Scheduling Security Model for a Cloud Environment

Abdullah Sheikh

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Sciences
Durham University
England

Dedicated to

I would like to dedicate this PhD to my parents for everything they have given me. Then I would like to dedicate this PhD to my wife Asrar, for her unlimited support and to my children Yazan and Aseel.

Scheduling Security Model for a Cloud Environment

Abdullah Sheikh

Submitted for the degree of Doctor of Philosophy

August 2019

Abstract

- **Context:** Scheduling in the cloud is a complex task due to the number and variety of resources available and the volatility of usage-patterns of resources considering that the resource setting is on the service provider. This is compounded further when security issues and Quality of Service (QoS) are also factored in.

- **Aim:** The aim of this research is to address limitations and gaps in current approaches of resource scheduling in cloud computing by developing a Scheduling Security Model (SSM).

- **Method:** Considering security as a key element that cloud services rely on which affects the services performance, cost and time concerns within the security constraints of the cloud service. Furthermore, this research will investigate and define the considerable challenges facing trusted and cost-effective resource scheduling in cloud computing environments. The SSM analyses the customer requirements for a service then works to schedule all tasks submitted to run over available resources depending on the security level. It then uses the SSM Algorithm and the Fast-Track technique to reduce the cost and the overall waiting time.

- **Results:** The worked examples of the SSM with different scenarios and comparing the SSM against other approaches in the same field show that the SSM can meet the customer requirements for cost effective and the QoS required.

- **Conclusions:** The advantages from the results show that the SSM can reduce the overall service cost compared to other approaches.

Publication List

- A. Sheikh, M. Munro and D. Budgen, ***SSM: Scheduling Security Model for a Cloud Environment***, International Conference on Cloud and Big Data Computing (ICCBDC 2018) ISBN 978-1-4503-6474-4, ACM, Vol. 2, No. 1, pp. 11–15, 2018.

- A. Sheikh, M. Munro and D. Budgen, ***Cost and Effect of Using Scheduling Security Model in a Cloud Environment***, International Conference on Cyber Security and Cloud Computing and Edge Computing and Scalable Cloud (CSCloud/EdgeCom), IEEE, pp. 95–101, 2019.

- A. Sheikh, M. Munro and D. Budgen, ***Systematic Literature Review (SLR) of Resource Scheduling and Security in Cloud Computing***, International Journal of Advanced Computer Science and Applications (IJCSA), IEEE, Vol. 10, No. 4, pp 35–44, 2019.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Sciences, University of Durham, England.
No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to thank my supervisor Professor Malcolm Munro for his invaluable advice, guidance, wisdom, and feedback during my PhD.

Also, I would like to thank my supervisor Professor David Budgen for his support and comments through all stages in my PhD.

Thank you to Durham University and the Department of Computer Science for facilitating and supporting my study during this PhD.

I would like to thank my parents for all support, and to all my wife's family for their support through my life.

My special thanks to my wife, and my children for their immeasurable and unlimited support, and their prayer for me to complete this PhD.

List of Abbreviations

DaaS Data as a Service

IaaS Infrastructure as a Service

IT Information Technology

NIST National Institute of Standards and Technology

PaaS Platform as a Service

QoS Quality of Service

SaaS Software as a Service

SLA Service Level Agreement

SLOs Service Level Objectives

SSM Scheduling Security Model

VM Virtual Machine

VMs Virtual Machines

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Cloud computing is used by a number of different sectors, predominantly by educational and business, as well as for personal use, for various purposes. Due to the rapid growth and the development in technology and facilities that cloud services can provide it has added a fascinating transformation to the Information Technology (IT) industry. Also, cloud computing provides convenient services enabling access to different computing resources such as networks, storage, and applications.

Cloud computing includes services such as data services, storage services, scheduling services, accessing to applications via the Internet, on-demand self-service, and service management. Data service is about all database services, processing, and data store. While storage services include using a cloud storage system to manage saving data remotely in a different storage location. The scheduling services include allowing customers to execute tasks on virtual resources and trying to allocate these tasks on these resources efficiently.

All these services can be provided on customer request without or with less service provider interaction. For example, a customer can request a storage space by submitting the request to a provider website. Then the customer can get the service by finalising service payment without any interaction from the service provider.

Cloud services bring various benefits to stakeholders (providers and customers). These benefits include wide access to software and applications over the Internet without any need to install any software on the customer terminal device. Moreover, using cloud services can be cost effective, as the cloud computing environment depends on reducing infrastructure cost.

Having said that, there are a number of different risks related to the use of cloud computing. For instance, risks pertaining to security and privacy (or lack thereof) are considered to be a big concern for all parties that are involved in the cloud-based services. Thus, any breach or failure in security will cause loss of customers and business. Another risk that, makes customers aware of the service they receive is that they will be locked into a contract with one provider until their contract is complete. As a result, service providers are more concerned and conscious of providing a better and more trusted service.

These considerations include the need of improving the Quality of Services (QoS) provided. QoS includes different aspects such as time, service performance, reducing cost, and some non-functional requirements like reliability and recovery [39]. The success of applying these QoS aspects will improve the cloud services to meet customer expectations.

With all the benefits of the cloud, security is still one of the main concerns that affect the use of the cloud service. Cloud providers will be subject to many threats at different level of the cloud. Similarly, customers have concern about security and they share some responsibility with the cloud providers to keep the service security at a high level. For example, a customer requests a cloud service with a set of tasks with different security levels, it is required to have a technique that can handle this request. This technique should be able to execute the tasks submitted in the right order combined with security and QoS aspects.

Executing tasks requires using a scheduling process that serves security as the main category, then uses priority to put tasks in right order. Security as a feature will be applied to all parts of the service, and the QoS will be applied to make the service more reliable and more efficient while the service is running. This complex request should be cost effective because the customer needs a cloud service that is secure, reliable, and with a very competitive cost compared to other service provider.

In this thesis, an investigation of the current situation shall be described, followed by a discussion on the requirement of having a cloud security model that is based on costs that can manage requests, focusing on security as a main feature that is associated with QoS aspects to meet the customer requirements. The model then seeks to execute scheduled tasks over allocated resources.

Moreover, this thesis defines a Scheduling Security Model (SSM) for a cloud environment that uses security as a main feature and the QoS to deal with customers' requests with different security levels.

1.2 About this Research

This thesis focuses on the issues related to scheduling and security within the cloud environment. In addition, it aims to address issues related to the cost of providing such a service, and also on how these overall costs can be reduced. Whilst, taking into consideration the customers' requirements.

In light of this, the thesis shall develop a model that can establish a cloud service that is based on a set of a predefined requirements from customers, which are achieved by identifying the security level required for each task.

1.3 Aims and Objectives

This section provides the research aims and objectives explaining why security is an important factor to be considered for scheduling in cloud computing.

1.3.1 Research Aim

After reviewing recent research and work in cloud security and how it affects scheduling in the cloud, the main aim of this research is to develop a model that determines security constraints to secure services, and then to use this model in scheduling to achieve a good QoS standard.

1.3.2 Research Objectives

In order to achieve the aim of this research, the following research objectives will be undertaken:

- To consider previous research for further investigation to understand the security constraints of scheduling.
- To identify the type of security constraints used by cloud providers and customers.
- To identify the evaluation metrics from recent research into scheduling in the cloud.
- To develop a scheduling security model that considers security as main factor for the resource scheduling process.
- To analyse and assess the scheduling security model using tools for scheduling in the cloud associated with the QoS.

1.3.3 Research Method

The method for this research is as follows:

- To develop a model that addresses some of the security limitations that affect resource scheduling such as: data security (confidentiality, intrusion, large data volume), service availability, and reliability.
- The model design should cover some resource scheduling limitations such as: performance, cost, resource scaling, provision of heterogeneous resources, security breaches on the running virtual machines (VMs).

1.4 Research Questions

Taking the aims and objectives into consideration, this research will attempt to answer the following main, secondary and evaluation questions:

1.4.1 Main Questions

1. To what degree can a Scheduling Security Model (SSM) be developed or adapted from existing models to incorporate security and scheduling?
2. What are the barriers to scheduling, in terms of security and how do they affect scheduling?
3. To what degree can any barriers identified to the use of the SSM be overcome?

1.4.2 Secondary Questions

The following questions are secondary questions that arise as a result of (and in relation to) the main research questions. They shall also be a part of the literature review process:

1. Within the field of cloud-based service, what previous research has been conducted on scheduling in reference to security constraints?
2. Aside from security, what other constraints need to be considered by cloud stakeholders in terms of scheduling on the cloud.

3. What different types of security constraints have been identified and what do they defend against?

4. What evaluation metrics have been used to help to evaluate recent research into Scheduling in the cloud?

5. Based on the analysed research identified, which form of security aware scheduling merits further research, and why is more research needed and what issues should the further research be addressing?

1.4.3 Evaluation Questions

The following questions are the evaluation questions that will be used to see how this research achieved its aims:

1. How does the SSM improve the security aspects of the cloud service?

2. How does the SSM impact resource scheduling and performance and security?

3. How well does the SSM help to achieve QoS?

1.5 Research Criteria for Success

To assess whether this thesis has fulfilled its objectives, the following aspects have been identified as the criteria for success.

1.5.1 Identification of Security Issues in Cloud Computing

It is very important to identify security issues in cloud computing as it is a major concern to all parties in the cloud environment. This will help to develop the new service model.

1.5.2 Development of a Scheduling Security Model

This thesis proposes a new service model considering the security as a main feature to drive the scheduling process in the service.

1.5.3 Evaluating the Model

To determine the overall effectiveness of the new model it will be evaluated.

1.5.4 Comparison Against Other Approaches

For more investigation, the model will be compared against other approaches to see the advantages and disadvantages of all approaches.

1.6 Thesis Outline

The work in this thesis discusses existing research that has been completely or partially obtained from journals or conference papers published during the period of completing this study.

The thesis is subsequently organised in the following manner:

- Chapter 2: presents the literature survey of research in Scheduling, Security, Cloud Environment, Scheduling in the Cloud, and Security in the Cloud.
- Chapter 3: defines and discusses the proposed Model, the SSM Components, the calculated components, the SSM algorithm, and shows some basic Examples of Costs.
- Chapter 4: presents the SSM Results, Examples and Scenarios used to examine the SSM. Then it shows a comparison of SSM and other Models.
- Chapter 5: discusses the evaluation, Example and Scenarios, and the Comparison of the SSM with other approaches.
- Chapter 6: concludes the thesis with a summary of main findings and discusses the future research.

1.7 Summary

This chapter presented an overview of this research background, then it has explained the research aims. After that, the research objectives have been addressed with the research method. Next, it discussed the research questions along with the evaluation questions. Then the research criteria for success were given.

Chapter 2

Literature Survey

2.1 Introduction

This chapter discusses and gives an overview of all the features that are involved to develop a new service model which are Scheduling, Quality of Service (QoS), Security, and Cloud Environment. The first part is Scheduling and it will be discussed to explain why scheduling is important process to make an efficient use of the cloud resources. Then it discusses the Quality of Service (QoS) and how it is important to have a better service quality considering the affecting factors. After that, as the security is a shared component for all parts it will show that how the materials need to be secured even physically or digitally and how it can be classified to different levels. Next, the Cloud Environment is discussed by defining the cloud and its characteristics and architecture. Then it discusses its service models and deployment models. Then in the following section, a discussion of the relation between scheduling and the cloud and what tools are used for cloud scheduling. Then the next section presents the relation between the security and the cloud and what security issues that need to be considered in the cloud service. Then it shows the security issues in the deployment models explaining the most common issues that need to be addressed. After a literature review has been completed to investigate the related work of this thesis finding the research direction and the assumption made to develop the new model.

2.2 Scheduling

Scheduling is a process of decision making to deal with allocating resources to tasks within a certain amount of time [50]. There are many types of resources and it can be a machine in a workshop, or resource in computing environment [38]. Scheduler has been classified as follows [50] [22]:

- Batch Scheduling: used to avoid any handling during the running time [28]. There are two types of batch scheduling: serial and parallel. In serial batching, tasks with same setting can be executed one by one on a machine. In parallel batching, a set of tasks can be grouped and executed at same time.
- Interactive: to allow decision making at running time to take an immediate response.
- Real Time: the ability to schedule tasks with specific time requirements.
- Parallel: tasks or group of tasks executed at the same time in one or more VMs [50].

There are three level of scheduling decisions:

1. Long Term: to control and decide what task execute first and to be supported once at anytime.
2. Medium Term: to control switching tasks for different criteria such as non active, fault, and low priority.
3. Short Term: to allow frequent interaction to make decisions in short time slot.

The main scheduling goals are:

1. Performance:
   The scheduling algorithms should be able to consider the following measures in order to get good performance behaviour:
   (a) maximise CPU Utilisation: to control the amount of tasks that can be processed.

(b) maximise Throughput: to execute as many tasks as possible in a certain amount of time.

(c) maximise Scheduling Efficiency: to execute all tasks without interruption.

