11-2016

Sla Management in a Collaborative Network Of Federated Clouds: The Cloudland

Asma Obaid Subaih AI Falasi

UAEU

جامعة الإمارات العربيـة المتحدة
United Arab Emirates University

United Arab Emirates University

College of Information Technology

SLA MANAGEMENT IN A COLLABORATIVE NETWORK OF FEDERATED CLOUDS: THE *CLOUDLEND*

Asma Obaid Hamad Saeed Subaih Al Falasi

This dissertation is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Under the Supervision of Dr. Mohamed Adel Serhani

November 2016

Declaration of Original Work

I, Asma Obaid Hamad Saeed Subaih Al Falasi, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this dissertation entitled "*SLA Management in a Collaborative Network of Federated Clouds: The CloudLend*", hereby, solemnly declare that this dissertation is my own original research work that has been done and prepared by me under the supervision of Dr. Mohamed Adel Serhani, in the College of Information Technology at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my dissertation have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this dissertation.

Student's Signature: ______________ Date: 13-12-2016

Approval of the Doctorate Dissertation

This Doctorate Dissertation is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Mohamed Adel Serhani
   Title:Associate Professor
   Department of E-Commerce
   College of Information Technology

   Signature____________________ Date 20/11/2016

2) Member: Khaled Shuaib
   Title: Professor
   Department of Information Security
   College of Information Technology

   Signature ____________________ Date 20/11/2016

3) Member: Saad Harous
   Title: Associate Professor
   Department of Intelligent Systems
   College of Information Technology

   Signature ____________________ Date 20/11/2016

4) Member (External Examiner):Ala Al-Fuqaha
   Title:Professor
   Department ofComputer Science
   Institution:Western Michigan University

   Signature ____________________ Date 11-20-2016

This Doctorate Dissertation is accepted by:

Dean of the College of Information Technology: Professor Omar El-Gayar

Signature ____________________ Date Dec. 14, 2016

Dean of the College of the Graduate Studies: Professor Nagi T. Wakim

Signature ____________________ Date 15/12/2016

Copy 11 of 12

Advisory Committee

1) Advisor: Dr. Mohamed Adel Serhani

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

2) Member: Dr. Elarbi Badidi

Title: Associate Professor

Department of Computer Science and Software Engineering

College of Information Technology

3) Member: Dr. Mohammad Mehedy Masud

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

Abstract

Cloud services have always promised to be available, flexible, and speedy. However, not a single Cloud provider can deliver such promises to their distinctly demanding customers. Cloud providers have a constrained geographical presence, and are willing to invest in infrastructure only when it is profitable to them. Cloud federation is a concept that collectively combines segregated Cloud services to create an extended pool of resources for Clouds to competently deliver their promised level of services. This dissertation is concerned with studying the governing aspects related to the federation of Clouds through collaborative networking. The main objective of this dissertation is to define a framework for a Cloud network that considers balancing the trade-offs among customers' various quality of service (QoS) requirements, as well as providers' resources utilization. We propose a network of federated Clouds, *CloudLend,* that creates a platform for Cloud providers to collaborate, and for customers to expand their service selections. We also define and specify a service level agreement (SLA) management model in order to govern and administer the relationships established between different Cloud services in *CloudLend*. We define a multi-level SLA specification model to annotate and describe QoS terms, in addition to a game theory-based automated SLA negotiation model that supports both customers and providers in negotiating SLA terms, and guiding them towards signing a contract. We also define an adaptive agent-based SLA monitoring model which identifies the root causes of SLA violations, and impartially distributes any updates and changes in established SLAs to all relevant entities. Formal verification proved that our proposed framework assures customers with maximum optimized guarantees to their QoS requirements, in addition to supporting Cloud providers to make informed resource utilization decisions. Additionally, simulation results demonstrate the effectiveness of our SLA management model. Our proposed *CloudLend* network and its SLA management model paves the way to resource sharing among different Cloud providers, which allows for the providers' lock-in constraints to be broken, allowing effortless migration of customers' applications across different providers whenever is needed.

Keywords: Cloud computing, cloud federation, game theory, SLA negotiation, SLA monitoring.

Title and Abstract (in Arabic)

تنظيم اتفاقيات مستويات الخدمة في شبكة تفاعلية للحوسبة السحابية المتكاملة

الملخص

تَعِدُ الحوسبة السحابية عملاءها بخدمات مرنة و متاحة و سريعة، لكن ليس بإمكان مزود خدمة سحابية واحد على الأغلب أن يحقق مثل هذه الوعود لجميع العملاء على حد سواء. فمزودوا الخدمات السحابية لهم امتداد جغرافي محدد، و قد لا يبادرون في تحمل تكاليف بنية تحتية إضافية في غير الحالات التي تضمن لهم تحقيق الربح المادي المتوقع. إن مفهوم تكامل خدمات الحوسبة السحابية يجمع مزودي خدمات الحوسبة السحابية للعمل معاً بصورة متكاملة مُشكلاً بذلك مصدراً لا متناهي من الموارد التي تُحقق متطلبات العملاء بكفاءة. تختص هذه الأطروحة بدراسة الجوانب الرئيسية المتعلقة بتحقيق تكامل خدمات الحوسبة السحابية من خلال الشبكات التفاعلية. و تهدف على وجه الخصوص إلى تحديد إطارعمل لشبكة خدمات الحوسبة السحابية يوازن بين متطلبات مستوى الخدمة من العملاء و تعزيز استغلال موارد مزودي الخدمات. من خلال هذه الأطروحة نقترح شبكةً من خدمات الحوسبة السحابية لتشكل منصة تكامل و تعاون بين مزودي الخدمة مما يزيد من اتساع نطاق اختيارات الخدمة بالنسبة للعملاء. ونقدم كذلك نموذجاً لإدارة اتفاقية مستويات الخدمة لتنظيم العلاقات المبنية بين مزودي الخدمة من خلال شبكة تكامل الخدمات المقترحة. يعتمد هذا النموذج على توصيف متعدد النطاقات يوفر تمثيلاً دقيقاُ لمتطلبات مستوى الخدمة. بالإضافة إلى استخدام نظرية المباراة للتفاوض على مستويات الخدمة، و الذي يساهم في الوصول إلى اتفاق مشترك بين العملاء ومزودي الخدمة على اتفاقيات مستويات الخدمة. يقدم النموذج المقترح أيضاً وكيلاً لمتابعة كفاءة مستوى الخدمة المقدمة يقوم بتحديد الأسباب الجذرية لأي انتهاكات لاتفاقيات مستويات الخدمة، ويعمل على إيصال أية تحديثات او تغييرات قد تطرأ على اتفاقيات مستوى الخدمة الجارية لجميع الأطراف ذات الصلة. أثبتت تجارب محاكاة الشبكة التفاعلية لخدمات الحوسبة السحابية فعالية الشبكة المقترحة و نموذج إدارة اتفاقيات مستوى الخدمة الخاص بها فعاليتهما في توفيرضمانات قصوى لمتطلبات جودة الخدمة لدى العملاء، بالإضافة إلى دعم مزودي الخدمة باتخاذ قرارت تؤدي إلى الاستغلال الأمثل للموارد المتوفرة.

مفاهيم البحث الرئيسية: الحوسبة السحابية، تكامل الحوسبة السحابية، نظرية المباراة، مفاوضة و متابعة اتفاقيات مستوى الخدمة.

Acknowledgements

First and foremost, I thank Allah for making it all possible for me, and for blessing me with all the people who stood alongside me during the last six years. Those who I would like to thank each and every one of them.

My genuine appreciation is expressed to my advisor Dr. Mohamed Adel Serhani for the endless support, motivation, encouragement, and valuable involvement. I could not ask for a better mentor for my Ph.D. study, his guidance was of a great help throughout my research and dissertation writing. My sincere thanks also goes to my advisory committee members: Dr. Yacine Atif, and Dr. Elarbi Badidi for their continuous support and advice during my dissertation preparation. I am also very grateful for all the support I received from Dr. Saad Harous, Dr. Rachida Dssouli, Dr. Najah Abu Ali, Dr. Ezedin Barka, and Dr. Farag Sallabi who contributed in teaching me the true aspects of research.

I was privileged to join the UAE University Ph.D. program as one of the first cohort of national Ph.D. candidates, and I will always be grateful to UAE University for providing me the opportunity to pursue my graduate study within an intellectual and professional environment in my home city of Al Ain. I am also indebted to my dearest family and friends for being there for me at all times.

Dedication

To my beloved family, my parents Obaid, and Kaltham. My siblings Alia, Hind, Rouda, Fatima, Mariam, Ahmed, Shiekha, Hamad, and Abdulla.

To my dear husband Hamad. My precious sons Buti and Obaid.

To my best friend Alia

This dissertation is dedicated for your belief in me.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
AW	Adjusted Winner
IaaS	Infrastructure as a Service
MMOG	Massively Multiplayer Online Gaming
PaaS	Platform as a Service
QoCS	Quality of Cloud Service
QoS	Quality of Service
SaaS	Software as a Service
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture

Chapter 1: Introduction

The Federation of Clouds is becoming increasingly appealing for both Cloud providers and customers. The concept of consolidating heterogeneous Cloud environments brings up wider service opportunities. It increases provider flexibility and expands customer choice, allowing custom mash-ups of Cloud services.

The focus of this dissertation is to study the feasibility of Cloud federation through collaborative networking. We identify interactions among Cloud customers and providers within the federation and highlight roles and responsibilities necessary to manage such a federation. In addition, we address issues related to: service selection, quality of Cloud service (QoCS) assurance, and service level agreement (SLA) management.

We propose a network of federated Clouds that is named *CloudLend* to create a platform for Cloud providers to collaborate, and for customers to extend their service opportunities. The proposed network allows customers to specify and negotiate their QoS requirements, which enables them to actively control the level of the provided service. *CloudLend* enables providers as well to gain access to a broader market share, and be exposed to more customers through the network, which enhances providers' resource utilization.

We define and specify a SLA management model that administers relationships established between different Cloud services in *CloudLend*. The proposed model is intended to administer Cloud federation environments, where Cloud services discovery, interaction and collaboration can be achieved. The SLA management model is composed of three correlated models: an XML-based SLA specification model, a game theory based SLA negotiation model, and an agent-based monitoring model. These SLA management building blocks aim to provide an assured collaboration platform for Cloud providers, where relationships are established to efficiently provide customers with quality assured services.

1.1 Background

This section surveys important information related to key concepts and knowledge used in this research. We first discuss social networking, and highlight the value it adds to the service provisioning on the Internet. Then we introduce Cloud computing, and the federation of Clouds. Furthermore, we define basic requirements for managing the SLA in federated Clouds. Finally, we introduce the game theory approach, and discuss the type of games that were adopted for SLA negotiation in *CloudLend*.

1.1.1 Social Collaborative Networks

Social collaborative network sites are defined as:

"*Web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system.*" (Boyd & Ellison, 2007)

The past few years have witnessed an undeniable exponential growth of social networks that has a notable influence on individuals and businesses. Recent statistics (Smith, 2016) show that social networking is the most popular online activity with an active 2.3 billion users, representing 72% of the Internet population. Moreover, 91% of retail brands use two or more social media channels. Besides, the massive amount of users' related data embedded within social networks allows for tailored advertising which is more likely to reach its intended audience than any other site on the Internet. This renders marketing the main source of revenue for social sites, followed by subscription fees. Such socially rich platforms generate a prominent trade power that is able to boost the amount of transactions being exchanged over social networks.

By exploiting their identities, interests, behavior, and particularly their relationships, people are considered the cornerstone of today's online social spaces. When relationships are established in a social network, interactions such as browsing, searching, messaging, content sharing and community formation will follow (Benevenuto, Rodrigues, Cha, & Almeída, 2009). Based on the burgeoning success of social networking among people, and bowing to the emerging notion of the Internet of Things (Atzori, Iera, & Morabito, 2010), we envisage a bigger platform of collaborative networking where colossal computing entities like the Clouds can collaborate and establish some relationships with its Cloud peers in order to produce a larger computing network we call *CloudLend*.

1.1.2 Cloud Computing

Cloud computing is an emerging trend for the provision of IT infrastructure as services. It is potentially transforming the way of offering business services, and developing software. The Cloud computing approach to service provisioning is to become prominent and accessible for all, without the hassle of investing in expensive hardware resources nor of managing or maintaining them.

Computing on the Cloud is perceived as an evident outcome of the recent expansion of the web as it grows into the Web of services. It is defined as: "a set of network enabled services, providing scalable, quality of service (QoS) guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way" (Wang, et al., 2010). This means accessing applications, services and IT infrastructure through QoS guaranteed Web services. Hence, Cloud computing enables users to utilize services without having to be aware of their complexity, nor to acquire the knowledge and expertise to actually consume the services. Basically, Cloud computing provides users with services to access hardware, software, and data.

Cloud computing is enabled by the enduring evolving technologies of Web and service oriented computing. It became popular nowadays because of the emergent necessity to provide complex IT infrastructure. Such resources are consumed by users for various applications, such as managing different software requirements, and handling the exponentially rising data size on the Internet. Furthermore, the common adoption of service oriented architecture (SOA), and web applications has increased the adoption of Cloud computing. SOA is considered as the underlying concept of the Cloud computing; as it enables remotely integrated services to be provided based on some specific end user requirements. According to the US National Institute of Standards and Technology (Mell & Grance, 2011), Cloud computing service models can be classified into three main categories:

1. Infrastructure as a Service (IaaS): the most straightforward form of Cloud computing, where providers offer infrastructure resources such as virtual machines as a service.
2. Platform as a Service (PaaS): a service model where customers are offered applications development environments as a service, such as: operating systems, databases, and web server.

3. Software as a Service (SaaS): a service model where customers are offered access to applications installed, and operated by Cloud providers.

1.1.3 Federation of Clouds

The Cloud federation can be defined as the aggregation of several Cloud services provided by different providers in order to achieve a specified goal (e.g. maximize profit, achieve high competitiveness, and guarantee a share in the market). Cloud federation is often misleadingly associated with the concept of Cloud portability. Conversely, portability of a Cloud refers to its ability to migrate a Cloud service to different providers. However, despite migration, Cloud services perform as intended, without the need to be reconstructed to fit the new Cloud environment. Williams (2009) defines three levels of portability: the first stage is the portability of virtual machines; which is concerned with the import and export of virtual machines across federated Clouds. The second stage is the portability of virtual machines along with the network setting, while the third one is the portability of APIs. On the other hand, Oberle & Fisher (2010) classify portability solutions into three categories: functional, data, and service enhancement.

1.1.4 SLA Management in the Federation of Clouds

Cloud computing presents a pay-as-you-go model for resources that can be invoked and tailored as per customer's QoS requirements. It is essential that customers receive guarantees on service delivery from Cloud providers. Such guarantees are provided through SLAs in order to govern and control service provisioning between customers and Cloud providers. SLA in Cloud Computing is defined as: "*The Cloud provider's contractually agreed-to level of performance for certain aspects of the services"* (Buyya, Broberg, & Goscinski, 2011). Many recent research efforts have invested in the adoption of SLA management approaches of Grid computing, Web services, and SOA to govern Cloud services provisioning. However, currently implemented SLA models do not fully satisfy most of the Cloud service provisioning requirements. These models are unable to manage flexible, elastic, and varying type of services. Therefore, new Cloud-specific SLA management models are required in order to provide accurate service definition, negotiation, deployment, monitoring, and even enforcement.

1.1.5 Game Theory

Game theory began with the work of Neumann and Morgenstren (1944) and it is defined as: "The study of mathematical models of conflict and cooperation between intelligent rational decision makers" (Myerson, 2013). It supports understanding, and resolving situations that involve two or more individuals making decisions that will affect one another's welfare. Game theory resolves such situations through general mathematical techniques.

In this research we consider the problem of SLA negotiation in *CloudLend* as a Fair Division game (Brams & Taylor, 1996). These involve players in a sequential game, in which they need to decide on how to divide an item, like a property ownership, or time-share to access a resource. Every player values the item to be shared among them differently. An example of a Fair Division game is called Fair Cake-cutting (Brams & Taylor, 1996). A cake with different toppings must be divided among many players, who have different preferences over different parts of the cake. The division needs to be fair to every player. In this case, each player receives a slice that he believes to be a fair share. In cases where a set of items is to be divided among players, yet these items themselves need to be kept as a whole, a proportional and envy-free division procedure is used (Brams & Taylor, 1996). The adjusted winner procedure (AW) (Brams & Taylor, 2000) is one of the proportional and envy-free division procedures. AW describes a fair division of a set of *n* items that can be shared between two players. Each player examines the *n* items, and assigns a rate for each individual item, out of a total of 100 points among them all. These points are a relative preference of the players for the various rated items.

In *CloudLend*, Cloud services are engaged in playing the SLA negotiation game in order to reach the best collection of SLA terms that would satisfy all players' requirements. The outcome of the game is basically a measure of the value a Cloud service gains by establishing a relationship with other players. During the SLA negotiation game, the SLA contract is considered as a whole entity that consists of several SLA terms. Therefore, during negotiation, players bargain over the value of SLA terms that make up the utility gain of the whole SLA contract. Eventually, both players need to decide on the impact every SLA term has on the total value of the SLA contract.

1.2 Motivation

The current status of the Internet shows that many Cloud service providers offer resources that are accessible via a wide spectrum of platforms. Common usages of online services include: messaging, applications downloading, Internet browsing, in addition to multimedia streaming, while other sophisticated computing applications became widely available with the appearance of Cloud services. Such services include: resource sharing, collaborating, multitasking and scheduling.

Services composition enables the seamless realization of new services through the configuration of some basic services based on customers' fluctuating requirements. In Cloud computing, services composition corresponds to Cloud federation which can provide solutions to prominent issues in Cloud computing. Most importantly, it allows resource sharing among different Cloud providers, so that even small Cloud providers with limited resources can offer a wider range of services without further investing on platform, and infrastructure. This helps to cut their costs of IT infrastructure and data center establishment, and allows them to enter a market dominated by leading Cloud providers.

Moreover, the composition of Cloud services through federation of Clouds will benefit the Cloud providers not only in cutting costs, but will also contribute to building a massive repository of customers' data, related to their requirements and activities. This data can be of a great help towards the shift to a customer-centric approach of Cloud service provisioning. Cloud providers can gain valuable understanding of customers' needs, which results in improved revenues, and enhanced customer satisfaction. In addition, the federation of Clouds allows for breaking the providers' lock-in constraints, enabling the effortless migration of customers' applications across different providers whenever is needed, and leading the way towards a competitive market.

Cloud federation is diffidently present in confined environments, such as governments, and enterprises where distributed data centers tend to have foreseeable collaboration aspects among their services. Additionally, specifications and implementation details of federated Clouds are required to be clearly defined, and agreed upon by participating parties prior to establishing the federation. Thus, in such environments, Cloud services can only interact with other predetermined services, and for a limited period of time.

In Cloud computing, many challenges can affect the quality of the provided services, such as: issues related to network connections, data security, service availability, or even changes in SLA conditions. Although Cloud federation offers many advantages for both Cloud customers and providers, providers are still reluctant to adopt the federation approach. This is due to their concerns related to the lack of SLA regulation and management. SLAs are important to set the expectations of both Cloud providers and customers, as well as to plan future changes in the provided service.

Challenges in federated Cloud SLA management come from the need to provide different SLAs, for different customers to integrate with their own business processes. This requires a clear and specific definition of SLA parameters and metrics, dynamic SLA negotiation and automated service monitoring, in addition to clear SLA enforcement measures. We provide hereafter an ample description of two key challenges faced by SLA management in the federated Cloud environment that are related to the following: a chain of interconnected services, and automatic adaptation to environment changes.

1.2.1 Chain of Interconnected Services

Cloud services interconnect with other services within the same or from other providers in order to fulfill tasks required by customers' QoS. Such interconnections are not necessarily confined to a single level. They can be extended to reach further services in order to carry out minor subtasks. Additionally, a Cloud service can maintain connections with one or more other Cloud services at the same time. This results in a chain of interconnected services that are bounded by multi-level SLAs. Therefore, considering SLA specification in a federated Cloud environment requires a comprehensive deliberation of the multi-level nature of connections among Cloud services. SLAs need to be defined in an aggregated manner, so that the complexity of the SLA chain is hidden from the customer. Yet, SLA terms parameters need to be specifically defined, and each mapped to its contributing services. Similarly, SLA negotiation requires a particular emphasis, as it is not restricted to Cloud provider-customer only. SLA negotiation occurs also between interconnected services, and it requires the design of proper mechanisms to facilitate communication, and to manage service-to-service negotiation. SLA monitoring is also very challenging in a federated Cloud environment. It necessitates measurements of SLA parameters using a set of metrics that are measured against thresholds on multiple dynamic levels. Therefore, monitoring approaches designed for federated Cloud environments are required to implement specific mechanisms, which are able to capture and monitor the aggregated, and fluctuating nature of interconnected SLAs.

1.2.2 Automatic Adaptation to Environment Changes

Connections among services in a federated Cloud environment are dynamic. Cloud services can frequently initiate, abandon, or fail relationships. Therefore, SLAs in such an environment are required to be dynamic, and automatically adapt to the underlying contract changes, since any alteration to established relationships will have a *cascading effect* on other interconnected services, and hence on agreed SLAs. When a SLA specification distinctively captures the multi-level nature of federated Cloud environments, adaptation to changes becomes feasible. Yet, it requires some autonomous mechanisms to detect relationship changes, and to revise SLA specifications accordingly. Furthermore, fluctuating relationships among Cloud services entail SLA renegotiation at multiple levels. This also requires convenient communication channels, and coordination protocols among federated Cloud services. Similarly, once SLAs are updated for any reason, they will need to be redeployed following the newly agreed changes. Additionally, methods to validate, and distribute SLAs to the involved parties are required on different service levels. Implemented monitoring measures will need to be notified as well, as the originally defined parameters' thresholds will probably change too, following any changes on agreed SLA specifications. Moreover, SLA enforcement measures need to cope with SLA updates as well by tracing violations, not only to figure out inducing services, but also to identify time slots during which SLA violations have occurred. This is to facilitate the realistic enforcement of corrective actions on both previously and newly contracted services.

1.3 Problem Statement and Key Contributions

This dissertation addresses the following problem: *How to create a Cloud market place that mitigates the heterogeneity of Cloud providers in order to provide Cloud customers with variant choices of services, despite the dynamic and aggregated nature of the Cloud federation environment, and eventually maximizes customers' satisfaction without compromising Cloud providers' profit* who *collaborate together to provide value-added services.*

In the context of this problem, we propose a Cloud services provisioning model that intends to convey the federation of Clouds to the public market, and we introduce a specific SLA management model to be incorporated with a network of federated Clouds, *CloudLend*. Furthermore, we investigate how different phases of the SLA life cycle affect the way Cloud providers advertise their services, and how services form interconnections, synergize, and provide value-added services to the end users.

1.3.1 Problem Statement

The problem of enabling Cloud federation through portable APIs in order to provide value-added services is considered among the Cloud research community as a dynamic and complex one. Selecting the most appropriate Cloud service, considering a set of properties (e.g. acceptable quality, cost effective, fully available) to participate in a federation is a complex, multi-criteria, and multi-decision problem. A federation is required to match customer's requirements, through an aggregated selection of individual Cloud services from different providers who have different interests. Besides, connections among services in a federated Cloud environment are dynamic. Cloud services can frequently initiate, abandon, or fail relationships. Therefore, SLAs in such an environment are required to be dynamic, and automatically adapt to the underlying contract changes, since any alteration to established relationships will have a cascading effect on other interconnected services, and hence on agreed SLAs.

Additionally, in a federated Cloud environment, Cloud services interconnect with other services in order to fulfill tasks required by customers' QoS requirements. Such interconnections are not necessarily confined to a single level. They can be extended to reach further services in order to carry out minor subtasks. Also, a Cloud service can maintain connections with one or more other Cloud services at the same time; this will result in a chain of interconnected services that are bounded by multi-level SLAs. In such settings a SLA requires specific considerations, as follows:

1. A SLA specification in a federated Cloud environment requires a comprehensive description of the multi-level nature of connections among Cloud services. SLAs need to be defined in an aggregated manner, so that the complexity of the SLA chain is hidden from the customer. Yet, SLA's constituent parameters need to be specifically defined, and mapped each to its contributing services.
2. SLA negotiation requires a particular emphasis, as it is not restricted to Cloud provider-customer only. SLA negotiation occurs also between interconnected services, it requires the design of proper mechanisms to facilitate communication, and to manage service-to-service negotiation.
3. SLA monitoring: QoS parameters in a federated Cloud environment are monitored on multiple levels using a set of metrics, measured against thresholds on multiple levels. Therefore, monitoring methods designed for federated Cloud environments are required to implement specific mechanisms, which are able to capture and monitor the aggregated nature of interconnected SLAs.

