
--------------------------------------------------------------------------------
As per the SLA between CSP-1 and CSP-4 ,the Number of VMs agreed is '30';
but the current requirement is '44'.
Also the Instance Type agreed is 'small'; but the current requirement is 'large'.
```

Figure 5.11 Selection of CSPs in the Cloud Federation

federation is assigned a trust value between 0 and 1 which shows how trustworthy that particular CSP is to the CSP-1.

The figure 5.11 shows the selection of CSPs in the cloud federation by the CSP-1 in order to meet the resource requirements of the user. The selection process considers the Rank Table (figure 5.9) and the Trust Table (figure 5.10) created by the CSP-1. As shown in the figure 5.11, even though CSP-19 is having the highest rank, this CSP is not selected because the trust value of this CSP (0.534) is less than the trust threshold (0.65) specified by the CSP-1. Hence, the CSP with the next highest rank (CSP-4) is selected from the Rank Table. Now, CSP-1 verifies that the QoS requirements of the user and the QoS features offered by the CSP-4 match, and also the trust value of CSP-4 (0.684) is above the trust threshold maintained by the CSP-1 (0.65). Hence, CSP-4 is selected for meeting the resource requirements of the user, and the request for 44 VMs is transferred to CSP-4. Now, the CSP-4 checks the SLA agreed between CSP-4 and CSP-1 in the federation. As shown in the figure 5.11, as per the SLA, the number of VMs agreed between them is 30; but the current requirement is for 44 VMs. Also, the instance type of the VMs agreed between them is small; but the current requirement is for 'large'.

Upon receiving the resource request from the CSP-1, as per the proposed approach, in order to deal with this resource request of 44 VMs from CSP-1, CSP-4 calculates the trust value of CSP-1 in the federation so that SLA renegotiation is avoided at that time, and the resource request may be accepted. Firstly, the CSP-4 calculates the local trust value of CSP-1.

The figure 5.12 shows the calculation of local trust of the CSP-1 by CSP-4. As ex-

```
####################################
Calculation of Local Trust of the CSPs
####################################
1. Probablilty of Success = 0.916
2. History of Interaction = 0.832
3. Existing Trust         = 0.855
4. Degree of Association  = 1.0
5. QoS Value              = 0.859

Local Trust               = 0.892
Trust Decay Factor        = 0.625
Final Local Trust         = 0.558

Local Trust < Trust Threshold (0.6)
```

Figure 5.12 Calculation of Local Trust

plained before, for calculating the local trust of the CSP-1, the trust parameters such as Probability of Success, Degree of Association, History of Interaction, Existing Trust and the QoS values are considered. As shown in the figure, the values of these parameters for the local trust calculation of CSP-1 are 0.916, 1.0, 0.832, 0.855, and 0.859 respectively. To calculate the local trust of the CSP-1, the average value of all the above parameters is taken and it is found to be 0.892 as shown in the figure. The Trust Decay Factor for CSP-1 is calculated as 0.625, and hence the final local trust of CSP-1 is 0.558. But, this local trust is less than the trust threshold maintained by the CSP-4 (0.6) for granting the resource request. Hence, the CSP-4 calculates the recommended trust of the CSP-1.

The recommended trust is calculated by taking feedback from the trusted CSPs of CSP-4. Hence, the CSP-4 calculates the trust values of all the relevant CSPs in the federation considering the parameters such as Probability of Success, Degree of Association, History of Interaction, Existing Trust and the QoS values. The trust table generated by the CSP-4 upon receiving the access request from the CSP-1 is shown in the figure 5.13. From this trust table, the set of CSPs having trust value greater than a predefined trust threshold (0.85) is identified. This table is shown in the figure 5.14 as Trusted CSPs of CSP-4. In our case, the number of CSPs having trust value greater than the threshold is 11. These CSPs are asked for recommendation of the CSP-1.

The figure 5.15 shows the Recommendation Table of the CSP-4, and as shown in the figure, the number of CSPs responded with the trust values of CSP-1 is 8. The other three

```
##########################################
Calculation of Recommended Trust of the CSPs
##########################################
------------------------------------------
Trust Table
------------------------------------------
CSP_ID          Trust Value
  2               0.804
  3               0.894
  5               0.86
  6               0.802
  7               0.87
  8               0.763
  9               0.799
  10              0.815
  11              0.859
  12              0.872
  13              0.86
  14              0.805
  15              0.778
  16              0.862
  17              0.799
  18              0.81
  19              0.79
  20              0.884
  21              0.809
  22              0.881
  23              0.87
  24              0.776
  25              0.875
```

Figure 5.13 Trust Table of CSP-4

CSPs may not have the history of interaction with the CSP-1 to calculate its trust value as required by the CSP-4. The table shows the trust values of various CSPs who have responded with the required recommendation, and their corresponding returned trust value of the CSP-1. Now, the trust value of the responded CSPs and their returned trust values are multiplied to get the recommended trust values of CSP-1. In our work, in order to filter the recommendation values given by the CSPs, we have implemented the outlier filtering algorithm proposed by Azzedin et al. (Azzedin and Ridha 2010). Hence, the resulting filtered recommendation table is shown in the figure 5.16. In the filtered recommendation table, the number of recommendations considered is 5, and the 3 recommendations are filtered out. From the filtered recommendation table, the total recommended trust is calculated as the average of the recommended trust of the filtered CSPs, and in our case, the

```
CSP_Trust_Threshold = 0.85
------------------------------------------------------------
Trusted CSPs
------------------------------------------------------------
CSP_ID          Trust Value
  3               0.894
  5               0.86
  7               0.87
  11              0.859
  12              0.872
  13              0.86
  16              0.862
  20              0.884
  22              0.881
  23              0.87
  25              0.875

Total number of CSPs above the CSP_Trust_Threshold : 11
------------------------------------------------------------
```

Figure 5.14 Trusted CSPs

```
------------------------------------------------------------------------
Recommendation Table
------------------------------------------------------------------------
CSP_ID  Trust Value     Returned Trust Value    Recommmended Trust
3       0.894                   0.836           0.747
5       0.86                    0.773           0.665
7       0.87                    0.881           0.766
11      0.859                   0.85            0.73
12      0.872                   0.788           0.687
16      0.862                   0.763           0.658
22      0.881                   0.885           0.78
25      0.875                   0.856           0.749

------------------------------------------------------------------------
Total number of CSPs responded : 8
```

Figure 5.15 Recommendation Table

total recommended trust is calculated as 0.754 as shown in the figure 5.16. Now, the total trust value is calculated considering the local and recommended trust values and it is found to be 0.656. Since this trust value is greater than the trust threshold (0.6), the VM request from CSP-1 is accepted by the CSP-4, even though there is a QoS violation between them (as shown in the figure 5.11).

```
-----------------------------------------------
Filtered Recommendation Table
-----------------------------------------------
CSP_ID          Recommmended Trust
3               0.747
7               0.766
11              0.73
22              0.78
25              0.749