(d) minimise Waiting Time: to reduce the amount of time that is needed for executing tasks for users.

(e) minimise Energy: to control and reduce the power consumption of resources.

2. Fairness:

   One of the important goals of scheduling is to treat all tasks to run in a reasonable time.

   (a) Equal CPU consumption: to allocate tasks the same processing time in the CPU.

   (b) Fair per(user, process, thread): giving all the same characteristics for execution.

   (c) CPU bound, I/O bound: to allow direct priority to task from a user.

3. Unfair:

   Sometimes the scheduling process tends to be unfair by giving advantage to tasks over another for a specific aim.

   (a) Priority System: to allow tasks with higher priority to run before the lower priority one.

   (b) Avoid starvation: to prevent that any task stays with no processing.

2.3 Quality of Service

Quality of Service (QoS) is one of the important factors that can help to improve any services, software, and applications [39]. So, the QoS means that the essential services features should meet all customer requirements.

According to Ramadan et al. [39], to have a good QoS there are some factors that need to be considered which affect the overall service as follows:

- Flexibility: It is all about managing all main function without any harm to the system.
- Maintainability and Readability: Similar to the flexibility but it is more focusing on error correction and making any modification needed.
- Performance and Efficiency: It is all about the response time and making sure there is no delay or unexpected waiting time.
- Scalability: It is about responding to customers' activities in reasonable amount of time.
- Availability and Robustness: It is all about the availability all the time even if a failure has occurred.
- Usability and Accessibility: It is all about making the user interface the most visible side by making it very comfortable for the customer and easy to use.
- Platform Compatibility: For better quality the service should be made to run on as many different platforms as it can on with different system environments such as operating systems, and internet browsers.
- Security: It is the most important factor that needs to be considered in any service, and for QoS there is a need to apply security policies to make sure there are no security breaches at any level.

2.4 Security

Security is a concept that the process protect from physical or digital unauthorised use of any asset [32]. Also, security is a critical feature for any Service. The service must be secure and trusted for both customer and provider as they have both agreed to the Service Level Agreement (SLA) [55]. Security issues can affect Data, Networks, Communications, Privacy, unauthorised access and most things connected

via the internet.

All of these variables need to be protected, and each one requires a different way of security. So, these variables can be classified into different security levels from high to low. This classification depends how much valuable information is included in each asset. For example, storing very important government data requires a very high security level. This security level includes physical security measures and secure network connection and secure encrypted data storage. Also, it may require a limited access control to manage the process of retrieving and storing this data.

So, Security is a very critical point that needs to be aware of all kind of information for all levels such as individuals, academic, business, and government even if it is digital or non digital materials.

2.5 Cloud Environment

This section serves as a background and a general view of cloud computing, basic cloud architecture and cloud features. Also, it shows the considered top ten obstacles that are facing the growth of cloud computing and how they are related to the security in the cloud. Then it presents the research method for this Systematic Literature Review (SLR). After that, it explains why this SLR is needed to be performed.

2.5.1 Cloud Definition

There are many different definitions of cloud computing. The National Institute of Standard and Technology (NIST) [30] gives a basic definition of cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

To obtain a cloud service a customer needs to contact a service provider. This communication process makes the customer and the provider reach an agreement of the level of the service. This agreement is referred as the Service Level Agreement (SLA) [35]. This SLA is the basis for the expected level of the service between the customer and the provider. The Service Level Agreement (SLA) is an agreement between a service provider and a customer, that specify the level of the service provided [1] [31] [33]. Also, both provider and customer follow the rules and conditions of this agreement to keep the service secure without any security issues. The provider of a cloud architecture can offer various services to a customer. Quality of Service (QoS) refers to the cloud stakeholders' expectation of obtaining a desirable service meeting requirements such as timeliness, scalability, high availability, trust and security specified in the Service Level of Agreement SLA [33]. For this research, Quality of Service (QoS) includes the following concerns:

- **Security:** Security is a shared responsibility between cloud providers and customer to ensure that the level of security is at a desired level. Customers need to be aware of security from their side and protect their service from any threats. Cloud providers are able to achieve better scalability by running multiple virtual machines on physical machines. They have to defend the service against any security risks from any unauthorised physical access, data security, security software, and resource security. Other cloud providers who do not use virtual machine have to secure servers and data storage from any security risks. Then any security risks in the virtualisation technology that allows co-occupant virtual machines to make unauthorised access could compromise the information assets of customers [31].

- **Service Performance:** A customer that requires a certain level of the service performance will need provider guarantees to run the required service in the cloud. As a result, the service quality is included in the SLA and the service provider must provide a service with good behaviour, stable network connection, process time with no delay, and reduced cost.

Services can vary both in terms of functionality (such as storage capacity or pro-

cessor count) or in terms of the Quality of Service (QoS) provided [48]. In terms of the QoS a provider will offer a defined SLA which the customer can use when determining the 'best provider for their needs.

According to Mell and Grance [30], the cloud architecture is a combination of the following three components:

1. **Essential Characteristics:** The essential characteristics refer to a set of cloud features that allow providers and customers managing, accessing, and measuring the cloud services and resources. These characteristics provide cloud providers and customers with different level of control to measure the provision of the service. From a security prospective, each characteristic has a different security concern for both provider and customer. These security concerns include access control and data security [5]. Access control includes accessing, managing the service, and access availability. Data security includes data confidentiality, date integrity, and data availability.

   The five essential cloud characteristics are:

   1. **On-demand self-service:** A customer can manage and control the service resources such as sever time and network storage without any physical interactions by the provider.
   2. **Broad network access:** Customers can access and use the cloud service from anywhere across the network.
   3. **Resource pooling:** Providers can serve customers with different resources according to customer demand. Resources such as storage, processing, physical machine, and network bandwidth [5]. Customers do not need to be concerned about physical location of resources.
   4. **Rapid elasticity:** Resources can be rapidly scaled outward or inward at any time according to customer demand.
   5. **Measured service:** Measured service is the ability to track and control the usage of the resources which can be performed by customers.

2. **Service Models:** Service models in the cloud define to a customer the type of the system management and system operations, and the type of the access to cloud systems. According to Nallur et al. [31] service availability, security and performance are the main elements that are considered to affect the cloud service in the service models. Based on the SLA, customers have to trust the provider on the service availability. The only concern is if there is any downtime to the service it will be the time of the service recovery to obtain the service again. The recovery process is the responsibility of the service provider based on the SLA. Both provider and customer are involved in security such as data security and protection. The provider is concerned about providing a secure and reliable service via the network. Service performance means that the service provided to customers at a satisfactory level and good quality. There are three types of service model, each service model provides different capabilities to obtain the service:

    1. **Infrastructure as a Service (IaaS):** To provide a basic form of the service such as a Virtual Machine (VM), virtual storage and network bandwidth [31]. Customers have to configure the setting and install any needed operating system and software before running the service. One of the main security concerns in IaaS is that the provider has to check that there is no VMs interfering while the service is running.

    2. **Software as a Service (SaaS):** Here, software and applications are provided by the cloud provider which let customers use these applications [30]. Customers can have access to the service from different devices via different interfaces such as web browser or a program interface. One security concern that needs to be considered is web browser security. The level of the browser security is very important, weak browser security can let an attacker get important information or hijack the customer resources and data.

    3. **Platform as a Service (PaaS):** In this form, the cloud provider provides a platform that allows the customer to develop their application but

the cloud provider is still responsible for maintenance and all upgrades of the platform [30].

3. **Deployment Models:** Deployment models describe how the cloud services deliver to customers. According to Dillon et al. [15] there are many security concerns on the cloud deployment models including data privacy and trust, policies, and data transfer. As a result of these concerns providers have to secure cloud services. Also, providers need to apply security policies that can handle data access and security. The four deployment models, which specify the availability of using cloud service [30], are:

    1. **Public:** To specify that the cloud service is accessible with no restriction for all users.
    2. **Private:** To make the cloud services available to a particular single group.
    3. **Community:** To make the cloud services shared between limited groups sharing similar concerns.
    4. **Hybrid:** A hybrid cloud includes services using multiple cloud combined together, for example joining services and making some parts private and other parts public or community [20].

2.6 Scheduling in the Cloud

Scheduling is a process or mechanism applied to minimise wasting limited resources by efficiently allocating them among all active nodes [63]. Nodes or Virtual Machines (VMs) are the virtual resources that are assigned to customers for running the service and executing tasks [5]. Scheduling is a very complex operation in cloud computing and it is used to allocate resources, improve server utilisation, enhance service performance, and execute tasks [54].

Scheduling can be either static or dynamic for scheduling resources in cloud computing, which can provide sufficient use of cloud resources to meet QoS re-

quirements [18]. Furthermore, using scheduling techniques can avoid conflicts of allocating active resources. For example, scheduling can avoid duplication of allocating the same virtual resource in the same time. Also, it can manage limited resources by handling high demand of requests by using a dynamic method that can update the system regularly and to execute tasks over resources. However, there are some issues that need to be considered such as security, limited resources, virtual machines and applications.

Executing and running tasks over the allocated resources raises some security issues that need to be considered such as data security, and service security. Data security includes privacy, integrity, and protection from any threats and attacks. Service security includes resource security, and privacy. So, there is a need to consider these issues and the security constraints include data security, and availability to get an optimised resource schedule. For this research, the main focus will be on the resource scheduling mechanisms where security is factored into the cloud model.

According to Singh et al. [51], there are two main objectives of resource scheduling as follows:

- To identify suitable resources for scheduling workloads on time and to enhance the effectiveness of resource utilisation. Workloads refer to the tasks that customers want to run over the resources.
- To identify heterogeneous multiple workloads to fulfil the QoS requirements such as CPU utilisation, availability, reliability and security.

2.6.1 Tools for Cloud and Scheduling

There are many software tools used for testing, managing and provisioning resources in Cloud Environments and in self-hosting [6]. However, one of main differences is that cloud service provides a huge number of resources via sharing features that is not provided by a self-hosting environment [56]. Using resources in a cloud environment provides better performance in terms of large or small scale, and resource

allocation using dynamic or static techniques which is different from a self-hosting environment.

There are many different techniques that can be used for cloud systems. Each technique is different in implementation and can be used for different purposes. These techniques include simulation, service mocking, test job parallelisation, and environment virtualisation [6].

- **Simulation:** Simulation is used to deploy a cloud service in a dependant environment, which can reduce the cost of running a real cloud environment to do experiments or utilisation [24]. It also allows testing the service and getting results of any tests that can be compared to a real cloud deployment. That would be more essential to focus on problems in different scenario with less complexity to conduct better cloud services.

  Many simulation tools and systems have been implemented and developed for cloud computing such as CloudSim [12]. CloudSim is a toolkit used for modelling and simulating cloud computing environments and evaluation of resource provisioning algorithms. CloudSim is a useful tool for investigating and testing and deploying in a Cloud Environment before applying it in a real environment. It can support obtaining an overview of the service performance and overall activities that need to be optimised.

- **Service Mocking or Service Oriented:** Service mocking or service oriented [23] is used to divide a task, so it can be tested independently. Service mocking replaces a remote service with a simulated one which behaves as if the real one is called [56].

- **Test job parallelisation:** Test job parallelisation is a technique that splits tasks from the service and execute them individually in a large parallel system at one time. This would be more efficient in time and cost but not for running

the all service together.

- **Environment virtualisation:** helps to maintain and test a cloud environment. In this technique resources such as virtual machines are used for fast processing which is beneficial for testing service performance and to reduce testing cost. Also, this technique provides various testing tools for security and performance testing.

2.7 Security in Cloud

This section presents the main seven security risks in cloud services. Then it discusses some security issues in the Cloud especially in the service models. After that it shows some common security issues in the deployment models.

With all the benefits that providers can offer to customers using cloud services, security is still a major concern that affects a Cloud Environment [58]. According to Che et al. [14] there are seven security risks shown in Figure 2.1, which have been extended from [55], that need to be considered in a cloud computing service. These issues are:

1. Abusing the cloud service and privilege user access.
2. Insecure interfaces and APIs.
3. Malicious inside the service.
4. Sharing technology issues and service recovery.
5. Losing Data or leakage including data security and location.
6. Account or service hijacking.
7. Unknown risk profile

Security in cloud computing is a shared responsibility between providers and customers [31]. Che et al. [14] analysed the security concerns between cloud providers and customers. Customers' security concerns include data security, user access, and data leakage. Cloud providers' security concerns include availability, long term system security, how to defend against hacking, data centre security, secure transaction, resources security, and access control and management system.