1.3.2 Scope and Assumptions

The scope of this dissertation is related to SLA management in a network of federated SaaS Clouds. This type of Cloud computing service is concerned with providing software licenses to customers through different payment options, such as subscription, service on demand, or "pay-as-you-go" model. In addition, we examine SLA specification, monitoring, and negotiation phases of the SLA life cycle. Furthermore, the mechanisms needed to technically implement the Cloud services federation in the *CloudLend* network is not a concern in this research and the Cloud federation is assumed to be managed by the network.

1.3.3 Research Questions

This work is intended to answer the following research questions:

1. Can different Cloud providers collaborate to achieve communal benefits?
2. How are members' activities carried out in a collaborative federated Cloud environment?
3. How would a connection among Cloud services within *CloudLend* network be governed?
4. Will Cloud customers be privileged to imply their QoS requirements to Cloud providers?
5. How can Cloud services be portrayed within the *CloudLend* network?
6. How can a customer find the best service offer for his QoS requirements?
7. How can a provider evaluate different customers' requests to achieve efficient resource utilization?
8. How can service provisioning within the network be evaluated?

1.3.4 Research Contributions

This dissertation creates the following research contributions associated with the application of collaborative networking, and game theory concepts for SLA management in federated Cloud environments while realizing controverting objectives of Cloud providers and customers:

1. Proposes a collaborative-based Cloud federation network named *CloudLend* that is intended to enrich service provisioning for Cloud customers and providers.
2. Studies the life cycle of a Cloud service within *CloudLend*, and highlights the added value of participation in the network for both Cloud customers and providers.
3. Proposes an SLA management model for the *CloudLend* network:
    a. Defines a multi-level SLA specification model to describe QoCS.
    b. Defines a game theory-based automated SLA negotiation model which is capable of:
        i. Balancing the trade-offs among customers' various QoS requirements, as well as providers' resources utilization.
        ii. Prioritizing SLA terms, which are more important to both Cloud customers, and providers.
        iii. Supporting both customers, and providers in negotiating SLA terms, and guiding them towards signing a contract.
        iv. Assisting customers in service selection, by enabling evaluation of different service alternatives based on a computed utility gain.
    c. Defines an adaptive agent-based SLA monitoring model which:
        i. Evaluates Cloud services performance.
        ii. Identifies root causes of SLA violations.
        iii. Impartially distributes any updates and changes in established SLAs to all involved parties.
4. Provides a formal specification of the *CloudLend* network.

5. Provides a formal specification of the SLA negotiation, and defines customer's, and provider's objective functions.
6. Evaluates the efficiency of the proposed SLA management model within *CloudLend* by:
    a. Implementing a *CloudLend* network simulator that offers the following features:
        i. Creates a random *CloudLend* graph populated with members, and their specified profiles.
        ii. Evaluates the accuracy of the SLA negotiation model by measuring the satisfaction level of *CloudLend* members.
        iii. Evaluates the elasticity, accuracy, and autonomicity of the SLA monitoring model.

1.3.5 Research Questions to Contributions Mapping

Table 1: Research Questions to Contributions Mapping

<table>
<tr><th></th><th>Research Question</th><th>Contribution</th><th>Related publication</th></tr>
<tr><td>1</td><td>Can different Cloud providers collaborate to achieve communal benefits?</td><td>Proposed a collaborative-based Cloud federation network named CloudLend that is intended to enrich service provisioning for Cloud customers and providers.</td><td rowspan="2">(Al Falasi, Serhani, & Elnaffar, The sky: a social approach to clouds federation, 2013)</td></tr>
<tr><td>2</td><td>How are members' activities carried out in a collaborative federated Cloud environment?</td><td>Studied the life cycle of a Cloud service within CloudLend, and highlighted the added value of participation in the network for both Cloud customers and providers.</td></tr>
</table>

3	How would a connection among Cloud services within such a network be governed?	Proposed an SLA management model for the *CloudLend* network that includes: 1. A multi-level SLA specification model to describe QoCS. 2. A game theory-based automated SLA negotiation model. 3. An adaptive agent-based SLA monitoring model.	(Al Falasi, Serhani, & Dssouli, 2013) (Al Falasi, Serhani, & Hamdouch, 2015) (Al Falasi & Serhani, 2016)
4	Will Cloud customers be able to dictate their QoS requirements to Cloud providers?	Provided a formal specification of the *CloudLend* network.	Asma Al Flasi, M. Adel Serhani, "End-to-End QoS management in federated Clouds: *CloudLend*", will be submitted to the IEEE Transaction on Service computing, November 2016
5	How can Cloud services be portrayed within the *CloudLend* network?	Provided a formal specification of the *CloudLend* network, and the SLA game.	
6	How can a customer find the best service offer for his QoS requirements?	Defined a game theory-based automated SLA negotiation model which is capable of: 1. Balancing the trade-offs among customers' various QoS requirements, as well as providers' resources utilization. 2. Prioritizing SLA terms, which are more important to both Cloud customers, and providers. 3. Supporting both customers, and providers in negotiating SLA terms, and guiding them towards signing a contract. 4. Assisting customers in service selection, by enabling evaluation of service alternatives based on a	(Al Falasi & Serhani, 2016)

		computed utility gain.	
7	How can a provider evaluate different customers' requests to achieve efficient resource utilization?	Defined customer's, and provider's objective functions.	Asma Al Flasi, M. Adel Serhani, "End-to-End QoS management in federated Clouds: *CloudLend*", will be submitted to the IEEE Transaction on Service computing, November 2016
8	How can service provisioning within the network be evaluated?	Evaluated the efficiency of the proposed SLA management model within *CloudLend* by implementing a *CloudLend* network simulator that offers the following features: 1. Creates a random *CloudLend* graph populated with members, and their specified profiles. 2. Evaluates the accuracy of the SLA negotiation model by measuring the satisfaction level of *CloudLend* members. 3. Evaluates the elasticity, accuracy, and autonomicity of the SLA monitoring model.	(Al Falasi & Serhani, 2016)

1.3.6 Dissertation Organization

The rest of the dissertation is organized as follows: chapter 2 summarizes the relevant research works. Chapter 3 introduces the proposed federated Cloud network *CloudLend*. Chapter 4 presents the SLA specification and monitoring models for

CloudLend, while, chapter 5 introduces the game theory based SLA negotiation model. Meanwhile, chapter 6 provides a formal description of *CloudLend* and the SLA negotiation model. Subsequently, chapter 7 highlights the implementation and evaluation of SLA management models in *CloudLend*. Finally, chapter 8 concludes this dissertation and points out some future research directions.

Chapter 2: Review of Related Work

Cloud services are an evolved version of Web services, and composition of Web services is a form of service federation, that was experienced in the context of Web services. However, Web services composition is realized within a limited number of organizations that are participating in the static, and tightly-coupled, service federation. In contrast, the federation of Cloud services is considered a dynamic, loosely-coupled Internet-scale type of service composition (Zhou, Athukorala, Gilman, Riekki, & Ylianttila, 2012). This section summarizes the existing work on the composition of web services, it also surveys the federation of Cloud services, and finally reviews the SLA management on federated Cloud environments.

2.1 Federation of Web Services

Nowadays, Web services play a vital role in the world of businesses integration and collaboration by providing a distinct aspect of collaboration through their ability to be composed. The last decade marked exhaustive research efforts on approaches towards the composition of Web services. It has been considered to be a promising solution that would change the software engineering vision. The composition of Web services is defined as a set of atomic services together with the control and data flow among the services (Claro, Albers, & Hao, 2005). Essentially, it depends on the SOA model to transform granular individual services into value-added composite services in order to fulfill specific end-user’s preferences. Conversely, Web service composition is considered a very complex task to be handled manually by humans.

Generally, this complexity is explained by Dustdar & Schreiner (2005) through the following reasons: there exists a huge amount of Web services on the Web, and the number is increasing every day, which leaves us with an enormous Web services repository to dig into. Also, the ability to create and update web services on the fly forces composition systems to perceive such updates at runtime; since decisions should be made based on the latest information. Additionally, Web services can be described by different models, as they are developed by different organizations. This trend has led to a significant number of research efforts on the selection and composition of Web services, both from academia and industry (Ter Beek, Bucchiarone, & Gnesi, 2007), who have put forward various compositions' methods, techniques and algorithms that can be based on wrappers, workflows, languages, ontologies and declarations (Alamri, Eid, & El Saddik, 2006). Furthermore, with the rise of social networks and collaborative environments, an informal and more dynamic perspective of service composition has been introduced (Maamar, Hacid, & Huhns, 2011) (Maaradji, Hacid, Daigremont, & Crespi, 2010).

2.2 Federation of Cloud Services

This section classifies surveyed related work on the federation of Cloud services based on the common categorizations of federation: horizontal federation, and hybrid federation. It also reviews the research into a newly evolving class of federation known as social Cloud.

2.2.1 Horizontal Federation

This approach to federation refers to when multiple Clouds join a federation to share their resources; it takes place on a single level of the three Cloud deployment models, including SaaS, IaaS, or PaaS. The Reservoir project (Rochwerger, et al., 2011) is a European research initiative that falls into this category. This project aims to design a Cloud computing architecture, which serves as a potential foundation for delivering IT services as utility services over the Internet. They define a model and an open architecture for Cloud federation. The basic principle of their model is that each IaaS provider is an independent business entity, which can federate with other providers based on its own requirements. For example, during service provisioning resources involved in a federation may be moved to other providers based on performance, or availability considerations. However, the Reservoir model considers federation at IaaS level.

mOASIC (Petcu, Macariu, Panica, & Crăciun, 2013) is another project that can be classified under this category of federation, and it aims to provide a set of language independent APIs to enable portability across different Clouds. They provide a four-tier architecture (data, business, load balancing, and presentation) through which developers can build their applications with portability in mind. They aim to support the on-demand grouping of different Cloud services through a broker by postponing the decision of service selection until service run time. Nonetheless, mOASIC deals with portability at PaaS level mainly.

(Buyya, Ranjan, & Calheiros, 2010) is also a research initiative that aims to define the architectural elements of InterCloud: a utility-oriented federation of Cloud computing environments. They argue that the key elements to enable InterCloud federation are Cloud Coordinators, Brokers, and Exchange. Cloud Coordinators manage Cloud services and their federation with other cloud services through the implementation of resource management functionalities: scheduling, allocation, monitoring, discovery and composition. Meanwhile, Cloud Brokers act on behalf of customers to identify suitable Cloud services providers through Cloud Exchange, and to negotiate with other Cloud’s coordinators for the best allocation of Cloud resources that shall meet the required QoS. Cloud Exchange is a Cloud services directory that stores information on Cloud services. Nevertheless, the suggested framework requires Cloud services to be part of a predefined federation platform and to initiate a cloud brokering service beforehand. In contrast, The *CloudLend* promotes on the fly federation of Cloud services. Bernstein, et al., (2009) are also trying to promote the concept of InterCloud by constructing two Cloud Computing environments, one with proprietary hypervisors -virtual machine managers - and the other using open source hypervisors. They are investigating protocols and formats, which can implement Cloud interoperability between the two environments. These include: service addressing, service naming, identity management and trust, presence and messaging protocols, as well as virtual machine management. Cisco's vision of Cloud computing future involves promoting the term Inter-Cloud (Urquhart, 2009) or "the Cloud of Clouds", in which every Cloud is anticipated to be able to use the capabilities of the virtualized infrastructure of all other Clouds. Nevertheless, efforts by Buyya, et al., (2010), and Urquhart (2009) are considered visionary and are still under research.

2.2.2 Hybrid Clouds

In this category, applications are based on several services from different providers. Celesti, et al., (2010) propose a three-phase model for a cross-Cloud federation, which depends on specific agents assigned to perform Cloud's discovery, match-making, and authentication. However, their main focus is on issues related to Clouds authentication, and the ability to establish a secure connection among federated Clouds. Meanwhile, Keahey, et al., (2009) have introduced a similar concept called Sky Computing, where distributed IaaS resources are overlaid by a virtual site that constructs a Sky environment. Their work, however, is limited to interconnecting compatible IaaS resources over a private network that is used by academics for scientific research projects only. CompatibleOne (Yangui, Marshall, Laisne, & Tata, 2014) is an open source project that provides interoperable broker to describe federated Cloud services. The project defines an object-based model to describe IaaS, and PaaS Cloud resources, and its monitoring capabilities lacks adaptation to dynamic SLA changes.

Conversely, the *CloudLend* discussed herein aims to create a network of Clouds that interact among each other to form a collaborative network regardless of their type or class. In addition, by establishing ties across *CloudLend*, Cloud services are able to federate, and thus provide fused services to be used for various applications.

2.2.3 Social Cloud

There has been little discussion on enabling Cloud federation by the power of social networks. A few studies, such as those by Chard, et al., (2012); John, et al., (2011) and Mohaisen, et al., (2011) address the potential of resources sharing among the members of a public social network on a Cloud-based model, which is quite the contrary to the emerging class of collaboration that the *CloudLend* network introduces. Chard, et al., (2012), and John, et al., (2011) authors adopt the common social network model of members' grouping and classification in order to provide the basis for different trust levels to control and restrict resources allocation.

Mohaisen, et al., (2011) present a design for the Social Cloud, in which connected nodes are engaged in a contract-based relationship along with a local task scheduling utility. In such a relationship, one node is an outsourcer, while the other node is considered a worker. However, the established relationships are limited to one-hop neighbors only and do not go beyond immediately connected nodes. On the other hand, Iosup, et al., (2010) address the concept of utilizing Cloud computing to enrich the social networking experience. They present an architecture for the continuous analysis of Massively Multiplayer Online Gaming (MMOG) utilizing resources on Cloud. The architecture mines information from the web using APIs provided by the social network operators, and then integrates information into datasets, analyzing those datasets, and finally presenting application specific results. Essentially, their architecture utilizes Cloud services to manage, collect, store and process data from a social network.

2.3 SLA Management in Cloud Environments

Related work on SLA management in Cloud environment can be classified into two categories. The first includes research efforts conducted on Cloud specific SLAs management, and the second includes research efforts initiated on federated Cloud specific SLAs.

2.3.1 Cloud-specific SLAs

Alhamad, et al., (Conceptual SLA framework for Cloud Computing, 2010) discuss an architecture for SLA management in a Cloud environment. They focus on the definition of SLA parameters by specifying metrics for every Cloud computing service model (IaaS, PaaS, SaaS). When it comes to SLA negotiation, and monitoring, the authors suggest an agent-based architecture in Alhamad, et al., (SLA-based trust model for Cloud Computing, 2010). Following the selection of the desired Cloud providers using a Cloud services directory, the customer signs the SLA contract with the SLA agent and proceeds with the selected Cloud provider. The SLA agent is also responsible for monitoring the performance of the Cloud providers, in order to update the providers’ reputation accordingly. Their approach assigns SLA negotiation, and monitoring management to be handled by a single agent. CSLA (Nie, Xueni, & Chen, 2012) is another SLA model for Cloud services that is based on WSLA (Keller & Ludwig, 2003). The proposed model depends on a coordination model to manage customer's QoS definition, where agents handle the mapping of multiple requirements definitions required by the customer into one aggregated SLA document. Once the SLA is agreed upon, and service provisioning commences, a management model provides the means to: deploy the SLA, measure performance, evaluate the SLA, and handle billing. This model does not provide a clear specification of the SLA negotiation scheme. Also, the design of the CSLA model does not consider the dynamic nature of SLAs in Cloud environments. An architecture to support SLA-based service provisioning in the Cloud is introduced by Buyya, et al., (2011). An SLA resource allocator handles interactions between users and Cloud resources, by examining requested QoS, and controlling admission of requests to available resources. It also provides mechanisms for service pricing, and SLA monitoring. When it comes to SLA management, SLAs are defined in terms of time-specific deadlines to execute applications. Once a user request is received, it is examined for QoS, then matching resources will be scheduled accordingly. Resources are frequently monitored to assure that SLAs will not be violated. This architecture provides efficient SLA-based resource scheduling mechanisms, which guarantee no SLA breaches. However, it does not specify methods to define SLAs, or a scheme for users and providers to negotiate SLAs. Additionally, it lacks identification of SLA enforcement mechanisms.

2.3.2 Federated Cloud Specific SLAs

Torkashvan & Haghighi, (2012) propose a WSLA-based SLA management framework for inter-cloud environments, where SLA parameters are defined using an XML-based language that is specifically designed to represent SaaS-related metrics. The framework also includes a monitoring component that is responsible for detecting SLA violations based on data included in an SLA log, and for inspecting the specified thresholds for every parameter. However, the framework does not specify how SLA negotiation is carried out, and furthermore the effect of federated Cloud services on SLA management is only considered on the definition phase of the SLA life cycle.

SLA@SOI (Wieder P. , Butler, Theilmann, & Yahyapour, 2011) is a framework that aims to introduce a holistic SLA management solution for service-oriented environments, which covers the complete SLA life cycle. The proposed architecture depends fundamentally on two models. One describes SLAs to allow the specification of both functional and non-functional QoS requirements, while the other facilitates communication among components involved in the SLA management. Additionally, SLA@SOI architecture is realized by different components that are used to: 1) manage business relations and policies, 2) manage SLA templates, negotiation, provisioning, and adjustment, 3) retrieve predictions of service performance, 4) invoke service implementations, and orchestrate provisioning activities, and 5) observe and monitor service status. The SLA@SOI project addresses key issues in SLA management. However, it is adopted by business entities as a supporting service management model in controlled enterprise-like environments. Additionally, although management of composed SLAs is considered within the framework, service discovery and binding are assumed to be arranged in advance between the business entity and Cloud provider.

To the best of our knowledge, there is no holistic SLA management model that specifies and administers SLAs in a federated Cloud services environment. Due to the complexity inherited with such environments, we assume that an SLA management model for a federated Cloud environment should consider the following requirements:

- Reflect the composite nature of the federated Cloud environment in managing multi-level SLAs.
- Be able to hide the complexity of SLA management from both customers and providers of Cloud services.
- Be able to identify origin of service interruptions caused by SLA violations in a chain of SLAs.
- Implement adaptive SLA mechanisms, which cope with dynamic underlying changes of SLAs, and relationships.
- Implement dynamic SLA validation and deployment methods.

Ensuring the exhaustive study of research efforts in the area, the *CloudLend* introduced here conversely aims to create a network of heterogeneous Clouds, which interact among each other to form a collaborative network of Clouds. By establishing ties across the network, Cloud services are able to federate, and thus provide fused services to be used for various applications. Further, the proposed SLA management model aims to address these objectives in a collaborative-based federated Cloud environment.

2.3.3 Game Theory for SLA Negotiation

Game theory is intended to optimize negotiation outcomes using various initiation conditions (Binmore & Vulkan, 1999). Research in this area is not concerned with the characteristics of the negotiation process itself, nor with the interaction between involved parties. Conversely, the emphasis is mainly on the outcome of the negotiation process. Hence, game theory outcomes are utilized to evaluate the satisfaction level of different options of an optimal solution, to any given negotiation game. This section reviews the research efforts on the adoption of game theory for SLA negotiation in Cloud, Grid, and Web service computing.

A bargaining game approach by Zheng, et al., (2010) describes an automated one-to-one web services SLA negotiation mechanism, while Alsrheed, et al., (2013) apply another bargaining game for an automated SLA negotiation in Cloud computing. Both approaches consider a game of only two players, and assume that players have complete information on the possible strategies, in addition to corresponding outcomes of their opponents. In reality, such an assumption is not always true. Figueroa, et al., (2008) introduce a mathematical negotiation model for high-performance computing (HPC). Their approach is based on signaling game in two rounds. Unlike our strictly competitive Fair Division approach, signaling is either competitive or cooperative. Silaghi, et al., (2012) address the problem of resource allocation in competitive grids. Their negotiation strategy can achieve a fair resource allocation. Nevertheless, SLA negotiation in grid, and HPC is not the same as SLA negotiation in Clouds. It is less complicated, as it involves specific users interested in some resource. Whereas in Cloud environments, complexity of negotiation is driven by market competition. Yaqub, et al., (2011) describe a generic

SLA negotiation platform for the SLA@SOI (Wieder P. , Butler, Theilmann, & Yahyapour, 2011), a framework for service-oriented environments. Yet, they address the enabling of SLA negotiation protocols. In this work, we focus on SLA negotiation strategies. A dynamic game for SLA negotiation in Cloud is presented in (Chen, Liu, Xu, & Wang, 2016), where a Cloud customer, and provider negotiate a single SLA term through a broker by measuring their satisfaction degree at every round of the game, until satisfaction difference is minimized.

To the best of our knowledge, there is no automated SLA negotiation model that assures the fairness and efficiency of the SLA negotiation in a federated Cloud services environment. Our approach is based on the Fair Division game (Brams & Taylor, 1996), which is a sequential game that allows multiple players, and assumes complete knowledge of all previous events that have occurred prior to a player's decision. The properties of this game makes it very suitable to be implemented in a dynamically changing, and complex, environment with multi-level relationships such as *CloudLend*.

Chapter 3: *CloudLend* a Network of Clouds

This chapter encompasses our first two contributions and introduces the concept of *CloudLend*: a federation of Clouds. It also defines its main components, describes how its community evolves, and introduces its SLA management model.

3.1 *CloudLend* Overview

CloudLend is a Cloud federation network that implements collaborative networking principles to optimize Cloud services provisioning for both Cloud customers and providers. The *CloudLend* concept is different than the concept of Social Cloud, which has been defined by different authors in the literature. Pezzi, (2009), Chard et al., (2010), (2012), (2016) and Chard & Caton (2015), they all refer to the particular notion of social-based service provisioning where human members of a social network site are able to publicize their computing resources while creating a Cloud service model on top of the social network site. However, we define a collaborative-based Cloud federation network, *CloudLend* as: "a network of Cloud services that are able to interact and collaborate; creating a next generation Internet where resources are infinite, data is boundless and transactions are human-less."

To characterize the *CloudLend* network we use an analogous approach to social networking. *CloudLend* can be perceived as a merge of classic social network sites such as Twitter, Facebook, and LinkedIn, augmented with a business model that facilitates exchanging services among the peer Clouds forming the members of the network. Looking up Cloud services to utilize, collaborate with or compete against are examples of key operations that can take place in this Cloud marketplace. To analyze the features of *CloudLend*, we examine the prominent features of human-based social networks, which are summarized by Boyd & Ellison, (2007) as follows:

1. **Profile construction:** a user profile generated from questions upon signing up.
2. **Connections identification**: initiating relationships with users on the network through network lookup, profile similarities or importing user's profile from other online communities such as Open ID, (2012) and OpenSocial, (2012) that provide cross-social network sites interoperability.
3. **Connection maintenance:** users tend to dynamically modify their relationships with others based on experience, change of interests, or the discovery of new connections.
4. **Privacy controls:** determines to what extent user identity is exposed to other members.

The collaborative network infrastructure is a key aspect of the *CloudLend* network as it represents the platform on which Cloud services communication is realized. However, prior to proposing an architecture of the *CloudLend* network, it is useful to examine the existing structure of traditional collaborative networking sites first. According to Kim, et al., (2010) a typical architecture of collaborative network sites implements the classical multi-tier client/server architecture, augmented with load balancers, memory caches and partitioned databases in order to scale up efficiently and meet performance requirements. This architecture supports all commonly known functions such as users profiling, connection management, user collaboration, search and exploration.

3.2 *CloudLend* Architecture

The *CloudLend* network incorporates basic collaborative network functions by which Cloud services exploration, listing and matching are carried out. This inspires us to adopt an architecture that is similar to the traditional collaborative networks. More specifically, the proposed architecture depicted in Figure 1 is comprised of two modules: *Collaboration* and *Federation*, administered by a *CloudLend Broker* who receives federation requests from customers, and handles communications across the *Collaboration* and *Federation* modules.