The number of Filtered Recommendations is 5

Total Recommended Trust = 0.754
Total Trust Value       = 0.656

Total Trust value > Trust Threshold (0.6)
VM Request from CSP-1 is accepted by CSP-4
```

Figure 5.16 Filtered Recommendation Table

5.6.4 Results and Analysis

In order to test and validate the proposed approach in the cloud federation environment, we have implemented the cloud federation of 25 CSPs using the CloudSim toolkit (Calheiros et al. 2011). Sample database is created and used as the database for testing our algorithm. We have considered the resource request in such a way that there is chance for QoS violation between the CSPs so that the proposed approach can be used to deal with the dynamic QoS violations. The figure 5.17 shows the number of resource requests of CSP-1 accepted/rejected in the cloud federation environment. Here, the X-axis shows the type of trust used for managing the QoS violations. At the same time, the Y-axis shows the number of resource requests accepted or rejected. That is, the Y-axis shows the number of resource requests accepted considering the local trust alone, and the local and recommended trust together, and it also shows the number of resource requests rejected when the calculated trust value is less than the trust threshold. The figure shows three cases: the first one indicates the total number of times the resource requests of CSP-1 is accepted considering only the local trust of CSP-1. Second case shows the total number of times the local trust value of CSP-1 alone was not sufficient, and hence the CSPs had to calculate the recommended trust of CSP-1, and the total trust was sufficient to accept the resource requests of CSP-1. The third case shows the total number of times the resource requests of

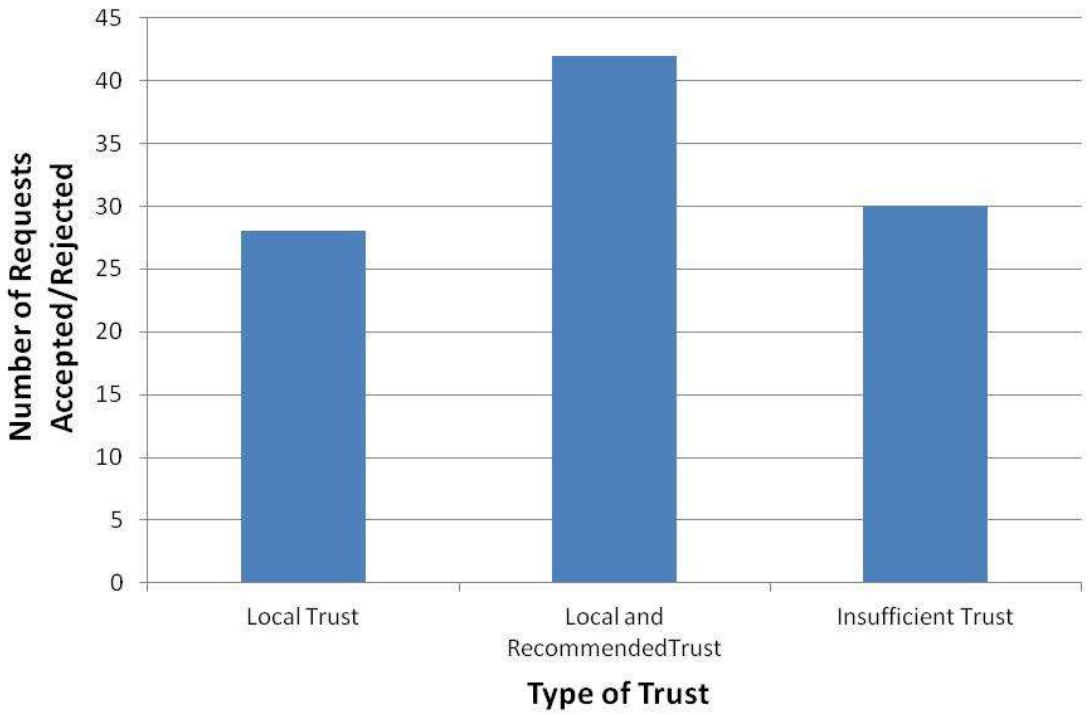

Figure 5.17 Analysis of the Accepted Requests in the Federation

CSP-1 were rejected as the total trust value was less than the trust threshold maintained in the system. As shown in the figure, out of 100 requests from CSP-1, the service requests were accepted 28 times using local trust, and 42 times using recommended trust of the CSP-1. 'Insufficient Trust' means the case when the service request is rejected even with the recommended trust. Hence in our simulation, 30 times the requests got rejected due to insufficient trust values. From the figure, it is seen that reputation of the CSP plays an important role in the cloud federation. As compared to local trust, recommended trust also plays an important role in solving the dynamic QoS violations, and thereby accepting the resource requests from CSP-1.

The figure 5.18 shows the average time taken for the service decision by a CSP when it gets the resource requests from CSP-1 in the federation, and also when there is a QoS violation between the CSPs. The figure shows the average time taken in two cases of service decision, considering 100 service requests. The first one shows the average time taken considering only the local trust of the requesting CSP-1. The second case shows the average time taken for the service decision, considering the local and the recommended trust values of CSP-1. As shown in the figure, the average time taken for the service

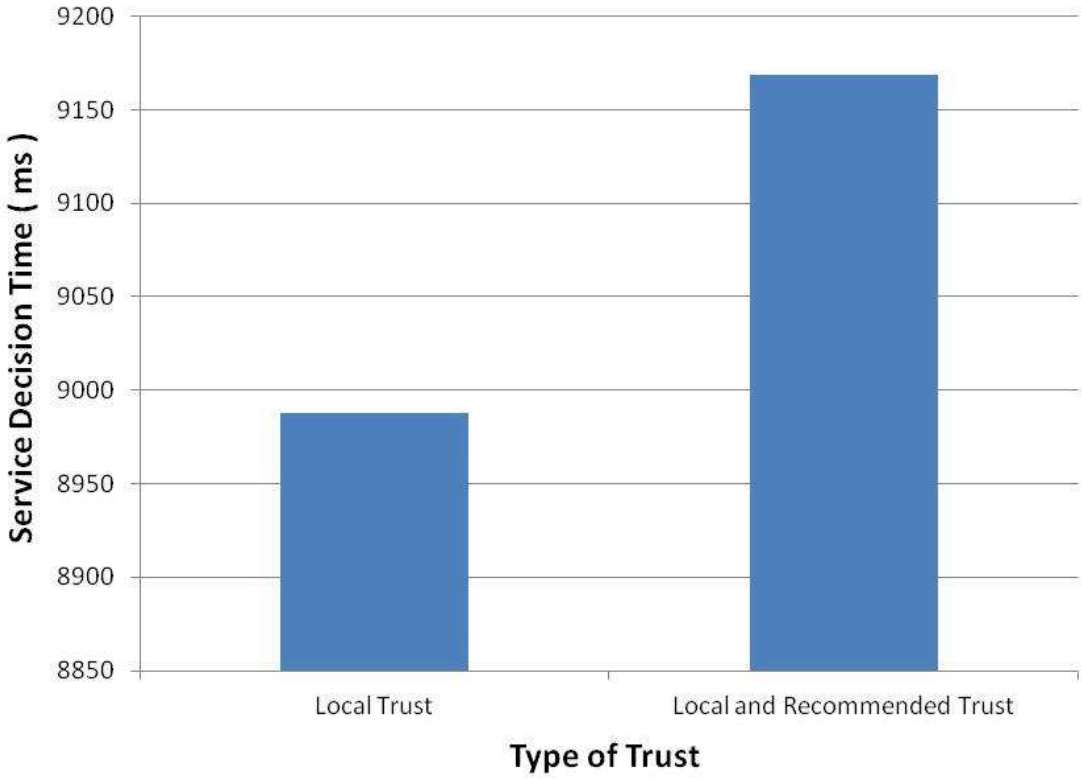

Figure 5.18 Analysis of the Service Decision Time

decision using local trust is 8989 ms and using the recommended trust is 9169 ms. Even though, calculation of the recommended trust takes longer compared to the calculation of the local trust alone, the performance of the cloud federation is improved in such a way that more user requests are satisfied. Hence, the economic benefits and the reputation of the partner CSPs in the cloud federation are improved.

In the real time cloud federation environments, SLA renegotiation between two parties (CSPs) involves the following steps. The user (CSP-A in the cloud federation) submits the resource request specifying the QoS parameters required or to be changed, to other CSP (CSP-B) in the federation. The CSP-B then proposes the initial offer based on its current availability and service features to fulfil the service request submitted by CSP-A. On receiving the initial offer, the requesting CSP-A can prepare the counter offer (if needed) which is sent to the CSP-B. The CSP-B then evaluates the counter offer (proposal). If the counter offer cannot be accepted, then that CSP proposes another counter offer. Finally, the negotiation or renegotiation process is terminated by the CSPs upon reaching mu-

tual agreements regarding the services and QoS, or when there are no mutual agreements reached between the parties. If mutual agreements are reached, then the SLA is created using the templates, and it is signed by the parties. Thus, it becomes the modified SLA after the renegotiation process. Since the renegotiation involves several steps or processes as explained, it is expected to take longer than the time taken to calculate the trust values of a CSP in the federation. Hence the time taken in our prototype simulation is considered to be better.

5.6.5 Pros and Cons of the Approach

The major advantage of the proposed approach of partner selection in the cloud federation environment is that it optimizes the search for partners in the cloud federation environment. When a CSP in the federation is running out of resources, this mechanism helps the CSP to identify the suitable partner for offloading the resource requests of the clients. The SSO approach implemented in the cloud federation is both secure and efficient as we have used AES-256 algorithm and the FHMQV protocol. The proposed trust based approach helps to solve the dynamic QoS violations in the cloud federation environment without requiring the SLA renegotiation at run time. Thus, the approach improves the performance, responsiveness, efficiency of the CSPs, and thereby the reputation and profits of the CSPs in the cloud federation environment. In the proposed trust-based approach, we consider the trusted CSPs of any CSP to get the recommendation of any other CSP in the federation. Here, we have assumed that the trusted CSPs of any CSP in the federation have a good transaction history with the specific CSP. Also, we have used the specified filtering algorithm to remove any outlier(s) among the recommended values. Here, unlike the stand-alone CSPs, the cloud federation is existing in a cooperative and mutually-benefiting manner, and hence, it is assumed that majority of the trusted CSPs of any CSP won't behave maliciously. Thus, our approach helps to meet the clients' requirements of a CSP during emergency situations ensuring timely and efficient service to the clients without requiring the SLA renegotiation in the cloud federation.

Thus, by calculating the local and the recommended trust values of the requesting CSP, the CSP takes a decision as to whether the resource request from that CSP can be accepted or not even when there is a QoS/SLA violation between the CSPs. As far as we know, this

is the first work that employs the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. Since there are no similar works available that deals with the management of dynamic QoS violations in the cloud federation domain, we were not able to compare our approach with other approaches.

5.7 Summary

In this chapter, we have implemented the trust-based approach for the management of dynamic QoS violations in the cloud federation environment. We have also implemented the partner selection approach for a CSP using the AHP and the TOPSIS methods, when it does not have enough resources to meet the resource requirements of its users. Also, we have implemented the SSO approach in the cloud federation environment using the AES-256 algorithm and the FHMQV protocol. In this work, various parameters are identified for calculating the trust values of CSPs in the cloud federation environment. Trust Decay Factor of a CSP also is considered and the proposed trust based approach shows that by calculating the local trust and the recommended trust values of the CSPs, the dynamic QoS violations can be effectively solved. The proposed approach was validated using the CloudSim toolkit. The analysis of the results obtained shows the effectiveness of the proposed approach in improving the efficiency, responsiveness, reputation and economic benefits of the CSPs in the cloud federation environment.

5.8 Topics Covered in Next Chapter

The next chapter discusses the trust-based approach for the management of break-glass access in the cloud federation environment. It shows how the legitimacy of the break-glass access request can be effectively determined by calculating the risk value of the access request, and also the trust value of the users in the cloud federation environment. The proposed approach is implemented using the CloudSim toolkit, and the analysis of the results is provided.

Chapter 6

Trust and Risk-Based Break-Glass Access Management in the Cloud Federation

This chapter presents the proposed trust and risk-based approach for the management of dynamic break-glass access in the cloud federation environment. PHR management and the overall flow of the proposed approach are discussed. The details of the risk calculation and the local and recommended trust calculation of the requesting user are presented. Experimental set up, results and the analysis are discussed to draw the inferences. Pros and cons of the approach, summary and the pointer to the topics covered in the next chapter are also given towards the end of the chapter.

Electronic Medical Records controlled and managed by the patients are known as Personal Health Records (PHRs). By utilizing the cloud based health care applications; the various users such as patients, doctors, nurses, other medical professionals etc. can access the medical data of the patients anytime, anywhere. Since the PHR data are treated as highly sensitive, proper access control mechanisms need to be enforced in dealing with access requests involving PHR data, in order to permit only the authorized users to access the data. With the emergence of cloud and the cloud federation paradigms, the PHR service providers find it effective to shift their applications and storage to the cloud, in order to reduce the operational cost. By using the multi-cloud based health care services, the quality of the health care given to patients can be improved while reducing the overall health care cost. Cloud mashups in the health care domain combine different services from multiple cloud providers into a single service or application. This service composition helps the CSPs offer more efficient services and functionalities to clients at lower costs.

The medical records of the patients contain private and sensitive information. Generally, these data cannot be accessed by all medical professionals in a hospital other than the consulting doctor or the ones explicitly permitted by the PHR owner. During emergency situations, availability of the healthcare data is more important than confidentiality, and hence relevant medical data should be made available to the concerned people irrespective of the employed access control model. But, in the case of an emergency, in order to save a patient's life, a PHR user such as a nurse needs to be permitted to access the PHR data of the patient, if the consulting doctor is unavailable at that time. If the access control system does not have this required break-glass mechanism, in this case, either the nurse may not be able to perform the emergency service, or a doctor's access rights may be given to the nurse, which may result in some misuse. In an emergency situation of a patient, he may not be able to give access rights to a requesting user after deciding his legitimacy to access his data. Also, practically it is not possible for a patient to predict and plan in advance which specific person(s) will request his PHR data during emergencies.

The researchers have been working in the area of PHR management in the cloud environment. Considering the various approaches proposed by the researchers (details are given in the section 2.4), it is seen that, how to identify the legitimate access request is an issue to be solved in the multi-cloud based healthcare domain. Considering the importance of the health care management using services from multiple cloud service providers, there should be an effective mechanism to deal with the emergency access requests or break-glass access requests from the PHR users in such a domain. Hence, we discuss the proposed mechanism for the management of dynamic break-glass access requests in the cloud federation environment in this chapter.

In this work, we are proposing a trust and risk-based framework for finding the legitimacy of the emergency access requests in the cloud federation environment. The proposed approach shows that by calculating the dynamic trust of the requesting user in the federation, break-glass access requests can be effectively managed. The proposed approach calculates the risk value of the access request made, and then the local and the recommended trust values of the requesting user. Then, based on the calculated trust value, the proposed trust-based approach helps to effectively decide whether the emergency access

request should be permitted or not in the cloud federation environment. As far as we know, this is the first work that employs the trust and risk-based mechanism for finding the legitimacy of the emergency access requests in the cloud federation environment.

The example scenario considered in our work is that of a PHR service provider who uses multiple cloud services from different CSPs to create a single application or service. By this combination of services, the PHR service provider is able to provide better and more efficient services to its clients such as hospitals or health care providers. We also assume that the different CSPs offering services are part of a cloud federation. The proposed mechanism calculates the risk involved in the access request and takes a suitable access decision by calculating the trust value of the user. The workflow of the proposed approach is discussed. We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results is also given. The analysis shows that the proposed approach is efficient in dealing with the break-glass access requests in the cloud federation environment.

6.1 PHR Management in the Cloud Federation

The context discussed in this chapter is that of a cloud federation based healthcare scenario as shown in the figure 6.1. In this case, it is assumed that the health care provider, such as hospitals aggregate the services from more than one CSP, and the combined services or the application is used by the PHR owners for the storage and processing of their health data. Also, it is assumed that the CSPs whose services are aggregated are part of a cloud federation. Now, the combined service is accessed by different users such as doctors, nurses, lab staff etc. As shown in the figure, three CSPs (Cloud Service Provider-A, B and C) are part of the cloud federation and the health care services such as Cloud Service-1 and Cloud Service-2 from the CSP-B and CSP-C respectively are combined and used by the Health Care Provider. This combined service is then used by the various PHR users (User-1, User-2, ..., User-N) such as doctors, nurses, lab staff etc. as shown in the figure.

Our proposed access control mechanism needs to be implemented and enforced by the CSPs in the federation offering different services to the PHR service provider who combines the various services, and offers the combined services to different PHR users. The PHR owners have to use the required encryption mechanism to ensure the fine-grained