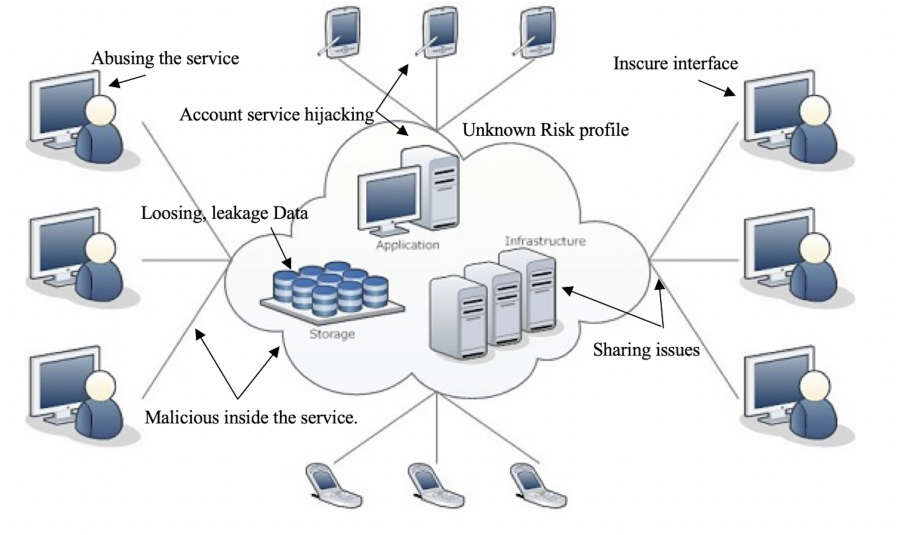

Figure 2.1: Security Risks in Cloud Service, adopted from [55]

2.7.1 Security Issues in Service Models

Table 2.1 shows the main security issues that exists for each service model [53]. From Table 2.1, SaaS has the most security issues because it is more complex than the other service models. PaaS and IaaS have less security issues compared with SaaS because they have better control over the security and they are not involved in the application level.

Table 2.1 shows the responsibility perspective for the the security issues for providers and customers. These issues are different in terms of responsibilities from the providers and customers. The table shows that most responsibility to ensure the security level of the service is on the providers. The providers' responsibili-

ties include data (security, locality, segregation, confidentiality), network security, authentication and authorisation, vulnerability in virtualisation, availability, and identity management. Using secure web applications to access the service is mostly the responsibility of the customers. The other security issues such as data access, data breaches and backup are shared responsibilities for providers and customers. These security issues affect differently each service model. These issues are:

- Data Security: Providers need to use good techniques to secure data access such as encryption and decryption.
- Network Security: To secure the data flow through the network from any security breach or leakage.
- Data Locality: To manage storing customers' data in a reliable location and to protect it from any risks.
- Data Integrity: To make sure that data is stored and then it correctly and accurately flows through the database over the service.
- Data Segregation: To secure the data flow, and data storage from any intrusions hacking the system on each level of the service.
- Data Access: To control data access for customers.
- Authorisation, Authentication: To manage accessing to the service or database.
- Data Confidentiality: To control and protect the data flow on each level of the service.
- Web Application Security: Customers need to ensure their web applications are secure to access the service.
- Data Breaches: Providers need to protect data and prevent any indirect access.
- Vulnerability in Virtualisation: Providers need to ensure that each tasks executed separately from each other to reduce security risks that could occur.
- Availability: Providers need to ensure that the service is delivered on demand.

- Backup: The backup information is important and if it has been hacked then any unauthorised accessed will cause a security issues for the customers. Providers need to ensure that backup is taken regularly and be secured and encrypted to make the service more reliable and fast recovery when it required.
- Identity Management: To control and check the identity of accessing to the service and resources by identifying all information that used to log in.

Table 2.1: Security Issues in the Service Models [53]

Security Issues	Service Models			Responsibility Perspective	
	IaaS	PaaS	SaaS	Providers	Customers
Data Security	√	√	√	√	
Network Security	√	√	√	√	
Data Locality	√		√	√	
Data Integrity		√	√	√	
Data Segregation			√	√	
Data Access	√	√	√	√	√
Authorisation, Authentication	√		√	√	
Data Confidentiality			√	√	
Web Application Security			√		√
Data Breaches	√		√		√
Vulnerability in Virtualisation	√	√	√	√	
Availability	√	√	√	√	
Backup			√	√	√
Identity Management	√		√	√	

Subashini and Kavitha [53] claim that the security issues in the service models such as data security and network security make a significant trade-off to each service model to obtain a reliable, trusted and secure services.

These service models offer different features to customers and providers to operate the service. SaaS offers many significant benefits to customers such as service efficiency improvement and reduced costs. In SaaS providers do all provisioning for hardware, data storage, power, virtual resources. As a result, customers have to pay for what they use, and there is no upfront cost for anything else. With all the benefits that are provided in SaaS, it has some issues such as lack of visibility of data stored and security.

In PaaS users can build their application on top of the platform, but this feature raises the security risks for all the services. Building applications on top of the platform increase security risks such as data security and network intrusion by unlocking the way to intruders trying any unauthorised actions [53]. For example, hackers can attack the applications code and run a very large amount of malicious programs to attack the service. In IaaS, customers can get services with less cost with basic security configuration and less load balance. Providers have to ensure that the service infrastructure is highly secure for, data storage, data security, data transmission, and network security.

2.7.2 Security Issues in Deployment Models

The common security issues that need to be addressed for the deployment models are Authentication, Authorisation, Availability, Access Control and Data Security [53]. These security issues are so important because each deployment model has a different security level. For example, the public cloud is less secure than the other cloud model, so it is more likely to be attacked by malicious hackers to get information that can used then to be hack at the private level. Providers are responsible for service security and they have to stop any unauthorised access or any malicious attacks of the service. Suspicious behaviour includes any malicious attacks and abuse of the service. Customer take responsibility for information security and data security such as integrity, confidentiality, authorisation and authentication.

There is a list of the top ten obstacles facing cloud computing summarised in

Table 2.2 [5]. Armbrust et al. [5] indicate that the consideration for each obstacle will vary from one stakeholder to another (customer and provider).

Table 2.2: The Top Ten Obstacles and Category [5]

No	Obstacles	Category	Stakeholder Perspective
1	Service Availability	Cloud Service Availability	Customer
2	Data Storage	Data, Data Boundaries	Provider
3	Data Confidentiality	Data, Data Boundaries	Customer
4	Data Transfer	Data, Data Boundaries	Customer
5	Performance Unpredictability	Performance, Scalability	Customer
6	Scalable Storage	Performance, Scalability	Customer
7	Error of Large Scale	Performance, Scalability	Provider
8	Quick Scaling	Performance, Scalability	Customer
9	Service Level Agreement (SLA)	Service Policies	Provider, Customer
10	Software Licence	Service Policies	Provider, Customer

The first obstacle is Service Availability which has multiple sides. One side is cloud providers offer multiple sites to improve availability, however, customers may choose to use multiple providers to increase availability. As a result, some parts of the services may become unavailable for some customers for any time.

There are many reasons that can cause service unavailability such as crashed applications, high loads in the service, and service hijack [40]. Then the customers will think that the service was down and it is not available to be used. However, services with multiple clouds or multiple sites give more opportunities for an attacker to cause a security threat. An attacker can use a public service to get to unauthorised access to resources or by doing many malicious activities that affect the service. One way to defend this issue is to use quick scale-up method and security monitoring [5]. Scaling method in the cloud is used to control cloud resources, which include two

type of scaling, horizontal and vertical [36]. The vertical or scale up is used to increase the virtual resources for restoring and improving performance also known as scaling outward. Service Availability is an issue that can be addressed using this method if any virtual resource becomes unavailable. The horizontal method is to scale upward by running the service in one physical resource. Providing the service from one physical resource or one site is an issue of the service availability.

The second, third and fourth obstacles are about data boundaries between platforms and Data Storage, Data Confidentiality, and Data Transfer. There are many security implications that should be considered including loosing data, data leakage, transferring data, and data security. The fifth, sixth, seventh, and eighth obstacles are more technical being related to performance, Scalable Storage, removing errors in a large scale distributing system, and how services can be established with quick scaling getting an overview of service costs. Quick scaling could cause unavailability of the service if there is a very high load tasks which needs to be considered as a security implication of this method. The ninth and tenth obstacles are about service policies for Service Level Agreement (SLA) and Software Licence. The concern here is about the eligibility or the authorisation of using the software and to ensure there is no misuse of the licence [5].

2.8 Related Work

This section discusses recent related approaches in the area of cloud security such as Data storage approaches that are related to Data as a Service (DaaS) data storage moving from a single cloud to a multi-cloud, and security models. It also provides some approaches in resource management that used static and dynamic methods focusing on performance.

A review of recent cloud models has been performed to get an overview of the models categories shown in Table 2.3 Models have been classified to categories related to the main focus of the approaches including Data as a service (DaaS), In-

frastructure as a Service (IaaS) and cloud storage. The DaaS models focus on all data security and different from cloud storage which is concerned about data centre security. The IaaS models focus on the infrastructure security.

In addition, Table 2.3 shows that there are some issues have less attention than others such as Authentication, Accountability, Intrusion, and Reliability. The most focused areas are Integrity, Availability, and Security. Most approaches are related to cloud storage and DaaS which make IaaS need more work especially in security.

The DepSky System [7] addresses the availability and the confidentiality of data in their storage system by using multi-cloud providers, combining Byzantine quorum system protocols, cryptographic secret sharing and erasure codes. Whereas NetDB2-MS [2] is a Model to ensure privacy level in DaaS based on data distributed to different service providers and to employ Shamirs secret algorithm [43].

The BlueSky System [57] has extended the DepSky System to be more reliable and deal with large storage volume from a cloud provider and to avoid a dedicated hardware server. Similarly, the SafeStore system [25] is more focused on availability not on performance and cost which is quite different than the other systems.

Other approaches like HAIL [8], ICStore [11], SPORC [16], Depot [29], Data storage Models [34] [49] have focused on cloud storage including some data security aspects such as security and data integrity and data confidentiality. They also have similar limitations such as data intrusion and availability.

There are some models at the deployment level which deal with the security risks but with limitations in confidentiality and integrity such as Separation Model, Migration Model, Availability Model, Tunnel Model, and Cryptography Model [65].

Data privacy is still a big concern in other Models like Jerico Formus Cloud Cube Model [13], Hexagon model [13], Multi-Tenancy Cloud Model [14], Cloud Risk Ac-

cumulation Model [9], and the Mapping Model [13] [14]. The logging approach [61] ensures that the log files can mitigate the risks to benefit both sides of accountability, security, performance, and scalability.

Other work in scheduling such as Tripathy and Patra [54] brought into consideration tasks priorities then assign them to be executed over the allocated resources. If there more than one task for each resource, it will be scheduled with different methods depending on what is better for each resource. Then it will use parallel running for all tasks. This work assigned dependant tasks first to run first then non dependant one that to minimise the deadlock situation.

Table 2.4 shows some approaches that related to resource management used static and dynamic methods and focusing on performance.

Approaches by Li et al. [27] and Yazir et al. [64] relate to resource scheduling using static and dynamic mechanism but they did not include any security factors to avoid any security risks. Static scheduling mechanism such as the approach introduced by Jiayin et al. [26] offers a static scheduling solution to improve service performance over virtual machines. The tasks are executed on certain cloud resources based on the static resource allocation. It aims to regulate many resources utilisation of Service Level Objectives (SLOs) of applications. Also, Qiu et. al. [26] propose an algorithm that adjust resource allocation based on updating the actual task executions which helps to recalculate the finishing time that assigned to the cloud.

Walsh et al. [59] proposed a utility function as a solution by dividing the architecture into two-layers (local and global). The local layer is responsible for calculating resource allocation dynamically. Whereas, the global layer computes the near optimal configuration of allocating resources based on results provided by the local layer, and to fix the load balancing with the server cluster which also helps applications scalability.

Table 2.3: Review of Cloud Models

Ref	Category			Main Focus												
	DaaS	IaaS	Cloud storage	Availability	Confidentiality	Reliability	Intrusion	Integrity	Fault tolerance	Recovery fail	Cost	Scalability	Performance	Accountability	Latency	Security	Authentication
DepSky [7]	√			√	√												
Bluesky [57]	√																
SafeStore [25]	√				√												
NetDB2-MS [2]	√			√			√	√									
NCCloud [21]			√						√	√	√						
HAIL [8]			√	√				√									
ICStore [11]			√				√										
SPORC [16]			√	√													
Depot [29]			√	√						√					√		
Logging Solutions [61]		√										√	√	√			
Venus [49]			√					√									
TCCP [41]		√		√				√									
CCM [13]																√	
Hexagon Model [13]																√	
MTCM [13]																√	
CSA [13]																√	
Mapping model [13]																√	
Separation Model [65]									√								
Migration Model [65]								√								√	
Availability Model [65]				√												√	
Tunnel Model [65]								√								√	
Cryptography Model [65]					√											√	
NDSM [3]	√		√													√	
Cloud Trust Model [42]	√															√	
DSM [62]	√				√											√	
DSSM [34]	√		√													√	√
SC [54]													√				

Table 2.4: Approaches for Resource Scheduling

Ref	Resource Management	Static	Dynamic	Performance
Adaptive management of virtualised resources [27]	√	√		√
Adaptive resource allocation [26]	√	√		√
Dynamic resource allocation [64]	√		√	√
Resource allocation for multi-tier [19]	√		√	√
Resource Allocation Policies [52]	√		√	√

Other approaches that use dynamic mechanism such as Yazir et al. [64] and Slegers et al. [52] include a comparison of static and four heuristic dynamic policies. They showed some differences and presented benefits and weaknesses of using each type in terms of using and managing cloud resources. A price model was introduced by Sharma et al. [44] for dynamic resource management and low cost of cloud service but they did not include the security factor and indicate saving cost on physical resources and maintenance cost as limitations of their model.