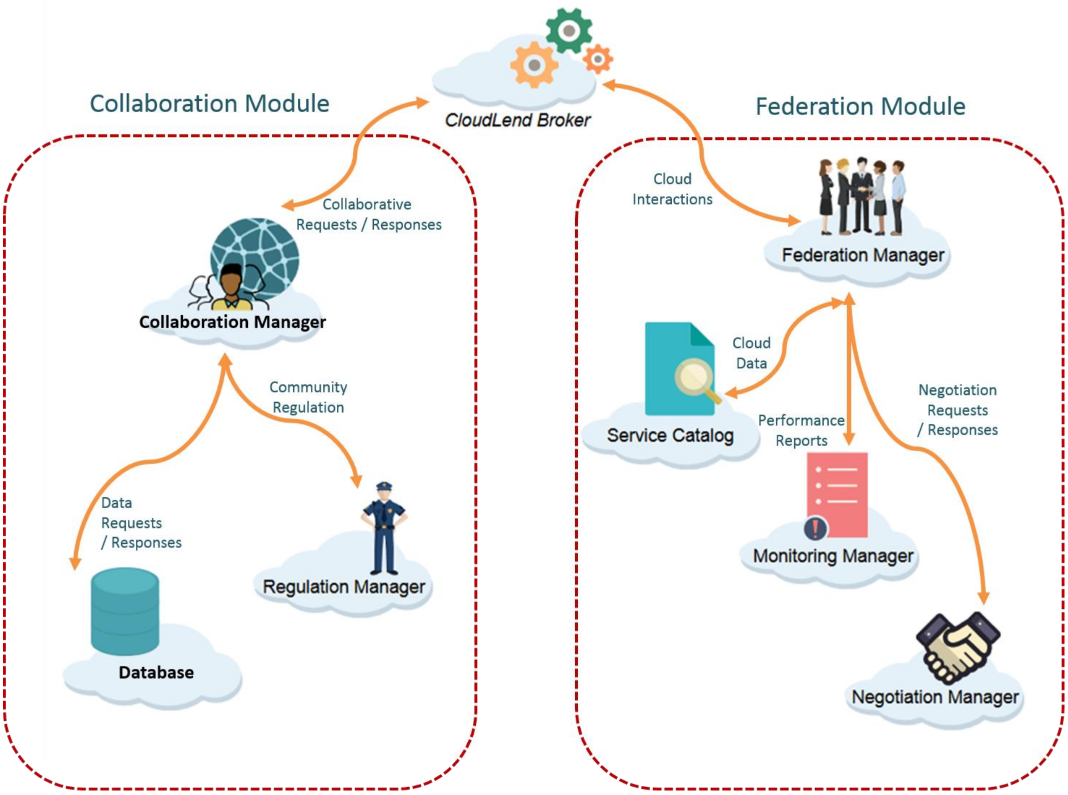

Figure 1: *CloudLend* Architecture

The brokering architectural pattern assures decoupling of the *CloudLend* modules, hence components are able to perform independently. This architecture also facilitates cross-modules communications. Each *CloudLend* module has its specific functions, and responsibilities. The *Collaboration Module* sets the ground for Cloud services to collaborate with each other by providing essential collaborative networks features and properties. It is mainly composed of three entities:

1. **Collaboration Manager:** implements the *CloudLend* business model, by managing links, and overseeing community evolvement.
2. **Database:** stores and manipulate data pertaining to the identity of Cloud services, their relationships, and the collaborative activities they perform on the network.
3. **Regulation Manager:** is responsible for regulating the *CloudLend* community by managing memberships, enforcing rules, granting rewards, intercepting violations, and imposing penalties.

On the other hand, the *Federation module* realizes the actual federation among Cloud services, and is composed of three entities:

1. **Service Catalog:** An information directory for Cloud services that manages their SLA profiles. SLA profiles hold information on Cloud services interfaces, and identifiers.
2. **Negotiation Manager:** Receives negotiation requests from the *CloudLend Broker*, and manages SLA contract negotiation by processing the initiator's SLO and the attendant's SLA, then implementing a game theory negotiation technique in order to reach a mutual agreement.
3. **Federation Manager:** An agent that receives federation requests from the *CloudLend Broker*, and performs the necessary tasks to enable the federation of Cloud services. Generally, a *Federation Manager's* tasks include the following:
    a. **Cloud services APIs retrieval:** The *Federation Manager* communicates with the *Service Catalog* in order to obtain the respective APIs of both peers involved in the relationship: the initiator, and the attendant Cloud services.

b. **Federation administration:** The *Federation Manager* initiates service federation, and forwards APIs, contracts, and other required credentials to the *Monitoring Manager* to supervise run-time operations. Additionally, upon completing the federation execution, the *Federation Manager* reports back to the *CloudLend Broker* in order to process billing, and relay performance results to the *CloudLend Regulation Manager*.

4. **Monitoring Manager:** Maintains federation throughout service provisioning. It monitors performance, ensures that SLAs are honored, and QoS attributes are maintained.

The interaction among different components of the *CloudLend* network is illustrated in Figure 2, and is discussed in the following section.

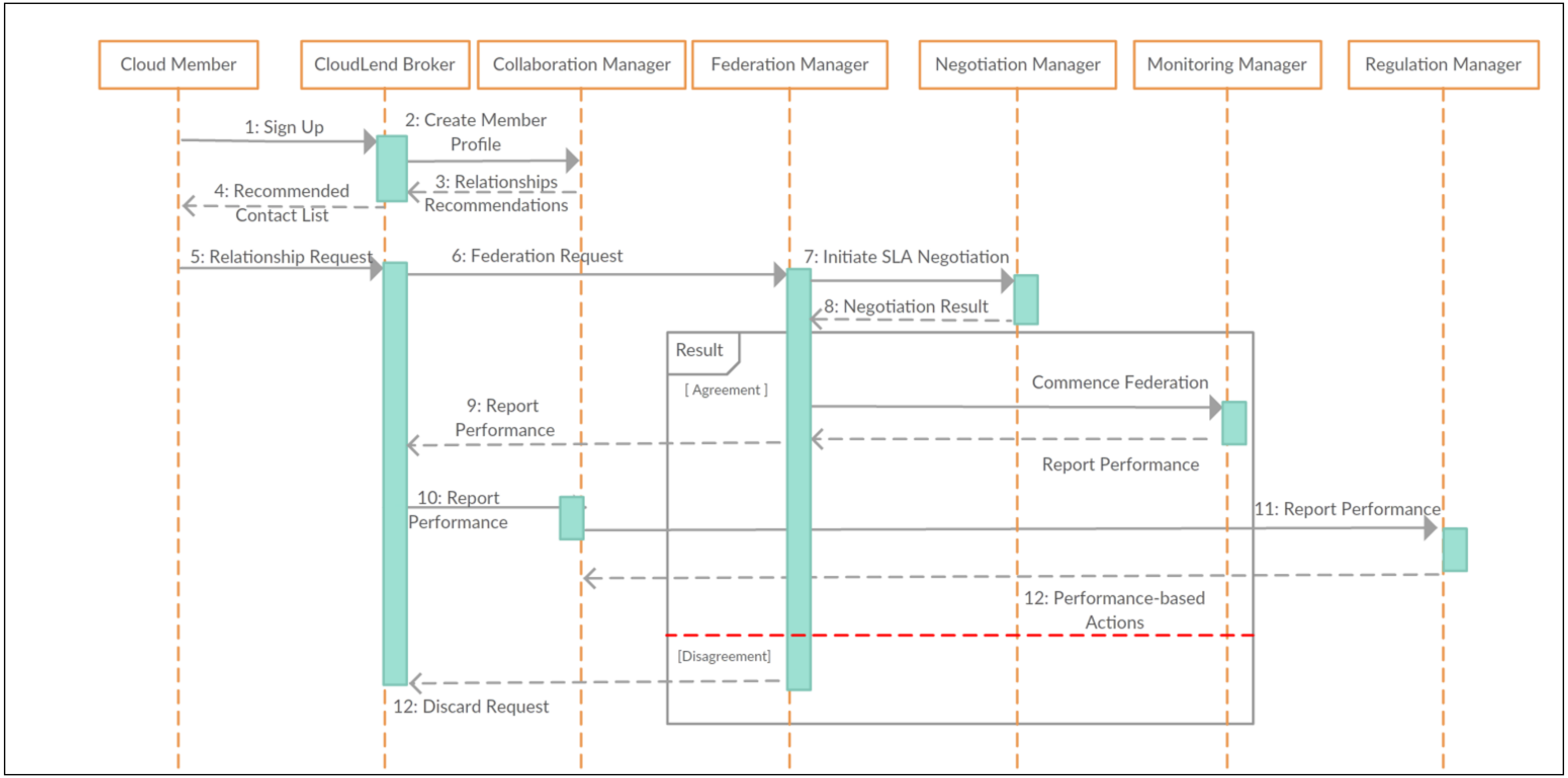

Figure 2: *CloudLend* Components Sequence Diagram

3.3 *CloudLend* Community

Cloud services with their distinct capabilities, and interfaces constitute the population of *CloudLend*. Residing in the *CloudLend* community gives each single Cloud service the opportunity to tap into the web of ties with other Clouds, in order to respond to persistent needs, such as service replacement, load balancing, request delegation, and performance guarantees. Like any community, *CloudLend* consists of regular members (Cloud services, customers, and providers), governing entities (e.g., *federation, collaboration, and regulation managers*), in addition to policies that keep the dynamics of the community under control.

3.3.1 Member's Profile Construction

Cloud customers, and providers first sign up to be members of the *CloudLend* community. *CloudLend* members then start populating their profiles with different information. Cloud customers provide information on service interests, while providers provide information on their collaboration interests, as well as information on their offered SaaS resources. The provider completes a Cloud service profile that includes both technical, and networking related information, such as service identifiers, APIs, description, areas of application and contexts, and possible areas of collaboration with others.

3.3.2 Relationships Identification

Relationships in a typical collaborative network are usually classified to reflect how members perceive each other. Individuals may identify others on the network as a family member, friend or a colleague, whereas in *CloudLend*, relationships can be looked at from the viewpoint of usage and involvement. In general, relationships in *CloudLend* can be classified as follows:

1. **Collaboration**: working together with an affiliate to achieve a certain goal.
2. **Competition**: striving to exclusively win a client's contract.
3. **Substitution**: replacing a service by another implementation of an equivalent or better service.
4. **Recommendation**: predicting counterpart services that the user had not yet considered based on his behavior on the network.
5. **Supervision**: directing affiliate services during the performance of a composite task.

Based on a community member's profile, and interests, *CloudLend* network recommends a list of Cloud services that might appeal to the newly joined member. The proposed list is generated by the *Collaboration Manager* so that these members get to select services to establish a relationship with. The *CloudLend* network also offers community members the opportunities to explore, and search for new peers to initiate new links with, based on provided QoS requirements.

3.3.3 SLA Negotiation

Prior to any relationship establishment, SLA negotiation takes place. The *Negotiation Manager* initiates a negotiation session based on a customer request, including customer's SLO, and provider's SLA. A customer's SLO can be composed of one or more QoS measurements. For example, an availability SLO may depend on several components, each of which can have a QoS availability measurement.

If the negotiation session ends up by reaching an agreement, then a relationship is established. On the other hand, in the case of disagreement, then members can choose either to reconsider their QoS, and renegotiate, or they may decide to discard the whole negotiation.

3.3.4 Service Provisioning and Monitoring

Service provisioning in *CloudLend* takes place after relationship establishment, when a federation request is received by the network, and subsequently forwarded to a *Collaboration Manager*. That in turn will provide the required information to be communicated afterwards to a *Federation Manager*. This will ensure that requirements of the federation are met, and will commence the federation. Next, the federation is handed over to a *Monitoring Manager* that monitors the service's performance in order to ensure that contracted SLAs are respected. In case a Cloud provider fails to conform to an established SLA, *CloudLend* network replaces the failing provider with another replacement Cloud provider. SLA migration is executed during a period that is equivalent to the service downtime specified in the initial SLA. Therefore, the SLA migration process is transparent to the customer.

When the federation is released, a performance report is prepared by the *Monitoring Manager*, and is communicated to a *Regulation Manager* in order to react by either granting rewards or imposing penalties based on the performance report.

3.3.5 Community Regulation

In human collaborative networks, people typically rely on trust derived from real world relationships. However, such a sentiment-based mechanism of trust is not applicable in the computing model of *CloudLend*. Rather, the *CloudLend* community banks on quantitative measures that are derived from the Cloud services' observed QoS, and the reputation of each Cloud service involved in a relationship. Based on these measures, a community member may choose to terminate a relationship with another member, leave the community (e.g. be unable to remain competitive), or initiate new ties with other evolving members. Looking after these relationships is important, and therefore, *CloudLend* provides its community members with the tools that allow them to track their subscription lists by sending change/update notifications, configuring performance thresholds required to maintain a relationship, and periodically recommending potential candidates in order to broaden the scope of interactions.

Furthermore, to ensure the sustainability of a community, members need motivation to remain actively engaged, to behave properly, and to add value to the whole community. Some incentives that can exist in *CloudLend* are precedency to specific community services, reputation gain, monetary rewards or access to exclusive functions, and services. Likewise, penalties can be applied to members who misbehave or violate contracted SLAs. Examples of penalties are monetary fines, reputation degradation, services deprivation, or even exclusion from the community.

3.4 *CloudLend* SLA Management Model

In any federated environment, it is essential for customers to receive guarantees on service delivery from Cloud providers. Such guarantees are provided through SLAs. SLAs govern, and control service provisioning between customers and Cloud providers. Therefore, we have introduced an SLA management model for federated Cloud environments. We studied the life cycle of SLA in our *CloudLend* network as an example of a federated Cloud environment, and proposed SLA specification, monitoring, and negotiation models. Components of the SLA management model in *CloudLend* network are deployed within the *Federation Module,* as illustrated in Figure 3.

For the purpose of this research, we consider SLA as a formal specification of both functional and non-functional QoS requirements, by which services are to be provided. Likewise, we describe SLA management as the management of SLA through its life cycle to fulfill QoS terms that bind a relationship between two Cloud services. In SaaS Cloud deployment model non-functional QoS attributes may include: availability, down time, response time, repair time, denial of service, user threshold level, and data requests threshold level. Besides other applications specific functional QoS attributes.

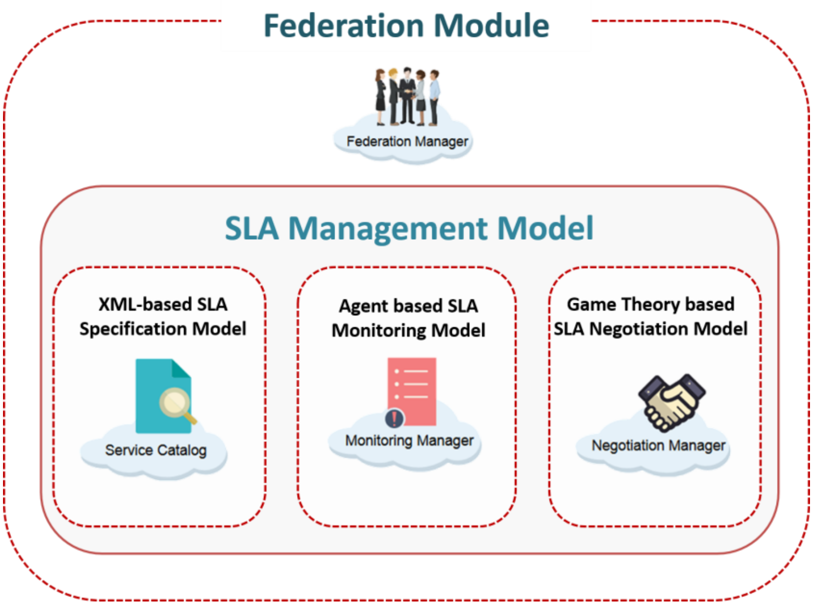

Figure 3: Components of *CloudLend's* SLA Management Model

3.4.1 SLA Specification Model

SLAs in *CloudLend* are specified using XML (W3C, 2015). XML provides a common syntax for interoperability among Cloud services within the network. It is used to accurately describe a *CloudLend* member's profile, identifies its QoS requirements, and defines its relationships. We propose an XML-based SLA specification model that contemplates the challenging characteristics of *CloudLend*. The SLA specification model is developed and thoroughly detailed in chapter 4.

3.4.2 SLA Monitoring Model

Monitoring Cloud services bounded by multi-level SLAs is very challenging. A monitoring model needs to: retrieve metrics from multiple levels, perform SLA parameters calculation, and then measure them against different thresholds. Not to mention the complexity added by the dynamic nature of the *CloudLend* network, where relationships can be created or terminated spontaneously. We introduce and describe an SLA monitoring model that address these challenges in chapter 4.

3.4.3 SLA Negotiation Model

SLA negotiation takes place prior to federation establishment. It is a mutual decision making process for the purpose of resolving providers' and customers' conflicting objectives (Dastjerdi, 2013). Many recent research efforts in SLA negotiation in Clouds have invested in the adoption of SLA negotiation approaches of Grid computing (Silaghi, Şerban, & Litan, 2012), Web services (Nie, Xueni, & Chen, 2012), (Torkashvan & Haghighi, 2012), and SOA (Wieder P. , Butler, Theilmann, & Yahyapour, 2011). Some others have opted for intelligent software agents (Alhamad, Chang, & Dillon, 2010), (Buyya, Garg, & Calheiros, 2011) and game theory (Alsrheed, El Rhalibi, Randles, & Merabti, 2013), (Zheng, Martin, Powley, & Brohman, 2010). The SLA negotiation model we propose in chapter 5 employs the principals of game theory in order to result in an efficient SLA negotiation.

3.5 Summary

CloudLend is described as a network of federated Clouds along with their customers. The architecture of this network is mainly comprised of the *Collaboration* and *Federation* modules, which are supervised by the *CloudLend Broker*. This architecture leads to the creation of service computing community which has its own life cycle and dynamics represented by the existence and fading of relationships among its service members. Managing SLAs in a federated Cloud environment such as *CloudLend* is a challenging problem, as it involves heterogeneous QoS definitions, conflicting members' objectives, and collective performance measures. An SLA management model for *CloudLend* was introduced in this chapter, and will be further explained in Chapters 4 and 5.

Chapter 4: SLA Specification and SLA Monitoring Models in *CloudLend*

Considering SLA specification in a federated Cloud environment requires a comprehensive deliberation of the multi-level nature of connections among Cloud services. This chapter includes two parts of the third contribution of this thesis, and tackles the complexity of managing multi-level SLAs during Cloud service provisioning within *CloudLend*. SLAs need to be defined in an aggregated manner, so that the complexity of the SLA chain is hidden from the consumer. Yet, constituent parameters need to be specifically defined, and mapped to each of its contributing services. Likewise, SLA monitoring is also very challenging in a federated Cloud environment. It necessitates measuring SLA parameters against different thresholds on multiple levels. Therefore, monitoring methods designed for federated Cloud environments are required to implement specific mechanisms, which are able to capture the aggregated nature of interconnected SLAs.

Therefore, we propose a multi-level SLA specification model that captures the aggregated nature of SLAs in *CloudLend*. We also propose an agent based monitoring model that contemplates the challenging characteristics of *CloudLend*. SLA management in the *CloudLend* network takes place within the *Federation Module*, where a *Federation Manager* handles SLA definition, and provisioning. However, a *Monitoring Agent* is responsible for SLA monitoring.

4.1 SLA Specification Model

SLAs in *CloudLend* are specified using XML (W3C, 2015). We have selected XML language to specify SLAs since it provides a common syntax for interoperability among Cloud services within the network. It is used to accurately describe a *CloudLend* member's profile, identifies its QoS requirements, and defines its relationships. SLA specification takes place within the *Federation Module,* where a *Federation Manager* handles SLA definition to be published in the *Service Directory*. We propose an XML-based SLA specification model that provides an SLA definition scheme as described below:

1. For every Cloud service in *CloudLend*: an SLA profile is published for other Cloud services to view. This public profile is used by the *CloudLend* network to facilitate Cloud services selection, and match making. Once a customer selects a Cloud service to utilize, SLA negotiation, and binding, occurs. The public SLA profile illustrated in Figure 4 includes the following specifications:
    a. Information related to the Cloud service: service name, type, provider, and reference to the service implementation interfaces.
    b. Information on QoS terms, and their assigned weights: terms' weights indicate the percentage of how much a *CloudLend* member values preserving his specified parameters of each SLA term, out of the total provided SLA terms.
2. For every relationship established between two Cloud services within a service federation in *CloudLend*, an SLA document is generated. The latter includes the following specifications:
    a. Information on both services engaged in the relationship: service name, type, provider, and reference to the service implementation interfaces as described in Figure 5.

b. Information on the agreed relationship: reference, type, initiator service, hired service, time of creation, and validity period as described in Figure 5.

c. Information on QoS terms: name, parameters, and allocations as described in Figure 6.

If the hired services - parent service - itself needed to hire another service - child service - to realize an SLA term, then the originally hired service shall maintain a reference to that sub-SLA document. References to both parent and child SLAs are maintained by an *SLA Management Service* implemented by the *CloudLend* network, which holds records on all established relationships on the network. In case of unexpected relationship changes, the relevant SLA management service instance is notified, so that the required measures are taken to revise affected SLA terms.

```
<interfaceDeclrs>
    <sla:InterfaceDeclr>
        <name>servicename</name>
        <provider>xxx</provider>
        <consumer>xxx</consumer>
        <endpoints>
            <sla:Endpoint>
                <name>epx</name>
                <location>xyz.com</location>
                <protocol>xxx</protocol>
            </sla:Endpoint>
        </endpoints>
    </sla:InterfaceDeclr>
</interfaceDeclrs>

<agreementTerms>
    <sla:AgreementTerm>
        <name>AT1</name>
        <id>AT1_id</id>
        <weight>30</weight>
        <guarantees>...</guarantees>
    </sla:AgreementTerm>
    <sla:AgreementTerm>
        <name>AT2</name>
        <id>AT2_id</id>
        <weight>70</weight>
        <guarantees>...</guarantees>
    </sla:AgreementTerm>
</agreementTerms>
```

Figure 4: A sample of a public Cloud service SLA profile specification

```
<participants>
    <sla:Participant>
        <name> service_x </name>
        <id> x_id </id>
        <role>provider</role>
    </sla:Participant>
    <sla:Participant>
        <name> service_y </name>
        <id> y_id </id>
        <role>customer</role>
    </sla:Participant>
</participants>

<relationship>
    <sla:Relationship>
        <id> rcs_xxx </id>
        <agreedAt> 12/02/2016</agreedAt>
        <effectiveFrom> 01/03/2016 </effectiveFrom>
        <effectiveUnitl> 30/03/2016 </effectiveUnitl>
    </sla:Relationship>
</relationship>
```

Figure 5: A sample of the post-negotiation relationship specification

```
<agreementTerms>
    <sla:AgreementTerm>
        <name>AT1</name>
        <id>AT1_id</id>
        <allocations>
            <sla:allocation>
                <id>x_id</id>
                <allocationPortion>35%</allocationPortion>
                <guarantees></guarantees>
            </sla:allocation>
            <sla:allocation>
                <id>y_id</id>
                <allocationPortion>65%</allocationPortion>
                <guarantees></guarantees>
            </sla:allocation>
        </allocations>
    </sla:AgreementTerm>
</agreementTerms>
```

Figure 6: A sample of a post-negotiation SLA terms specification

4.3 SLA Monitoring Model

Monitoring Cloud services bounded by multi-level SLAs is very challenging. A monitoring model needs to retrieve metrics from multiple levels, perform SLA parameters calculation, and then measure them against different thresholds. Not to mention the complexity added by the dynamic nature of the *CloudLend* network, where relationships can be created or terminated spontaneously. Therefore, we propose an agent-based SLA monitoring model that considers the above challenges. Agents are independent, problem solving computational entities that are capable of effective interaction and collaboration with other agents in dynamic and open environments (Luck, Ashri, & d'Inverno, 2004). Agent-based software development provides a level of autonomy for distributed and dynamic systems like *CloudLend.*

The proposed agent-based monitoring model is described in Figure 7 as follows:

1. For every established SLA, an instance of the *Monitoring Manager* is created, and two monitoring services are initiated:
    a. **Detection Service:**
        i. To collect necessary relationship runtime information.
        ii. To perform periodic SLA inspection with the SLA management service, so that changes in SLA terms can be detected.
    b. **Evaluation Service:**
        i. To evaluate performance parameters against specified SLAs.
        ii. To report relationship evaluation results to the *Monitoring Coordinator.*

Information on other dependent SLAs in a federation is retrieved from the *SLA Management Service*, which triggers a *Monitoring Coordinator* that is responsible for:

1. Collecting different dependent SLA evaluation reports.

2. Performing the required analysis in order to make sure that parent SLA terms are maintained.
3. Broadcasting any updates and changes in established SLAs to all involved parties.