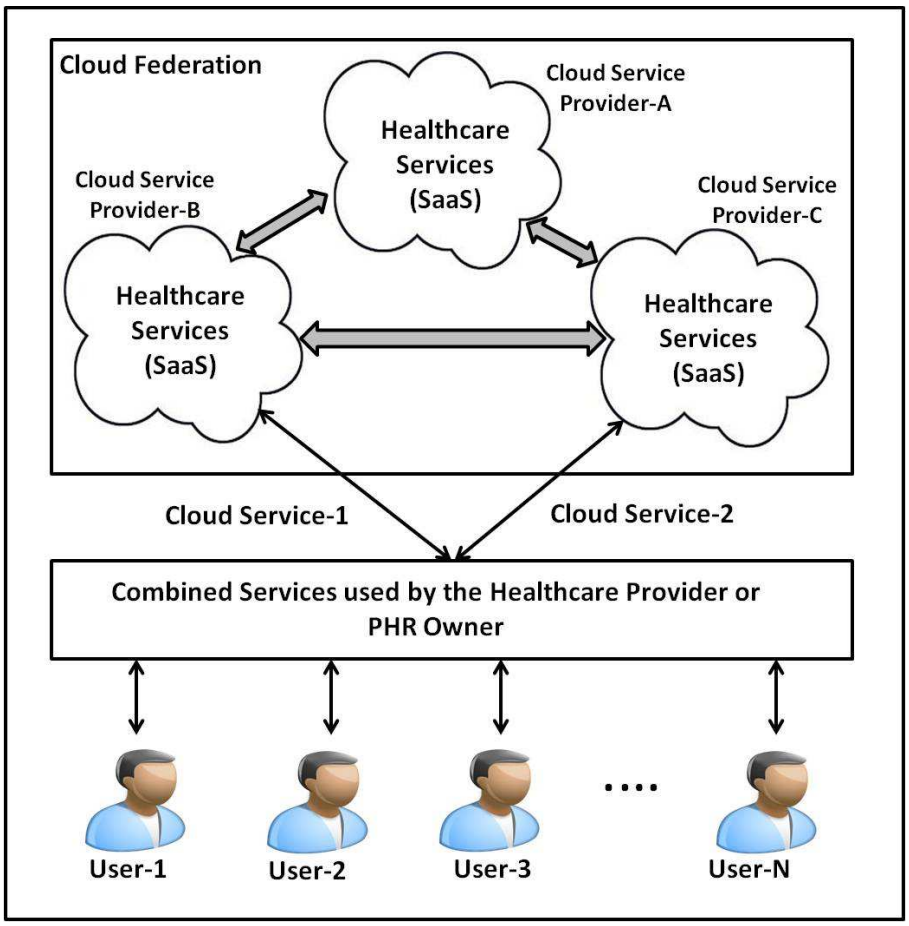

Figure 6.1 PHR Management in the Cloud Federation

access control of their personal health data stored in the cloud servers. In our proposed approach, we do not deal with the encryption mechanisms to be used for protecting the health data of the patients. Our proposed approach deals with emergency access requests and takes a decision as to whether the request should be permitted or not considering the various parameters. The concept of 'emergency staff' is not used in this work; and the legitimacy of the access requests of different medical staff are analysed dynamically considering the risk value of the access request made, and also the trust value of the user requesting the break-glass access.

6.1.1 Risk Management in the Access Control

In traditional access control models, every access request is evaluated based on the pre-established policies. In dynamic access control systems, every access request is analyzed dynamically, considering not only the security policies, but the context, attributes of the

entities and also the risks involved in granting the access request. The risk of allowing a process in a system is defined as the potential damage that can happen due to that process, and it is calculated as the product of the probability of occurrence of an undesired event and its impact on the system (Diep et al. 2007). Before taking an access decision, risk-based access control systems conduct a risk analysis of the access requests made, and a numeric value is assigned to the risk. Then, depending on the risk threshold maintained in the system, access is either permitted or rejected.

6.2 Access Control Framework

The overview of the proposed access control framework with the break-glass management as implemented in our work is shown in the figure 6.2. In this model, a PHR user makes

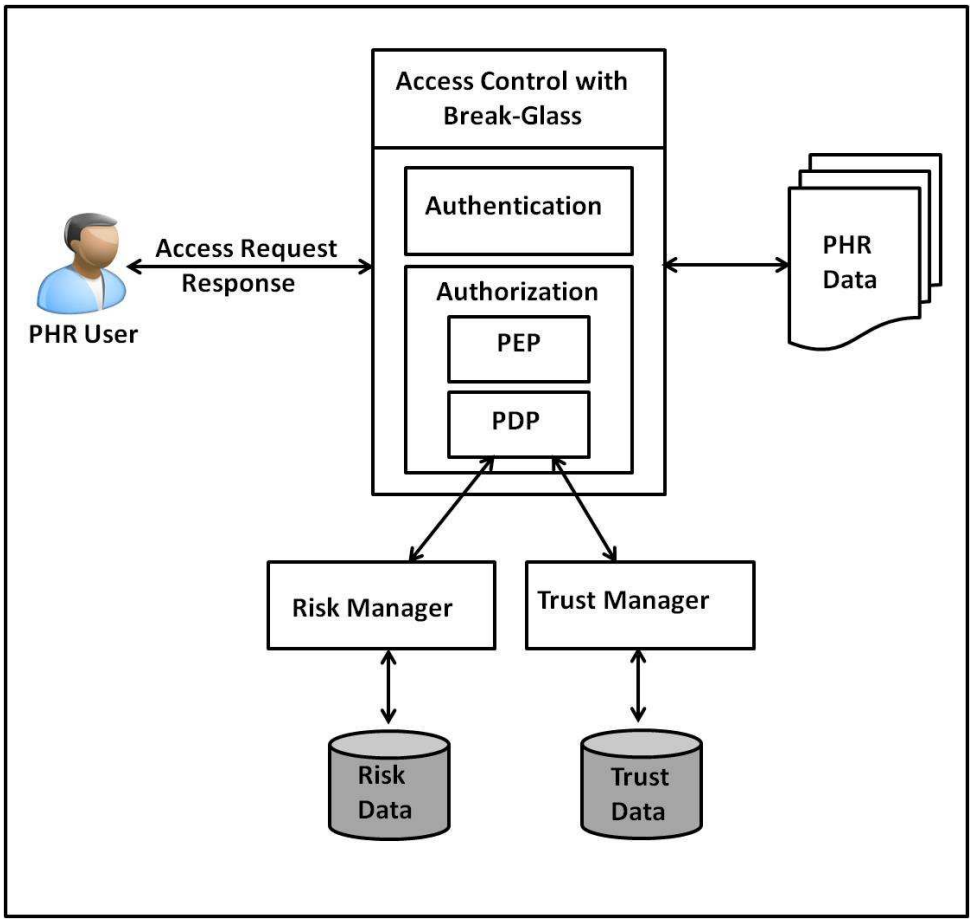

Figure 6.2 Overview of the Access Control Framework with Break-Glass Mechanism

the request to access the PHR data stored with any of the CSPs in the federation, and this

access request is handled by the module, access control with the break-glass mechanism, as shown in the figure. The various functional components in this framework are:

6.2.1 Authentication

Before allowing a break-glass access to the PHR data of patients, the user who wants to access the data should be authenticated. Every user has a user name-password pair which is encrypted using AES-256 algorithm and this is used for the authentication of the corresponding user.

6.2.2 Authorization

This component verifies the access rights of the requesting user and takes a decision as to whether the access request should be permitted or not. This component has two modules, PEP (Policy Enforcement Point) and PDP (Policy Decision Point). The PEP contacts the PDP for access decision, and implements the access decision taken by the PDP. Whenever a user makes a break-glass access request, this component verifies the request and takes a decision as to whether the request need to be permitted or not. This component contacts the Risk Manager module for calculating the risk value of allowing the access request. It also contacts the Trust Manager module for calculating the trust value of the requesting user in the cloud federation environment.

6.3 Proposed Approach for the Management of Break-Glass Access Requests in the Cloud Federation

The various functional components in the proposed approach for dealing with the break-glass access requests are shown in the figure 6.3. Thus, the various components in the implementation are:

6.3.1 User Authentication

This component verifies the identity of the person trying to access the health data of the patient. In the proposed approach, the health care service provider uses encryption mechanism to store the passwords of the PHR users. The identity details of the PHR users are stored with the CSP hosting the application or service. The passwords of the PHR users are

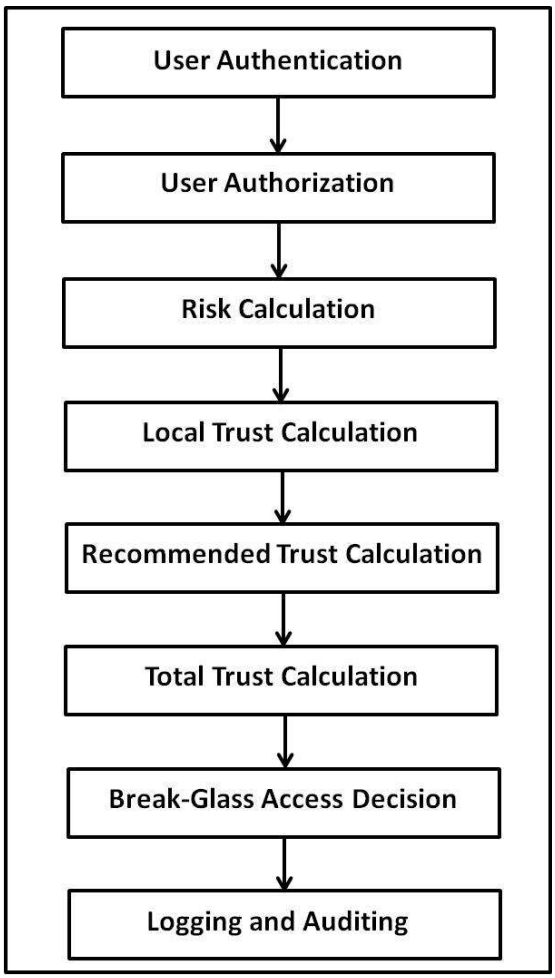

Figure 6.3 Proposed Approach for the Break-Glass Mechanism

encrypted using the Symmetric Encryption scheme, AES-256, and this serves the purpose of securing the stored passwords against various attacks.

6.3.2 Authorization

This component verifies the access rights of the user trying to access the patient's information. In our implementation, a user requesting emergency access to a patient's data is allowed to do so in two cases. In the first case, a user is allowed access when he has the break-glass access privilege granted explicitly by the PHR owner. This may be the case such as the user trying to access the patient's health information is the consulting doctor of the patient or even a medical professional allowed by the PHR owner. In the second case, the emergency access requests are handled by the proposed trust-based approach, and a suitable access decision is taken accordingly considering the various parameters.

In order to implement the first case, we have considered a parameter 'Degree-of-Bias' in our implementation which can be set to any value of -1, 1 or 0 by the PHR owner. For a particular user and a health file, the value of -1 for the 'Degree-of-Bias' parameter means that the PHR owner has not given permission to the specific PHR user for accessing the health file of the patient under any circumstances including emergency situations. A value of 1 for the parameter indicates that the specific PHR user is allowed to access the health data of the patient under all circumstances. A value of 0 for the parameter means that the PHR owner has not given any explicit preferences for the user; and hence when an access request is made by the user, appropriate decision is taken by the mechanism proposed in this chapter. If the requesting user is not permitted to access the patients' data according to the normal access policies of the system, the proposed approach takes a suitable decision as to whether the access request should be permitted or not considering the risk and the trust values associated with the request.

6.3.3 Calculation of Risk and Trust

Before taking a decision on whether to allow or deny the break-glass access request of a user, firstly, the risk value of the access request is calculated. Based on the risk value, the trust-threshold is selected. Then, the local and the recommended trust values of the user are calculated. Based on the total trust value of the requesting user, the access request is either accepted or rejected. The details of the calculation of risk and trust values are given in the following sections.

6.3.4 Logging and Auditing

In our implementation, before the access request is permitted, a warning message will be sent to the user requesting the break-glass access. After every permitted break-glass access, the details of the access request such as the User-ID, date and time of access, and the specific resource accessed etc. are entered into a log file. Also, auditing is used to verify whether the break-glass access was genuine or not. The auditing of the permitted break-glass access can be performed either by contacting the consulting doctor of the patient or the administrator of the combined multi-cloud health care application. The result of the audit is recorded in the corresponding table in the database, which could be used for taking

the break-glass access decision for the same user in the future. In our work, we assume that a break-glass access can be practically audited within 24 hours. Since we are taking the access decision considering the current trust value of the user, a user will not be allowed to have break-glass access multiple times in a row, if the previous break glass access is not audited by the system. If a user performs not-genuine break-glass access, his trust will get reduced and finally, he will not have enough trust to make the break-glass access in the future. Also, certain administrative actions such as terminating the user can be taken if the number of not-genuine break-glass access exceeds a limit.

6.3.5 Trust Update of the User

As already mentioned, auditing is performed by the CSP after every break-glass access made by the users. Hence, after a break-glass access and the subsequent auditing, if it is found that the access was genuine, the trust value of the user who performed the break-glass access is updated as:
New Trust=Old Trust+[(x/y)*Old Trust]/10, where x is the number of genuine break-glass access made by the user, and y is the number of total break-glass access permitted by the user.

Also, after a break-glass access and the subsequent auditing, if it is found that the access was not genuine, the trust value of the user who performed the break-glass access is updated as:
New Trust=Old Trust-[(a/y)*Old Trust], where a is the number of not-genuine break-glass access made by the user, and y is the number of total break-glass access permitted by the user.

6.4 Risk Calculation

In order to calculate the risk value of an access request, the following three factors are considered in our work.

6.4.1 The impact factor (Factor-1)

This factor calculates the impact of allowing the access request made by the user in terms of possible violation of confidentiality, integrity and availability of the data. In our im-

plementation, in order to show the working of the proposed approach, we have considered four access operations such as read, write or update, download and delete operations. Thus, the impact factor is calculated as: $\frac{\sum_{i=1}^{3} P_i * I_i}{3}$, where P_i (i=1, 2, 3) corresponds to the probability of having a violation of the Confidentiality (C), Integrity (I) and the Availability (A) respectively and I_i (i=1, 2, 3) takes value of 1 if there is a violation of C, I or A because of the access operation, and I_i takes a value of 0 if there is no violation of C, I or A respectively. The initial probability for an access request to violate C, I and A of the PHR data is considered equal and taken as 0.33

6.4.2 The sensitivity factor (Factor-2)

This factor calculates the sensitivity of the data being accessed. In our work, we have considered five types of reports of patients containing details such as their basic information, allergy details, X-ray, lab and scan reports. Depending on the sensitivity of the information contained in these reports, each of these reports is assigned a sensitivity value ranging from 1 (lowest) to 5 (highest). So, whenever an access request is received for a report of a patient, the sensitivity factor is calculated as:

Sensitivity = (Assigned Sensitivity / Maximum Sensitivity)

The sensitivity value of a PHR file is assigned by the consulting doctor. When he uses the cloud service to upload or store a specific PHR file of a patient, the sensitivity value is assigned in the specified range, because only the consulting doctor knows how sensitive the information contained in the file is.

6.4.3 The probability of malicious access factor (Factor-3)

This factor shows the probability of the current access request being malicious. Access is deemed malicious, when it was not used for the intended purpose. Here, we consider auditing after each break-glass access. After auditing, if it was found that there was no need of the break-glass access made by that PHR user for the specified PHR file, the access is deemed malicious. The proposed system, after auditing, records all the malicious accesses to the data being protected. Thus, the total number of malicious accesses made to the data of a particular patient by a specific user is available. Hence, based on the past history, the probability of malicious access to a particular data is calculated as:

The probability of malicious access=(Total number of Malicious Access/Total number of Access Permitted)

Hence, the value of the Total Risk of the current access request is calculated as:

Total Risk= (Factor-1 + Factor-2 + Factor-3)/3.

Now, depending on the value of the total risk calculated corresponding to an access request, the current trust threshold is dynamically selected by the CSP in the federation.

6.5 Local Trust Calculation of the User

The calculation of local trust of the user requesting break-glass access involves the following parameters:

6.5.1 Probability of Success

This parameter considers the number of successful break-glass access made by the user in the past. In our implementation, every permitted break-glass access is recorded, and the process of auditing can be used to verify the break-glass access as successful or not. Thus, success is assessed through the audit mechanism. After every break-glass access, an auditing process is carried out to verify if there was a need for the specific PHR user to access the specified PHR file. If there was a need, the access is termed as successful break-glass access or genuine break-glass access. After auditing, if it is found that there was no need, then the access is termed as unsuccessful break-glass access or not-genuine break-glass access. Hence, this parameter is calculated as: Probability of Success=(x/z), where x is the total number of successful break-glass access permitted and z is the total number of break-glass access requested by that specific user.

6.5.2 Degree of Association

For calculating the local trust of the user, the total period of association of the user showing how long he has been associating with the hospital or healthcare provider is taken into account by considering the date and time of joining of the user (Doctor, Nurse, Lab Technician etc.) with the hospital or the health care centre. Based on the date and time of joining the federation, the Degree of Association is given a value x for the user, where $x \in [0, 1]$. In our simulation, the parameter x takes values 0, .1, .2, .4, .6, .8 and 1 corresponding to seven

ranges of the time period such as less than one month, between 1 month and 2 months, 2 months and 4 months, 4 months and 6 months, 6 months and 9 months, 9 months and 12 months (1 year) and greater than 12 months respectively.

6.5.3 History of Interaction

This shows the lead of the number of successful break-glass accesses over the number of unsuccessful break-glass accesses made by a user in the past. It is calculated as: History of Interaction=$(x-a)/y$, where x is the total number of successful break-glass access permitted, a is the total number of unsuccessful break-glass access made and y is the total number of break-glass access permitted by the specific user.

6.5.4 Existing Trust

This shows the existing trust value of a user before the current trust value is calculated. As this factor indicates, a user with a higher existing trust value is expected to have a more positive impact on the calculation of the current trust value than a user having a lower existing trust value, or a user joined the healthcare provider recently.

6.5.5 Access Level

In our work, we have considered three categories of users such as doctors, nurses and lab technicians, and each category of users is assigned a numeric value ranging from 1 to 3 showing the access level associated with them. Thus, the category of doctors is given the access level of 3, nurses given the access level of 2, and the category of lab technicians is given the access level of 1. Hence, the Access Level factor of the requesting user is calculated as: Access Level=(Assigned Level/Maximum Level)

6.5.6 Access Right