2.8.1 Research Focus

In this research, an investigation has been performed to understand what scheduling technique would be best suited to serve the purpose of this study. It has been found that the most appropriate scheduling technique would be use the scheduling priority method, as it will help to execute tasks over the allocated resources as it will be combined with the security as a main feature.

Cloud security is necessary in order to classify tasks and divide the resources to match this classification. Which is very important to keep the security as it required all the time during any service requests.

As a part of any service, QoS factors will be considered as it discussed in Section 2.3. It needs to address the affecting factors to have a better quality for the service provided.

The service via cloud environment needs all the parts discussed (Scheduling, QoS, and Security), Figure 2.2. shows where this thesis focused on and it combined these features in the model which depends on the literature review that has been done.

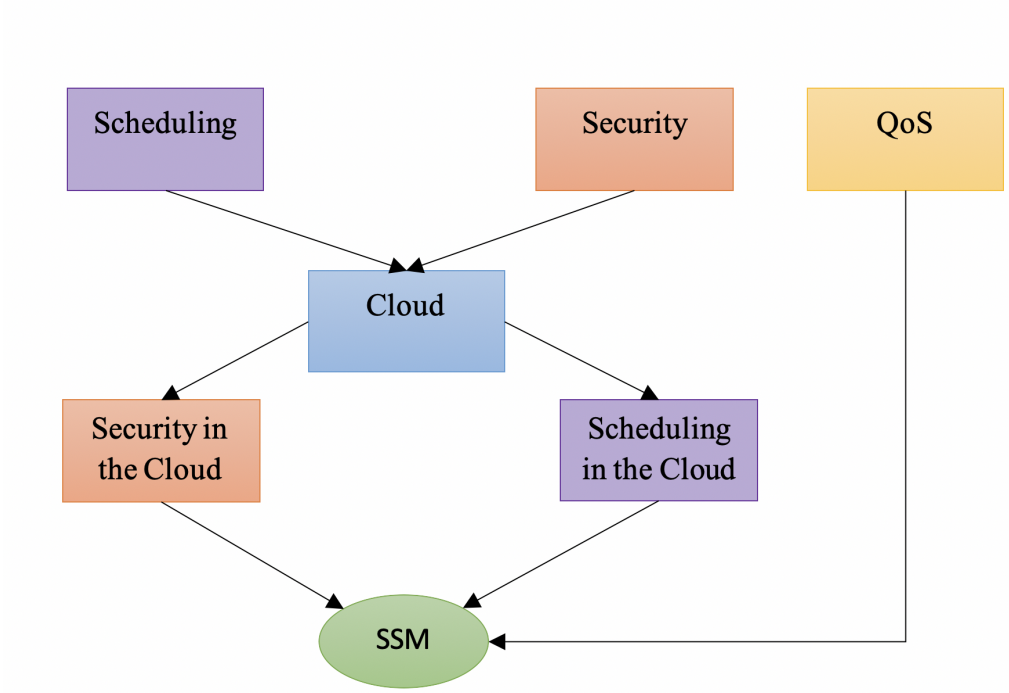

Figure 2.2: Features of the SSM

2.9 Assumptions for New Model

This section discussed the initial proposed new model Scheduling Security Model (SSM) and discusses a set of assumptions as follows:

- Security level is the main driver to group tasks.
- Task importance to control ordering tasks within resources.
- Execution time and elapsed time are used for Re-calculation.
- Tasks are not paused once started.

- Resources are not duplicated.
- Task size is ignored.
- Fast-Track list and Execution list in each resource for control executing tasks.
- A resource is available for each security level.
- Infinite number of tasks are allowed.
- Resource cost is per hour.

2.10 Summary

This chapter discussed and gave an overview of all parts that involved were to develop a new service model. The first part is scheduling and it has been discussed as it is an important process to make an efficient use of the cloud resources. Then it discussed the QoS and how it is important to have a better service quality considering the affecting factors. After that, as the security is a shared component for all parts it has been shown that how the materials need to be secured even physical or digitally secured and it can be classified to different levels. Next, the cloud environment has been discussed with defining the cloud and its characteristics and architecture. Then it discussed its service models and deployment models. Then in the following section, a discussion of the relation of scheduling and the cloud and what tools are used for cloud scheduling have presented. Then next section presented the relation with the security and the cloud and what security issues that need to be considered in the cloud service. Then it showed the security issues in the deployment models explaining the most common issues that need to be addressed. After a literature review has been done to investigate the related work to this thesis finding the research direction and the assumption made to develop the new model.

Chapter 3

Scheduling Security Model (SSM)

3.1 Introduction

In this Chapter, Security and its level will be explained in the context of the Scheduling Security Model (SSM). Also, this chapter defines the SSM and its components. Then it explains how the SSM calculates the costs for a service requested. Next, it defines the SSM and the Scheduling Function steps. It presents some examples of costs to show how the SSM works. Then it points to some limitations of the SSM.

3.1.1 Security Definition

Before identifying the model components there is a need to define what security level means in this context. According to Watson [60] the overall security level can be considered to be applied for executing tasks from trusted public resources to highly trusted private resources. So, the SSM defined the security level as shown in Figure 3.1. Then the SSM security levels considered as the following:

1. Level 1: is trusted public resource with basic security that can execute public tasks.
2. Level 2: is more trusted public resource with more security setting that can execute public tasks.
3. Level 3: is highly trusted public resource with security that can execute public tasks.

4. Level 4: is trusted private resource with security level that can execute private tasks.

5. Level 5: is highly trusted private resource with more security setting that can execute private tasks.

As a result, each task will be given a security level from 1 to 5. Then the resources will be created depending on how many security levels are identified by the tasks.

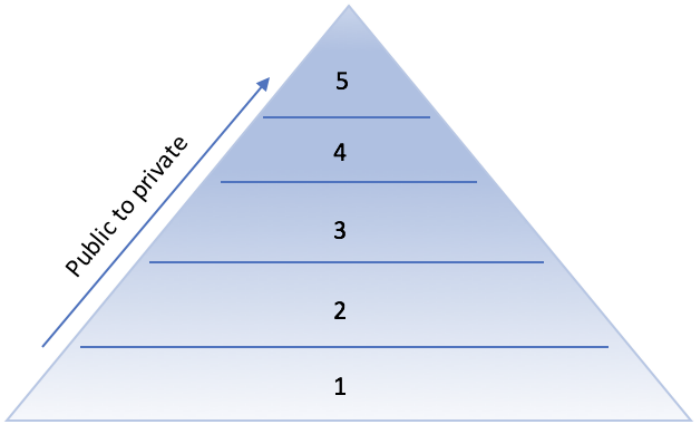

Figure 3.1: Security Levels from Public to Private Resources, adopted from [60]

For example, a customer requests a service that includes two tasks. One task is to analyse general data with no security requirement. The other task is to save private data that requires higher security level. So, the service will require two resources one is trusted public resource with basic security feature to execute the first task, and another trusted private resource with security feature such as more secure fire-wall to execute the second task.

3.2 The SSM

This section presents the Scheduling Security Model (SSM) components and how the costs for executing a service can be calculated. Also, it defines the SSM model and the scheduling function steps.

3.2.1 SSM Components

The suffixes used in the definitions are specified as follows:

i : Tasks

k : Resources

j : Security Level

l : Task Importance

The values in the definitions are:

- T: Set of tasks, Where M is the number of tasks. Each task has a security level and importance level t_i (h_j, p_l)

  t_i: tasks i

- tm: Time cost

  tm_i: time cost for task i

- q: Quality of Service for the service.

- b: Customer budget for the service.

- h: Security level for a task.

  h_j: Security level $\in \{1, 2, 3, 4, 5\}$

- hw: Security weight for the security level (for each task).

  hw_j: Security weight for security level j.

- Each task (t_i) has a security level h_j

  (and then a security weight hw_j)

- R: Set of resources (R_k), $k \in \{1, 2, 3, 4, 5\}$

  (can use up to to 5 resources $\{R_1, R_2, R_3, R_4, R_5\}$)

then, used resources will be numbered from 1 to N

N: is the number of resources used (N determined later)

- R_k: is a set of tasks, where the $hw_j = Rw_k$

- Rw: Resource security weight (for each resource)

  Rw_k: Security weight for resource k (R_k)

  $Rw_k = hw_j$

- p: Tasks Importance, p_l: is the importance $\{1, 2, 3\}$

  task t_i has importance p_l

- M: Number of tasks.

- N: Number of resources.

- e: Maximum time.

The SSM consists of the following components:

- Tasks Set T: The customer will specify all the tasks that need to be executed, where M is the total number of tasks.

$$T = \{t_1, t_2, t_3, ..., t_M\} \tag{3.2.1}$$

- Time cost: The elapsed time tm_i will be associated with each task to be executed on the allocated resource. The calculated total time cost is:

$$tm = tm_1 + tm_2 + tm_3 + ... + tm_M \tag{3.2.2}$$

- Quality of Service (QoS): SSM supports quality of service which allows the scheduler to adjust the service scheduling tasks to achieve the required quality of service **q** within the customer budget. Depending on the customer QoS target, **q** will be selected from a set of options associated with different costs. QoS can be a value in the range from 0, low quality, and incremented by 0.1 to reach 1.0 which is the highest quality level of service:

$$q \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\} \tag{3.2.3}$$

- Customer Budget: A customer will submit the overall budget, **b** for the service request.

- Security Level: The security level in this context means that the required level of the security that can be applied to each task. The customer will submit the security level required for each task. The security level, h_j, will be a number between 1 (low security) to 5 (high security), each level is the security level required for the task to run over a resource see Figure 3.1. SSM will then map this security level to security weight, hw_j for task t_i.

$$h_j \in \{1, 2, 3, 4, 5\} \tag{3.2.4}$$

  Then the associated weight for each level of security will be as follows by indexing the values:
  For tasks:

$$hw_j = [0.0, 0.25, 0.5, 0.75, 1.0] \tag{3.2.5}$$

  All tasks will be allocated and executed on the same Resource with the same security weight. Then the associated security weight for Resources will be as follows:

$$Rw_k = [0.0, 0.25, 0.5, 0.75, 1.0] \tag{3.2.6}$$

  The SSM will map the associated security weight from the equivalent security level. So that, for example if h_1= 2 then hw_1 = 0.25. This will be used to calculate the estimated cost to establish the service and to create the required resources with the security level for each resource (public with low security or private with higher security weight).

- Tasks Importance: There are many techniques that can be used to prioritise tasks for execution, and in the SSM there will be three levels of task importance p_i which will help ordering the tasks queue. The main reason for specifying

the task importance to three levels is to order the task in a Resource R. These importance levels will be considered after classifying tasks based on the security levels given by the customer and to be ordered as first come first served. The customer will submit the importance level required for each task:

$$p_l \in \{1, 2, 3\} \tag{3.2.7}$$

The reason for making the Task Importance in three levels is that the scheduling in the SSM is serving the security as a category, then there is a need to give each task within the same category an order to be executed. So, the order will be as identified with this three levels but if there are some tasks with the same Task Importance level then the scheduling will be for first come first served. This means that tasks with the highest value of $p_i = 3$ is the most important for the customer.

For example, a customer has submitted tasks t_1 with security level $h_1 = 1$ and importance $p_1 = 2$, and t_2 with security level $h_2 = 1$ and importance $p_2 = 3$. That means t_2 will be executed before t_1 because t_2 has the highest importance. Also, the Tasks will be allocated and to be executed on one Resource R_1 as they have the same Security Level see Table 3.1.

Table 3.1: Example of Ordering Tasks

Security Level(Weight)/Importance	1	2	3	R_k
1 (0.00)		t_1	t_2	R_1
2 (0.25)				
3 (0.50)				
4 (0.75)				
5 (1.00)				

After submitting all Tasks details, the SSM will ask the customer if there any task decencies, if there are dependencies between tasks the customer needs to enter the dependant task for each task submitted. That will help executing Tasks in the scheduling process using the Fast-Track technique.

Another simple example with tasks dependencies will be discussed in details later in the Scheduling Process. A customer has submitted tasks t_1 with security level $h_1 = 1$ and importance $p_1 = 2$, t_2 with security level $h_2 = 1$ and importance $p_2 = 3$, and t_3 with security level $h_3 = 3$ and importance $p_3 = 2$. Then the Task dependency t_3 depends on t_1. The SSM will analyse this input and allocates Tasks to two Resources R_1 and R_2. Tasks t_1 and t_2 will be allocated to R_1 and t_3 to R_2. But with the dependencies required t_1 will assigned to the Fast-Track list. That means t_1 will be executed first then t_3 depends on t_1 and it has the highest Security Level that lets the SSM to list first see Table 3.2.