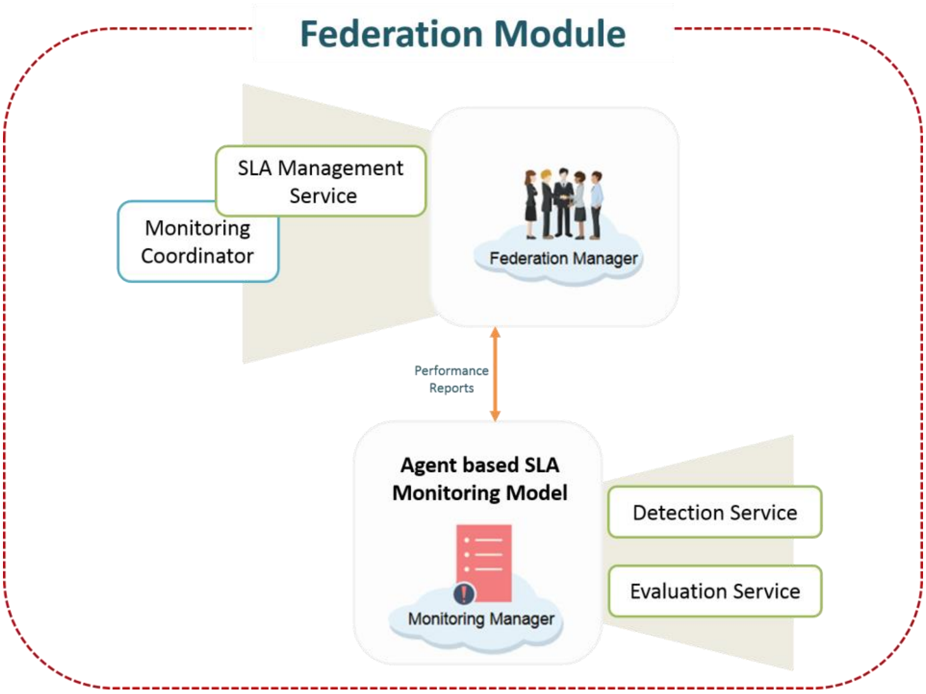

Figure 7: Componenets of *CloudLend's* SLA Monitoring Model

4.4 Summary

This chapter introduced two components of the SLA management model for *CloudLend*: an XML-based SLA specification model, and an agent-based SLA monitoring model. The SLA specification model distinctively captures the multi-level nature of federated Cloud environments. Hence, adapting to fluctuating relationships becomes feasible, while the SLA monitoring model performs periodic SLA inspections, reacts efficiently to identify the source of any violation, performs the required analysis to make sure that parent SLA terms are maintained, and updates all dependent Cloud services.

Chapter 5: A Game Theory based Automated SLA Negotiation Model

Typically, in Cloud computing, Cloud providers define their SLAs, and publish them for customers in a take-it-or-leave-it manner. Customers are not privileged with an adequate SLA negotiation opportunity that enables them to impose their QoS requirements on Cloud providers. In a federated Cloud environment such as *CloudLend*, this problem is magnified, since *CloudLend's* members intend to establish connections with others across the network. Such interconnections can be multileveled and established concurrently which results in a chain of interconnected services that are bounded by multi-level SLAs.

Henceforth, there exists a need for an automated negotiation model that fairly enables federated Cloud services to review SLAs, respond to SLA offers, and eventually sign an SLA contract. A negotiation model is required to facilitate the negotiation process while considering the complexity of services interconnections within the *CloudLend* network, ensuring that negotiation on multiple levels does not burden the federated network, and does not impede the resources utilization of Cloud providers, nor overlooks customers' QoS requirements. Thereupon, motivated by the lack of automated SLA negotiation models in federated Clouds, this chapter highlights the key contribution of this thesis, and aims to achieve the following research objectives:

1. Propose a game theory-based automated SLA negotiation model in *CloudLend* network, which is capable of:
    a. Balancing the trade-offs among customers' various QoS requirements, as well as providers' resources utilization.

b. Prioritizing SLA terms, which are more important to both Cloud customer, and provider.

c. Supporting both customers and providers in negotiating SLA terms, and guiding them towards signing a contract.

d. Assisting customers in service selection, by enabling the evaluation of different service alternatives based on a computed utility gain.

e. Evaluating the efficiency of the proposed SLA negotiation model in a federated Cloud environment, *CloudLend*.

5.1 Model Description

This section describes and illustrates the game of SLA negotiation in *CloudLend*. In such a network, Cloud services participate in the SLA negotiation game not to ultimately win the game. Conversely, they aim to reach the best collection of SLA terms that would satisfy all players' requirements. The outcome of the game is basically a measure of the value a Cloud service gains by establishing a relationship with other players. In game theory, this outcome is known as *utility*. Players negotiate SLAs to evaluate the expected utility from the anticipated relationships, which is used then to make the decision whether to establish the relationship or not. We introduce an SLA negotiation model that considers the SLA contract as a whole entity that consists of several SLA terms. Therefore, during negotiation, players bargain over the value of SLA terms that make up the utility gain of the whole SLA contract. Every player values each individual SLA term differently. Eventually, both players need to decide on the impact every SLA term has on the total value of the SLA contract.

We consider the SLA negotiation problem in *CloudLend* to be a fair division game (Brams & Taylor, 1996). Such games involve players in a sequential game, where they need to decide on how to divide an item. Every player values the item to be shared among them differently. An example of a fair division game is called *Fair Cake-cutting* (Brams & Taylor, 1996), in which a cake with different toppings must be divided among many players, who have different preferences over different parts of the cake. The division needs to be fair to every player. In this case, each player receives a slice that he believes to be a fair share. In our case, the SLA contract is a resource that is compiled of several different SLA terms. During negotiation, players will evaluate every SLA term differently. Each player knows the value of a single SLA term to him. Eventually, players need to reach a consensus on how much of a value is assigned to every single SLA term, out of the overall value of the SLA contract. In such situation, where a set of items is to be divided among players, yet these items themselves need to be kept as a whole; a proportional and envy-free division procedure is used (Brams & Taylor, 1996). The Adjusted Winner procedure (AW) (Brams & Taylor, 2000) is one of the proportional and envy-free division procedures, assuming players are rational. Once played out, the outcome is proven to exhibit three important properties:

1. Pareto optimal: any alternative allocation of items that improves one player's outcome will worsen the others.
2. Envy-free: each player is allocated a share of items that is at least as large, or at least as desirable as that received by any other player.
3. Equitable: every player believes that his allocation is valued the same as the other player's (based on their declared ratings).

The AW procedure describes a fair division of a set of ***n*** items that can be shared between two players. Each player examines the ***n*** items, and assigns a rate for each individual item, out of a total of 100 points among them all. These points are a relative preference of the players for the various rated items. We adopt the AW procedure as the most appropriate model of SLA negotiation in *CloudLend*. A list of essential elements of such an SLA negotiation game is described as follows:

1. **Players**: are the decision makers. Each has a goal to maximize his/her utility by his/her choice of actions. In *CloudLend* players are: Cloud customers, and Cloud providers.
2. **Actions**: are choices available for players to make. In *CloudLend* players' possible set of actions includes: place an SLA offer, accept an SLA offer, reject an SLA offer, place an SLA counter-offer, and end an SLA negotiation.
3. **Strategy**: of a player is a rule that tells him which action to choose at each instant of the game, given his information set about the game and other players. In *CloudLend* a player's strategy is represented by: ratings of SLA terms.
4. **Outcome**: the result of a player deciding to settle on a particular strategy, measured numerically. In *CloudLend* the outcome of the negotiation game is: allocation of SLA terms.

5.1.1 Example of SLA Negotiation using Fair Division Game

The following example explains how the AW procedure works when implemented in the SLA negotiation process within *CloudLend*. Let two Cloud services S_1 , and S_2 be negotiating an SLA contract. The contract specifies 6 different SLA terms $T = \{ t_1, t_2, t_3, t_4, t_5, t_6 \}$. Both services S_1 , and S_2 are players in the SLA negotiation game that goes as follows:

1. Both services S_1, and S_2 rate every term in T out of 100 score among them all, as described in Table 2.

Table 2: Services' Ratings of SLA Terms

SLA Term	S_1 Ratings	S_2 Ratings
t_1	25	30
t_2	12	15
t_3	30	30
t_4	6	9
t_5	7	9
t_6	20	7

1. Start the initial allocation of SLA terms to players S_1, S_2. S_1 is allocated the SLA terms which he rated more than S_2 and vice versa. In case of identical ratings, assign the SLA term to the first player S_1.
2. Let T_1 be the set of all SLA terms that S_1 rated more than S_2, in addition to the terms with identical ratings. $T_1 = \{t_3, t_6\}$.
   a. Sum up all of S_1 scores for all SLA terms $\in T_1$. Total S_1 score is 50.
3. Let T_2 be the set of all SLA terms that S_2 rated more than S_1. $T_2 = \{ t_1, t_2, t_4, t_5 \}$.
   a. Sum up all of S_2 scores for all SLA terms $\in T_2$. Total S_2 score is 63.
4. S_2 is assigned all SLA terms $\in T_2$. And S_1 is assigned all SLA terms $\in T_1$.
   a. Order SLA terms assigned to S_2 as follows:
      i. Create a ratio $\frac{S_2}{S_1}$ for each SLA term $i \in T$. Calculated ratios are listed in Table 3.

Table 3: $\frac{S_2}{S_1}$ Ratio for all SLA Terms

SLA Term	$\frac{S_2}{S_1}$ Ratio
t₁	1.2
t₂	1.25
t₃	1
t₄	1.5
t₅	1.28
t₆	0.35

ii. Since S_2 has a greater total score. T_2 is rearranged so that SLA terms with the smallest ratio are first, followed by the one with the second smallest ratio, and so on. $T_2 = \{ t_1, t_2, t_5, t_4\}$.

5. To make the assignment more equitable, transfer SLA terms allocations or fractions of SLA terms allocation from S_2 to S_1; starting with terms with the smallest ratio.

   a. An appropriate fraction is the one that brings both player's total score to the same level. We must transfer part of t_1 to T_1. Let x be the portion of t_1 that will be transferred to T_1. We must solve the following equation for x:

   $$63 - 30x = 50 + 25x$$

   $$x = 0.24$$

   Thus We transfer 24% of t_1 from T_2 to T_1.

   b. T_1 calculated rating for t_1 is: $25 \times \frac{24}{100} = 6$

   And T_2 calculated rating for t_5 is: $30 - (30 \times \frac{24}{100}) = 22.8$

   Total score assigned to each player is: 55.91

6. The final division:

a. S_1 wins all of SLA terms t_3 , t_6 all the time, and is allocated t_1 for 24% of the time of the SLA contract.

b. S_2 wins all of SLA terms t_2, t_4, t_5 all the time, and is allocated t_1 for 76% of the time of the SLA contract.

The final allocation of terms is the outcome of the SLA negotiation game.

5.2 Tie-Breaking Rule for AW Procedure

In the case of non-rational behavior of players sharing identical utilities for all negotiated items, the AW procedure yields an allocation that is characterized to be equitable but not Pareto optimal, nor envy-free due to the tie-breaking method used by the AW procedure. It resolves ties in an arbitrary deterministic way which starts by allocating all terms to one player, then starts transferring terms of lower order to the other player, until equality is attained (AW Procedure, 2003).

The following example explains how the AW procedure works when both players equally strive to obtain the same SLA terms, and submit identical weights for all negotiated SLA terms. Let two Cloud services S_1, and S_2 be negotiating an SLA contract. The contract specifies 6 different SLA terms $T = \{ t_1 , t_2 , t_3 , t_4 , t_5 , t_6 \}$. The game goes as follows:

1. Both services S_1, and S_2 rate every term in T out of 100 score among them all, as described in Table 4.

Table 4: Services' Identical Ratings of SLA Terms

SLA Term	S_1 Ratings	S_2 Ratings
t_1	10	10
t_2	5	5
t_3	15	15
t_4	7	7
t_5	8	8
t_6	55	55

2. Start the initial allocation of SLA terms to players S_1, S_2. S_1 is allocated the SLA terms which he rated more than S_2 and vice versa. In case of identical ratings, assign the SLA term to the first player S_1.
3. Let T_1 be the set of all SLA terms that S_1 rated more than S_2, in addition to the terms with identical ratings. $T_1 = \{ t_1 , t_2 , t_3 , t_4 , t_5 , t_6\}$.
    a. Sum up all of S_1 scores for all SLA terms $\in T_1$. Total S_1 score is 100.
4. Let T_2 be the set of all SLA terms that S_2 rated more than S_1. $T_2 = \emptyset$.
    b. Sum up all of S_2 scores for all SLA terms $\in T_2$. Total S_2 score is 0.
5. S_2 is assigned all SLA terms $\in T_2$. And S_1 is assigned all SLA terms $\in T_1$.
    c. Order SLA terms assigned to S_1 as follows:
        i. Create a ratio $\frac{S_2}{S_1}$ for each SLA term $i \in T$. Calculated ratios are listed in Table 5.

Table 5: $\frac{S_2}{S_1}$ Ratio for all Identical SLA Terms Ratings

SLA Term	$\frac{S_2}{S_1}$ Ratio
t_1	1
t_2	1
t_3	1
t_4	1
t_5	1
t_6	1

6. To make the assignment more equitable, transfer SLA terms or fractions of SLA terms from S_1 to S_2; starting with terms with the smallest ratio.
    a. Transfer all of t_1 to T_2. S_1 new total is 90; and S_2 new total is 10.
    b. Transfer all of t_2 to T_2. S_1 new total is 85; and S_2 new total is 15.
    c. Transfer all of t_3 to T_2. S_1 new total is 70; and S_2 new total is 30.
    d. Transfer all of t_4 to T_2. S_1 new total is 63; and S_2 new total is 37.

e. Transfer all of t_5 to T_2. S_1 new total is 55; and S_2 new total is 45.

f. Need to transfer part of t_6 to T_2. Let x be the portion of t_6 that will be transferred to T_2. We must solve the following equation for x:

$$55 - 55x = 45 + 55x$$

$$x = 0.09$$

Thus we transfer 9% of t_6 from T_1 to T_2.

g. T_2 calculated rating for $t_6$is: $55 \times \frac{9}{100} = 5$

And T_1 calculated rating for t_1 is: $55 - (55 \times \frac{91}{100}) = 50$

Total score assigned to each player is: 50

7. The final division:

a. S_1 is allocated t_6 for 9% of the time of the SLA contract.

b. S_2 wins all of SLA terms t_1, t_2, t_3, t_4, t_5 all the time, and is allocated t_1 for 9% of the time of the SLA contract.

This allocation is equitable, however in this case one player gets the highly rated SLA term, and the other gets the least rated ones. Consequently, as the number of negotiated SLA terms increases, chances are one player is assigned the most highly weighted terms, and the other is assigned the least weighted terms. This is an equal allocation, yet not fair. Therefore, we propose implementing a tie-breaking rule, which in case both players share identical weights for all SLA terms, gives every player 50% of every SLA term in the negotiation contract. SLA terms allocation under this rule is Pareto optimal, envy free, and equitable, since it splits all SLA terms between both players, giving half of every SLA term to each player (Aziz, Brânzei, Filos-Ratsikas, & Frederiksen, 2015)

5.3 Summary

This chapter described an automated SLA negotiation model for federated Cloud services based on game theory. The model applies a Fair Division game called Adjusted Winner within the process of SLA negotiation in *CloudLend*. We demonstrated how SLA negotiation is performed using the AW procedure, and we also introduced a tie breaking rule to enhance the results of AW in the case of highly competitive SLA negotiations. The proposed SLA negotiation model is suitable to for distributed Cloud environments such as *CloudLend*, since members of the network can be engaged in as many negotiation games they need without affecting the outcome of an ongoing negotiation game.

Chapter 6: Formal Definition of the SLA Negotiation Model

This chapter proposes a formal representation of the *CloudLend* network and links establishment among its members. Furthermore, the SLA negotiation problem in the network is also formally defined. SLA negotiation is described as a Fair Division game, and objective functions of *CloudLend* members are formalized for the optimized satisfaction for customers, and an informed resources' utilization for providers.

6.1 Formal Definition of *CloudLend* Network

The *CloudLend* network is composed of a set of federations $F_i(i = 1, 2, \dots, I)$, where i refers to the number of federations. Each federation F_i is composed of a set of Cloud providers $P_{ij}(j = 1,2, \dots, J_i)$, where j refers to the number of providers. Each Cloud provider P_{ij} can offer a subset of services $S_{ij} \subset S$, where $S = \{s_1, s_2, \dots, s_n\}$ is the set of all service types that can be offered within *CloudLend*. A service consumer $C_i(i = 1, 2, \dots, I)$, can be a Cloud customer, or another Cloud provider.

6.1.1 Links Establishment Between a Cloud Customer and Provider

The *CloudLend* network can be seen as a global network of networks in which each node S_{ijm} represents the m^{th} service offered by $P_{ij}(1 \leq m \leq M_{ij})$, where M_{ij} is the number of service types offered by P_{ij}. A customer request R_k ($k = 1,2, \dots, K$) includes a set of QoS requirements $Q_k = \{q_1, q_2, \dots, q_n\}$. For every request R_k received by *CloudLend*, a set of corresponding service offers formed by a set of providers $S_k = \{P_1, P_2, .., P_n\}$ is created. Each customer request R_k will generate a subnetwork $G_k \subset G$ in which each node $s_{ijm} \in S_{ij} \cap S_K$. Once a customer decides on an offer from S_k, a relationship $r(C_i, P_i)$ is established between the customer and the selected provider. Each relationship $r(C_i, P_i)$ is bounded by an SLA contract. Which includes a set of agreed SLA terms between customer C_i and provider P_i $L_{C_i, P_i} = \{t_1, t_2, \dots, t_n\}$.

6.1.2 Links Establishment Among Different Cloud Services

A link between two cloud services $(s_{ijm}, s_{i'j'm'})$ describes a relationship between two services offered by different Cloud providers P_{ij}, and $P_{i'j'}$ in a Cloud federation F_i. Where $s_{ijm} = s_n, and\ s_{i'j'm'} = s_{n'}$ with $n \neq n'$. A relationship r can be established only if both services involved in the relationship are available. Each relationship $r\ (s_{ijm}, s_{i'j'm'})$ between two services is bounded by an SLA contract, which includes a set of agreed SLA terms $L_{s_{ijm}, s_{i'j'm'}} = \{t_1, t_2, \dots, t_n\}$.

6.2 SLA Negotiation as a Fair Division Game

SLA negotiation in *CloudLend* is represented as iterations of bids exchanges between the Cloud service provider, and the customer until reaching the final agreement on the provided SLA terms. Both parties involved in the negotiation process exchange their bids during negotiation rounds. A negotiation round is the period of time through which one party offers a bid, while the other reviews that bid to either accept or place a counter offer. Hence, starting another negotiation round. N_r represents the number of SLA negotiation rounds before reaching an agreement, where N_r is bounded to: $N_{min} \leq N_r \leq N_{max}$, assuming that N_{min} and, N_{max} are the minimum, and the maximum number of SLA negotiation rounds set by the *CloudLend* network.

In *CloudLend*, negotiation usually occurs in the following cases:

1. Between a customer $C_i \in C$, and one or more Cloud providers $P_j \in P$.
2. Between a Cloud provider P_j, and another Cloud provider(s) in P when forming a federation.
3. Between service S_{ijm} offered by $P_j \in P$, and other service(s) offered by other providers within a Cloud federation $F_i \in F$.

For each customer request R_k the objective of the service provider is to minimize the possible number of SLA negotiation rounds N_r. Hence, providers aim to seal the deal within at least N_{min} number of rounds, because prolonged negotiation sessions might lead to unused Cloud resources that remain idle during the period of negotiation, which directly affects provider's profit, while the objective of a customer is to maximize his satisfaction measured in terms of QoS. At any given negotiation round, P_j aims at having minimum changes made to the offered ratings of SLA terms, while C_i aims at winning SLA terms that are of high importance to him. The level of C_i and P_j satisfaction is measured by the utility gained by the Fair Division game, AW, which is played at every negotiation round.

6.2.1 The Adjusted Winner Game

In the context of *CloudLend* the SLA negotiation game is played between a customer and a provider. For every negotiated relationship $r\ (x, y)$ between any two *CloudLend* members, x and y are players of an AW game. During the game, the two players try to split a set $L = \{q_1^{x,y}, q_2^{x,y}, \ldots, q_n^{x,y}\}$ of SLA terms of the negotiated SLA contract. Let $a = (a_1, a_2, \ldots, a_n)$, and $b = (b_1, b_2, \ldots, b_n)$, indicate rating vectors, where a_i and b_i are the rating assigned by x and y respectively for $q_i^{x,y} \in L$. An allocation vector $W = (W_A, W_B)$ is an assignment of portions of SLA terms to the players, where $W_A = (w_A^1, \ldots, w_A^n) \in [0,1]^n$ and $W_b = (w_b^1, \ldots, w_b^n) \in [0,1]^n$ are the allocations of x and y respectively. Both Players have additive utility over SLA terms. The utility of player x for his allocation W_A given its rating a, is:

$$u_a(W_A) = \sum_{i=0}^{n} a_i \cdot w_A^i \tag{1}$$

And the utility of player y for his allocation W_B given its rating b, is:

$$u_b(W_B) = \sum_{i=0}^{n} b_i \cdot w_B^i \tag{2}$$

After each round of the game, players' utilities are weighted, and SLA terms allocations are modified, until both utilities reach an equilibrium.

$$\sum_{i \in N} a_i = \sum_{i \in N} b_i \ \forall\, q_n^{x,y} \in L \tag{3}$$

6.3 Objective Functions of *CloudLend* Customers and Providers

CloudLend members negotiate SLAs to evaluate the utility they expect to gain from the anticipated relationships. This utility is perceived differently by Cloud customers, and providers. For a Cloud customer, utility gain is used to support the customer in making the decision of selecting the most satisfactory relationship among different alternative relationships with other *CloudLend* providers. While for Cloud providers, utility gain is used to support in making the decision of prioritizing relationships in order to achieve most efficient utilization of provider's resources.

6.3.1 Customer Satisfaction Optimization using Fair Division Game

Customer satisfaction is generally measured by the degree to which his QoS requirements are guaranteed. The proposed *CloudLend* network provides customers with the ability to evaluate all alternative providers' offers using the SLA negotiation model. By comparing utility gains that result from all negotiation sessions, a customer can then decide to sign the contract with the provider that yielded the maximum utility gain.

Let $O_c = \{u_{p_1}, u_{p_2}, \dots, u_{p_n}\}$ a set of utility gains resulted by SLA negotiation games played out between customer C_i and service providers $P_n: \forall\, P_n \in S_k$. Based on the *Fair Division Theorem* (Brams & Taylor, 1996), AW produces an allocation of the negotiated items based on the players' announced valuations that is efficient, equitable and envy-free. We conclude that every $u_{p_n} \in O_c$ is optimized as it exhibits the following properties:

1. $\boldsymbol{u_{p_n}}$ **is efficient:** any other allocation that is strictly better for one player is strictly worse for the other.
2. $\boldsymbol{u_{p_n}}$ **is equitable:** C_i SLA terms allocation is the same as P_J SLA terms allocation.
3. $\boldsymbol{u_{p_n}}$ **is envy-free:** neither player would trade his SLA terms allocation for that of the other.

As a result, *CloudLend* network finds customer's overall utility as:

$$U_{C_i} = Max_{n=1}^{n}\, u_{p_n}: \forall\, u_{p_n} \in O_c \tag{4}$$

This implies that C_i gets an offer that provides maximum optimized guarantees to his QoS requirements other than any alternative offer.