This parameter shows the access right value of the user requesting the break-glass access. In our work, we have considered four access operations such as read, download, write and delete with respect to the PHR data of the patients, and each operation is given a numeric weightage such as read=1, download=2, write=3 and delete=4. Hence, this parameter is calculated as: Access Right=(Assigned Right/ Maximum Right)

6.5.7 Permitted Factor

When a user is making a break-glass access request, the Permitted Factor is considered to show how many break-glass access requests made by the user were permitted in the past. Hence, this parameter is calculated as: Permitted Factor=(y/z), where y is the total number of break-glass access requests permitted and, z is the total number of break-glass access requests made by the user.

6.5.8 Genuine Factor

This shows the ratio of the number of genuine break-glass access made by a user to the total number of break-glass access permitted by the user in the past. Hence, this is calculated as: Genuine Factor=(x/y), where x is the total number of genuine break-glass accesses made, and y is the total number of break-glass access requests permitted by the specific user.

Hence, the local trust value of the requesting user is calculated as:

Trust Value=(Probability of Success + Degree of Association + History of Interaction + Existing Trust + Access Level + Access Right + Permitted Factor + Genuine Factor)/8

6.5.9 Trust Decay Factor of the User

In the cloud federation domain, the trust value of a user is considered to be dynamic and the calculated trust value decays over time. Hence, we have considered the Trust Decay Factor while calculating the trust value of the requesting user in the federation. This decay factor is selected depending on when the requesting user had the last transaction with the CSPs in the federation. The decay factor is adjusted in such a way that the trust value gets decremented more when the date and time of the last transaction of the user with a CSP becomes older. In our implementation, this decay factor is represented as $1/x$, where $x \in [1, 2]$, depending on the date and time of the last transaction.

Hence, the final value of the Local Trust of the requesting user is calculated as:

Local Trust Value=Trust Value X Trust Decay Factor

We have selected the decay factor as $1/x$ to show the variation in the trust value of a user, where x depends on the time elapsed since the last transaction of the requesting user with any CSP in the federation. In our prototype simulation, the parameter x takes

values 1.1, 1.2, 1.4, 1.6, 1.8 and 2 corresponding to six ranges of the elapsed time since the last transaction, such as less than one month, 1-3 month(s), 3-6 months, 6-9 months, 9-12 months and greater than one year respectively. In real time implementation, the parameter x is also CSP-specific. Practically, different CSPs can use different values for x for the same time period. It also depends on how long the cloud federation has been in existence, and also how long the requesting user has been a member of this federation (health care provider). Accordingly, a CSP in the federation can decide the value of x.

6.6 Recommended Trust Calculation

In order to calculate the recommended trust of the requesting user, the trusted CSPs in the federation are identified. Then, the feedback of the requesting user is collected from the trusted CSPs.

6.6.1 Selection of Trusted CSPs

When a CSP gets an emergency access request, the CSP calculates the trust value of other CSPs in the federation to identify the trusted CSPs, and from these trusted CSPs, the feedback of the requesting user is collected. In order to select the trusted CSPs, a CSP considers the parameters such as Probability of Success, History of Interaction, Existing Trust, Degree of Association and QoS Value.

The Probability of Success of a CSP with any other CSP in the federation is calculated as: Probability of Success=(v/w), where v is the total number of successful transactions between the CSPs and w is the total number of transactions initiated between them. History of Interaction shows the lead of the number of successful transactions over the number of unsuccessful transactions with a particular CSP. It is calculated as: History of Interaction=$(v-u)/w$, where v is the total number of successful transactions, u is the total number of unsuccessful transactions and w is the total number of transactions by a specific CSP with another CSP in the federation. Existing Trust shows the existing trust value of a CSP towards another CSP in the federation, before the current trust value is calculated. As this factor indicates, a CSP with a higher existing trust value is expected to have a more positive impact on the calculation of the current trust value than a CSP having a lower existing trust value, or a CSP joined the federation recently. For calculating the trust of a

CSP in the federation, the total period of association of the CSP with the federation also is taken into account, by considering the date and time of joining of the CSP with the cloud federation. Based on the date and time of joining the federation, the Degree of Association is given a value x for the CSP, where $x \in [0, 1]$.

While calculating the trust value of a CSP in the federation, the QoS parameters are considered separately to distinguish one CSP from another in the federation. It shows the details of the previous interaction of a CSP with another CSP in the past. Hence, this calculation involves the following factors:

Availability Factor: Availability Factor is calculated as (p/w), where p is the total number of times the service from a specific CSP was available when requested and, w is the total number of service requests made to that CSP.

Reliability Factor: Reliability Factor is calculated as (q/p), where q is the total number of times the service was reliable and, p is the total number of times the service was available from that CSP.

Confidentiality Factor: Confidentiality Factor is calculated as (r/p), where r is the total number of times the confidentiality was intact with the service from a CSP and, p is the total number of times service was available from that CSP.

Integrity Factor: Integrity Factor is calculated as (s/p), where s is the total number of times the integrity was intact with the service from the CSP and, p is the total number of times service was available from that CSP.

Response Time Factor: Response Time Factor is calculated as (t/p), where t is the total number of times the response time was within the promised limit and, p is the total number of times service was available from that CSP.

Hence, the final QoS Value corresponding to a CSP in the federation is calculated as:
QoS Value=(Availability Factor + Reliability Factor + Confidentiality Factor + Integrity Factor + Response Time Factor)/5

Thus, the Trust Value of the CSP is calculated as:
Trust Value= (Probability of Success + History of Interaction + Degree of Association + Existing Trust + QoS Value) /5

While calculating the QoS values, we have considered five factors such as availability,

reliability, confidentiality, integrity and response time factors. In real time implementation, the weightage given for each parameter may be different from one CSP to another depending on their business objectives. Also, it depends on the type of service a CSP offers to its users. For example, a CSP may require some application with very high response time, another CSP might require a service with a high degree of confidentiality, and a third one might require a service with a high degree of availability etc. Hence, by considering these factors, the various parameters can be given suitable weights by a CSP in the cloud federation. In our prototype simulation, just to show the working of our approach, we have given equal weights to all the parameters. In real time cloud federation environment, it will vary from one CSP to another.

6.6.2 Trust Decay Factor of the CSP

In the cloud federation domain, the trust value of a CSP is considered to be dynamic and the calculated trust value decays over time. This decay factor is selected depending on when a CSP had the last transaction with any other CSP in the federation. In our implementation, this decay factor is represented as $1/x$, where $x \in [1, 2]$, depending on the date and time of the last transaction.

Hence, the Total Trust Value of a CSP in the federation is calculated as:
Total Trust Value=Trust Value X Trust Decay Factor

After calculating the trust values of all the possible CSPs, those CSPs with trust value greater than a specific threshold are selected into the list of trusted CSPs.

6.6.3 Recommended Trust Calculation of the User

After selecting the trusted CSPs in the federation, a CSP contacts the CSPs in the list of trusted CSPs regarding the feedback of the requesting user. The CSP contacts those CSPs (m out of n CSPs, where m is the number of trusted CSPs, n is the total number of CSPs in the federation and $m <= n$) and each of the m CSPs calculates its current trust value of the user specified, and communicates that trust value to the CSP that asked for it. The CSP then aggregates the trust values collected from the trusted CSPs to calculate the final recommended trust of the requesting user in the federation. After calculating the final recommended trust value, the CSP calculates the total trust value of the requesting user as:

Total Trust Value =(Local Trust + Recommended Trust)/2

Based on this total trust value of the requesting user, the CSP decides to either accept or reject the break-glass access request from that user.

In our simulation, total trust value of the user is calculated as the average of the local trust and recommended trust values. Local trust value calculated by a CSP is based on its own experience of working with a particular user, and the recommended trust value is based on the feedback from other trusted CSPs. In our implementation, in order to calculate the recommended trust of a PHR user, initially the trusted CSPs are selected. Here also for selecting the trusted CSPs, the trust-threshold used is CSP-specific. Generally, it can be reasonably high (0.7 in our case). Then, the feedback regarding a specific user is collected from these CSPs. Hence, this recommended trust value also assumes importance in the calculation of the total trust value of the requesting user. That is the reason why we have given equal weightage to local trust and the recommended trust. Again, in the real time cloud federation implementation, a CSP can use different weights such as 0.6 for the local trust value and 0.4 for the recommended trust value. In our prototype simulation, just to show the working of the proposed mechanism, we have used equal weights (0.5) for both the local trust and the recommended trust values.

6.7 Workflow of the Proposed Approach

The workflow of the proposed approach for the management of break-glass access requests is shown in the figure 6.4. When a CSP receives the break-glass access request from a user, it carries out the authentication and the authorization processes. Authentication is performed by verifying the user name and the password of the requesting user. The identity credentials of the users are stored after encryption using Symmetric Encryption technique, AES-256 for ensuring the required security against the possible attacks. If the authentication is failed, the access request is rejected and the message is communicated to the requesting user. If the authentication of the user is successful, then the access rights of that particular user for the requested PHR data are checked to verify if the user has been given explicit access rights by the data owner himself. If the requested access rights are allowed by the PHR owner, then the access to the required PHR data is permitted. If there is no explicit authorization given by the data owner for the requesting user, then

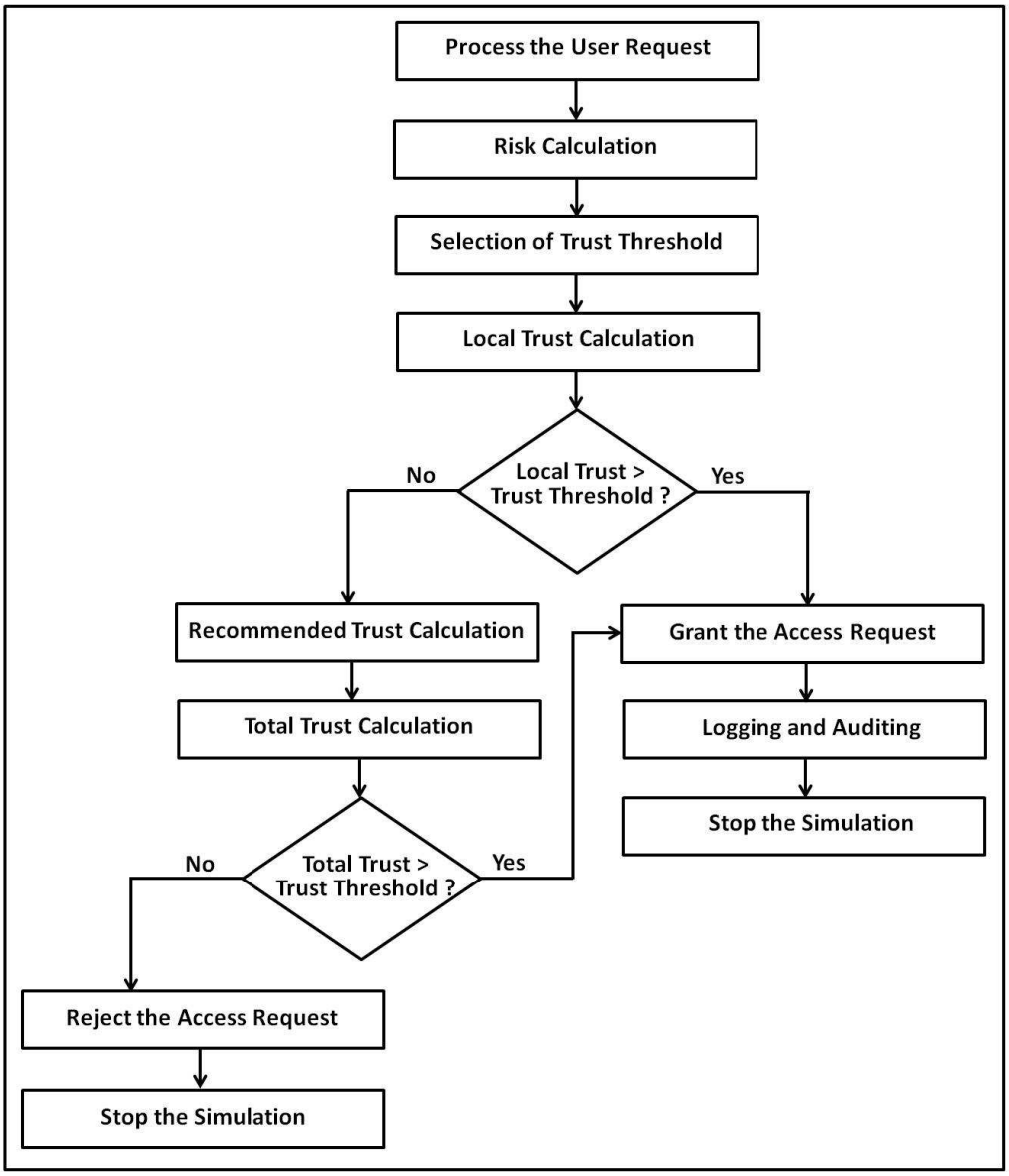

Figure 6.4 Workflow of the proposed Approach for Break-Glass Access Management

the proposed break-glass access mechanism is executed to decide whether the requested access needs to be permitted or not.

As shown in the figure, when an access request is made by a user, the risk value corresponding to that access request is calculated. Depending on the calculated risk value, the trust threshold is decided. Then, the local trust calculation of the requesting user is performed. If the local trust value of the user requesting the break-glass access is greater than the trust threshold selected, then the access request is permitted. The details of the permitted access are logged and then audited for determining whether the break-glass access

was genuine or not. Also, the database is updated accordingly to record the details of the permitted access and the modified trust value of the user who made use of the break-glass access, into the corresponding tables in the database.

If the local trust value of the requesting user is less than the trust threshold, then the recommended trust value is calculated as explained in the previous section. Then, the CSP calculates the total trust value of the user considering both the local and the recommended trust values. If the total trust value is greater than the threshold value, then the break-glass access request of the user is permitted. Then, the details of the permitted access are logged and audited for determining whether the break-glass access was genuine or not, and the database is updated accordingly to reflect the details of the access request permitted and the modified trust value of the user in the required tables in the database. If the total trust value of the user is still less than the trust threshold, the requested break-glass access request is rejected.

6.8 Experimental Results

The main objective of this experiment is to implement and test the proposed approach for dynamically managing the break-glass access requests in the cloud federation environment. Our test scenario consists of 25 CSPs in the cloud federation.

6.8.1 Experimental Setup

We have carried out the simulation experiments on a system with Intel (R) Core (TM) i7-3770, CPU 3.40 GHz, 8.00 GB RAM and 32-bit Operating System (Ubuntu 14.04). Softwares used for the implementation include CloudSim-3.0.3, Eclipse IDE version 3.8, MySQL Workbench Community (GPL) for Linux/Unix version 6.0.8 and Java version 1.7.0_55.

6.8.2 Break-Glass Mechanism in the Cloud Federation Environment

In our work, we have considered 25 CSPs in the simulated cloud federation environment. We assume that the PHR service provider combines the healthcare services from these CSPs in the federation. We also assume that there are SLAs among the CSPs in the federation who offer cloud-based health care services, to share the trust values of users among

them. In order to show the working of the proposed approach, we have considered 150 users in three categories such as doctors, nurses and lab staff who make requests to access the PHR data of 100 patients. We have considered 5 types of reports/files of patients, and 4 types of access operations such as read, download, write and delete operations. In the example shown in this chapter, we have considered the scenario when User-4 (doctor) requests 'write' operation on File-3 of Patient-1 stored at CSP-1. In this case, the PHR owner has not given explicit 'write' permission to the requested file.

Thus, when CSP-1 receives the access request, it calculates the risk value of the break-glass access request made. The figure 6.5 shows the Risk Calculation of the break-glass access request made by User-4 requesting 'write' operation on File-3 of Patient-1. In our