Table 3.2: Example of Ordering Tasks with Dependencies

Security Level(Weight)/Importance	1	2	3	R_k
1 (0.00)		t_1^{FT}	t_2	R_1
2 (0.25)				
3 (0.50)		t_3		R_2
4 (0.75)				
5 (1.00)				

Table 3.3 shows a summary of the components that specify the customer requirement for requesting a service. Then SSM will analyse the requirements for the calculating step.

3.2.2 Calculated Components

- Table 3.3 shows a summary of the customer inputs for the SSM.

- Resources Required for a set of tasks: First the model will categorise the tasks into categories depending on their security level. Then it creates, N, the number of resources required, which will be equal to the number of the task categorise and each resource will take the security level of that category and it will mapped to the resources security weight Rw_k.
  For example, if the customer submitted tasks (t_1: with h_1=2, p_1=2 and t_2: h_3

Table 3.3: Summary of the customer inputs

Component	Values	Range
Budget	b	$b > 0$
Maximum Time	e	$0 < e \leq 60$
QoS	q	0.0, 0.1, 0.2,..,1.0
Tasks	t_1, t_2, t_3, ... t_i, indexed by $_i$, $i \in \{1 - M\}$	M Tasks
Task Security Level	h_1,h_2, h_3,.... h_j, indexed by $_j$	$j \in \{1, 2, 3, 4, 5\}$
Task Importance	p_1, p_2,p_3,... p_l, indexed by $_l$	$l \in \{ 1, 2, 3 \}$

=0.5,p_2=1), q=0.0, and e=60 minutes, the model will categorise these tasks into two categories with two different security levels. After that the model creates two resources N=2, R_1 takes security weight Rw_1 0.25, and R_2 with security weight Rw_2 0.5 then assign each category to the similar resources with same security weight as shown in Table 3.4.

Table 3.4: Example of a Service Required

Security Level(Weight)/Importance	1	2	3	R_k	RC_k
1 (0.00)					
2 (0.25)		t_1		R_1	20
3 (0.50)	t_2			R_2	20
4 (0.75)					
5 (1.00)					

- Actual Cost: The Actual Cost calculated for the service calculated depends on the customer requirement.

  - Resource Cost = Cost of resources for RT_k hours.

$$Resource\ Cost(RC) = \sum_{k=1}^{N} (RC_k * RT_k) \tag{3.2.8}$$

Where RC_k is Resource Cost for Resource k per hour and RT_k is the actual time used by Resource k in hours.

- Quality of Service Cost = Resource Cost * Quality of Service required

$$QoS\ \ Cost = RC * q \tag{3.2.9}$$

- Security Cost for each Resource

$$SC_k\ \ = RC_k * Rw_k * RT_k \tag{3.2.10}$$

Where Rw_k is Security weight for Resource k.

- Security Cost for all Resources

$$SC = \sum_{k=1}^{N} SC_k = \sum_{k=1}^{N} (RC_k * Rw_k * RT_k) \tag{3.2.11}$$

- *Actual Cost (AC)* $= RC + SC + QoS\ Cost$

Therefore

$$AC = \sum_{k=1}^{N} (RC_k * RT_k) + q * \sum_{k=1}^{N} (RC_k * RT_k) + \sum_{k=1}^{N} (RC_k * Rw_k * RT_k) \tag{3.2.12}$$

Then

$$AC = \sum_{k=1}^{N} ((RC_k * RT_k) * (1 + q + Rw_k)) \tag{3.2.13}$$

3.2.3 The Model

The main idea of the SSM is to categorise the submitted tasks on their security level. Then it calculates the estimated cost of the service. Next, it asks for customer confirmation to establish the service. After getting the customer confirmation the service will be established, then it tries to recalculate the cost for possible cost reduction. The model has the following stages, shown in Figure 3.2:

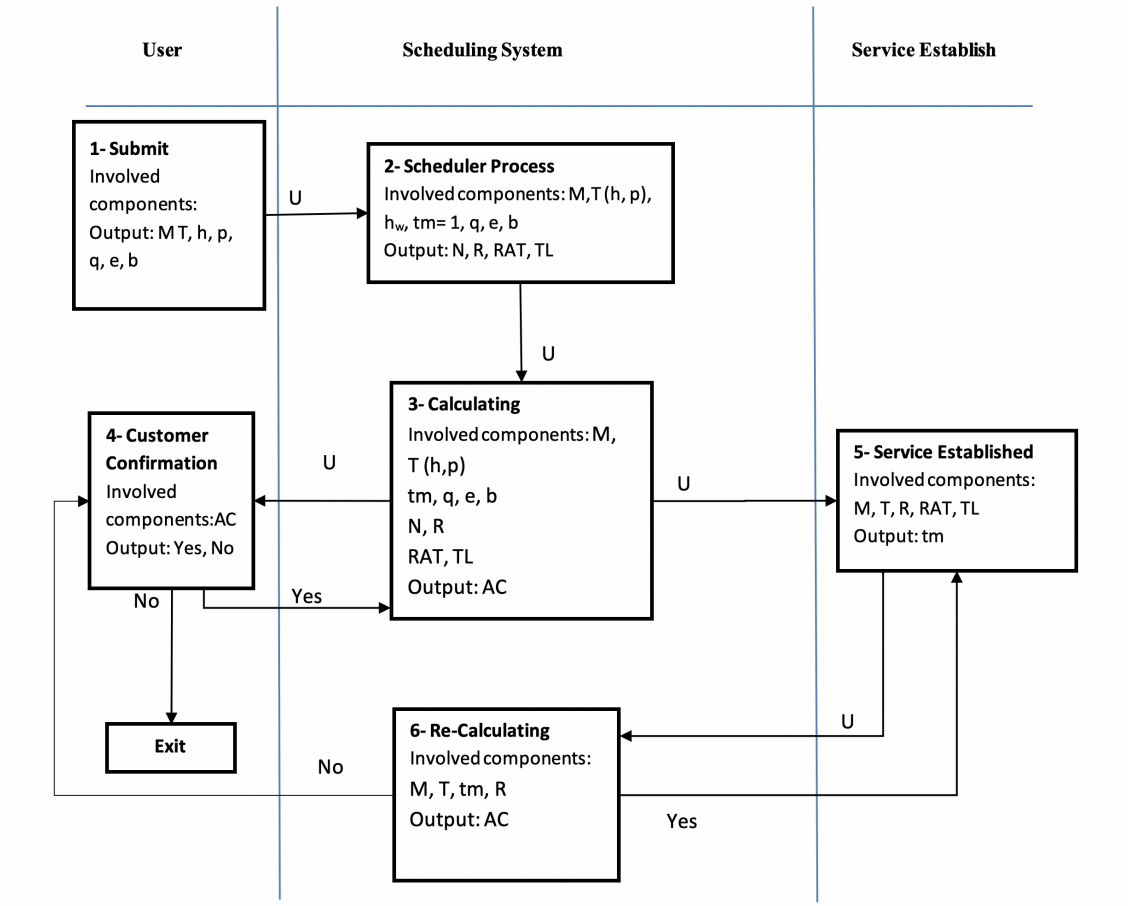

Figure 3.2: Scheduling Security Model (SSM)

1. **Submit:**

   A customer requests a service and provides information on all attributes of the requirement needs. They submit attributes: Over all Time of the service, Number of tasks, tasks importance, overall budget, security level for each task, and QoS required.

   - Budget b.
   - The Maximum Time Required e: Minimum 60 minutes (one hour).
   - The QoS q.
   - The Number of Tasks M.
   - For each Task t_i the information: Security Level: $h_j \in \{1, 2, 3, 4, 5\}$, and Tasks Importance: $p_l \in \{1, 2, 3\}$.

2. **Scheduler Process:**

   The SSM analyses the customer requirements given in the Table 3.2. Then it categorises the tasks depending on the security level of each task. The number of categories is equal to the number of the security levels required. For each category, tasks will be ordered depending on the tasks importance. The task with higher task importance will be run first down to the task with lowest task importance. If there are more than one task with the same task importance the SSM will put them in the task number order.

3. **Calculations:**

   In this stage, the SSM scheduler calculates the Actual Cost and compares it to the customer budget. This will help to identify the service attributes and resources (VMs). The service attributes are sent to the customer for confirmation.

4. **Customer Confirmation:**

   The customer will receive an overview of all costs to take the final decision to go ahead before establishing the service.

5. **Establishing the Service:**

   After Customer Confirmation, the SSM will establish the service and execute the tasks on the allocated resources.

6. **Re-Calculating:**

   At run time, for each Resource, the SSM optimises time cost and calculate the Resource actual time usage and not the elapsed time. The difference in the cost will be discussed in Examples of Costs section. In general, the elapsed time considers the running time of each resource without calculating any waiting time. So, if there are any dependencies over different Resources it will cause a delay to the service. But by considering the actual running time that will calculate that existed from waiting the dependent tasks to finish.

   Considering the Resource actual time usage after the tasks execution process there may a chance to add some credit to the customer budget b to reduce

the cost. So, the SSM will calculate the sum of actual times for each task in a Resource, then reflect the actual time on the resource cost.

The SSM will check the cost of actual resource time against the resource cost. If the actual time is less than the required time there will be credit added to the customer, and if the actual time is more than the required time then the SSM will stop the service until the customer adds more credit to continue with the service.

3.2.4 SSM Scheduling Function

This section explains the SSM Scheduling Function and what components are involved in each step. Then it explains what change will occur, and what it is look like after the current step. The SSM Scheduling Function includes the following steps:

1. **Categorise:**

   From the attributes given by the customer this step uses the security level for each task to categorise tasks into categories. Each category represents a security level required for running tasks in this category. So this step does the following:

   - **Involved Components:**

     All Tasks: $T = \{t_1, t_2, t_3, ..., t_M\}$, Security Level: $h_j \in \{1, 2, 3, 4, 5\}$, Tasks Importance: $p_l \in \{1, 2, 3\}$. Also, the components budget: b, and QoS: q will be considered for the Calculation step to check that there enough credit before establishing the service. The time: e will be checked as the minimum time of the service should be 60 minutes see Table 3.5.

   - **The Change:**

     First the SSM takes all the tasks and categorises them depending on the security level, and allocates them in categories which have the same security level.

   - **Components after the Process:**

     Tasks are categorised to different Category depending on Tasks Security Level. For example, if tasks are categorised into two categories that

Table 3.5: Categorise Involved Components

Component	Values	Range
Budget	b	$Estimated\,Cost \leq b$
Time	e	$0 < e \leq 60$
QoS	q	0.0, 0.1, 0.2,..,1.0
Tasks	t_1, t_2, t_3, ..t_i.. t_M, indexed by $_i$, $i \in \{1 - M\}$	M Tasks
Task Security Level	h_1,h_2, h_3,.... h_j	h_j {1, 2, 3, 4, 5 }
Task Importance	p_1, p_2,p_3,... p_l	{ 1, 2, 3 }

means there were two security levels for all tasks. Then the number of categories N will be used to identified the number of Resources required to establish the service see Table 3.5. The number of Categories is equal to the number of Security Levels required. For example, if there are three Security Levels required that means there will be three Categories then after customer confirmation there will be three Resources See Table 3.6.

Table 3.6: Categorise Components after Scheduler Process

Component	Values	Range
Budget	b	$Estimated\,Cost \leq b$
Time	e	$0 < e \leq 60$
QoS	q	$q \in \{0.0, 0.1, 0.2,..,1.0\}$
Tasks	t_1, t_2, t_3, ..t_i.. t_M, indexed by $_i$, $i \in \{1 - M\}$	M Tasks
Task Security Level	h_1,h_2, h_3,.... h_j	$j \in \{1, 2, 3, 4, 5\}$
Task Importance	p_1, p_2,p_3,... p_l	$l \in \{1, 2, 3\}$
Categories	1,2,3,...N	{1, 2, 3, 4, 5}
Categories Security Level	Security Level	{1, 2, 3, 4, 5}

2. **Compare and Order:**

In each category Tasks will be ordered by the Task Importance level. So, the

SSM compares Tasks in each category to put them in right order preparing them to be executed in establishing the service step. Task with higher Task Importance will be ordered first then the task with lower Task Importance. Then Resources will be identified to be the same number of the categories with equivalent security level. So this step does the following:

- **Involved Components:**

  All Tasks: $T = \{t_1, t_2, t_3, ..., t_M\}$ each task has a Security Level: $h_j \in \{1, 2, 3, 4, 5\}$, and Tasks Importance: $p_l \in \{1, 2, 3\}$. Then budget b after analysing the customer inputs will be compared with the estimated cost.

- **The Change:**

  For each Category, tasks will be compared and ordered depending on the Task Importance p_l. Task with higher Task Importance will come first then the Task with lower Task Importance.

- **Components after the Process:**

  All Categories created and ordered and will be ready to send to the customer to confirm the service. The Number of the category will be used to identified how many Resources needed for the service requested. Also, the SSM will check the estimated cost if it less than customer budget b, if it less the customer will be asked to add more credit if the service confirmed.