6.3.2 Informed Resource Utilization for Cloud Providers

Being a member of *CloudLend* offers Cloud providers greater exposure to potential customers through the network, besides the ability to strategically negotiate their SLA terms with prospect customers based on information gained from previous negotiations. Therefore, chances of Cloud providers' resources utilization are maximized. Let $O_p = \{u_{c_1}, u_{c_2}, \dots, u_{c_n}\}$ a set of utility gains resulted by SLA negotiation games played out between provider P_j and different customers C_n: $\forall\, C_n \in C$. Unlike Cloud customers, providers don't get to choose customers, and they don't negotiate the same exact SLA terms with different customers. Providers simultaneously negotiate different SLA terms, with different customers who have different QoS requirements. Thus, comparing provider's utility gains u_{c_n} of different SLA negotiation games is impractical. Nevertheless, a Cloud provider's overall utility gain U_{P_j} results from comparing individual SLA terms allocation $\left(w^1_{P_j}, \dots, w^n_{P_j}\right)$ across all negotiation games. Assuming that each SLA terms maps to a specific provider's resource, and that provider implements his own resource scheduling mechanism that is independent from the *CloudLend* network. Let $T = \{t_1, t_2, .., t_n\}$ a set of all SLA terms offered by a Cloud provider P_j who participates in a set of concurrent SLA negotiation games $V = \{v_1, v_2, \dots, v_i\}$ with different customers in a specific period of time. For every game v_i, P_j offers a set of his SLA terms T_{v_i} , where $T_{v_i} \subseteq T$. As a result of participating in V, P_j is assigned a set of SLA terms' allocation per SLA negotiation game: $W_{v_i} = \{w^{t_1}_{P_j}, \dots, w^{t_n}_{P_j}\} \in [0,1]^n\ \forall\, v_i \in V$. The set of allocations W_{v_i} represents P_j utility gains from v_i for every individual SLA term $t_n \in T_{v_i}$. As a result, the *CloudLend* network finds utility gain per individual SLA term u_{t_n} :

$$u_{t_n} = max\left(w^{t_n}_{P_j}\right)\ \forall\, v_i \in V \qquad (5)$$

Let $Y = \{u_{t_1}, u_{t_2}, \dots, u_{t_n}\}$ a set of individual SLA terms' utility gains $u_{t_n} \forall\, t_n \in T$. $f(t_n, v_i)$ represents a mapping between an SLA term t_n, and the game v_i with

$\max(u_{t_n})$ for t_n, $f: t_n \mapsto v_i \; \forall \; t_n \in T$. The *CloudLend* network uses $f(t_n, v_i)$ to find a game v_i that has the highest frequency of occurrence in $f(t_n, v_i)$, which is the game that offers the most resources utilization among all other negotiated games in V. Let $D = \{v_1, v_2 \dots, v_n\}$ the set of all $v \in f(t_n, v_i)$, P_j overall utility is offered by the most occurring v in D:

$$U_{P_j} = Mode\ D \tag{6}$$

Assuming that utility gains evaluation occurs in a fixed period of time for all concurrent negotiation games. U_{P_j} has a positive impact on provider's resources utilization because of the following reasons:

1. It enables the prioritizing of different customers' requests based on their impact on providers' resources, which aids providers with decisions regarding resources' scheduling.
2. It allows providers to be informed on the amount of current, and prospective demand on particular resources, which supports providers in making decision regarding their negotiation strategies with prospect customers.

6.4 Summary

This chapter provided a formal description of the *CloudLend* network and relationship establishment among its members, in addition to the formal description of the SLA negotiation problem as a fair division game. Finally, objective functions of Cloud providers and customers were defined and evaluated.

Chapter 7: *CloudLend* SLA Management Models Evaluation

This chapter describes the implementation of the *CloudLend* simulator along with the structure of its main modules and their distinguished features. In addition to describing the evaluation of the proposed SLA management models including SLA negotiation, and monitoring in *CloudLend* network.

7.1 *CloudLend* Network Simulator

To provide a proof of concept for the *CloudLend* proposed architecture, and to validate the efficiency of the proposed SLA management model, a simulation environment was required to simulate the *CloudLend* network. Therefore, we examined some open source visual analytics tools such as GINY (Instiute for Systems Biology, 2013), Prefuse (UC Berkeley, 2012), and JGraph (Alder, 2016) that provide powerful graph visualization features. However, for *CloudLend* evaluation we seek a robust analysis of functionalities in addition to graph visualization, therefore we considered social network analysis tools like GUESS (Adar, 2007), and JUNG (O'Madadhain, 2010). GUESS is an experimental data analysis and visualization tool for graphs and networks that contains a domain-specific embedded language called Gython (an extension of Python), and uses open source software like JUNG. It is a Java software library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network, which is precisely what is required for the evaluation purposes. Hence, we used JUNG to implement a *CloudLend* network simulator.

Figure 8 describes the components of the *CloudLend* simulator which implements the following features:

1. **Network generator:** creates a random *CloudLend* graph of ***n*** members, each has ***m*** SLA terms.
2. **Built in SLA specification mechanism:** specifies appropriate SLA terms' weights based on selected test cases.
3. **Negotiation manager:** implements the game theory based SLA negotiation model, and finds out the actual SLA terms allocations gained by executing the enhanced AW procedure.
4. **Federation manager:** evaluates a potential relationship with any given *CloudLend* members, and manages established relationships.
5. **Monitoring Agent:** monitors an established federation within *CloudLend*, detects changes during service provisioning, and reports any SLA violations.

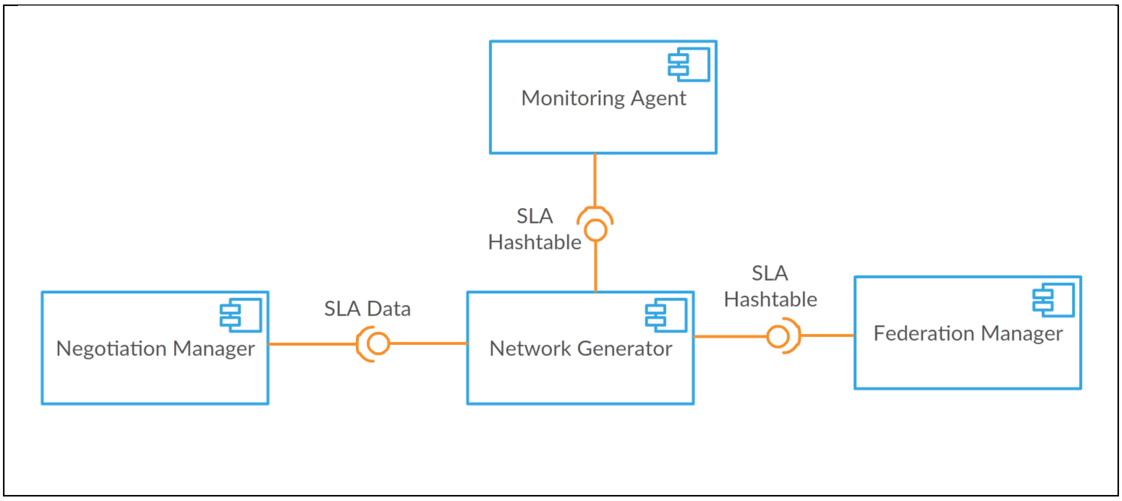

Figure 8: The *CloudLend* simulator component diagram

Figure 9: The *CloudLend* simulator sequence diagram illustrates a sequence diagram of messages passed across the components of the *CloudLend* simulator.

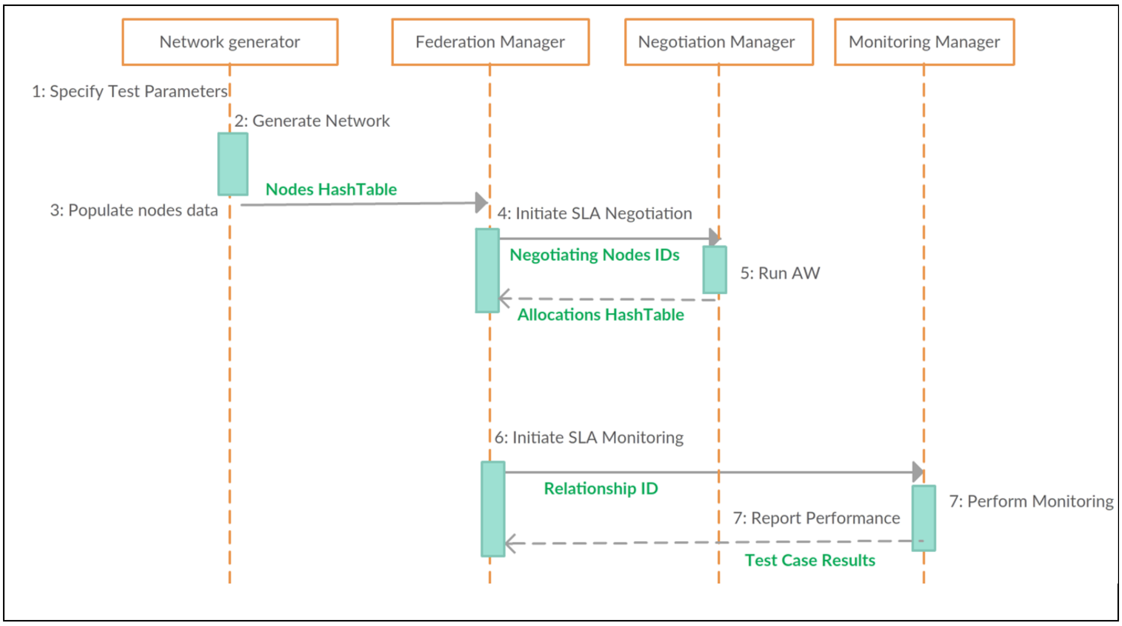

Figure 9: The *CloudLend* simulator sequence diagram

7.2 Multi-level SLA Specification Mechanism

CloudLend simulator implements the proposed multi-level SLA specification model, SLA terms are specified randomly based on selected test cases once the *CloudLend* network is generated. For every node of the *CloudLend* network *SLAData* object is created to hold information on the node itself, its established relationships, and its SLA terms weights and specifications. All *SLAData* objects are stored in a *Hashtable* that is accessible to other components of the simulator. For example, the *Federation Manager* gets a reference to all *CloudLend* members participating in a federation, along with their dependencies. Also, the *Monitoring Agent* gets a reference to SLA terms specification and their dependencies within the network.

7.3 SLA Negotiation Model Evaluation

To validate the efficiency of the proposed negotiation model, we ran several SLA negotiation test cases that evaluate the relationship establishment decision between two selected *CloudLend* members. Figure 10 illustrates the evaluation process, which consists of specifying the number of nodes in the network, the number of SLA terms to be negotiated, and selecting a test case. The simulator generates a *CloudLend* network accordingly, and two negotiating nodes are selected. Consequently, the AW procedure starts the SLA terms allocation game, and the final set of allocated SLA terms is evaluated against user's expected SLA terms allocation.

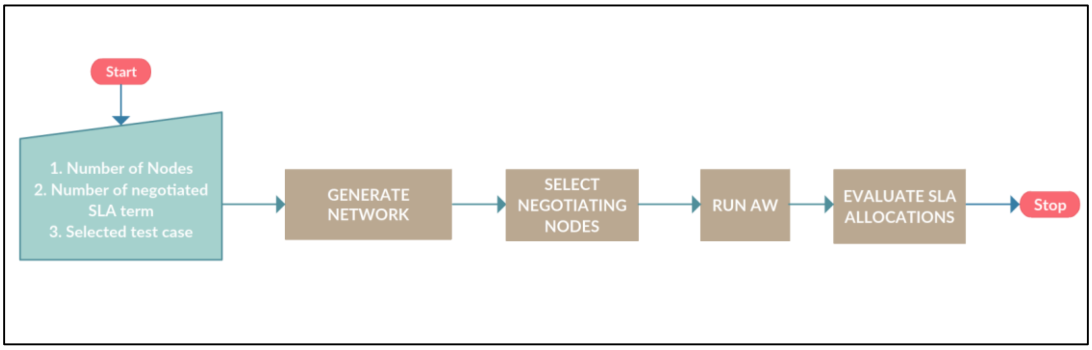

Figure 10: Flowchart diagram of the SLA negotiation model evaluation

The SLA allocations' evaluation algorithm illustrated in Figure 11 begins with finding the expected SLA terms allocation for a negotiated SLA contract based on players' submitted weights. Next, the actual SLA terms allocation is found by running the AW procedure. Both allocations are finally compared to obtain the model's accuracy level which is calculated as follows:

$$\sum_{t=0}^{n} \frac{z_{a_x,a_c}^{t_n}}{n} * 100 \quad (7)$$

Where t_n is the SLA term being evaluated, a_x is the expected term allocation as per submitted weights, and a_c is the actual term allocation as per AW.

A sample SLA negotiation test run in the *CloudLend* simulator is illustrated in Figure 12.

Algorithm 1 Evaluate relationship in CloudLend simulator

Input: Array a_t includes all negotiated SLA terms weights

procedure EXPECTED-SLA-ALLOCATION(a_t)
    for each term t_i in a_t **do**
        Set array $a_x(t_i)=$ ISEXPECTED($w^{t_i}_{p1,p2}$)
    end for
    Return a_x
end procedure
procedure ACTUAL-SLA-ALLOCATION(a_t)
    Set array $a_c=$ AWPROCEDURE($w^{t}_{p1,p2}$)
    Return a_c
end procedure
procedure ACCURACY-LEVEL(a_x, a_c)
    for each term t_i in a_t **do**
        if $a_x(t_i) = a_c(t_i)$ **then**
            $z^{t_i}_{a_x,a_c} = 1$
        else if $a_x(t_i)a_c(t_i)$ **then**
            $z^{t_i}_{a_x,a_c} = 0$
        else if $w^{t_i}_{p1} = w^{t_i}_{p2}$ **then**
            $z^{t_i}_{a_x,a_c} = 0.5$
        end if
    end for
    $AccuracyLevel = \frac{\sum_{t=0}^{n} z^{t_n}_{a_x,a_c}}{n} * 100$
    Return $AccuracyLevel$
end procedure

Figure 11: Algorithm1 Evaluate relationship in *CloudLend* simulator

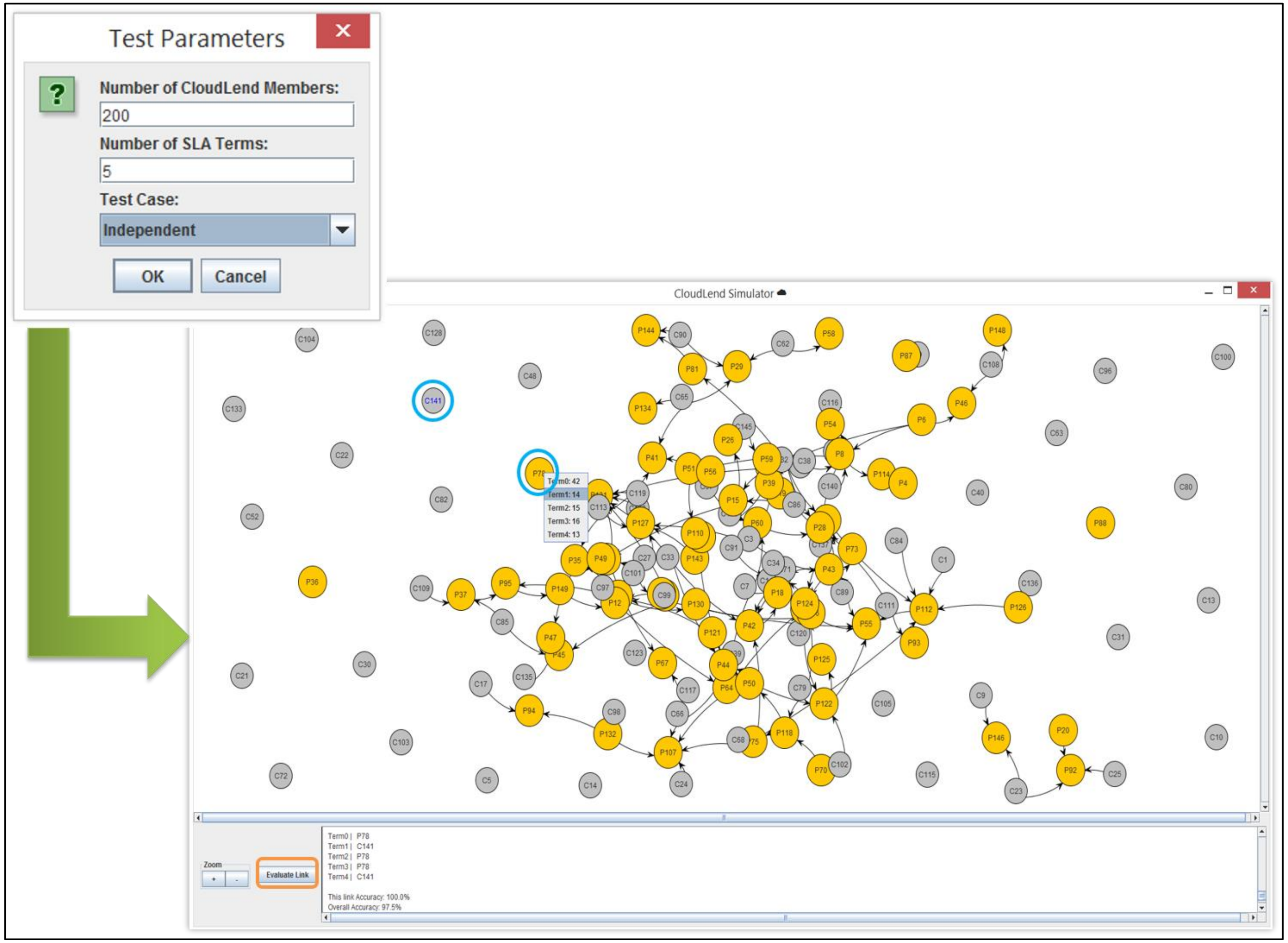

Figure 12: *CloudLend* simulator snapshot of SLA negotiation evaluation

7.3.1 Experimental Setup and Test Strategies

We ran the *CloudLend* simulator with several number of SLA terms, and different SLA terms' weights comparing the traditional AW performance against the enhanced AW with tie-breaking rule.

For every test run we calculated the model's accuracy level, which indicates the closeness degree between players' expected SLA terms allocation, and actual SLA terms allocated by the proposed model. During any particular negotiation round initiated between a Cloud customer, and a provider, the negotiated SLA contract shall include ***n*** number of SLA terms. In this test, we considered 5, 20 and 50 terms through 100 runs. Several runs of the same test are required because simulation results will differ depending on the random network generation in every run. In order to decide on the appropriate number of simulation runs, we performed 1000, 500, 200, and 100 runs while observing the mean, and the variance of every run. For our case a 100 simulation runs provided representative sample.

The null hypothesis H_0 states that the model provides equal means of accuracy level for all the various number of negotiated SLA terms, $H_0: \mu_1 = \mu_2 = \mu_3$. Where μ_1, μ_2, μ_3 are mean accuracy levels when SLA terms are 5, 20, and 50 respectively. The alternative hypothesis H_1 states that the mean accuracy level of at least an amount of SLA terms is significantly different.

The SLA terms weights provided by the two players; customer, and provider shall be experimented with the following test strategies:

1. Strategy 1: Each player provides different and independent weights of all SLA terms.

2. Strategy 2: Both players provide identical weights for all SLA terms.
3. Strategy 3: Each player adopts a single-choice strategy, where a player allocates most of his weights to a single SLA term and neglects the others.

7.3.2 Results and Discussion

7.3.2.1 SLA Negotiation with Traditional Adjusted Winner

Performing a One-Way ANOVA test to analyze the variance of the three different amounts of SLA terms, with a 95% confidence level yielded p-value < 0.05. Therefore, the decision rule was to reject H_0 for all three test strategies, and we concluded that different number of negotiated SLA terms have different means of accuracy levels.

Figure 13, Figure 14, Figure 15, Figure 16, and Figure 17 summarize the simulation results of the SLA negotiation model using the traditional AW, for the different numbers of negotiated SLA terms, when adopting different negotiation strategies.

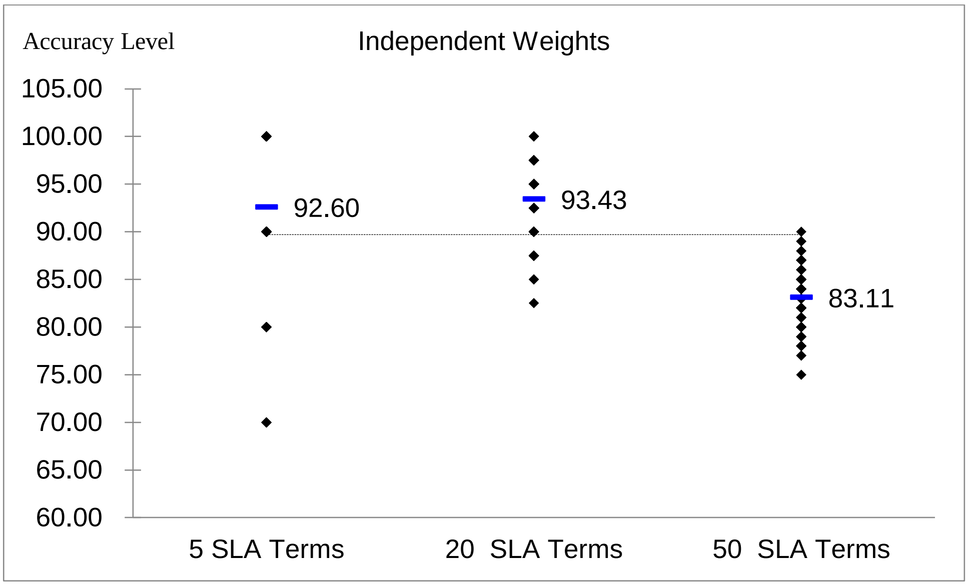

Figure 13: Analysis of variance for the independent weights with the traditional

AW

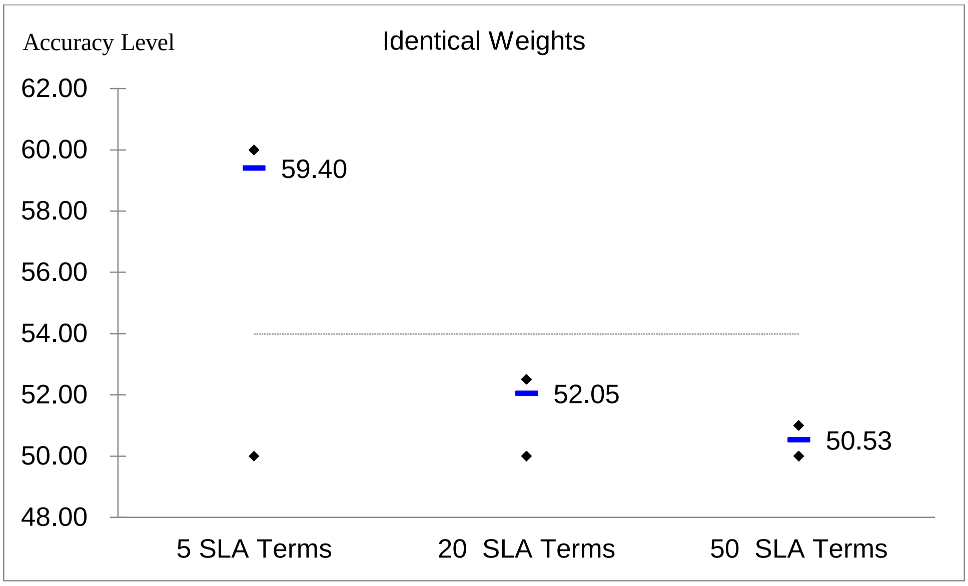

Figure 14: Analysis of variance for the identical weights with the traditional AW

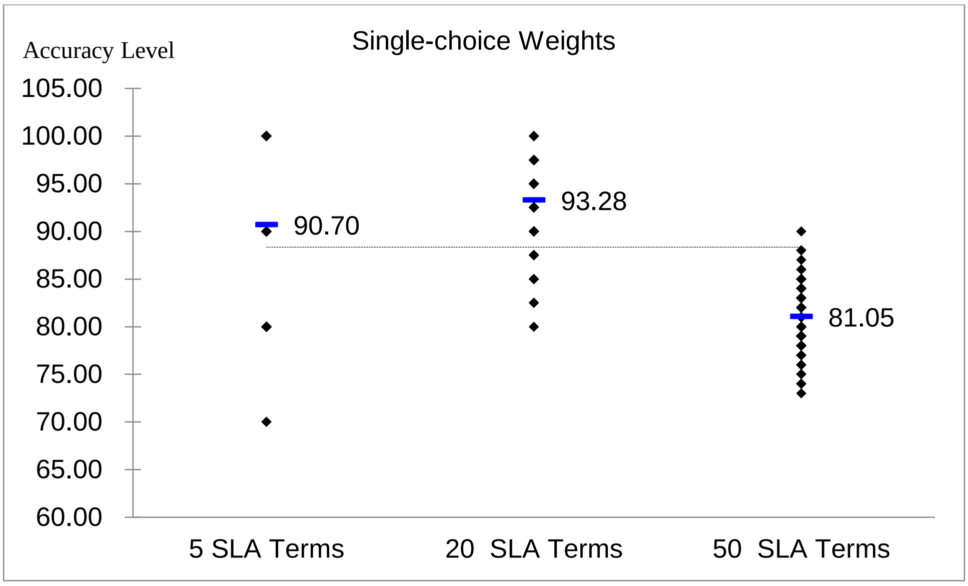

Figure 15: Analysis of variance for the single-choice weights with the traditional AW