```
--Risk Calculation--------
Risk Factor-1 = 0.330
Risk Factor-2 = 0.600
Risk Factor-3 = 0.011
Final Risk Value = 0.314
```

Figure 6.5 Risk Calculation

work, we have considered 3 risk factors, and they are calculated as explained in the section 6.4. For this implementation, we have used the sample database and as shown in the figure 6.5, the Risk Factor-1 is calculated as 0.33, the Risk Factor-2 is calculated as 0.6 and the Risk Factor-3 is calculated as 0.011. The final risk value corresponding to the access request made is calculated as the average of the above three factors and in our case, it is 0.314.

The figure 6.6 shows the Risk-Trust Table maintained by CSP-1 where the user has

```
-- Risk Trust Table ------------
Risk Value  Trust Threshold
0.0-0.2          0.55
0.2-0.4          0.6
0.4-0.6          0.7
0.6-0.8          0.75
0.8-1.0          0.8
Selected Trust Threshold = 0.6
```

Figure 6.6 Risk-Trust Table

requested the break-glass access. This table shows the trust threshold to be selected corresponding to a particular risk score. In our case, since the calculated risk score is 0.314 (as shown in the figure 6.5), the selected trust threshold is 0.6.

The figure 6.7 shows the calculation of local trust of the user by CSP-1. The various parameters considered for the trust calculation are Probability of Success, Degree of Association, History of Interaction, Existing Trust, Access Level, Access Right, Permitted Factor and Genuine Factor, and they are calculated as explained earlier in section 6.5.

```
--Calculation of Local Trust of the User ---
1. Probability of Success = 0.848
2. Degree of Association = 1.000
3. History of Interaction = 0.728
4. Existing Trust = 0.616
5. Access Level = 1.0
6. Access Right = 0.400
7. Permitted Factor = 0.967
8. Genuine Factor = 0.876
-----------------------------------------
Trust Decay Factor = 0.714
Final Local Trust = 0.575
Local Trust < Trust Threshold
```

Figure 6.7 Calculation of Local Trust of the User

Thus, as shown in the figure 6.7, the calculated values of the various parameters are 0.848, 1.0, 0.728, 0.616, 1.0, 0.4, 0.967 and 0.876 respectively. Now, the average of all these eight factors is multiplied by the calculated Trust Decay factor (0.714) to get the final local trust value (0.575) of the user. Since the final local trust of the user is less than the trust threshold (0.6) maintained by the system, the CSP-1 calculates the recommended trust of the requesting user.

In order to calculate the recommended trust of the requesting user, CSP-1 identifies the trusted CSPs. The figure 6.8 shows the CSP-Trust table generated by CSP-1 to which the user has made the break-glass access request. This table shows the current trust values of CSP-1 towards every other CSP in the federation, along with the corresponding Trust Decay Factor. These trust values are calculated by considering the parameters Probability of Success, Degree of Association, History of Interaction, Existing Trust and the QoS values as explained in the section 6.6. From this table, all the CSPs whose calculated

```
---------------Calculation of Recommended Trust--------------
-- CSP_Trust Table -----------------------------
CSP_ID          Trust_Value  Decay_Factor  Total_Trust_Value
2               0.886           0.833          0.739
3               0.772           0.833          0.643
4               0.935           0.833          0.779
5               0.804           0.833          0.670
6               0.905           0.909          0.823
7               0.907           0.833          0.756
8               0.775           0.833          0.646
9               0.874           0.833          0.728
10              0.866           0.833          0.721
11              0.680           0.833          0.566
12              0.870           0.833          0.725
13              0.845           0.833          0.704
14              0.874           0.833          0.728
15              0.633           0.833          0.528
16              0.866           0.833          0.721
17              0.821           0.833          0.684
18              0.792           0.833          0.660
19              0.931           0.909          0.847
20              0.859           0.909          0.781
21              0.791           0.833          0.659
22              0.895           0.833          0.746
23              0.889           0.833          0.741
24              0.882           0.833          0.735
25              0.810           0.833          0.675
```

Figure 6.8 CSP-Trust Table

trust values are above the threshold (CSP_Trust_Threshold=0.7) are selected to the set of Trusted CSPs of CSP-1.

The figure 6.9 shows the Trusted CSPs of CSP-1, along with their corresponding trust values. From the figure, it is seen that the total number of CSPs in the set of trusted CSPs is 15. Now, from these trusted CSPs, CSP-1 asks for the recommendation regarding the requesting user.

The figure 6.10 shows the Recommended Trust Table of CSP-1, and as shown in the figure, the number of CSPs responded is 12. Other CSPs may not have any data to calculate the trust value of the requesting user. This table shows the trust value of every responded CSP and the corresponding trust value of the user as returned by it. Also, as shown in the figure, these two values are multiplied to get the recommended trust value of the user. The average value of all the recommended values returned by the responded CSPs is calculated

```
Trusted CSPs
------------------------------------------------------------
 CSP_ID         Trust_Value
 2              0.739
 4              0.779
 6              0.823
 7              0.756
 9              0.728
 10             0.721
 12             0.725
 13             0.704
 14             0.728
 16             0.721
 19             0.847
 20             0.781
 22             0.746
 23             0.741
 24             0.735
CSP_Trust_Threshold Value = 0.7
Total Number of CSPs above the CSP_Trust_Threshold Value:15
------------------------------------------------------------
```

Figure 6.9 Trusted CSPs

```
---Recommended_Trust Table-----

CSP_ID  Trust_Value     Returned Trust_Value    Recommended_Trust
7       0.756           0.845                   0.639
2       0.739           0.831                   0.614
9       0.728           0.843                   0.614
6       0.823           0.856                   0.705
20      0.781           0.842                   0.658
10      0.721           0.875                   0.631
4       0.779           0.850                   0.662
13      0.704           0.877                   0.617
16      0.721           0.839                   0.605
14      0.728           0.880                   0.641
12      0.725           0.855                   0.620
19      0.847           0.842                   0.713

No. of CSPs Responded : 12
Total Recommended Trust = 0.643
```

Figure 6.10 Recommended Trust Table

to get the total recommended trust value of the user. As shown in the figure, the total value of the recommended trust of the requesting user is calculated as 0.643.

Now, the average of the total local trust (0.575 as shown in the figure 6.7) and the total

recommended trust (0.643 as shown in the figure 6.10) values are taken to get the total trust value of the user. Hence, the total trust of the requesting user is calculated as 0.609. In our work, while calculating the total trust value of the user, equal weightage (0.5) is given to both the local trust and the recommended trust as the recommendations are taken from the trusted CSPs of CSP-1.

Now, since the total trust value (0.609) of the requesting user is greater than the trust threshold (0.6), the break-glass access request is permitted with a warning message as shown in the figure 6.11. In our work, the details of the break-glass access are logged into a file after encrypting with AES-256 algorithm.