3. **Assign:**

   After ordering all tasks in each Category and receiving customer confirmation to establish the service, all category will be assigned to a Resource R_k that match the Security Level of Tasks in the Category. Also, if there any dependencies all dependent Tasks will be allocated in the Fast-Track list in each Resource.

   - **Involved Components:**

     All Categories and number of Categories N, all Tasks: $T = \{t_1, t_2, t_3, ..., t_M\}$, Security Level: $h_j \in \{1, 2, 3, 4, 5\}$, Tasks Importance: $p_l \in \{1, 2, 3\}$.

Table 3.7: Assign Components after Process

Component	Values	Range
Tasks	t_1, t_2, t_3, ..t_i.. t_M, indexed by $_i$, $i \in \{1 - M\}$	M Tasks
Task Security Level	h_1, h_2, h_3,.... h_j	$j \in \{1, 2, 3, 4, 5\}$
Task Importance	p_1, p_2, p_3,... p_l	$l \in \{ 1, 2, 3 \}$
Resources	1,2,3,...N	{ 1, 2, 3, 4, 5 }
Resource Security Weight	Rw_1, Rw_2, Rw_3,.. Rw_N	{0.00, 0.25, 0.50, 0.75, 1.00}

- **The Change:**

  All Security Levels will be mapped to correspondent Security Weight for each Task then to identify the Security Level for each Resource. Then all Resources R_k created, and each category including Tasks will be assigned to a Resource R_k with same Security Level.

- **Components after the Process:**

  All Categories including all Tasks assigned to the correspondent Resource R_k and to be ready for the next step.

4. **Execute:**

   All Tasks including Tasks in Fast-Track list ready to be Executed over the allocated Resources R_k.

   - **Involved Components:**

     All Tasks and Resources are ready to establish the service.

   - **The Change:**

     All Tasks will be in the executing process. The SSM will run the Fast-Track list for each Resource if there any Task dependencies then Tasks in the normal order depending on Task Importance.

   - **Components after the Process:**

     All Tasks executed and all Resources will be deleted after finishing the service.

3.2.5 SSM Algorithm

The SSM will use a basic algorithm that divide into multiple steps starting from getting the customer inputs including budget, QoS, maximum time, and all Tasks information. This information will be used to calculate the total cost of the service. Also, the task information such as Security Weight will be used to categorise Tasks into Categories. The Security is the feature that the SSM will use to schedule the Tasks over Resources then it uses Task Importance to order Tasks within the Resource. Moreover, if there are any Task dependencies the SSM will use Fast-Track technique. In the Fast-Track technique there will be an additional list in each Resource, this list will include all dependent Tasks ordered depending on the related Task with the higher Security Level in other Resource more details on how it works explained in Example 3.2.

3.2.6 Fast-Track

The Fast-Track technique is to give any task a high priority for execution depending on the security level required and it will be applied at the Execution step if there are dependencies between tasks. For example, a customer submitted three tasks requiring the same security level t_1, t_2, t_3 and importance level 1,2,3 respectively. The normal executing order is t_3, t_2, t_1. If t_3 depends on t_1, then applying the Fast-Track technique and the executing order will be t_2, t_1, t_3.

3.3 EXAMPLE OF COSTS

This section presents simple examples of calculating costs (Actual Cost, Time). Moreover, it shows the different between the cost with the actual time usage and the elapsed time. Also, how the SSM schedules and orders Tasks within the allocated Resources. Then it explains how all Tasks will be executed.

3.3.1 Example 3.1:

1. **Submit:**

   Customer has requested and submitted information as shown in Table 3.5, $q = 0.0$, $b =$ £100, and $e = 60$ minutes.

Table 3.8: Example 3.1 Customer Requirement for a Service

Security Level(Weight)/Importance	1	2	3	R_k
1 (0.00)				
2 (0.25)		t_1		R_1
3 (0.50)	t_2			R_2
4 (0.75)				
5 (1.00)				

2. **Scheduling Process:**

   The SSM will analyse the request from the information submitted by the customer. After the Categorise step and the Customer Confirmation, Tasks are assigned to Resources R_1 and R_2 as shown in Table 3.8. Then it identified the service requirements as follows:

   - For this service request, the submitted tasks have two security levels. The SSM will categories tasks into two categories. Each category represents different security level. Next, the SSM identified the number of Resources depending on the number of categories. So, the number of Resources is $N = 2$ Resources. The Resource Costs are $RC_1 = 20$, and $RC_2 = 20$.
   - For the tasks submitted the Resources associated as follows:
     - Task t_1 will be allocated to Resource R_1 with security weight 0.25.
     - Task t_2 will be allocated to Resource R_2 with security weight 0.5.

   Table 3.9. shows the result of the SSM analysing customer requirements for this service request, and the level of security for all tasks and Resources, and tasks allocated to Resources.

Table 3.9: Example 3.1 SSM Analysing Customer Requirement

Security Level(Weight)/Importance	1	2	3	R	RC_k
1 (0.00)					
2 (0.25)		t_1		R_1	20
3 (0.50)	t_2			R_2	20
4 (0.75)					
5 (1.00)					

3. **Calculation:**

   The SSM will calculate the initial cost for each resource and send it to the customer as follows:

   - Resource Cost:

     RC_k= Cost of R_k per hour

     $RT_k = 1$ for each R_k (i.e 60 minutes)

     Therefore:

     $RC_1 = 20$, RC_2= 20

     $RC = RC_1 + RC_2$

     $RC = 20 + 20 = 40$

   - QoS Cost:

     QoS Cost = $RC * q$

     QoS Cost = 40 * 0.0 = 0.0

   - Security Cost:

     Security Cost for Resources (SC):

     SC_1 for R_1 = (20 * 0.25) = 5

     SC_2 for R_2 = (20 * 0.50) = 10

     $SC = SC_1 + SC_2 = 5 + 10 = 15$

   - Actual Cost:

     Actual Cost AC = RC + q + SC

     Actual Cost AC = £40 + £0.0 + £15 = £55

   - Check if less than or equal to b:

$55 \leq 100 = \text{True}$

4. **Customer Confirmation:**

   The SSM will send all information about the service to the customer for confirmation as follows:

   Task t_1 requires one Resource (R_1) and task t_2 requires one Resource (R_2) with Time $e = 60$ minutes, and the Cost = £55 is less than or equal to $b =$ £100.

5. **Service Established over Resources R:**

   After receiving the confirmation from the customer, the SSM will start to execute the tasks over the allocated resources. Because there is no dependency R_1 and R_2 can be established at the same time.

6. **Re-calculating:** In this example the time has been calculated after the establishing the service for the tasks:

   $tm_1 = 1.67 \quad minutes$, $tm_2 = 3.33 \quad minutes$

   So, the SSM will Re-calculating the Actual Cost for t_1 over R_1 and t_2 over R_2, and then the Actual Cost will be as the following:

   - Resource Cost (RC):

     $RC_1 = 1.67/60 * 20 = 0.56$

     $RC_2 = 3.33/60 * 20 = 1.11$

     $RC = 0.56 + 1.11 = 1.67$

   - QoS Cost q:

     QoS Cost = 1.67 * 0.0 = 0

   - Security Cost (SC):

     $SC_1 = 0.56 * 0.25 = 0.14$

     $SC_2 = 1.11 * 0.50 = 0.56$

     $SC = 0.14 + 0.56 = 0.7$

   - Then AC = 1.67 + 0.0 + 0.7 = £2.37

Therefore, after the Re-calculating step the AC is less than the initial AC at the calculation step because less time is used.

This example shows that categorising Tasks depending on the Security Level required makes different cost to the Resources. But the cost can be reduced if there is no dependencies between tasks. Also, the Re-calculating step consider the elapsed time to calculate the Actual Cost and can be reflected on the customer budget and can add more credit to it.

3.3.2 Example 3.2:

1. **Submit:**

   Customer has requested and submitted information as shown in Table 3.10, q=0.0 and e = 60 minutes. Tasks dependencies as follows:

   t_6 depends on t_3, t_5 depends on t_1.

Table 3.10: Example 3.2 Customer Requirement for a Service

Security Level(Weight)/Importance	1	2	3	R
1 (0.00)				
2 (0.25)	t_1	t_2		
3 (0.50)				
4 (0.75)	t_3	t_4		
5 (1.00)	t_5	t_6		

2. **Scheduling process:**

   The SSM will analyse the request from the information submitted by the customer and shown in Table 3.11. Then it identifies the service requirements as follows:

   - For this service request, the submitted tasks have three security levels required. The SSM will categories Tasks into three Categories. Each Category represents different Security Level. The SSM identifies the number of Resources depending on the number of categories. So, the number of

Resources is $N = 3$ Resources. The Resource costs are $RC_1 = 20$, $RC_2 = 20$, and $RC_3 = 20$.

- For the tasks submitted the Resources associated as follows:
  - Tasks t_1, and t_2 will be allocated to Resource R_1 with security weight 0.25.
  - Tasks t_3, and t_4 will be allocated to Resource R_2 with security weight 0.75.
  - Tasks t_5, and t_6 will be allocated to Resource R_3 with security weight 1.00.

Table 3.11 shows the SSM analysing customer requirements for this service request, and the level of security for all tasks and Resources, and tasks allocated to Resources.

Table 3.11: Example 3.2 SSM Analysing Customer Requirement

Security Level(Weight)/Importance	1	2	3	R	RC_k
1 (0.00)					
2 (0.25)	t_1	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3	t_4		R_2	20
5 (1.00)	t_5	t_6		R_3	20

3. **Calculation:**

After analysing the requirement, the SSM sends the details to the customer to get the final confirmation to establish the service.

- Resource Cost:

  RC_k= Cost for one hour per Resource R_k

  Because $RT_k = 1$

  Therefore

  $RC_1 = 20$, $RC_2 = 20$, $RC_3 = 20$

  $RC = RC_1 + RC_2 + RC_3$

  $RC = 20 + 20 + 20 = 60$

- QoS Cost:

  QoS Cost = $RC * q$

  QoS Cost = 60 * 0.0 = 0.0

- Security Cost for one hour per Resources:

  Security Cost for each Resource:

  SC_1 for R_1 = (20* 0.25) = 5

  SC_2 for R_2 = (20* 0.75) = 15

  SC_3 for R_3 = (20* 1.00) = 20

  Security Cost for all Resources SC:

  $SC = SC_1 + SC_2 + SC_3$

  (SC)= 5 + 15 + 20 = 40

- Actual Cost:

  Actual Cost (AC) = 60 + 0 + 40 = £100

- Check if less than or equal to b:

  $100 \leq 200$ = True

4. **Customer Confirmation:**

   The SSM sends all information about the service to the customer for confirmation as follows:

   Tasks t_1, and t_2 require one Resource (R_1) and Tasks t_3, and t_4 require one Resource (R_2) and Tasks t_5, and t_6 require one Resource (R_3) with Time e = 60 minutes, and the Cost = £100 is less than or equal to b = £200.

   Then, the SSM waits the customer confirmation to establish the service.

5. **Service Established over Resources R:**

   After receiving the confirmation from the customer, the SSM will start to execute the tasks over the allocated resources as follows:

   - All Resources R_1, R_2, and R_3 will be established at the same time. The Tasks execution order for each Resource depends on the security and Importance Levels only and all Resources start at same time:
     - $R_3 : t_6, t_5$

- $R_2 : t_4, t_3$
- $R_1 : t_2, t_1$

- For the same service request in this example suppose that there are dependencies between tasks as follows:
  t_6 depends on t_3, and t_5 depends on t_1
  Tasks execution order will change and the SSM will use the Fast-Track technique see Table 3.12, and the new execution order will be as follows:
  - $R_2 : t_3^{FT}, t_4$
  - $R_1 : t_1^{FT}, t_2$
  - Then $R_3 : t_6, t_5$

Table 3.12: Example 3.2 SSM Analysing Customer Requirement with Fast-Track

Security Level(Weight)/Importance	1	2	3	R	RC_k
1 (0.00)					
2 (0.25)	t_1^{FT}	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3^{FT}	t_4		R_2	20
5 (1.00)	t_5	t_6		R_3	20

- The SSM will start running Tasks in the Fast-Track list for each resource. Then it will use time-out technique to ensure there will not be huge delay in the service and not affecting the other tasks.

Example 2 shows that the scheduling process taking Security Level as main driver to execute Tasks over different Resources is very complex when there are dependencies between Tasks. But by applying the Fast-Track technique to simplify the process when dependencies exist, should not causes any delay to other tasks.