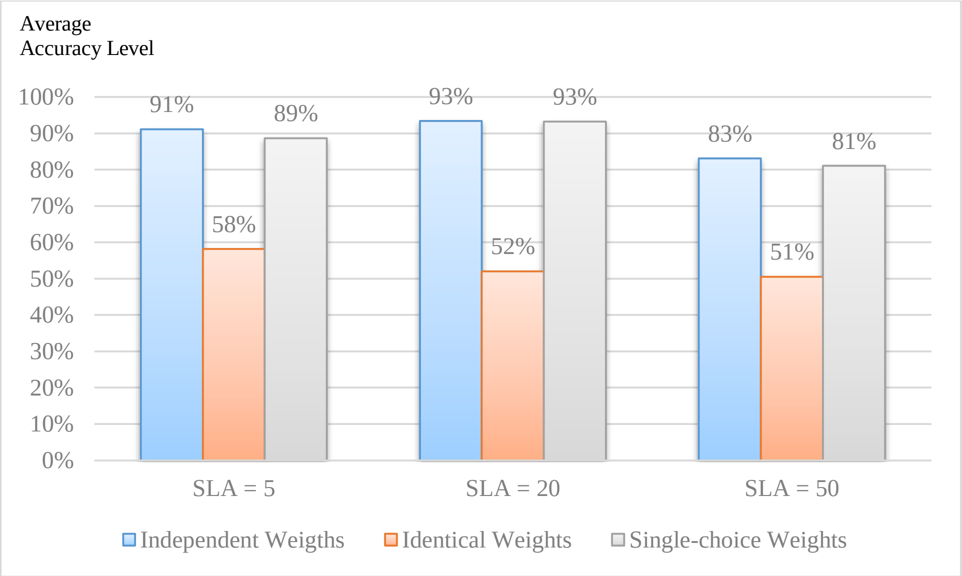

Figure 16: Average accuracy levels for the SLA negotiation model with the traditional AW

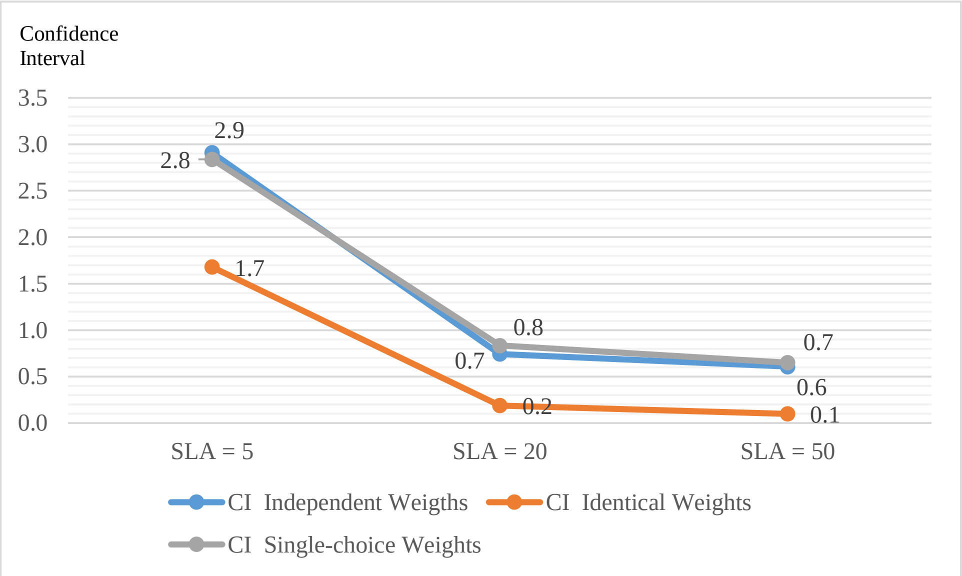

Figure 17: Confidence intervals for the SLA negotiation model with the traditional AW using 95% confidence level

We are 95% confident that when the number of negotiated SLA terms is 5, and when players provide different and independent weights, the negotiation model provides accuracy level that falls within the range of 90% - 94%. And, when the number of negotiated SLA terms is 20, the negotiation model provides 92%-94%

accuracy. Additionally, when the number of negotiated SLA terms is 50 the negotiation model provides 82%-83% accuracy.

However, when both players submit identical weights for all SLA terms, we are 95% confident that the model achieves a range of 58%-59% accuracy when the number of negotiated SLA terms is 5. The model achieves 51%-52% accuracy when the number of SLA terms is 20. Besides, it achieves 50.4%-50.6% accuracy when the number of SLA terms is 50.

Finally, when players adopt the single-choice strategy we are 95% confident that the model provides 89%-92% accuracy when the number of negotiated SLA terms is 5. Nevertheless, the model provides 92%-94 accuracy when the number of SLA terms is 20, and it provides 80%-81% accuracy when the number of negotiated SLA terms is 50.

Noticeably, the model's accuracy level drops when the submitted weights are identical. This is owing to the tie-breaking method used by the AW procedure; which starts by allocating all terms to a single player, then starts transferring terms of lower weights to the other player, until equality attained. Consequently, as the number of SLA terms increases, chances are one player is allocated more highly weighted terms, and the other is allocated more low weighted terms.

7.3.2.2 Adjusted Winner with the Tie Breaking Method

Performing a One-Way ANOVA test to analyze the variance of the three different amounts of SLA terms: 5, 20 , and 50, with a 95% confidence level yielded p-value < 0.05 for the test strategies 1, and 3, while it yielded p-value > 0.05 for the test strategy 2. Therefore, the decision rule was to reject H_0 for strategies 1, and 3, and we concluded that different number of negotiated SLA terms have different means of accuracy levels. On the other hand, for strategy 2 the decision rule was to accept H_0; which means in the case of identical weights the AW with tie breaking rule provides equal means of accuracy levels regardless to the number of SLA terms.

Figure 18, Figure 19, Figure 20, Figure 21, and Figure 22 summarize the simulation results of the SLA negotiation model using the tie breaking rule for the AW, for the different numbers of negotiated SLA terms, when adopting different negotiation strategies.

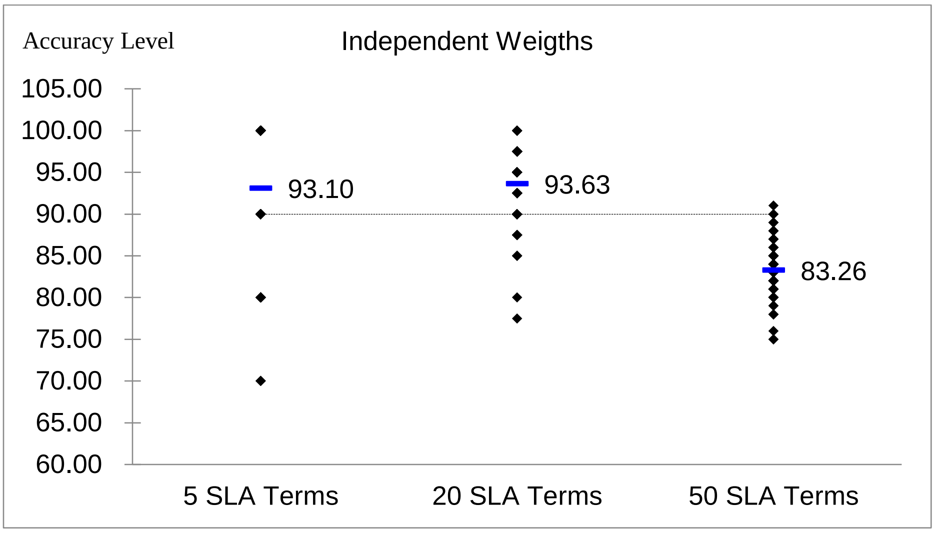

Figure 18: Analysis of variance for the independent weights with the tie breaking rule for the AW

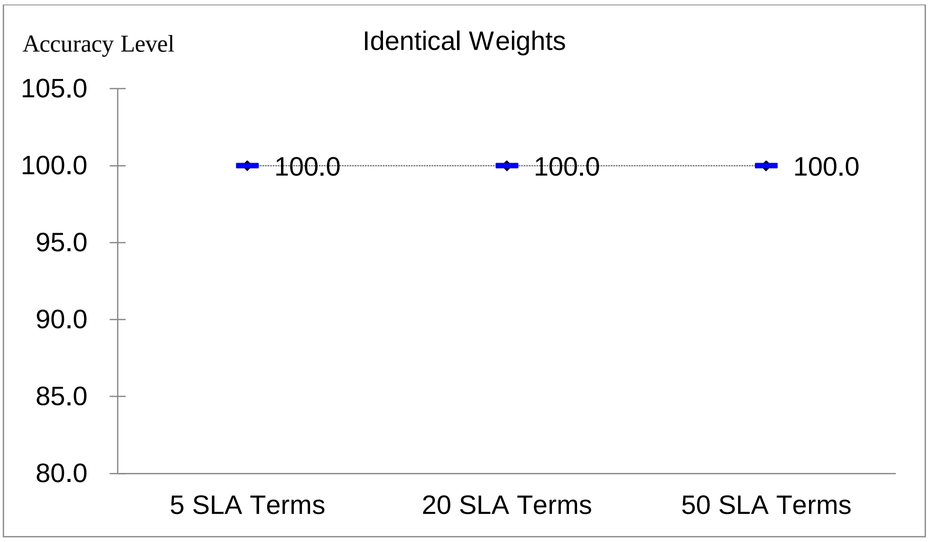

Figure 19: Analysis of variance for the identical weights with the tie breaking rule

for the AW

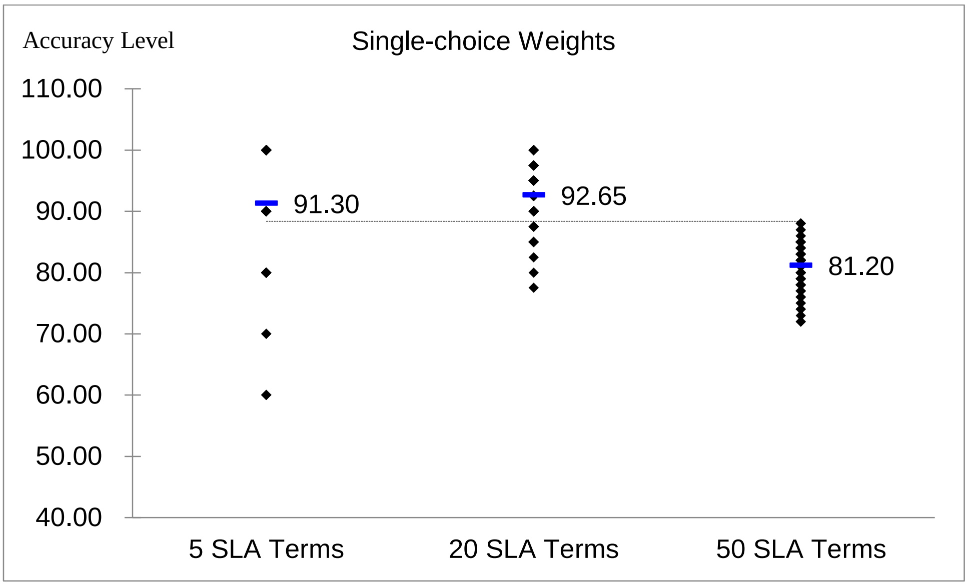

Figure 20: Analysis of variance for the single-choice weights with the tie breaking rule for the AW

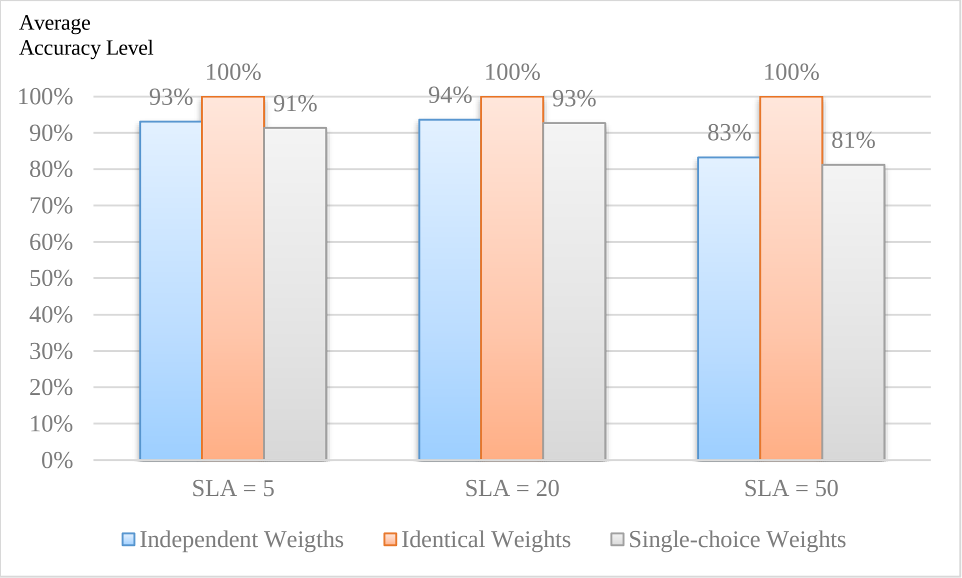

Figure 21: Average accuracy levels for the SLA negotiation model with the tie breaking rule for the AW

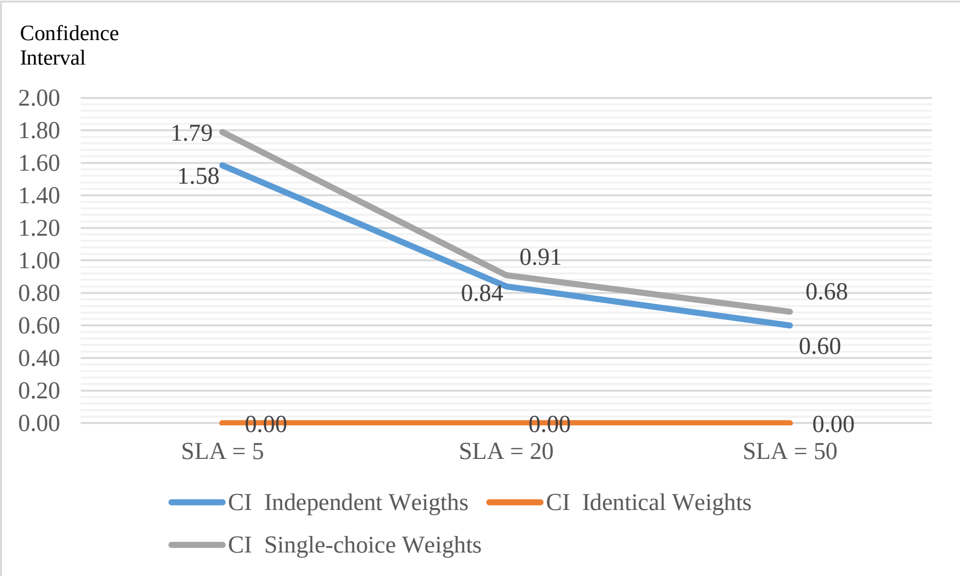

Figure 22: Confidence intervals for the SLA negotiation model with the tie breaking rule for the AW using 95% confidence level

We are 95% confident that when the number of negotiated SLA terms is 5, and when players provide different and independent weights, the negotiation model provides accuracy level that falls within the range of 91% - 94%. And, when the number of negotiated SLA terms is 20, the negotiation model provides 92%-94% accuracy. Additionally, when the number of negotiated SLA terms is 50 the negotiation model provides 82%-83% accuracy.

However, when both players submit identical weights for all SLA terms, we are 95% confident that the model achieves 100% accuracy regardless of the number of negotiated SLA terms. Finally, when players adopt the single-choice strategy we are 95% confident that the model provides 89%-93% accuracy when the number of negotiated SLA terms is 5. Nevertheless, the model provides 91%-93 accuracy when the number of SLA terms is 20, and it provides 80%-81% accuracy when the number of negotiated SLA terms is 50.

The model's accuracy level marginally drops as the number of SLA terms increases. This is due to the fact that the more terms to be included, the less the points, out of the total 100 points, are available for weighting. This decreases the closeness between submitted weights, and increases the error ratio when computing the expected terms allocation.

Table 6, and Figure 23 compare the average accuracy level for the traditional AW against the AW with tie breaking rule for the different numbers of negotiated SLA terms, while adopting different negotiation strategies. As a result, it is clear that the proposed SLA negotiation model based on the AW with tie breaking rule provides an improved steady accuracy level regardless of the played out strategy, or the number of the SLA terms included in the negotiation game.

Table 6: Accuracy Levels of Traditional Aw vs. AW with Tie Breaking

Test Cases	Average Accuracy Levels					
	Traditional AW			AW with Tie Breaking		
	SLA Terms					
	5	20	50	5	20	50
Strategy 1	92.6%	93.4%	83.1%	93.1%	93.6%	83.2%
Strategy 2	59.4%	52%	50.5%	100%	100%	100%
Strategy 3	90.7%	93.2%	81%	91.3%	92.6%	81.2%

Figure 23: Accuracy Levels of Traditional Aw vs. AW with Tie Breaking

7.4 SLA Monitoring Model Evaluation

A monitoring model for a distributed environment such as *CloudLend* is required to have some properties that confront challenges raised by the flexibility, competitiveness, and dynamicity of the Clouds federation. Therefore, we designed simulation experiments to test the proposed SLA monitoring model for three main properties which include: elasticity, accuracy, and autonomicity. These properties are further explained in section *7.4.1* Experimental Setup and Test Strategies. Hence, we ran three SLA monitoring test cases that detect changes, and violations in federations within the simulated *CloudLend* network. Figure 24 illustrates the evaluation process for the proposed SLA monitoring model. To begin with, we specify the number of nodes in the simulated network, and the *CloudLend* simulator generates the network accordingly, then one of the monitoring test cases is selected based on the selected monitoring model property. A sample SLA monitoring test run in the *CloudLend* simulator is illustrated in Figure 25.

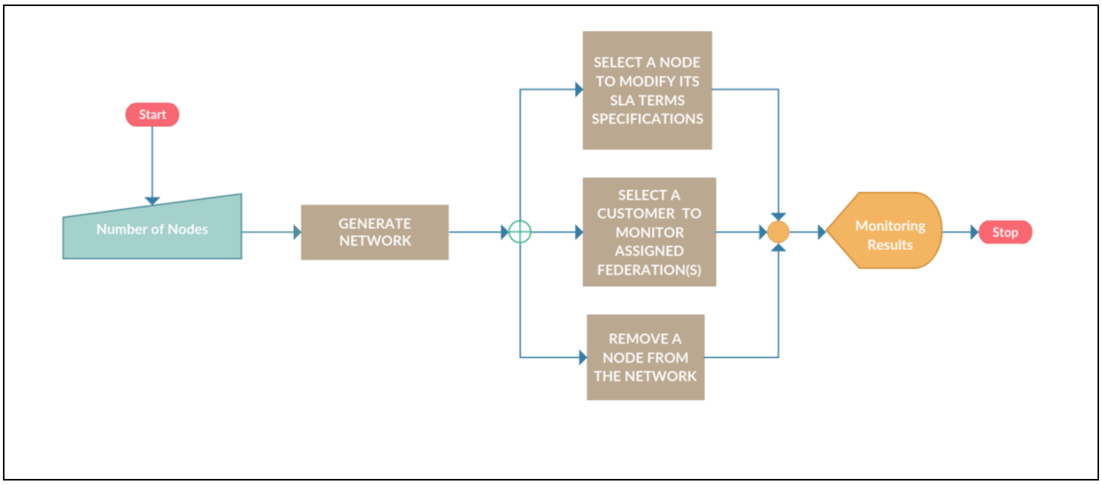

Figure 24: Flowchart diagram of the SLA monitoring model evaluation

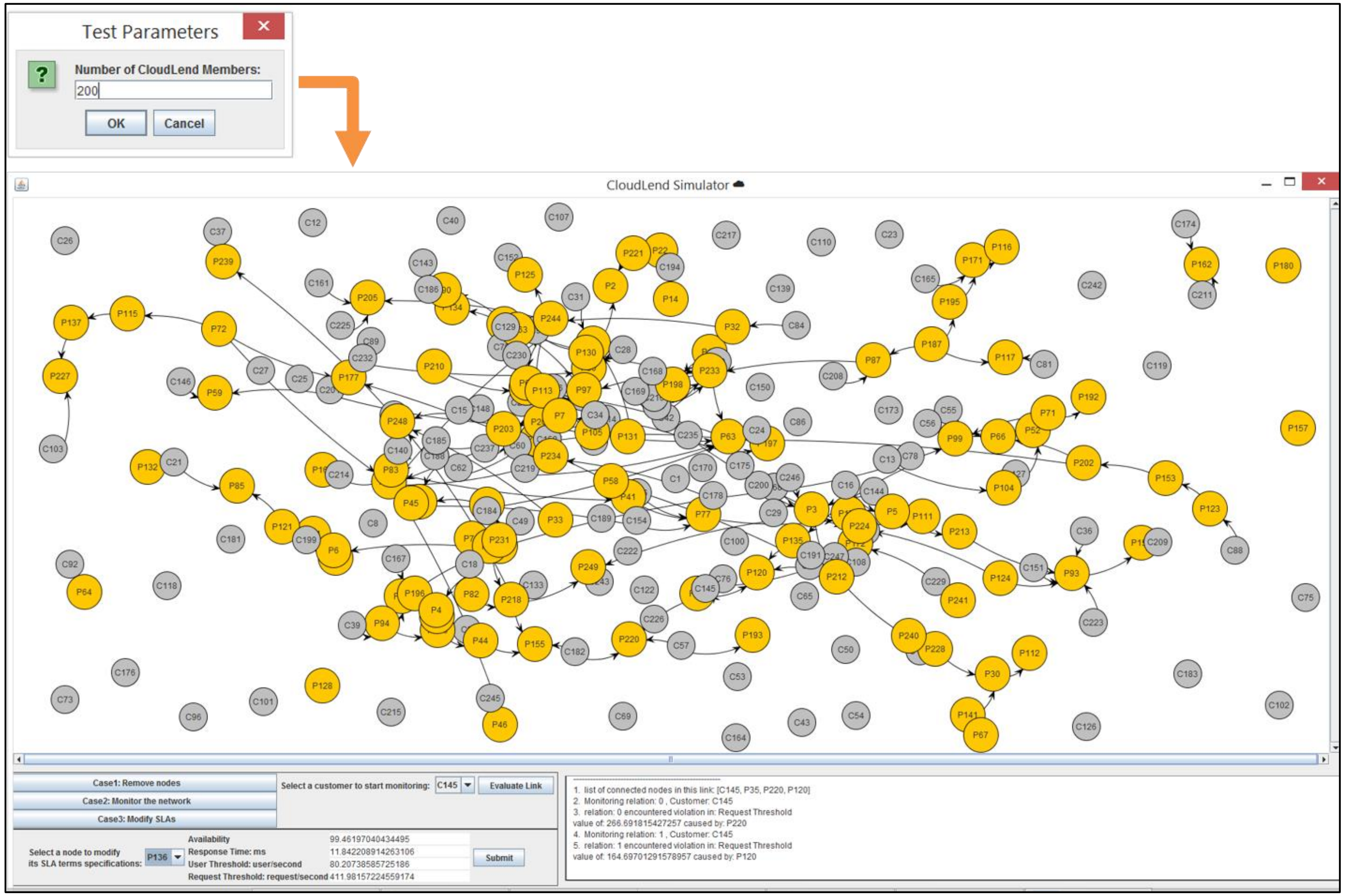

Figure 25: *CloudLend* simulator snapshot of SLA monitoring evaluation

7.4.1 Experimental Setup and Test Strategies

This evaluation aims to test three properties of the agent-based SLA monitoring model: 1) elasticity, 2) autonomicity, and 3) accuracy as follows:

Test Case 1: Elasticity indicates that the monitoring model is able to cope with dynamic changes of monitored *CloudLend* members (Clayman, Galis, & Mamatas, 2010). To test this property, we ran 100 iterations of the *CloudLend* simulator with different network sizes: 50, 200, and 500, while randomly removing 20% of the members of the network. We assume that 80% of the relationships are linked to 20% of the network members, Based on the 80/20 rule (Barabási & Frangos, 2014).