```
Total Trust Value = 0.609
------------------------------------------------------------------------
 Trust Value > Trust Threshold
 You are going to access a Report Which you are not authorized to.
 This access details will be logged in and communicated to the Admin.
 Do you want to proceed?
 Y
 Access Permitted
 Time Duration = 92111
```

Figure 6.11 Access Decision

6.8.3 Results and Analysis

In order to test and validate the proposed approach in the cloud federation environment, we have implemented the cloud federation of 25 CSPs using the CloudSim toolkit (Calheiros et al. 2011). Sample database is created and used as the database for testing our algorithm. We have considered the break-glass access request of a user in such a way that there is no explicit authorization given by the PHR owner for the specific user, and hence the proposed approach is executed to take the suitable access decision.

The figure 6.12 shows the number of accepted break-glass access requests of 5 users. In this figure, the X-axis shows the user requested the break-glass access, and the Y-axis shows the number of break-glass access requests accepted considering the local and the recommended trust, and also the number of break-glass access requests rejected when the calculated trust was less than the trust threshold. In our work, we have considered 100 break-glass access requests of 5 users. The figure shows the number of accepted break-

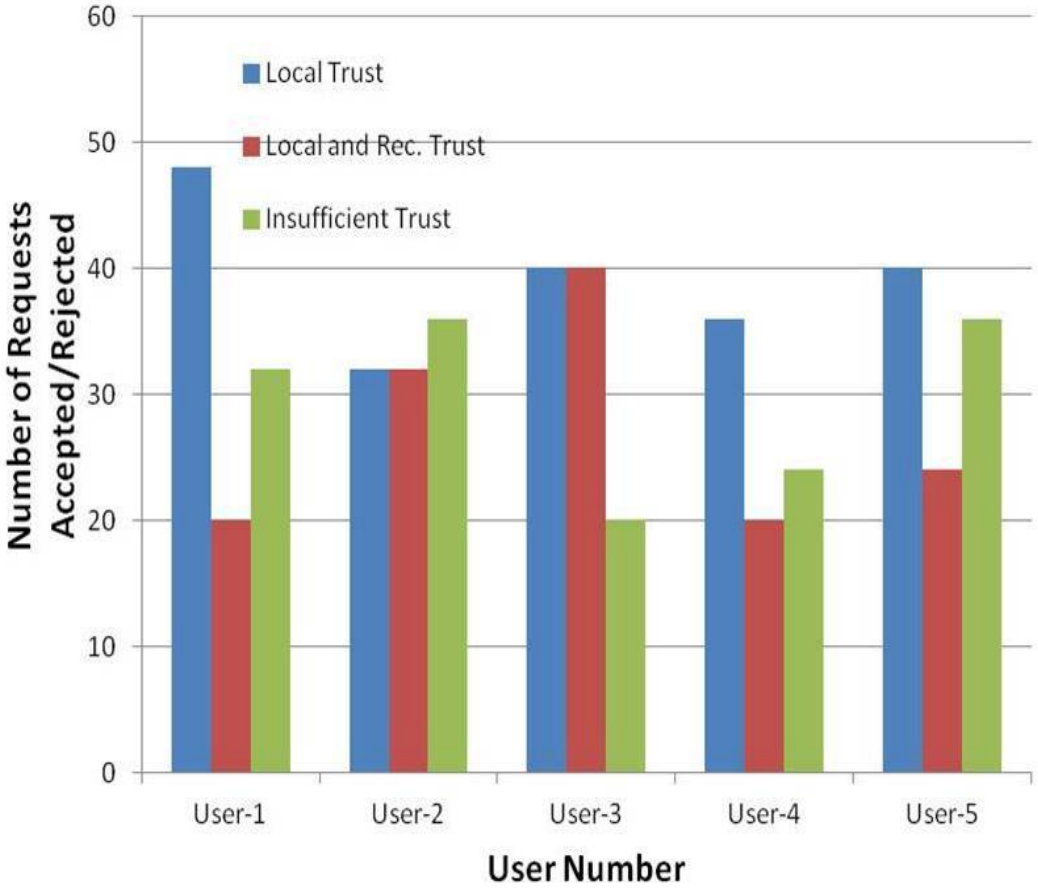

Figure 6.12 Analysis of the Accepted/Rejected Requests

glass requests of the users in three cases. The first case shows the number of requests accepted considering only the local trust of the user. The second case indicates the number of requests accepted considering the local and the recommended trust of the requesting user. The third case shows the number of requests rejected even after considering the local and the recommended trust of the user. As shown in the figure, out of 100 access requests made by User-1, 48 times the access requests were accepted using the local trust alone, and 20 times the access requests were accepted using both the local and recommended trust of the user. The figure also shows that, 32 times the break-glass access requests of User-1 were rejected due to insufficient trust even after considering the local and the recommended trust of the requesting user. As shown in the figure, the corresponding data for User-2, User-3, User-4 and User-5 are (32, 32, 36), (40, 40, 20), (36, 20, 24) and (40, 24, 36) respectively.

The results show that the total number of accepted break-glass requests of a user in-

creases with the recommendation from the trusted CSPs in the federation, and hence the overall efficiency of the cloud-based healthcare services is improved. Since we have not used any real time data in our simulation, we assume that the permitted break-glass accesses were genuine, as in real life cases, the auditing of a break-glass access proves the genuineness of the access made.

The figure 6.13 shows the average time taken for the break-glass access decision for 5 users in our simulation. The figure shows the service decision time taken for the users in two cases. The first case shows the time taken for the break-glass access decision considering only the local trust of the user. The second case shows the time taken considering both the local and the recommended trust values of the requesting user. As shown in the figure, the average time taken for the break-glass access decision considering only the local trust of the User-1 is 4317 ms and the average time taken considering the local and the recommended trust values of User-1 is 5475 ms. The corresponding data for User-2, User-3, User-4 and User-5 are (2776, 5161), (2867, 5217), (4648, 5827) and (2675, 5672) respectively. As shown in the figure, even though the calculation of the recommended trust of a user takes longer compared to the calculation of the local trust alone, the efficiency of the cloud-based healthcare services is improved as more break-glass access requests are accepted by the cloud based healthcare service.

6.9 Pros and Cons of the Approach

The major advantage of the proposed approach of break-glass management in the cloud federation environment is that it helps the CSP to take emergency access decision in the cloud federation environment. It helps a CSP in the federation to identify the PHR user requesting access to a patient's data as trustworthy or not. The proposed approach calculates the risk value of the access request made, and then the local and the recommended trust values of the requesting user. Then, based on the calculated trust value, the proposed trust based approach helps to effectively decide whether the emergency access request should be permitted or not in the cloud federation environment. Thus, the approach improves the performance, responsiveness and the efficiency of the healthcare services delivered by the CSPs in the federation. In the proposed trust-based approach, we consider the trusted CSPs

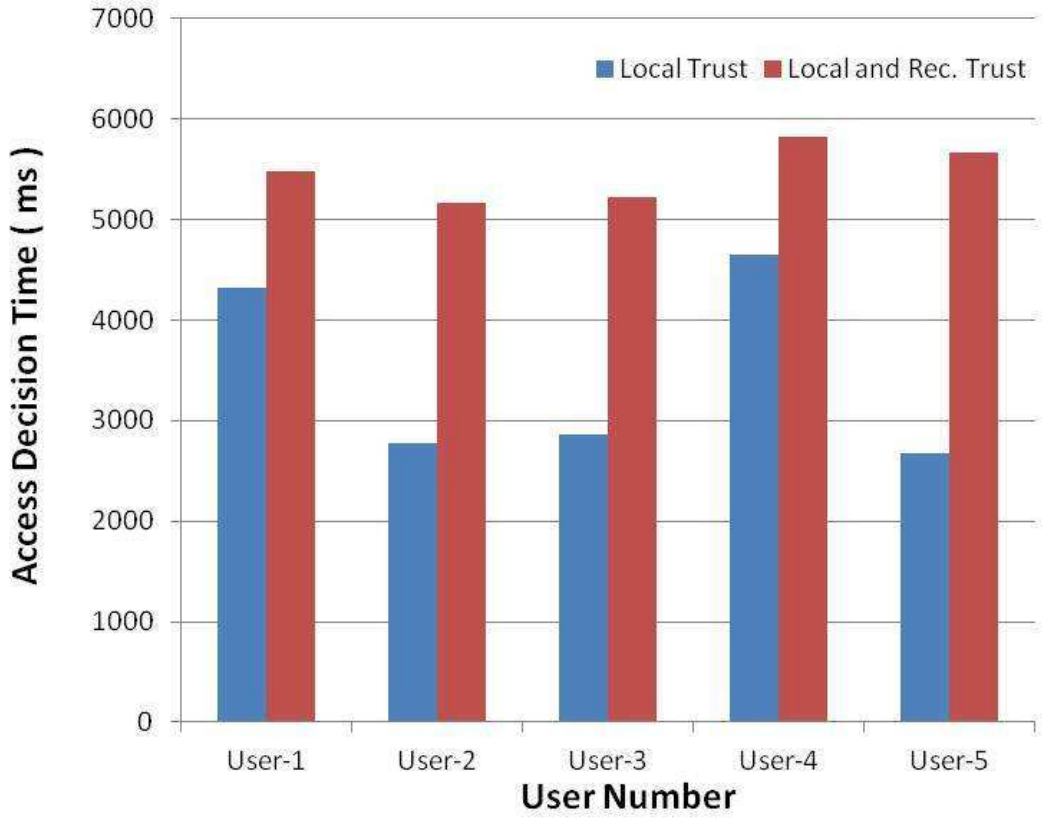

Figure 6.13 Analysis of the Access Decision Time

of any CSP to get the recommendation of a requesting user in the federation. Here, we have assumed that the specific CSP has a good transaction history with the trusted CSPs in the federation. Thus, our approach helps to take the access decisions efficiently during emergency situations, ensuring timely and efficient service to the clients. As far as we know, this is the first work that employs the trust-based approach for the management of dynamic break-glass requests in the cloud federation environment. Since there are no similar works available that deals with the management of dynamic break-glass requests in the cloud federation domain, we were not able to compare our approach with other approaches.

6.10 Summary

This chapter presents a novel trust-based approach for the management of break-glass access requests in the cloud federation environment. It shows that by calculating the dynamic trust of the requesting user in the federation, break-glass access requests can be effectively

managed. The proposed break-glass access mechanism calculates the risk value of the access request, and the local and the recommended trust values of the requesting user to evaluate the degree of trustworthiness of the user who is requesting emergency access to the personal sensitive health data of the patients. The proposed approach was validated using the CloudSim toolkit. The analysis of the obtained results shows the effectiveness of the proposed approach. Thus, considering the importance of the emerging cloud federation and the cloud-based healthcare services, the proposed approach is relevant and efficient in dealing with the emergency access requests of users where the sensitive health information has to be protected, at the same time made available during emergency situations.

6.11 Topics Covered in Next Chapter

The next chapter concludes this thesis, and also provides some future research directions in the domain of cloud federation.

Chapter 7

Conclusion & Future Directions

Cloud Federation is an emerging technology in which multiple cloud service providers collaborate to share the cloud infrastructure among them. This collaboration helps to share the information and resources among the partners in an efficient way so that availability, reliability and other QoS parameters of the services offered by the various CSPs in the federation are improved. It helps to achieve better resource utilization of the individual CSPs and thereby better revenues for them. The research in the field of cloud federation is in the nascent stage, and there are many research issues requiring effective solutions before the efficient utilization of this emerging technology. This thesis focuses on developing effective approaches for the identity and access management issues in the cloud federation environment.

In chapter 3, we have discussed the proposed Single Sign-On authentication approach in the cloud federation scenario, considering multiple identity providers and cloud service providers. Single Sign-On (SSO) is an authentication mechanism in which a service consumer is required to be authenticated only once while accessing various services from multiple service providers. The chapter discusses the three major steps in the approach proposed in this thesis such as registration of the user with the Identity Provider (IdP), registration of the user with the Cloud Service Provider (CSP) and requesting services from the CSP. We have also considered the security aspects of the data transferred between the various entities in the cloud federation during the SSO mechanism. In our implementation, for securing the data in transit such as during the transfer of the identity tokens of the cloud users between CSPs and also between a CSP and an IdP, we have used the Symmetric Key Encryption technique using Advanced Encryption Standard, AES-256. Also, we have used

Fully Hashed Menezes-Qu-Vanstone (FHMQV) key sharing protocol for key exchange between the entities in the simulation. We have simulated the proposed approach using the CloudSim toolkit, and the simulation results show that the SSO approach is highly beneficial while accessing multiple services from CSPs in the cloud federation, as it reduces the execution time of the user request for resources in the cloud federation environment. In the simulated federation set up, if a CSP alone cannot handle the access request, the access request is transferred to other CSPs in the federation. In our simulation, the maximum number of SSO associated with a single access request of a user in the federation is 20. The analysis of the results shows that the SSO approach reduces the average user response time considerably, besides providing the required security features. Also, by using the Single Sign-On authentication mechanism, it reduces the load of the cloud users and developers in dealing with multiple credentials while accessing services from various CSPs in the federation.

Because of the proposed approach for authentication, the service providers can concentrate more on their core services, since the identity management operations are taken care of by the identity providers. The cloud users are benefited in such a way that they will be able to seamlessly access the different services offered by the multiple CSPs in the federation without the need for providing the identity credentials again and again for accessing the services from different CSPs. Hence, the proposed mechanism achieves the effective identity management in the cloud federation environment while maintaining security against various attacks.

In chapter 4, we have discussed an effective mechanism for partner selection in the cloud federation environment. In the cloud federation scenario, if a CSP does not have enough resources to meet the requirements of its clients, it can get the resources from other partners in the federation. The cloud partner in the federation to which the user request can be transferred, should be selected in such a way that the QoS requirements of the users are not compromised and also the budgetary constraints of the users are taken care of. Literature review shows that there is no single efficient solution for partner selection in the cloud federation environment meeting the needs of the present cloud computing paradigm. The work proposed in this thesis systematically ranks the various CSPs in the

cloud federation for their effective selection. We have designed a mechanism to rank the CSPs in the federation using the AHP and the TOPSIS methods. The QoS values of various CSPs in the federation and the user requirements are used for ranking the CSPs in the federation. The QoS requirements of the users are given suitable weights using the AHP method. These weights are used in the TOPSIS method to rank the various CSPs. We have also incorporated the trust values of the CSPs in the federation into the partner selection approach in order to filter out untrustworthy CSPs. The proposed mechanism can be used by any CSP in the cloud federation to rank other CSPs whenever that CSP runs out of resources with its own cloud. The proposed mechanism is implemented using the CloudSim toolkit. The analysis of the results obtained highlights the advantages and the disadvantages of the proposed approach.

The major advantage of the proposed approach for partner selection is the fact that it helps a CSP in the federation to optimize its search for partners. Instead of searching all the CSPs in the federation to select the suitable one, now the search can be done based on the Rank Table generated. This is especially useful when there are so many CSPs in the federation with different cost and other QoS parameters. Another advantage of the proposed approach is the ability to specify suitable weights to the QoS attributes corresponding to the user requirements. Different users will have non-similar priorities, and they can be specified in the proposed approach while selecting the suitable partners in the cloud federation for offloading the users' requests to them. Also, the proposed approach considers the trust values of the CSPs in the federation, before selecting the suitable CSP from the federation for resource allocation, and his helps to avoid non-trustworthy or malicious CSPs from possible collaboration.

In chapter 5, we have proposed and implemented the trust-based approach for the management of dynamic QoS violations in the cloud federation environments. Normally, there will be QoS agreements between the partners in the federation for the resource sharing, and the process of SLA renegotiation is carried out among the CSPs to modify the QoS parameters of the services agreed among them. Now, if a request comes to a CSP from another CSP in the federation for some resources whose QoS features are not as per their prior agreement, how to dynamically deal with such a request in the federation without the

time consuming SLA renegotiation at that time is an issue to be considered. In this thesis, we propose a trust-based mechanism for the management of dynamic QoS violations, when one CSP requests resources from another CSP in the federation. As far as we know, this is the first work that employs the trust-based approach for the management of dynamic QoS violations in the cloud federation environment.