3.4 Benefits of the SSM

This section identifies points that have been addressed by the SSM:

- The SSM offers various options to request a service that should meet the customer requirements such as requiring different Security Levels and QoS which are applicable to the SSM.
- As indicated in the Cloud definition the service should be available on demand at any time. So, the SSM should provide the service at any time for any service request.
- The SSM provides Resources availability for all Security Levels required for any service request.
- Parallel execution over a Resource with dependency. The SSM lets Resource to run at the same time with other Resources if there is any Tasks dependencies to avoid unnecessary waiting time.
- The SSM limited the service to 5 security levels and 5 Resources to minimise the cost. That makes it more clearer for the customer to identify the service requirements.
- Task size is ignored by the SSM, and the SSM more focused on the Tasks Security Levels to establish the service.
- The Security Cost requires defining how many Security Levels to the service request which makes different to the total cost and no matter how many Tasks will be executed in each level, even if all Security Levels are required for a service.

3.5 Summary

This chapter defined the Scheduling Security Model (SSM), and its components. Also, it explained the security in the context of the SSM model as the main factor. After that, it introduced how the SSM model works in each stages explaining the scheduling function and the algorithm. Then it shows how the costs are calculated. At the end, it showed how the SSM works through some examples. Finally, it concluded with benefits that are addressed by the SSM.

Chapter 4

Results

4.1 Introduction

In this chapter, the Scheduling Security Model (SSM) will be examined through worked examples of customer submitted service requests. Each example is presented with some possible scenarios. The SSM shows the Actual Cost (AC), then it applies the time calculating steps in the Re-calculating stage to present the cost of both elapsed time and actual running time. Furthermore, it discusses the cost and effect of the service in each scenario. The main effect is that both the AC of the elapsed time and AC of actual running time are cheaper than the cost of establishing the service. Also, by applying the time Calculating steps the service will run with less waiting time. Finally, as a result of these worked examples and scenarios, it is suggested that comparing the SSM against similar scheduling models in same area would identify more implications and clarify other cases related to cost and effects on the service.

4.2 Examples and Scenarios

This section discusses the Re-Calculating step considering the elapsed time of the tasks compared to the running time. The following examples are to show Re-Calculating for different scenarios. It assumes the Quality of Service cost q is 0, and that all the times are in minutes. Also, a standard cost of £20 per hour has

been set for all examples, that will help to understand how security cost affects the Actual Cost for the service.

4.2.1 Example: 1

A customer submits a service request with the details showing in Table 4.1.

Table 4.1: SSM Customer Requirement for Example 1

Security Level(Weight)/Importance	1	2	3	Resource	RC_i
1 (0.00)					
2 (0.25)			t_1	R_1	20
3 (0.50)		t_2		R_2	20
4 (0.75)					
5 (1.00)					

Tasks submitted as follows:

- Task t_1: p_1= 3, $hw_1 = 0.25$
- Task t_2: p_2= 2, $hw_2 = 0.50$

After analysing the customer inputs, the SSM created a Resource for each Task as follows:

- R_1 for t_1
- R_2 for t_2

Scenario: 1.1

The calculated AC will be as follows:
From equation 3.2.13

$$AC = ((20*1) * (1+0+0.25)) + ((20*1) * (1+0+0.50))$$

$$AC = £25 + £30 = £55$$

For Example 1 Table 4.1: the actual running time each Resource as follows:

- $R_1 : t_1$ and $tm_1 = 18$ minutes
- $R_2 : t_2$ and $tm_2 = 13$ minutes

Tasks time line for Scenario 1.1 is shown in Figure 4.1. Here both actual time and the elapsed time will be equal because there are no dependencies between the tasks. As a result the AC will be the cost of actual running time for each resource. The actual running time for each Resource:

- $tm_1 = 18$ minutes, RT_1: Time for $R_1 = 18$ minutes
- $tm_2 = 13$ minutes, RT_2: Time for $R_2 = 13$ minutes

So, the SSM will use the actual running time for all Resources to Re-Calculating the Actual Cost.
Re−Calculating:
From equation 3.2.13

$$AC = ((20*18/60) * (1+0+0.25))+((20*13/60) * (1+0+0.50))$$

$$AC = £7.5 + £6.5$$

$$AC = £14$$

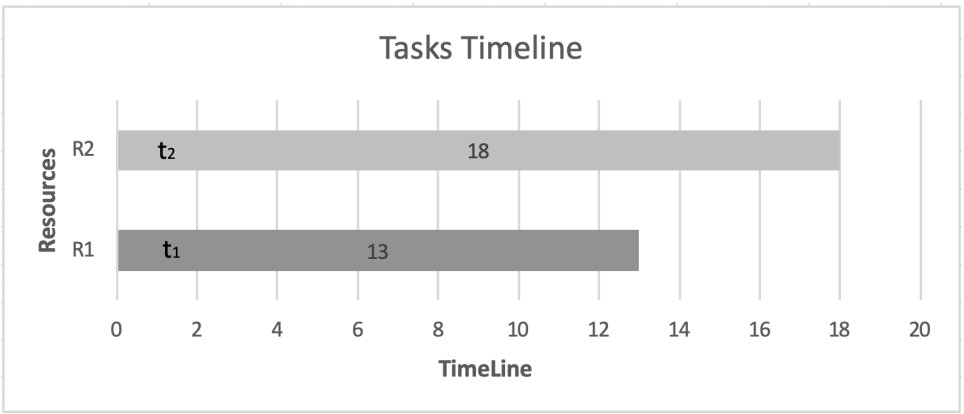

Figure 4.1: Tasks Time Line for Scenario 1.1

Table 4.2: SSM Customer Requirement for Example 2

Security Level(Weight)/ Importance	1	2	3	Resource	RC_i
1 (0.00)					
2 (0.25)	t_1	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3	t_4		R_2	20
5 (1.00)	t_5	t_6		R_3	20

4.2.2 Example: 2

A customer submitted a service request with the details shown in Table 4.2.

Scenario: 2.1

For Example 2 Table 4.2: Scenario: 2.1, the running time for each Resource, and the Tasks with ^{FT} indicates that is has been Fast-Tracked as follows:

- $R_1 : t_1^{FT}$ and $tm_1 = 18$ minutes, t_2 and $tm_2 = 15$ minutes
- $R_2 : t_3^{FT}$ and $tm_3 = 13$ minutes, t_4 and $tm_4 = 10$ minutes
- $R_3 : t_6$ and $tm_6 = 10$ minutes, t_5 and $tm_5 = 5$ minutes

The dependencies are: t_5 depends on t_1 and t_6 depends on t_3. If the SSM considers the running time for each Task, there will be a delay in executing Tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 18$ minutes, $tm_2 = 15$ minutes, RT_1 Time for R_1 = 18+15 = 33 minutes
- $tm_3 = 13$ minutes, $tm_4 = 10$ minutes, RT_2: Time for R_2 = 13+10 = 23 minutes
- $tm_6 = 10$ minutes, $tm_5 = 5$ minutes, RT_3: Time for R_3 = 10+5 = 15 minutes

Here there will be waiting time, so it will be added to RT_3: Time for $R_3 = 13 + 10 + 5 = 28$ minutes. The reason for adding tm_3 not tm_1 is that tm_3 is less than tm_1

which can let the related Task t_6 start just after it finishes.

Re−Calculating:

From equation 3.2.13

$$AC = £13.75 + £13.42 + £18.67$$

$$AC = £45.84$$

This is illustrated in Tasks time line in Figure 4.2. If, the SSM does not consider the waiting time and just calculates the elapsed time as follows:

- $tm_1 = 18$ minutes, $tm_2 = 15$ minutes, RT_1 Time for $R_1 = 18 + 15 = 33$ minutes
- $tm_3 = 13$ minutes, $tm_4 = 10$ minutes, RT_2: Time for $R_2 = 13 + 10 = 23$ minutes
- $tm_6 = 10$ minutes, $tm_5 = 5$ minutes, RT_3 Time for $R_3 = 10 + 5 = 15$ minutes

Re−Calculating:

From equation 3.2.13

$$AC = £13.75 + £13.42 + £10.00$$

$$AC = £37.17$$

As a result of calculating the elapsed time the AC is less than calculating the AC with the running time.

Scenario: 2.2

For Example 2 Table 4.2: Scenario 2.2, one of dependant Tasks finishes before the other Task that relates to a higher Security Level. So, the SSM will start executing the higher Security Task but in a different order depending on what dependant Tasks finishes first:

- $R_1 : t_1^{FT}$ and $tm_1 = 8$ minutes, t_2 and $tm_2 = 3$ minutes
- $R_2 : t_3^{FT}$ and $tm_3 = 10$ minutes, t_4 and $tm_4 = 4$ minutes

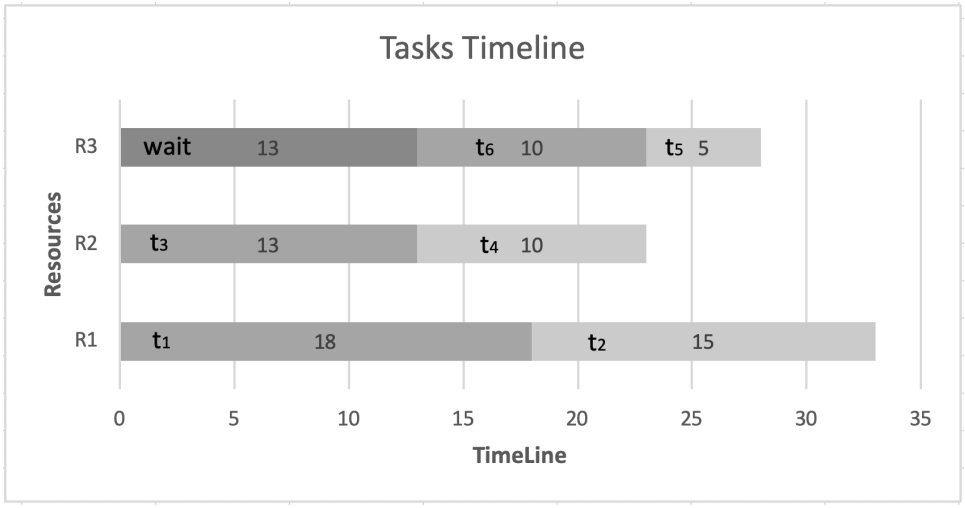

Figure 4.2: Tasks Time Line for Scenario 2.1

- R_3 : t_6 and $tm_6 = 5$ minutes, t_5 and $tm_5 = 7$ minutes

The dependencies are: t_5 depends on t_1 and t_6 depends on t_3. If the SSM considers the running time for each Tasks, there will be a delay of executing tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for $R_1 = 8 + 3 = 11$ minutes
- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for $R_2 = 10 + 4 = 14$ minutes
- $tm_6 = 5$ minutes, $tm_5 = 7$ minutes, RT_3: Time for $R_3 = 5 + 7 = 12$ minutes

Here there will be waiting time, so it will be added to RT_3: Time for $R_3 = 8 + 7 + 5 = 20$ minutes. The reason for adding tm_1 not tm_3 is that tm_1 less than tm_3 which can let the related Task t_5 start just after it finishes and is different from Scenario 2.1.

Tasks time line is shown in Figure 4.3. Also, it shows Tasks order on each Resource and the calculated waiting time until Tasks on R_3 start.

Re−Calculating:

From equation 3.2.13

$$AC = £4.58 + £8.17 + £13.33$$

$$AC = £26.08$$

If, the waiting time is not considered then the elapsed time is calculated as:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for $R_1 = 8 + 3 = 11$ minutes
- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for $R_2 = 10 + 4 = 14$ minutes
- $tm_6 = 5$ minutes, $tm_5 = 7$ minutes, RT_3: Time for $R_3 = 5 + 7 = 12$ minutes

Re−Calculating:
From equation 3.2.13

$$AC = £4.58 + £8.17 + £8.00$$

$$AC = £20.75$$

Again as a result of calculating the elapsed time the AC is less than calculating the AC with the running time.

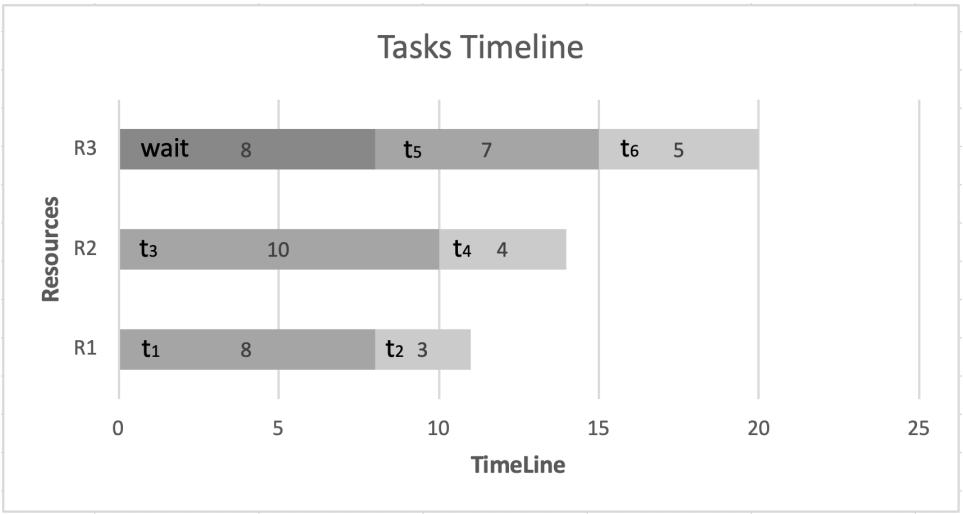

Figure 4.3: Tasks Time Line for Scenario 2.2

Scenario: 2.3

For Example 2 Table 4.2: Scenario 2.3, there are two Tasks dependent on the same Task. So, the SSM will start executing Tasks in same scheduling order considering the Task Importance.