Test Case 2: Autonomicity indicates that the monitoring model is able to react to irregular changes automatically, while hiding inherent complexity to relevant *CloudLend* members (Mian, Martin, & Vazquez-Poletti, 2013). To test this property, we ran 100 iterations of the *CloudLend* simulator with different network sizes: 50, 200, and 500, while randomly augmenting changes in SLA terms specifications of 20% of the members of the network.

Test Case 3*:* Accuracy indicates that the monitoring model is able to accurately detect events as they measure, and to identify the causes of the problem (Aceto, Botta, De Donato, & Pescapè, 2013). To test this property, we ran 100 iterations of the *CloudLend* simulator with different network sizes: 50, 200, and 500, while evaluating all federations established within the network following the events of removal of 20% of network members, and the modification of SLA terms specifications of 20% of the network members.

7.4.2 Results and Discussion

Test Case 1: Simulation results showed that when randomly removing 20% of members of the network the monitoring model instantly adjusts to the removal event regardless of the size of the simulated *CloudLend* network by: 1) identifying affected relationships, 2) replacing the removed member with a substitute member if exists in the network, and 3) updating relationship status with the *Federation Manager*.

The null hypothesis H_0 states that all the network sizes have equal percentage of mean replaced nodes, $H_0: \mu_1 = \mu_2 = \mu_3$. Where μ_1, μ_2, μ_3 are mean replaced nodes percentages for the network of sizes 50, 200, and 500 respectively. The alternative hypothesis H_1 states that the mean replaced nodes of at least one network is significantly different. Performing a One-Way ANOVA test to analyze the variance of the three different network sizes, with a 95% confidence level yielded p-value $<$ 0.05. Therefore, the decision rule was to reject H_0, and we concluded that different network sizes have different means of replaced nodes. Results also indicated that as the network size increases, the possibility of instantly replacing a failing node without customer intervention also increases. Figure 26, and Figure 27 depicts the results of this test case.

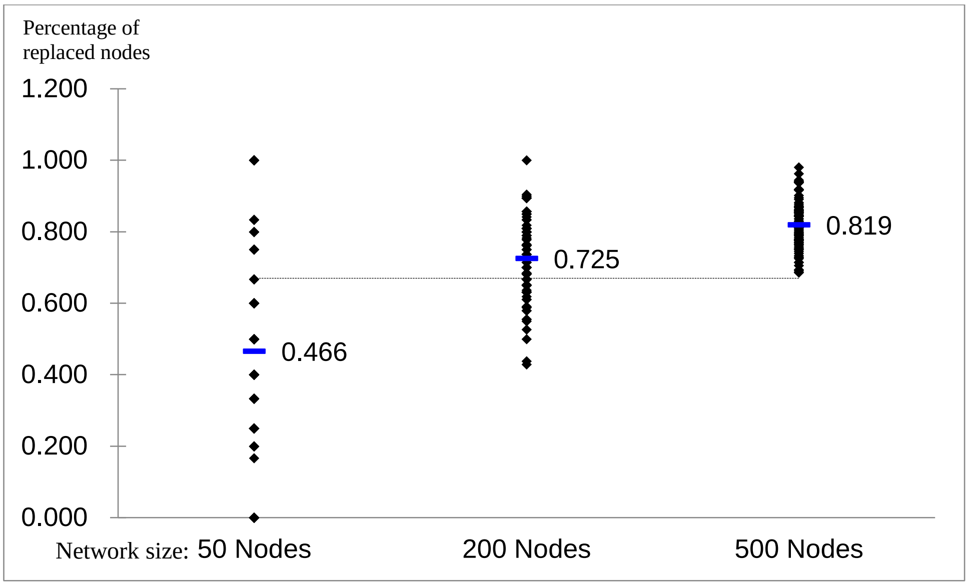

Figure 26: Analysis of variance for the percentage of replaced deleted nodes with variant network sizes using 95% confidence level

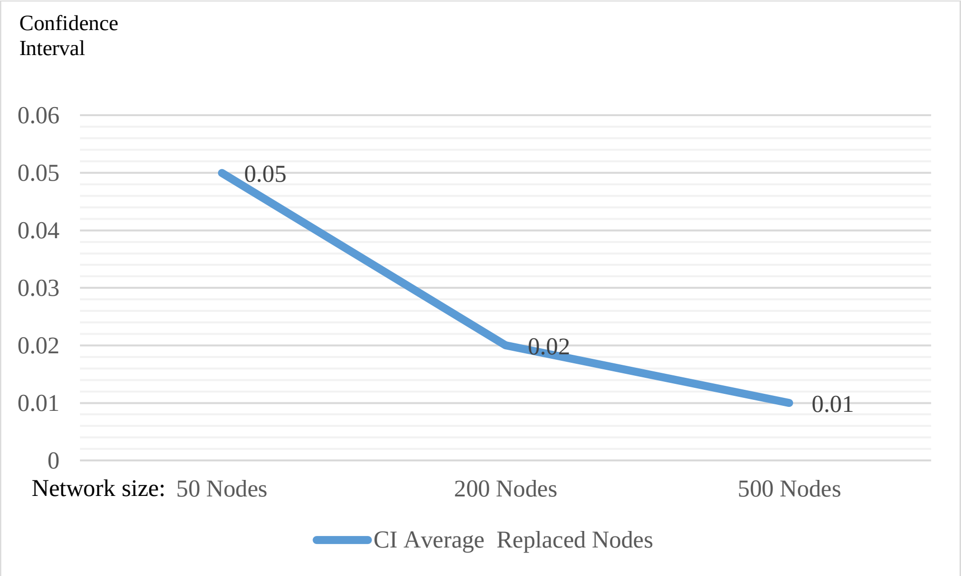

Figure 27: Confidence intervals of average replaced deleted nodes with variant network sizes using 95% confidence level

With a network of size 50 nodes we are 95% confident that 43%-51% of total 464 removed nodes were replaced with substitute nodes, while no replacement nodes were found in the network for 49-57% of the removed nodes. Furthermore, increasing the network size to 200 nodes, we are 95% confident that 71%-74% of the total 1964 removed nodes were replaced, while no replacement nodes were found in the network for 26%-29% of the removed nodes. Finally, we are 95% confident that 81%-83% of the total 4967 removed nodes were replaced with substitute nodes, while no replacement nodes were found for 17%-19% of the total removed nodes when the size of the network was increased to 500 nodes. This test shows that the proposed negotiation model exhibits elasticity, which also implies scalability that supports variation of the size of the monitored entities.

Test Case 2: When augmenting random changes in SLA terms specifications of 20% of members of the simulate *CloudLend* network, the monitoring model automatically react by evaluating affected relationships. It detects resultant SLA violations, identifies root causes, and reports SLA violations if any. Figure 28, and Figure 29 illustrate the performance of the monitoring model with regards to changes in network size.

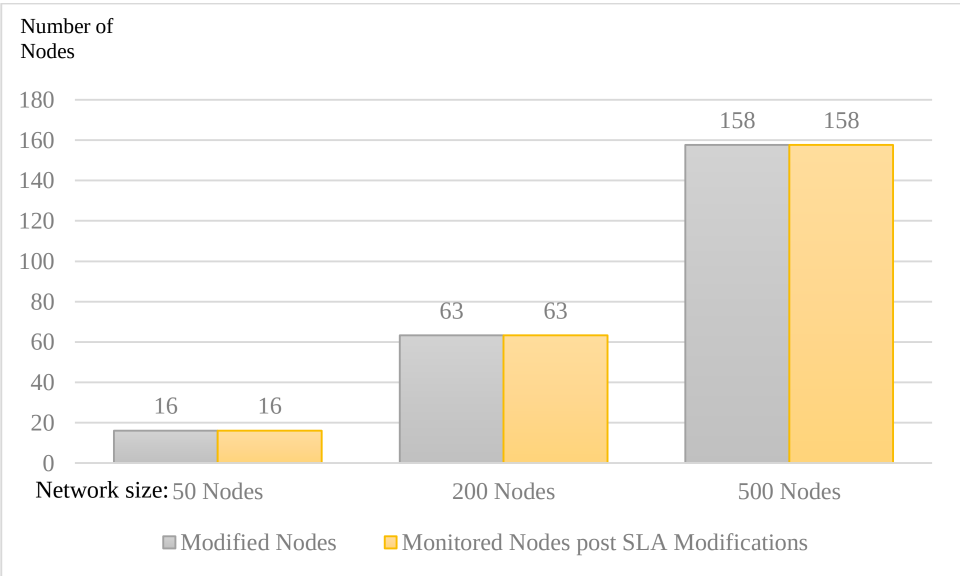

Figure 28: Adaptive monitoring post SLA modification with variant network sizes

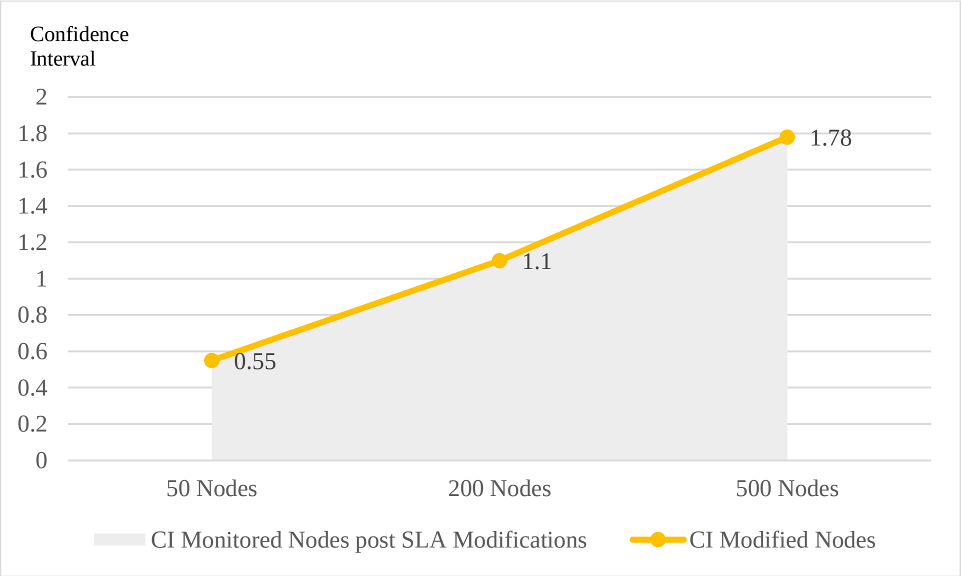

Figure 29: Confidence intervals for Adaptive monitoring post SLA modification with variant network sizes using 95% confidence level

We are 95% confident that the model was able to perform monitoring on an average of 15 -16 modified nodes in 100 networks of size 50 nodes. Moreover, we are 95% confident that 62-64 modified nodes were monitored in 100 networks of size 200 nodes, and an average of 155-159 modified nodes were monitored in 100 networks of size 500 nodes. This test shows that monitoring model is autonomic, and is able to react to irregular changes without manual interference.

Test Case 3: When evaluating all established federations within the simulated *CloudLend* network, the monitoring model achieved 100% accuracy, and was always able to detect, identify, and report nodes removal and SLA modification events instantly regardless of the size of the simulated network. Figure 30, and Figure 31 demonstrate the accuracy level of the monitoring model with regards to changes in network size.

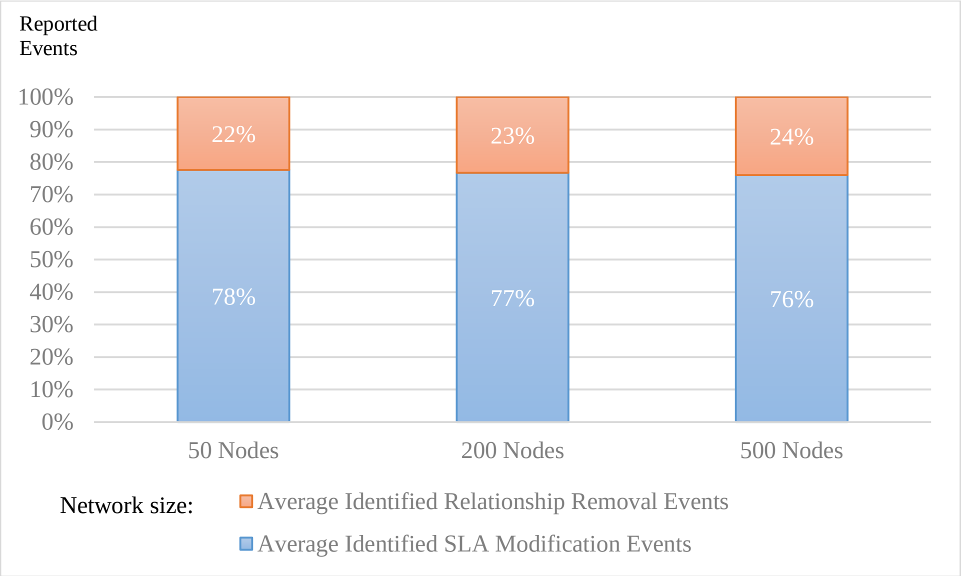

Figure 30: Reported events ratio with variant network sizes

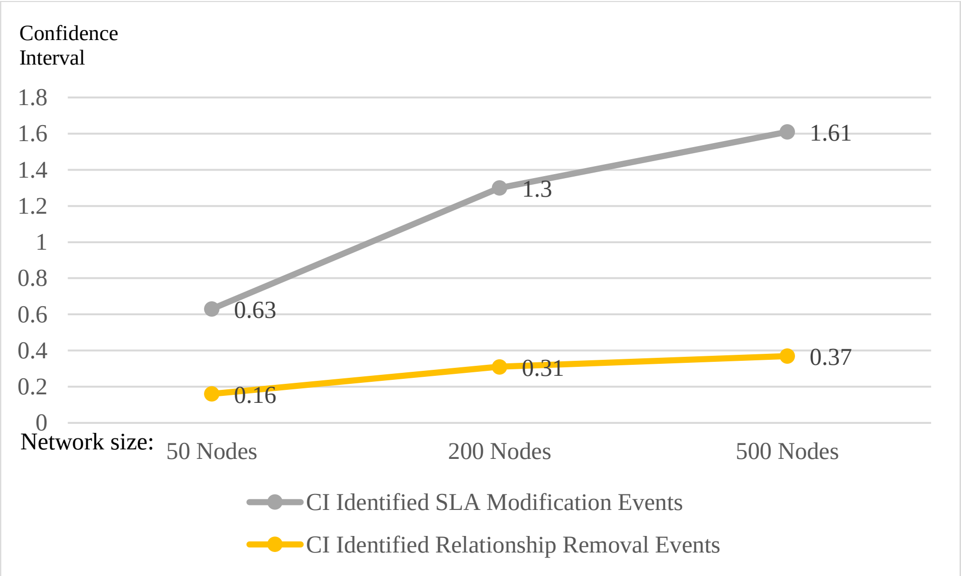

Figure 31: Confidence intervals for reported events ratio with variant network sizes using 95% confidence level

The model was able to perform monitoring on 2047 established federations in 100 networks of size 50 nodes, 100% of the simulated events were instantly reported. We are 95% confident that 21%-24% of the reported events include relationship removal events, and 75%-81% SLA terms modification events. Moreover, 8355 established federations were monitored in 100 networks of size 200 nodes, the model also instantly identified and reported all simulated events. We are 95% confident that 22%-24% of the reported events include relationship removal events, and 74%-80% SLA terms modification events. In addition, 20681 established federations were monitored in 100 networks of size 500 nodes. Similarly, all simulated events were identified and reported, we are 95% confident that 23%-25% of the reported events include relationship removal events, and 73%-79% SLA terms modification events. This test indicates that the monitoring model is timely and accurate, which is an important property of a monitoring system in a Cloud environment where Cloud providers are subject to monetary penalties in case of SLA violations. Table 7 outlines a sample *CloudLend* simulator's output results in respond to a couple of executed test cases.

Table 7: Sample of the *CloudLend* Simulator Output

TEST CASE	**SAMPLE SIMULATOR OUTPUT**
REMOVE 20% OF THE NETWORK	1. Service Provider P36 was deleted 2. Alternative Service Provider P10 was found 3. Alternative link between P31 and P10 has been established 4. Alternative link between C49 and P10 has been established 5. Service Provider P9 was deleted 6. No alternative service provider was found in the network
MODIFY SLAs OF 20% OF THE NETWORK	1. P41 Was modified 2. list of connected nodes in this link: [C22, P27, P41, P1] 3. Monitoring relation: 6, Customer: C22 4. relation: 6 encountered a violation in: Request Threshold value of: 245 req/s caused by: P1 5. Monitoring relation: 7, Customer: C22 6. No violations were found
POST EVENTS MONITORING OF ALL FEDERATIONS IN THE NETWORK	1. Service Provider P4 was deleted 2. Alternative Service Provider P44 was found 3. Alternative link between P20 and P44 has been established 4. Alternative link between C41 and P44 has been established 5. list of connected nodes in this link: [C41, P37, P20, P44] 6. Monitoring relation: 21, Customer: C41 7. relation: 21 encountered a violation in: User Threshold value of: 445 user/s caused by: P37 8. Monitoring relation: 22, Customer: C41 9. relation: 22 encountered a violation in: Request Threshold value of: 218 req/s caused by: P20 11. P33 Was modified 12. list of connected nodes in this link: [C43, P33]

	13. Monitoring relation: 25, Customer: C43 14. No violations were found

To the best of our knowledge, there are no recent work on evaluating the effect of collaborative networking on the SLA monitoring for Cloud services. Current research efforts in Cloud monitoring are focused on evaluating the actual Cloud services performance during run time using probes for sensing low-level metrics (e.g. CPU utilization, and memory consumption), (Buyya, Ranjan, & Calheiros, 2010), (Katsaros, et al., 2012), (Seo, Kim, Cui, Seo, & Lee, 2015), (Aversa & Tasquier, 2016). Furthermore, according to Assis & Bittencourt (A survey on cloud federation architectures: Identifying functional and non-functional properties, 2016), the examined Cloud federation approaches monitor the elements for the infrastructure, and application execution, but not the integrity of the federation. Whereas our evaluation of *CloudLend's* SLA monitoring model is specifically concerned with testing the model's properties within the context of a collaborative network of federated Cloud services. Table 8 indicates the properties of our SLA monitoring model with regards to the several properties that should be considered in a distributed monitoring system (Aceto, Botta, De Donato, & Pescapè, 2013).

Table 8: Key properties of distributed monitoring systems

Monitoring System Properties	***CloudLend***
Scalability **It can cope with a large number of probes**	Addressed * Scalability that supports variation of the size of the monitored entities
Elasticity **It can cope with dynamic changes of monitored entities**	Addressed
Adaptability **It can adapt to varying computational and network loads**	Not addressed
Timeliness **If detected events are available on time for their intended use**	Addressed
Autonomicity **Automatically reacting to unpredictable changes, while hiding intrinsic complexity to providers and consumers**	Addressed
Accuracy **The provided measures are accurate**	Addressed

7.5 Summary

This chapter presented the different components of the *CloudLend* simulator, in addition to illustrating the methods used to evaluate both: the SLA negotiation, and monitoring models. Obtained results proved the efficiency of the SLA negotiation model by attaining accurate expected SLA allocations for the negotiating parties. Besides evaluation results showed that the proposed SLA monitoring model exhibits three important properties: 1) elasticity, by adapting to network changes 2) accuracy by timely detecting SLA violations, and 3) autonomicity by independently reacting to changes in the network.

Chapter 8: Conclusion and Further Research Directions

This dissertation studied the problem of Cloud federation through collaborative networking, motivated by some current issues in Cloud computing such as providers' lock-in constraints, and Cloud computing infrastructure setup and running costs. This thesis proposed a collaborative network *CloudLend* that allows resource sharing among different Cloud providers, in addition to the effortless migration of customers' applications across different providers. It identified possible interactions among Cloud customers and providers within federations in *CloudLend*. Additionally, roles and responsibilities necessary to manage a federation were also highlighted. We have also addressed issues related to: service selection, and QoS guarantee in federation of Clouds. An SLA management model that administers relationships established between different Cloud services in *CloudLend* has also been defined, specified, and evaluated. This study is imperative as web applications and services nowadays demand global exposure, and customers are expecting an *always on* kind of service. Therefore, federation is considered the future of Cloud computing.

8.1 Findings with Regards to Research Questions

This research provided answers to the following research questions:

1. **Can different Cloud providers collaborate to achieve communal benefit?**

   Chapter 3 demonstrated how different Cloud providers can set their competition aside and collaborate with each other to attain joint advantages through the collaborative network *CloudLend*. We identified the decoupled components of the enabling architecture of the proposed network that is mainly comprised of the *Collaboration* and *Federation* modules, which are supervised by the *CloudLend Broker*. We also illustrated how these components interconnect to realize the collaboration of Cloud services.

2. **How are members' activities carried out in a collaborative federated Cloud environment?**

   Chapter 3 also explained the life cycle and dynamics of *CloudLend* community represented by the existence and fading of relationships among its members, starting with member's profile construction, relationship identification, SLA negotiation, service provisioning and monitoring, as well as community regulation enforcement.

3. **How would connection among Cloud services within such a network be governed?**

   Chapters 4 and 5 presented three important elements that are responsible for SLA-based relationship government in *CloudLend*: 1) XML-based SLA specification model that distinctively captures the multi-level nature of federated Cloud environments, so that adaptation to fluctuated relationships becomes feasible. 2) Agent-based SLA monitoring model that performs periodic SLA inspections, reacts efficiently to identify the source of any violation, performs the required analysis to make sure that parent SLA terms are maintained, and updates all dependent Cloud services. 3) SLA negotiation model based on game theory that fairly enables federated Cloud services to review SLAs, respond to SLA offers, and eventually sign an SLA contract.

4. **Will Cloud customers be able to dictate their QoS requirements to Cloud providers?**

   Chapter 5 explained the fair division game used for SLA negotiation by describing the players, their actions, strategies, and game outcome. This outcome enables Cloud customers to impose their QoS requirements on Cloud providers by reaching a consensus on the allocation of their weighted QoS requirements.

5. **How can Cloud services be portrayed within the *CloudLend* network?**

   Chapter 6 provided a formal representation of the *CloudLend* network and its Cloud members, as well as a customer-provider relationships establishment, in addition to a cross provider relationships establishment.

6. **How can a customer find the best service offer for his QoS requirements?**

   The proposed *CloudLend* network provides customers with the ability to evaluate all alternative providers' offers using the SLA negotiation model. By comparing utility gains resulted by all negotiation sessions, a customer can then decide to sign the contract with the provider that yielded the maximum utility gain. Chapter 6 introduced an objective function for optimizing customer satisfaction using a fair division game.

7. **How can a provider evaluate different customers' requests to achieve efficient resource utilization**?

Chapter 6 demonstrated the objective function of Cloud providers within *CloudLend* that results from comparing individual SLA terms allocation across all negotiation games. A Cloud provider's objective function represents an informed resources' utilization by which a provider is able to prioritize customers' requests, and predict the prospective demand on his resources.

8. How can service provisioning within the network be evaluated?

The proposed SLA management models were evaluated using a *CloudLend* simulator, as illustrated in chapter 7. The simulation experiments included network generation with augmented SLA specification mechanism, SLA negotiation evaluation, and SLA monitoring evaluation. Results showed that the SLA negotiation model provided an accuracy level of 94.77% regardless of the played out strategy, or the number of the SLA terms included in the negotiation game. Additionally, *CloudLend* simulation results demonstrated that the proposed agent-based SLA monitoring model guarantees three important properties, which are: elasticity, accuracy, and autonomicity.

8.2 Further Research Directions

Although the SLA enforcement phase falls out of the scope of this research; however, it imposes similar complications of SLA specification, negotiation, and monitoring when considered in federated Cloud environments. SLA enforcement is the last phase in the SLA life cycle and is carried out once an SLA violation is determined in order to trigger proper correction actions as specified in the SLA. Relationships among Cloud providers in a federated environment are basically dynamic chains of interconnected services that are bounded by multi-level SLAs. Therefore, SLA enforcement methods need to be designed with this complexity in mind, since it is not a trivial task in an intricate federated Cloud environment, where a service performance is influenced by its interconnected services. Hence, enforcement measures, which are capable of impartially distributing corrective actions among interconnected services, need to be thoughtfully designed. Additionally, we intent to address the possible overhead caused by service migration and replacement within *CloudLend*. As well as, revisiting the SLA negotiation problem considering situations where the Cloud providers initiate service requests with the customers, and where SLA terms can't be split between the Cloud customer and provider.