The proposed approach involves various steps such as partner selection, authentication, local trust calculation, recommended trust calculation and the total trust calculation. For authentication, we have implemented the SSO approach using the AES-256 algorithm and the FHMQV protocol as explained in the chapter 3. For the partner selection approach, we have used AHP and the TOPSIS methods. For the calculation of the local trust of the CSP, we have identified and formulated various parameters such as probability of success, history of interaction, existing trust, degree of association and QoS values. Thus, the trust value of a CSP is calculated as the average of these five factors. Also, the QoS value of a CSP is calculated by considering the average value of the five parameters such as availability, reliability, confidentiality, integrity and the response time. We have also considered the trust decay factor in our work in order to take into account the dynamic nature of the trust value. In order to calculate the recommended trust value of a CSP, firstly, the trusted CSPs are identified, and from them the feedback regarding the specified CSP is collected. In our work, we have eliminated any outlier in the feedback in order to increase the accuracy of the calculated trust value of a CSP in the federation. The final trust value of a CSP in the federation is calculated as the average of the local and the recommended trust values.

We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results is also given. The analysis of the total number of resource requests of a CSP accepted/rejected in the cloud federation environment shows important results. From the analysis, it is seen that reputation of the CSP plays an important role in the cloud federation environment. As compared to local trust, recommended trust also plays an important role in solving the dynamic QoS violations, and thereby accepting the resource requests from a CSP. Even though the calculation of the recommended trust takes longer compared to the calculation of the local trust alone, the performance of the cloud federation is im-

proved in such a way that more user requests are satisfied. Thus, the proposed trust-based approach shows that by calculating the local trust and the recommended trust values of the CSPs, the dynamic QoS violations can be effectively solved. The analysis of the results shows that the proposed approach improves the performance, responsiveness, efficiency, reputation and the profits of the CSPs in the federation.

In chapter 6, we have presented a trust-based approach for the management of dynamic break-glass access in the cloud federation environments. There should be an effective way to handle access request to PHR data during emergency situations, when the patients' information is stored in a multi-cloud or cloud federation environment. Even though there are many works that use cryptographic mechanisms to protect the personal health data of patients, to the best of our knowledge, the issue of identifying the legitimate access request taking trust into consideration in a cloud federation environment has not been addressed in a satisfactory manner. As far as we know, this is the first work that employs the trust and risk-based mechanism for finding the legitimacy of the emergency access requests in the cloud federation based healthcare environment.

The proposed approach shows that by calculating the dynamic trust of the requesting user in the federation, break-glass access requests can be effectively managed. The proposed approach calculates the risk value of the access request made, and then the local and the recommended trust values of the requesting user. Then, based on the calculated trust value, the proposed trust-based approach helps to effectively decide whether the emergency access request should be permitted or not in the cloud federation environment. In this work, the identity credentials of the users are encrypted using the AES-256 algorithm. In our implementation, we have considered the parameter 'Degree-of-Bias' which indicates whether the PHR owner has given explicit permission for a particular user and a health file. In our work, the risk calculation of the access request involves the three factors such as impact factor, sensitivity factor and the probability of malicious access factor. Local trust calculation of the user involves the eight factors such as probability of success, degree of association, history of interaction, existing trust, access level, access right, permitted factor and the genuine factor. We have also considered the trust decay factor of the user. Recommended trust calculation involves identifying the trusted CSPs in the feder-

ation, and then collecting feedback from the trusted CSPs regarding a specific PHR user. The total trust of the user is calculated by averaging the local and the recommended trust values of the user, and the break-glass access decision is taken depending on the total trust value of the PHR user.

We have implemented the proposed approach using the CloudSim toolkit, and the analysis of the results is also given. The analysis of the number of accepted break-glass requests of the users in the cloud federation environment is presented. The results show that the total number of accepted break-glass requests of a user increases with the recommendation from the trusted CSPs in the federation, and hence the overall efficiency of the cloud-based healthcare services is improved. In our simulation, the analysis of the average time taken for the break-glass access decision shows important results. Even though the calculation of the recommended trust of a user takes longer compared to the calculation of the local trust alone, the efficiency of the cloud-based healthcare services is improved as more break-glass access requests are accepted by the cloud-based healthcare service. Thus, considering the importance of the emerging cloud federation and the cloud-based healthcare services, the proposed approach is relevant and efficient in dealing with the emergency access requests of users. Thus, the approach improves the performance, responsiveness and the efficiency of the healthcare services delivered by the CSPs in the federation environment.

Thus, the major contributions in this thesis are:

1. Design and implementation of an effective authentication mechanism in the cloud federation environment that considers multiple identity providers. The security of the data transferred between the entities is taken care of by AES-256 and FHMQV protocols.

2. Design and implementation of an effective partner selection approach in the cloud federation environment for offloading the cloud users' requests for resources. The proposed method considers the QoS values offered by the CSPs, and also their trust values in the cloud federation, before selecting the suitable partner. AHP method is used to assign suitable weights to the QoS values according to the user requirements, and TOPSIS method is used to rank the various CSPs in the cloud federation for

meeting the resource requirements.

3. Design and development of a novel and efficient mechanism for the management of dynamic QoS violations while offloading the cloud users' resource requests in the cloud federation environment. The proposed mechanism uses the trust-based approach to manage the QoS violations without requiring the SLA re-negotiation at that time.

4. Design and development of an effective trust and risk-based approach for the management of dynamic break-glass access in the cloud federation environment. This work employs the trust and risk-based mechanism for finding the legitimacy of the emergency access requests in the cloud federation based healthcare environment.

Some possible future research directions are listed below:

In this thesis, we have worked at the horizontal cloud federation model. The work may be extended to vertical cloud federation model, also using real time implementation of cloud federation environment. The proposed approaches for Single Sign-On (SSO) authentication and the management of dynamic QoS violations can be implemented and the performance could be evaluated at the vertical cloud federation model as well. In our simulation, we have used the sample database created for testing the proposed approaches. Verification of the same can be carried out in a real cloud federation environment using real time test data. Also, for testing the trust and risk-based break-glass access management approach, in real life cases, auditing of a break-glass access can be used to prove the genuineness of the access request permitted by a user.

Bibliography

Abawajy, J. (2009). Determining service trustworthiness in intercloud computing environments. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, 784–788. IEEE.

Abawajy, J. (2011). Establishing trust in hybrid cloud computing environments. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 118–125. IEEE.

Agostinho, L., Feliciano, G., Olivi, L., Cardozo, E., and Guimaraes, E. (2011). A bio-inspired approach to provisioning of virtual resources in federated clouds. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, 598–604. IEEE.

Ahmed, M. and Xiang, Y. (2011). Trust ticket deployment: A notion of a data owner's trust in cloud computing. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 111–117. IEEE.

Anastasi, G. F., Carlini, E., and Dazzi, P. (2013). Smart cloud federation simulations with cloudsim. In *Proceedings of the first ACM workshop on Optimization techniques for resources management in clouds*, 9–16. ACM.

Armstrong, P., Agarwal, A., Bishop, A., Charbonneau, A., Desmarais, R., Fransham, K., Hill, N., Gable, I., Gaudet, S., Goliath, S., et al. (2010). Cloud scheduler: a resource manager for distributed compute clouds. *arXiv preprint arXiv:1007.0050*.

Azzedin, F. and Maheswaran, M. (2002). Towards trust-aware resource management in grid computing systems. In *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, 452–457. IEEE.

Azzedin, F. and Ridha, A. (2010). Feedback behavior and its role in trust assessment for peer-to-peer systems. *Telecommunication Systems*, 44(3-4), 253–266.

Benaloh, J., Chase, M., Horvitz, E., and Lauter, K. (2009). Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, 103–114. ACM.

Bermbach, D., Kurze, T., and Tai, S. (2013). Cloud federation: Effects of federated compute resources on quality of service and cost*. In *Cloud Engineering (IC2E), 2013 IEEE International Conference on*, 31–37. IEEE.

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., and Morrow, M. (2009). Blueprint for the intercloud-protocols and formats for cloud computing interoperability. In *Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on*, 328–336. IEEE.

Bernstein, D. and Vij, D. (2010a). Intercloud directory and exchange protocol detail using xmpp and rdf. In *Services (SERVICES-1), 2010 6th World Congress on*, 431–438. IEEE.

Bernstein, D. and Vij, D. (2010b). Intercloud security considerations. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 537–544. IEEE.

Bernstein, D. and Vij, D. (2011). Intercloud exchanges and roots topology and trust blueprint. In *Proc. of 11th International Conference on Internet Computing*, 135–141.

Bernstein, D., Vij, D., and Diamond, S. (2011). An intercloud cloud computing economy-technology, governance, and market blueprints. In *SRII Global Conference (SRII), 2011 Annual*, 293–299. IEEE.

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, 321–334. IEEE.

Brucker, A. D., Petritsch, H., and Weber, S. G. (2010). Attribute-based encryption with break-glass. In *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, 237–244. Springer.

Buyya, R., Ranjan, R., and Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and architectures for parallel processing*, 13–31. Springer.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23–50.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010a). How to enhance cloud architectures to enable cross-federation. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 337–345. IEEE.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010b). Security and cloud computing: intercloud identity management infrastructure. In *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on*, 263–265. IEEE.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010c). Three-phase cross-cloud federation model: The cloud sso authentication. In *Advances in Future Internet (AFIN), 2010 Second International Conference on*, 94–101. IEEE.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2011a). Federation establishment between clever clouds through a saml sso authentication profile. *International Journal on Advances in Internet Technology*, 4(1 & 2), 14–27.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2011b). Intercloud: the future of cloud computing. concepts and advantages. In *Cloud Computing: Methodology, Systems, and Applications*, 167–193. CRC Press, Taylor and Francis.

Chandrasekhar, S., Singhl, M., Tingjian, G., Krishnan, R., Joon Ahn, G., and Bertino, E. (2013). Collaboration in multicloud computing environments: Framework and security issues.

Chang, C.-C. and Chang, Y.-F. (2004). Yet another attack on a qr-based password authentication system. In *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on*, 2, 170–173. IEEE.

Chang, C.-C. and Lee, C.-Y. (2012). A secure single sign-on mechanism for distributed computer networks. *Industrial Electronics, IEEE Transactions on*, 59(1), 629–637.

Chang, H. and Choi, E. (2011). User authentication in cloud computing. In *Ubiquitous Computing and Multimedia Applications*, 338–342. Springer.

Chen, C., Gun, C., and Lin, H. (2011). A fair and dynamic password authentication system. In *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, 4505–4509. IEEE.

Chen, Y., Khoussainov, B., and Ye, X. (2013). A game theoretic approach to service discovery and selection. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, 4072–4079. IEEE.

Diep, N. N., Lee, S., Lee, Y.-K., and Lee, H. (2007). Contextual risk-based access control. *Security and Management*, 2007, 406–412.

Dong, C., Russello, G., and Dulay, N. (2008). Shared and searchable encrypted data for untrusted servers. In *Data and Applications Security XXII*, 127–143. Springer.

Ermakova, T. and Fabian, B. (2013). Secret sharing for health data in multi-provider clouds. In *Business Informatics (CBI), 2013 IEEE 15th Conference on*, 93–100. IEEE.

Fan, C.-I., Chan, Y.-C., and Zhang, Z.-K. (2005). Robust remote authentication scheme with smart cards. *Computers & Security*, 24(8), 619–628.

Fiorese, A., Matos, F., Júnior, O. C. A., and Ruppenthal, R. (2013). Multicriteria approach to select service providers in collaborative/competitive multi-provider environments. *Int. J. of Computer Science and Network Security*, 13(9), 15–22.

Foreman, J. (2006). At risk of exposure - in the push for electronic medical records, concern is growing about how well privacy can be sfeguarded. *Los Angeles Times, June 26*.

Fugkeaw, S., Manpanpanich, P., and Juntapremjitt, S. (2007). A robust single sign-on model based on multi-agent system and pki. In *Networking, 2007. ICN'07. Sixth International Conference on*, 101–101. IEEE.

Ghosh, N., Ghosh, S. K., and Das, S. K. (2015). Selcsp: A framework to facilitate selection of cloud service providers. *Cloud Computing, IEEE Transactions on*, 3(1), 66–79.

Goiri, I., Guitart, J., and Torres, J. (2010). Characterizing cloud federation for enhancing providers' profit. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 123–130. IEEE.

Goiri, Í., Guitart, J., and Torres, J. (2012). Economic model of a cloud provider operating in a federated cloud. *Information Systems Frontiers*, 14(4), 827–843.

Goriawala, S. (2013). Authentication and access control: Selecting the appropriate authentication method for your organization. *SmartSignIn (www.smartsignin.com)*.

Govil, S. B., Thyagarajan, K., Srinivasan, K., Chaurasiya, V. K., and Das, S. (2012). An approach to identify the optimal cloud in cloud federation. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 1(1), 35–44.

Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, 89–98. Acm.

Gunjan, K., Sahoo, G., and Tiwari, R. (2012). Identity management in cloud computing–a review. In *International Journal of Engineering Research and Technology*, 1, 1–5. ESRSA Publications.

Habib, S. M., Ries, S., and Mühlhäuser, M. (2010). Cloud computing landscape and research challenges regarding trust and reputation. In *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on*, 410–415. IEEE.

Haresh, M., Kalady, S., and Govindan, V. (2011). Agent based dynamic resource allocation on federated clouds. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, 111–114. IEEE.

Hassan, M. M. and Huh, E.-N. (2011). Resource management for data intensive clouds through dynamic federation: a game theoretic approach. In *Handbook of Data Intensive Computing*, 169–188. Springer.

Hassan, M. M., Song, B., Han, S.-M., Huh, E.-N., Yoon, C., and Ryu, W. (2009). Multi-objective optimization model for partner selection in a market-oriented dynamic collaborative cloud service platform. In *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, 637–644. IEEE.

Huda, M. N., Yamada, S., and Sonehara, N. (2009). Privacy-aware access to patient-controlled personal health records in emergency situations. In *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, 1–6. IEEE.

Hwang, C.-L., Lai, Y.-J., and Liu, T.-Y. (1993). A new approach for multiple objective decision making. *Computers & operations research*, 20(8), 889–899.

Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2), 618–644.

Juntapremjitt, S., Fugkeaw, S., and Manpanpanich, P. (2008). An sso-capable distributed rbac model with high availability across administrative domain. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, 121–126. IEEE.

Kanwal, A., Masood, R., and Shibli, M. A. (2014). Evaluation and establishment of trust in cloud federation. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, page 12. ACM.

Kertész, A., Kecskeméti, G., Marosi, A., Oriol, M., Franch, X., and Marco, J. (2012). Integrated monitoring approach for seamless service provisioning in federated clouds.

In *Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on*, 567–574. IEEE.

Khan, A. U., Oriol, M., Kiran, M., Jiang, M., and Djemame, K. (2012). Security risks and their management in cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 121–128. IEEE.

Kim, J. and Hong, S.-p. (2012). A consolidated authentication model in cloud computing environments. *International Journal of Multimedia and Ubiquitous Engineering*, 7(3), 151–160.

Kumar, S. and Cohen, P. R. (2000). Towards a fault-tolerant multi-agent system architecture. In *Proceedings of the fourth international conference on Autonomous agents*, 459–466. ACM.

Künzi, J., Koster, P., and Petkovic, M. (2009). Emergency access to protected health records. In *MIE*, 705–709.

Le, G., Xu, K., and Song, J. (2012). Gossip-based hybrid multi-attribute overlay for resource discovery in federated clouds. In *e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on*, 279–284. IEEE.

Lee, C.-C., Li, L.-H., and Hwang, M.-S. (2002). A remote user authentication scheme using hash functions. *ACM SIGOPS Operating Systems Review*, 36(4), 23–29.

Leung, K.-C., Cheng, L., Fong, A. S., and Chan, C.-K. (2003). Cryptanalysis of a modified remote user authentication scheme using smart cards. *Consumer Electronics, IEEE Transactions on*, 49(4), 1243–1245.

Li, M., Yu, S., Ren, K., and Lou, W. (2010). Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In *Security and Privacy in Communication Networks*, 89–106. Springer.

Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1), 131–143.

Li, W. and Ping, L. (2009). Trust model to enhance security and interoperability of cloud environment. In *Cloud Computing*, 69–79. Springer.

Lin, C.-W., Shen, J.-J., and Hwang, M.-S. (2003). Security enhancement for optimal strong-password authentication protocol. *ACM SIGOPS Operating Systems Review*, 37(2), 7–12.

McCallister, E., Grance, T., and Scarfone, K. A. (2010). Sp 800-122. guide to protecting the confidentiality of personally identifiable information (pii). *National Institute of Standards & Technology*.

Mihailescu, M. and Teo, Y. M. (2010). Dynamic resource pricing on federated clouds. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, 513–517. IEEE.

Mukhopadhyay, S. and Argles, D. (2011). An anti-phishing mechanism for single sign-on based on qr-code. In *Information Society (i-Society), 2011 International Conference on*, 505–508. IEEE.

Pub, N. F. (2001). 197: Advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197(441), 0311.

Radha, V. and Reddy, D. H. (2012). A survey on single sign-on techniques. *Procedia Technology*, 4, 134–139.

Ren, X. and Wu, X.-W. (2012). A novel dynamic user authentication scheme. In *Communications and Information Technologies (ISCIT), 2012 International Symposium on*, 713–717. IEEE.

Sánchez, R., Almenares, F., Arias, P., Díaz-Sánchez, D., and Marín, A. (2012). Enhancing privacy and dynamic federation in idm for consumer cloud computing. *Consumer Electronics, IEEE Transactions on*, 58(1), 95–103.

Sarr, A. P., Elbaz-Vincent, P., and Bajard, J.-C. (2010). A secure and efficient authenticated diffie–hellman protocol. In *Public Key Infrastructures, Services and Applications*, 83–98. Springer.

Siebenhaar, M., Lampe, U., Lehrig, T., Zöller, S., Schulte, S., and Steinmetz, R. (2011). Complex service provisioning in collaborative cloud markets. In *Towards a Service-Based Internet*, 88–99. Springer.

Singh, R. and Vipra Gupta, M. K. (2013). Dynamic federation in identity management for securing and sharing personal health records in a patient centric model in cloud. *International Journal of Engineering and Technology*, 5(3), 2201–2209.

Song, B., Hassan, M., Huh, E.-N., Yoon, C.-W., and Lee, H.-W. (2009b). A hybrid algorithm for partner selection in market oriented cloud computing. In *Management and Service Science, 2009. MASS'09. International Conference on*, 1–4. IEEE.

Song, B., Hassan, M. M., and Huh, E.-N. (2009a). A novel cloud market infrastructure for trading service. In *Computational Science and Its Applications, 2009. ICCSA'09. International Conference on*, 44–50. IEEE.

Stihler, M., Santin, A. O., Marcon Jr, A. L., and Fraga, J. D. S. (2012). Integral federated identity management for cloud computing. In *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*, 1–5. IEEE.

Sun, J., Zhu, X., Zhang, C., and Fang, Y. (2011). Hcpp: Cryptography based secure ehr system for patient privacy and emergency healthcare. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 373–382. IEEE.

Thummavet, P. and Vasupongayya, S. (2013). A novel personal health record system for handling emergency situations. In *Computer Science and Engineering Conference (ICSEC), 2013 International*, 266–271. IEEE.

Tong, Y., Sun, J., Chow, S. S., and Li, P. (2013). Towards auditable cloud-assisted access of encrypted health data. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, 514–519. IEEE.

Triantaphyllou, E. and Mann, S. H. (1995). Using the analytic hierarchy process for decision making in engineering applications: some challenges. *International Journal of Industrial Engineering: Applications and Practice*, 2(1), 35–44.

Truong-Huu, T. and Tham, C.-K. (2014). Competition and cooperation among providers in a cloud-of-clouds environment. *National University of Singapore, Tech. Rep., Jan*.

Tusa, F., Celesti, A., Paone, M., Villari, M., and Puliafito, A. (2011). How clever-based clouds conceive horizontal and vertical federations. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, 167–172. IEEE.

Vijayakumar, V. and Banu, R. (2008). Security for resource selection in grid computing based on trust and reputation responsiveness. *International Journal of Computer Science and Network Security*, 8(11), 107–115.

Vijayakumar, V., Wahida Banu, R., and Abawajy, J. H. (2012). An efficient approach based on trust and reputation for secured selection of grid resources. *International journal of parallel, emergent and distributed systems*, 27(1), 1–17.

Wang, G., Yu, J., and Xie, Q. (2013). Security analysis of a single sign-on mechanism for distributed computer networks. *Industrial Informatics, IEEE Transactions on*, 9(1), 294–302.

Wenge, O., Siebenhaar, M., Lampe, U., Schuller, D., and Steinmetz, R. (2012). Much ado about security appeal: cloud provider collaborations and their risks. In *Service-Oriented and Cloud Computing*, 80–90. Springer.

Wu, C.-S. and Khoury, I. (2012). Qos-aware dynamic research component composition for collaborative research projects in the clouds. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 883–888. IEEE.

Xin, L. and Datta, A. (2010). On trust guided collaboration among cloud service providers. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on*, 1–8. IEEE.

Yan, L., Rong, C., and Zhao, G. (2009). Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography. In *Cloud Computing*, 167–177. Springer.

Yoon, K. (1987). A reconciliation among discrete compromise solutions. *Journal of the Operational Research Society*, 277–286.

Yoon, K. P. and Hwang, C.-L. (1995). *Multiple attribute decision making: an introduction*, 104. Sage publications.

Zhang, Z., Wang, Y., and Wang, Z. (2013). A grey topsis method based on weighted relational coefficient. In *Grey Systems and Intelligent Services, 2013 IEEE International Conference on*, 550–557. IEEE.

Zwattendorfer, B. and Tauber, A. (2012). Secure cross-cloud single sign-on (sso) using eids. In *Internet Technology And Secured Transactions, 2012 International Conference for*, 150–155. IEEE.

Publications

International Journals

1. Manoj V. Thomas and K. Chandrasekaran, *"Agent-based approach for identity and access management in the inter-cloud environments"*, International Journal of Trust Management in Computing and Communications, Inderscience Publishers, ISSN: 2048-8378, Volume 2, Issue 2, pp. 125-149, 2014. (DBLP Computer Science Bibliography, Google Scholar, INSPEC(IET), ProQuest indexed) http://www.inderscienceonline.com/doi/abs/10.1504/IJTMCC.2014.064144

2. Manoj V. Thomas, Anand Dhole, and K. Chandrasekaran, *"Single Sign-On in Cloud Federation using CloudSim"*, International Journal of Computer Network and Information Security (IJCNIS), ISSN: 2074-9104, Volume 7, Issue 6, pp. 50-58, 2015. (Google Scholar, DOAJ, INSPEC (IET), ProQuest indexed) http://www.mecs-press.org/ijcnis/ijcnis-v7-n6/IJCNIS-V7-N6-6.pdf

3. Manoj V. Thomas and K. Chandrasekaran, *"Agent-Based Cloud Broker Architecture for Distributed Access Control in the Inter-Cloud Environments"*, International Journal of Information Processing (IJIP), ISSN: 0973-8215, Volume 8, Issue 1, pp. 107-123, 2014. (Google Scholar, ICI indexed) https://www.ijipbangalore.org/abstracts_8(1)/p10.pdf

4. Manoj V. Thomas and K. Chandrasekaran, *"Dynamic Partner Selection in Cloud Federation for ensuring the Quality of Service for Cloud consumers*, International Journal of Modeling, Simulation, and Scientific Computing, World Scientific Publishers, ISSN: 1793-9623, Volume 8, Issue 3, 2017. (Scopus and ESCI indexed). http://www.worldscientific.com/doi/pdf/10.1142/S1793962317500362

5. Manoj V. Thomas and K. Chandrasekaran, *"A Trust-Based Approach for Management of Dynamic QoS Violations in Cloud Federation Environments*, Open Journal of Cloud Computing (OJCC), ISSN: 2199-1987, Volume 2, Issue 2, pp. 21-43,

2015. (Google Scholar, OAI, Worldcat indexed) https://www.ronpub.com/OJCC_2015v2i2n03_Thomas.pdf

6. Manoj V. Thomas and K. Chandrasekaran, *"Trust and Risk-Based Approach for the Management of Dynamic Break-Glass Access in the Cloud Federation Environments*, International Journal of Computer Science and Information Security, ISSN: 1947-5500, Volume 14, Issue 7, pp. 141-152, 2016. (Google Scholar, DOAJ, ESCI indexed) https://sites.google.com/site/ijcsis/vol-14-no-7-jul-2016

Book Chapters

1. K. Chandrasekaran and Manoj V. Thomas, *Distributed Access Control in Cloud Computing Systems*, Encyclopedia of Cloud Computing, Wiley Publications, pp. 417-432, 2016. http://onlinelibrary.wiley.com/doi/10.1002/9781118821930.ch35/summary

2. Manoj V. Thomas and K. Chandrasekaran, *Identity and Access Management in the Cloud Computing Environments*, Identity Theft: Breakthroughs in Research and Practice, IGI Global Publishers, pp. 38-68, 2016. http://www.igi-global.com/chapter/identity-and-access-management-in-the-cloud-computing-environments/167219

International Conferences

1. Manoj V. Thomas and K. Chandrasekaran, *"Workflow Model for Distributed Access Control"*, in Proc. Third IEEE International Conference on Advances in Computing and Communications (ACC-2013), Cochin, India, pp. 363-366, IEEE, 2013. http://ieeexplore.ieee.org/document/6686409/

2. Manoj V. Thomas and K. Chandrasekaran, *"An Access Control Model for Cloud Computing Environments"*, in Proc. Second IEEE International Conference on Advanced Computing, Networking and Security (ADCONS-2013), Surathkal, India, pp. 226-231, IEEE, 2013. http://ieeexplore.ieee.org/document/6714168/

3. Manoj V. Thomas and K. Chandrasekaran, *"Research Focus on Distributed Access Control"*, in Proc. International Conference on Advanced Computing, Networking and Security (ADCONS-2011), Surathkal, India, 2011. https://scholar.google.com/scholar?cluster=6023913816908207531&hl=en&oi=\scholarr

4. Manoj V. Thomas and K. Chandrasekaran, *"Agent-Based Cloud Broker Architecture for Distributed Access Control"*, in Proc. Seventh International Conference on Communication Networks (ICCN-2013), Bangalore, India, Elsevier, pp. 189-197, 2013. http://searchdl.org/public/book_series/elsevierst/3/ICCN21.pdf

5. Manoj V. Thomas and K. Chandrasekaran, *"Agent-Based Approach for Distributed Access Control in Cloud Environments"*, in Proc. Second IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2013), Mysore, India, pp. 1628-1633, IEEE, 2013. http://ieeexplore.ieee.org/document/6637425/

BIO-DATA

Name : Manoj V. Thomas

Email : manojkurissinkal@gmail.com

Date of Birth : February 27, 1981

Address : Kurissinkal (H),

Attenganam (P.O.),

Kasaragod (Dt.),

Kerala-671 531.

Educational Qualifications:

Degree	Year of Passing	Institution	Class
M.Tech	2008	NITK, Surathkal.	First Rank with Gold Medal
B.Tech	2003	RIT, Kottayam.	Distinction