Further studies on federated Cloud environments shall also consider investigating open issues in Cloud computing that just got complicated with Cloud federation. Such issues are related to accounting, and security. Accounting in Cloud environments refer to the gathering, and processing of Cloud services usage reports for billing purposes. Deployment of the "pay-as-you-go" model promoted by Clouds may not be as easy as it sounds in the federated Clouds. Monitoring federated resource usage for billing purposes is required taking into account different billing schemes (postpaid, or prepaid) offered by various Cloud providers. Security assurance in federated Clouds is also an issue, since guaranteeing the security of individual Clouds in the federation does not necessarily guarantee cross-Cloud security. Security concerns may include trust management, as well as data confidentiality and integrity. Furthermore, the adaption of collaborative networking for Clouds federation rises other issues related to maintaining the welfare of the collaborative community. These issues include reputation management for building trust based on past experiences, or collected information. In addition to regulations specification that is essential to collaboration facilitation, such as mechanisms to enforce penalties and to prevent malicious behavior.

References

Aceto, G., Botta, A., De Donato, W., & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks, 57*(9), 2093-2115.

Adar, E. (2007). *GUESS: the graph exploration System*. Retrieved from http://graphexploration.cond.org/

Al Falasi, A., & Serhani, M. A. (2016). SLA Specification and Negotiation Model for a Network of Federated Clouds. *IEEE International Conference on Cloud and Big Data Computing.*

Al Falasi, A., Serhani, M. A., & Dssouli, R. (2013). A model for multi-levels SLA monitoring in federated cloud environment. *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC),* (pp. 363-370).

Al Falasi, A., Serhani, M. A., & Elnaffar, S. (2013). The sky: a social approach to clouds federation. *Procedia Computer Science, 19*, 131-138.

Al Falasi, A., Serhani, M. A., & Hamdouch, Y. (2015). A Game Theory Based Automated SLA Negotiation Model for Confined Federated Clouds. *International Conference on Cloud Computing*, (pp. 102-113).

Alamri, A., Eid, M., & El Saddik, A. (2006). Classification of the State-of the Art Dynamic Web Services Composition Techniques. *International Journal of Web and Grid Services, 2*(2), 148-166.

Alder, G. (2016). Retrieved from JavaScript Diagramming: http://www.jgraph.com/

Alhamad, M., Chang, E., & Dillon, T. (2010). SLA-based trust model for Cloud Computing. *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, (pp. 321-324).

Alhamad, M., Dillon, T., & Chang, E. (2010). Conceptual SLA framework for cloud computing. *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST).* Dubai.

Alhamad, M., Dillon, T., & Chang, E. (2010). Conceptual SLA framework for Cloud Computing. *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, (pp. 606-610).

Alhamad, M., Dillon, T., & Chang, E. (2010). SLA-Based Trust Model for Cloud Computing. *13th International Conference on Network-Based Information Systems (NBiS).* Takayama .

Alsrheed, F., El Rhalibi, A., Randles, M., & Merabti, M. (2013). Rubinstein's alternating offers protocol for automated cloud computing negotiation. *Proceedings of the 14th Annual PG Symposium on the Convergence of Telecommunications, Networking, and Broadcasting.*

Assis, M., & Bittencourt, L. (2016). A survey on cloud federation architectures: Identifying functional and non-functional properties. *Journal of Network and Computer Applications, 72,* 51-71.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks, 54*(15), 2787-2805.

Aversa, R., & Tasquier, L. (2016). Design of an Agent Based Monitoring Framework for Federated Clouds. *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA),* (pp. 115-120).

AW Procedure. (2003, October). Retrieved from Adjusted Winner Website: www.nyu.edu/projects/adjustedwinner/awProcedure.htm

Aziz, H., Brânzei, S., Filos-Ratsikas, A., & Frederiksen, S. K. (2015). The adjusted winner procedure: characterizations and equilibria. *arXiv preprint arXiv:1503.06665*.

Barabási, A.-L., & Frangos, J. (2014). *Linked: the new science of networks science of networks.* Basic Books.

Benevenuto, F., Rodrigues, T., Cha, M., & Almeída, V. (2009). Characterizing user behavior in online social networks. *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, (pp. 49-62).

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., & Morrow, M. (2009). Blueprint for the InterCloud-protocols and formats for Cloud computing interoperability. *Fourth International Conference on Internet and Web Applications and Services*, (pp. 328-336).

Binmore, K., & Vulkan, N. (1999). Applying game theory to automated negotiation. *Netnomics, 1*(1), 1-9.

Boyd, D. M., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication, 13*(1), 210–230.

Brams, S. J., & Taylor, A. D. (1996). *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press.

Brams, S. J., & Taylor, A. D. (2000). *The Win-Win solution: guaranteeing fair shares to everybody.* WW Norton \& Company.

Buyya, R., Broberg, J., & Goscinski, A. (2011). *Cloud Computing: Principles and Paradigms.* New Jersey: John Wiley & Sons, Inc.

Buyya, R., Garg, S. K., & Calheiros, R. N. (2011). SLA-oriented resource provisioning for Cloud Computing: Challenges, architecture, and solutions. *International Conference on Cloud and Service Computing (CSC),* (pp. 1-10).

Buyya, R., Garg, S., & Calheiros, R. (2011). SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions . *International Conference on Cloud and Service Computing (CSC).* Hong Kong .

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

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010). Three-phase cross-Cloud federation model: The Cloud SSO authentication. *Second International Conference on Advances in Future Internet (AFIN),* (pp. 94-101).

Chard, K., & Caton, S. (2015). Social Clouds: Crowdsourcing Cloud Infrastructure. In *Crowdsourcing* (pp. 191-217). Springer.

Chard, K., Bubendorfer, K., Caton, S., & Rana, O. F. (2012). Social Cloud Computing: A vision for socially motivated resource sharing. *IEEE Transactions on Services Computing, 5*(4), 551-563.

Chard, K., Caton, S., Rana, O. F., & Bubendorfer, K. (2016). Social clouds: a retrospective. *IEEE Cloud Computing, 2*(6), 30-40.

Chard, K., Caton, S., Rana, O., & Bubendorfer, K. (2010). Social Cloud: Cloud Computing in Social Networks. *in IEEE 3rd International Conference on Cloud Computing (CLOUD).*

Chen, H., Liu, X., Xu, H., & Wang, C. (2016). A Cloud Service Broker Based on Dynamic Game Theory for Bilateral SLA Negotiation in Cloud Environment. *International Journal of Grid and Distributed Computing, 9*(9), 251-268.

Claro, D. B., Albers, P., & Hao, J.-K. (2005, July). Selecting Web Services for Optimal Composition. *ICWS International Workshop on Semantic and Dynamic Web Processes, Orlando-USA.*

Clayman, S., Galis, A., & Mamatas, L. (2010). Monitoring virtual networks with lattice. *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, (pp. 239-246).

Dastjerdi, A. V. (2013, March). QoS-aware and semantic-based service coordination for multi-Cloud environments. *PhD Dissertation, The University of Melbourne*.

Dustdar, S., & Schreiner, W. (2005). A survey on Web Services composition. *International journal of web and grid services, 1*(1), 1-30.

Figueroa, C., Figueroa, N., Jofre, A., Sahai, A., Chen, Y., & Iyer, S. (2008). A game theoretic framework for SLA negotiation. *HP Laboratories Palo Alto, Tech. Rep. HPL-2008-5*.

Instiute for Systems Biology. (2013). *Graph Interface library*. Retrieved from GINY: http://csbi.sourceforge.net/index.html

Iosup, A., Lăscăteu, A., & Ţăpuş, N. (2010). CAMEO: enabling social networks for massively multiplayer online games through continuous analytics and cloud computing. *Proceedings of the 9th Annual Workshop on Network and Systems Support for Games*, (p. 7).

John, K., Bubendorfer, K., & Chard, K. (2011). A Social Cloud for Public eResearch. *IEEE 7th International Conference on E-Science (e-Science).*

Katsaros, G., Kousiouris, G., Gogouvitis, S. V., Kyriazis, D., Menychtas, A., & Varvarigou, T. (2012). A Self-adaptive hierarchical monitoring mechanism for Clouds . *Journal of Systems and Software , 85*(5), 1029-1041. Retrieved from http://www.sciencedirect.com/science/article/pii/S0164121211002998

Keahey, K., Tsugawa, M., Matsunaga, A., & Fortes, J. A. (2009). Sky computing. *Internet Computing, IEEE, 13*(5), 43-51.

Keller, A., & Ludwig, H. (2003). The WSLA framework: Specifying and monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management, 11*(1), 57-81.

Kim, W., Jeong, O.-R., & Lee, S.-W. (2010). On social Web sites. *Information Systems, 35*(2), 215-236.

Luck, M., Ashri, R., & d'Inverno, M. (2004). *Agent-based software development.* Artech House.

Maamar, Z., Hacid, H., & Huhns, M. N. (2011). Why Web Services Need Social Networks. *IEE Internet Computing, 15*(2), 90-94.

Maaradji, A., Hacid, H., Daigremont, J., & Crespi, N. (2010, May). Towards a Social Network Based Approach for Services Composition. *IEEE International Conference on Communications (ICC)*, (pp. 1-5).

Mell, P. M., & Grance, T. (2011). Sp 800-145. the nist definition of cloud computing.

Mian, R., Martin, P., & Vazquez-Poletti, J. L. (2013). Provisioning data analytic workloads in a cloud. *Future Generation Computer Systems, 29*(6), 1452-1458.

Mohaisen, A., Tran, H., Chandra, A., & Kim, Y. (2011). SocialCloud: Using Social Networks for Building Distributed Computing Services. *arXiv preprint arXiv:1112.2254*.

Myerson, R. B. (2013). *Game theory.* Harvard university press.

Neuman, J. v., & Morgenstern, O. (1944). Theory of games and economic behavior. *Theory of games and economic behavior*. Princeton.

Nie, G., E., X., & Chen, D. (2012). Research on Service Level Agreement in Cloud Computing. In *Advances in Electric and Electronics* (pp. 39-43). Heidelberg: Springer Berlin.

Nie, G., Xueni, E., & Chen, D. (2012). Research on Service Level Sgreement in Cloud Computing. In *Advances in Electric and Electronics* (pp. 39-43). Springer.

Oberle, K., & Fisher, M. (2010). ETSI CLOUD initial standardization requirements for Cloud Services. In *Economics of Grids, Clouds, Systems, and Services* (pp. 105-115). Springer.

O'Madadhain, J. (2010, January). *Java Universal Network/Graph Framework*. Retrieved from JUNG: http://jung.sourceforge.net/

Open ID. (2012). *Open ID*. Retrieved from http://openid.net

OpenSocial. (2012). *OpenSocial*. Retrieved from http://docs.opensocial.org

Petcu, D., Macariu, G., Panica, S., & Crăciun, C. (2013). Portable Cloud applications- From theory to practice. *Future Generation Computer Systems, 29*(6), 1417-1430.

Pezzi, R. (2009, September 20). *Information Technology Tools for a Ttransition Economy.* Retrieved from Socialcloud: http://socialcloud.net/papers/ITtools.pdf

Rochwerger, B., Breitgand, D., Epstein, A., Hadas, D., Loy, I., Nagin, K., . . . others. (2011). RESERVOIR - When one Cloud is not enough. *IEEE Computer, 44*(3), 44-51.

Seo, S., Kim, M., Cui, Y., Seo, S., & Lee, H. (2015). SFA-based cloud federation monitoring system for integrating physical resources. *2015 International Conference on Big Data and Smart Computing (BIGCOMP)*, (pp. 55-58).

Silaghi, G. C., Şerban, L. D., & Litan, C. M. (2012). A time-constrained SLA negotiation strategy in competitive computational grids. *Future Generation Computer Systems, 28*(8), 1303-1315.

Sim, K. M. (2009). Agent-based cloud commerce. *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, (pp. 717-721).

Smith, K. (2016). 96 Amazing Social Media Statistics and Facts for 2016. *96 Amazing Social Media Statistics and Facts for 2016*. (Brandwatch, Ed.) Retrieved from https://www.brandwatch.com/2016/03/96-amazing-social-media-statistics-and-facts-for-2016/

Ter Beek, M., Bucchiarone, A., & Gnesi, S. (2007). Web Service Composition Approaches: From Industrial Standards to Formal Methods. *Second International Conference on Internet and Web Applications and Services,* (pp. 15-20).

Torkashvan, M., & Haghighi, H. (2012). CSLAM: A framework for cloud service level agreement management based on WSLA. *Sixth International Symposium onTelecommunications (IST),* . Tehran.

Torkashvan, M., & Haghighi, H. (2012). CSLAM: A framework for Cloud Service Level Agreement management based on WSLA. *Sixth International Symposium on Telecommunications (IST)*, (pp. 577-585).

UC Berkeley. (2012). Retrieved from Prefuse, information visualization toolkit : http://prefuse.org/

Urquhart, J. (2009, April). The Inter-Cloud and Internet Analogies. *The Inter-Cloud and Internet Analogies*. Retrieved from http://blogs.cisco.com/datacenter/the_inter-cloud_and_internet_analogies/

Venticinque, S., Aversa, R., Di Martino, B., Rak, M., & Petcu, D. (2010). A cloud agency for SLA negotiation and management. *European Conference on Parallel Processing*, (pp. 587-594).

W3C. (2015). Extensible Markup Language (XML). *Extensible Markup Language (XML)*. Retrieved from https://www.w3.org/XML/

Wang, C., Schwan, K., Talwar, V., Eisenhauer, G., Hu, L., & Wolf, M. (2011). A flexible architecture integrating monitoring and analytics for managing large-scale data centers. *Proceedings of the 8th ACM international conference on Autonomic computing*, (pp. 141-150).

Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: a perspective study. *New Generation Computing, 28*(2), 137-146.

Wieder, P., Butler, J. M., Theilmann, W., & Yahyapour, R. (2011). *Service level agreements for cloud computing.* Springer Science \& Business Media.

Wieder, P., Butler, J., Theilmann, W., & Yahyapour, R. (2011). *Service Level Agreements for Cloud Computing.* Heidelberg: Springer.

Williams, J. L. (2009). An implementors perspective on interoperable Cloud APIs. *Cloud interoperability roadmaps session, OMG Technical Meeting.*

Wu, L., Garg, S. K., & Buyya, R. (2015). Service Level Agreement (SLA) based SaaS Cloud Management System. *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on*, (pp. 440-447).

Yangui, S., Marshall, I.-J., Laisne, J.-P., & Tata, S. (2014). CompatibleOne: The open source cloud broker. *Journal of Grid Computing, 12*(1), 93-109.

Yaqub, E., Wieder, P., Kotsokalis, C., Mazza, V., Pasquale, L., Rueda, J. L., . . . Chimeno, A. E. (2011). A generic platform for conducting SLA negotiations. In *Service Level Agreements for Cloud Computing* (pp. 187-206). Springer.

Zheng, X., Martin, P., Powley, W., & Brohman, K. (2010). Applying bargaining game theory to web services negotiation. *Services Computing (SCC), 2010 IEEE International Conference on*, (pp. 218-225).

Zhou, J., Athukorala, K., Gilman, E., Riekki, J., & Ylianttila, M. (2012). Cloud architecture for dynamic service composition. *International Journal of Grid and High Performance Computing (IJGHPC), 4*(2), 17-31.

List of Publications

Falasi, Asma Al, M. Adel Serhani. “SLA Specification and Negotiation Model for a Network of Federated Clouds: *CloudLend*” IEEE International Conference on Cloud and Big Data Computing, July 2016.

Falasi, Asma Al, M. Adel Serhani, and Younes Hamdouch. “A Game Theory based Automated SLA Negotiation Model for Confined Federated Clouds.” CLOUDCOMP 2015, 6th International Conference on Cloud Computing, October 2015.

Falasi, Asma Al, M. Adel Serhani, and Rachida Dssouli. "A Model for Multi-levels SLA Monitoring in Federated Cloud Environment." Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC). IEEE, 2013.

Falasi, Asma Al, M. Adel Serhani, and Said Elnaffar. "The Sky: A Social Approach to Clouds Federation." Procedia Computer Science 19 (2013): 131-138.

Asma Al Falasi, M. Adel Serhani (2011) A Framework for SLA-based cloud services verification and composition, 287-292. In 2011 International Conference on Innovations in Information Technology.

Asma Al Flasi, M. Adel Serhani, Ezedin Barka (2011) A collaborative reputation-based vetting model for online certification of businesses, 1-6. In Proceedings of the Second Kuwait Conference on e-Services and e-Systems - KCESS 2011.

Asma Al Flasi, M. Adel Serhani, “End-to-End QoS management in federated Clouds: *CloudLend*”, will be submitted to the IEEE Transaction on Service computing, November 2016.

Appendix

CloudLend Simulator Class Diagrams

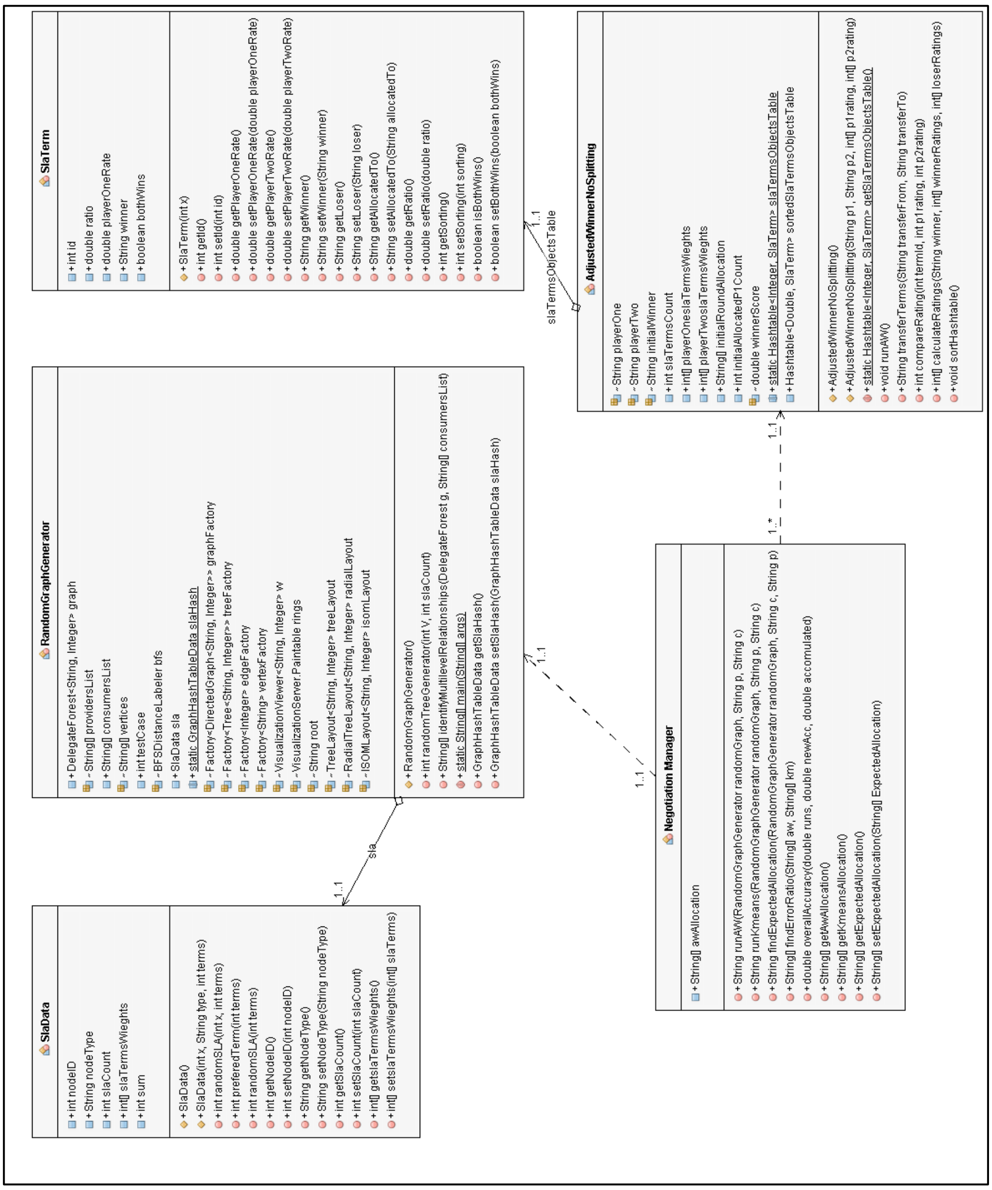

Figure 32: SLA negotiation model Class diagram

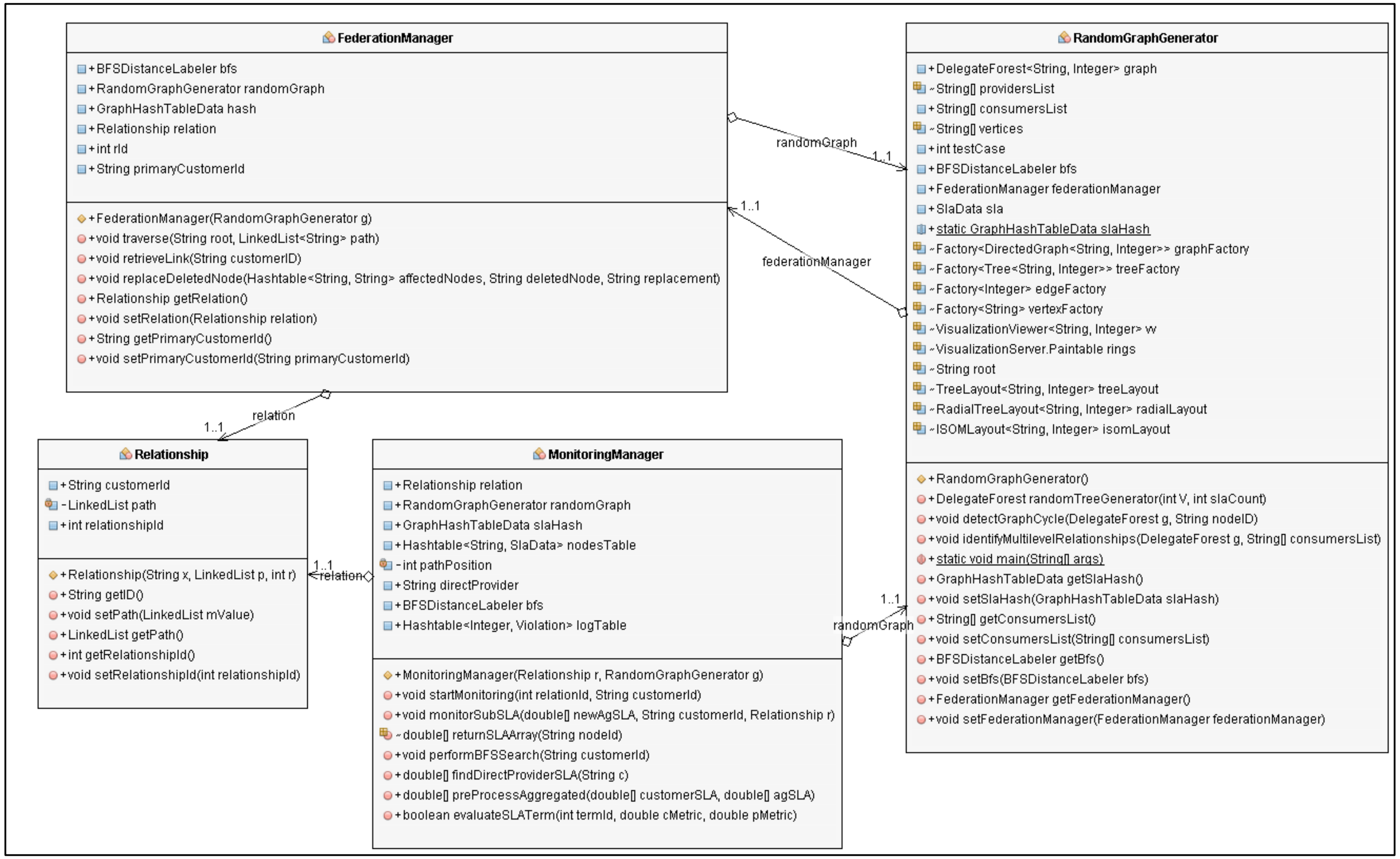

Figure 33: SLA monitoring model Class diagram