- R_1 : t_1 and $tm_1 = 5$ minutes, t_2 and $tm_2 = 5$ minutes
- R_2 : t_3^{FT} and $tm_3 = 5$ minutes, t_4 and $tm_4 = 5$ minutes
- R_3 : t_6 and $tm_6 = 5$ minutes, t_5 and $tm_5 = 5$ minutes

The dependencies are: t_5 depends on t_3 and t_6 depends on t_3. If the SSM considers the running time for each Tasks, there will be a delay of executing tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 5$ minutes, tm_2= 5 minutes, RT_1: Time for R_1 = 5+5 = 10 minutes
- $tm_3 = 5$ minutes, tm_4= 5 minutes, RT_2: Time for R_2 = 5+5 = 10 minutes
- tm_6= 5 minutes, tm_5= 5 minutes, RT_3: Time for R_3 = 5+5= 10 minutes

Here there will be waiting time, so it will be added to RT_3.
Then, $RT_3 = 5 + 5 + 5 = 15$ minutes. Tasks time line is shown in Figure 4.3. Also, it shows Tasks order on each Resource and the calculated waiting time until Tasks on R_3 start.
Re−Calculating:
From equation 3.2.13

$$AC = £4.17 + £5.83 + £10.00$$

$$AC = £20$$

If, the SSM does not consider the waiting time and just calculates the elapsed time as follows:

- $tm_1 = 5$ minutes, $tm_2 = 5$ minutes, RT_1: Time for R_1 = 5+5 = 10 minutes
- $tm_3 = 5$ minutes, $tm_4 = 5$ minutes, RT_2: Time for R_2 = 5+5 = 10 minutes
- $tm_6 = 5$ minutes, $tm_5 = 5$ minutes, RT_3: Time for R_3 = 5+5 = 10 minutes

Re−Calculating:
From equation 3.2.13

$$AC = £4.17 + £5.83 + £6.67$$

$$AC = £16.67$$

As a result of calculating the elapsed time the AC is less than calculating the AC with the running time.

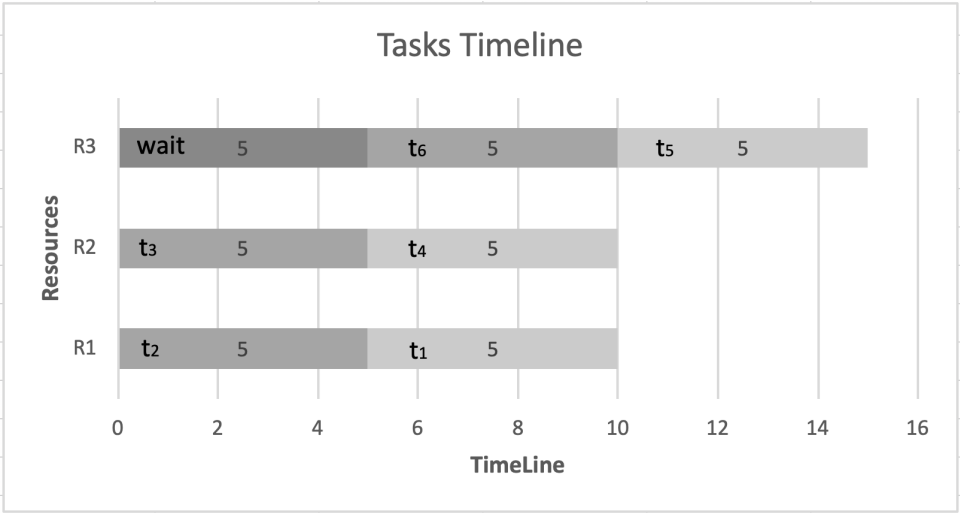

Figure 4.4: Tasks Time Line for Scenario 2.3

4.2.3 Example: 3

A customer submitted a service request with the details shown in Table 4.3.

Table 4.3: SSM Customer Requirement for Example 3

Security Level(Weight)/Importance	1	2	3	Resource	RC_i
1 (0.00)					
2 (0.25)	t_1	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3	t_4		R_2	20
5 (1.00)	t_5, t_7	t_6		R_3	20

Scenario: 3.1

For Example 3 Table 4.3: Scenario 3.1, one of the dependant Tasks finishes before the other Tasks that are related to a higher Security Level. So, the SSM will start executing the higher Security Task in a different order depending on what dependant Tasks finishes first:

- $R_1 : t_1^{FT}$ and $tm_1 = 8$ minutes, t_2 and $tm_2 = 3$ minutes
- $R_2 : t_3^{FT}$ and $tm_3 = 10$ minutes, t_4 and $tm_4 = 4$ minutes
- $R_3 : t_5$ and $tm_5 = 7$ minutes, t_6 and $tm_6 = 5$ minutes, t_7 and $tm_7 = 7$ minutes

The dependencies are: t_5 depends on t_1 and t_6 depends on t_3. If the SSM considers the running time for each Tasks, there will be a delay of executing Tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for $R_1 = 8{+}3 = 11$ minutes
- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for $R_2 = 10{+}4 = 14$ minutes
- $tm_6 = 5$ minutes, $tm_5{=}$ 7 minutes, $tm_7{=}$ 7 minutes, RT_3: Time for $R_3 = 7{+}5{+}7{=}$ 19 minutes

Here, there will be waiting time, so it will be added to $RT_3 = 8 + 7 + 5 + 7 = 27$ minutes. The reason for adding tm_1 not tm_3 is that tm_1 less than tm_3 which can let the related Task t_5 start just after t_1 finishes.
Re−Calculating:
From equation 3.2.13

$$AC = £4.58 + £8.17 + £18.00$$

$$AC = £30.75$$

If, the waiting time is not considered then the elapsed time is calculated as:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for $R_1 = 8 + 3 = 11$ minutes

- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for $R_2 = 10 + 4 = 14$ minutes
- $tm_5 = 7$ minutes, tm_6= 5 minutes, tm_7= 7 minutes, RT_3: Time for R_3= 7+5+7= 19 minutes

Tasks time line is shown in Figure 4.5. Also, it shows Tasks order on each Resource and the calculated waiting time until Tasks on R_3 start.

Again, as a result of calculating the elapsed time the AC is less than calculated AC with the running time. But, what if the SSM lets t_7 run first to reduce the total running time as follows:
The waiting time will be less in both cases in the running time:
For Resource R_3: (8-7) + 7 + 5 + 7 = 20 minutes
Re−Calculation:
From equation 3.2.13

$$AC = £4.58 + £8.17 + £13.33$$

$$AC = £26.08$$

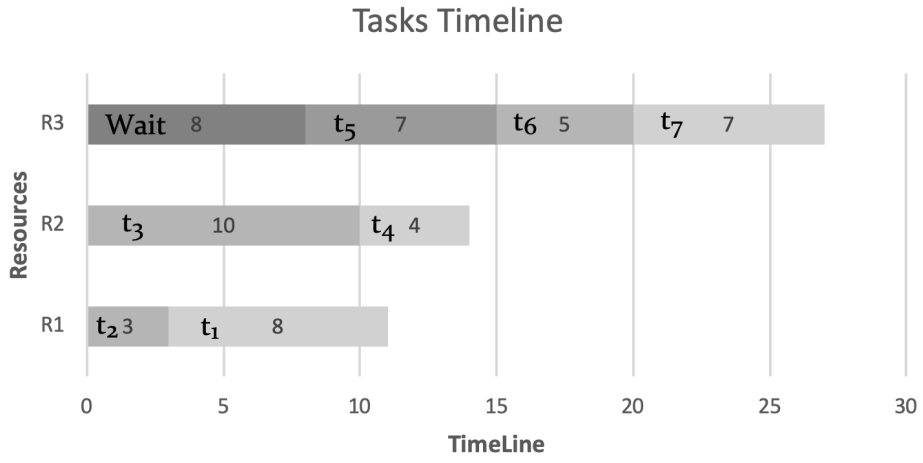

Figure 4.5: Tasks Time Line for Scenario 3.1

Scenario: 3.2

For Example 3 Table 4.3: Scenario 3.2, one of the dependant Tasks finishes before the other Tasks that are related to a higher Security Level. So, the SSM will start

executing the higher Security Task in a different order depending on what dependant Tasks finishes first:

- R_1 : t_1^{FT} and $tm_1 = 8$ minutes, t_2 and $tm_2 = 3$ minutes
- R_2 : t_3^{FT} and $tm_3 = 10$ minutes, t_4 and $tm_4 = 4$ minutes
- R_3 : t_5 and $tm_5 = 7$ minutes, t_6 and $tm_6 = 5$ minutes, t_7 and tm_7= 12 minutes

The dependencies are: t_5 depends on t_1 and t_6 depends on t_3. If the SSM considers the running time for each tasks, there will be a delay of executing tasks t_6 and t_5 because of the dependencies. In this case, the calculated running time for each Resource will be as follows:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for R_1= 8+3= 11 minutes
- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for R_2= 10+4= 14 minutes
- $tm_5 = 7$ minutes, $tm_6 = 5$ minutes, tm_7= 12 minutes, RT_3: Time for R_3= 7+5+12 = 24 minutes

Here there will be waiting time, so it will be added to RT_3 = 8+7+5+12 = 32 minutes.
The reason for adding tm_1 not tm_3 is that tm_1 is less than tm_3 which can let the related Task t_5 start just after t_1 finishes.
Re−Calculating:
From equation 3.2.13

$$AC = £4.58 + £8.17 + £21.33$$

$$AC = £34.08$$

Tasks time line for Scenario 3.2 shown in Figure 4.6. Also, it shows Tasks order on each Resource and the calculated waiting time until Tasks on R_3 start. If, the waiting time is not considered then the elapsed time is calculated as:

- $tm_1 = 8$ minutes, $tm_2 = 3$ minutes, RT_1: Time for $R_1 = 8 + 3 = 11$ minutes
- $tm_3 = 10$ minutes, $tm_4 = 4$ minutes, RT_2: Time for $R_2 = 10 + 4 = 14$ minutes

- tm_5= 7 minutes, tm_6= 5 minutes, tm_7= 12 minutes, RT_3: Time for R_3=5+7+12= 24 minutes

Here, if the SSM lets t_7 run first there will be a delay for running t_5 and t_6 with no waiting time.
Re−Calculating:
From equation 3.2.13

$$AC = £4.58 + £8.17 + £16.00$$

$$AC = £28.75$$

Also, in this scenario, t_6 will run after t_7 then t_5 as it has a higher Security Level, and it is depending on Task t_3.

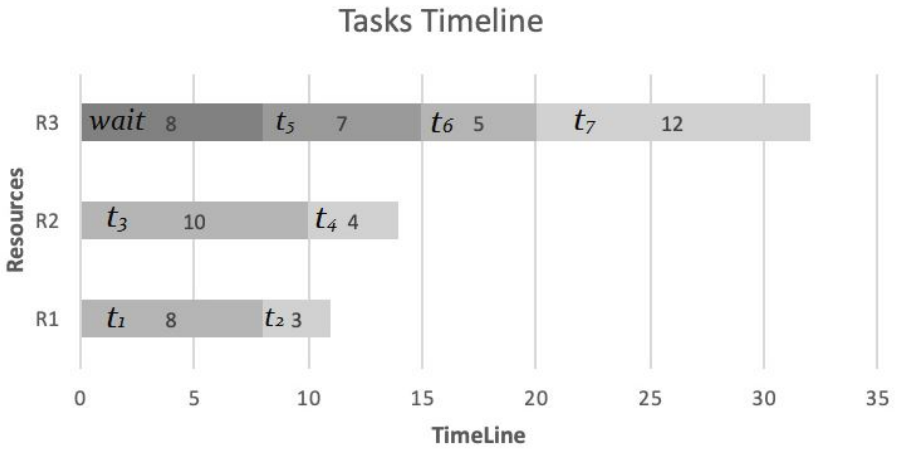

Figure 4.6: Tasks Time Line for Scenario 3.2

4.2.4 Example: 4

A customer submitted a service request with the details shown in Table 4.4.

Scenario: 4.1

For Example 4 Table 4.4: Scenario 4.1 where more than one Task depends on each other. The dependencies are: Task t_5 depends on Task t_3, and Task t_3 depends

Table 4.4: SSM Customer Requirement for Example 4

Security Level(Weight)/Importance	1	2	3	Resource	RC_i
1 (0.00)					
2 (0.25)	t_1	t_2		R_1	20
3 (0.50)					
4 (0.75)	t_3	t_4		R_2	20
5 (1.00)	t_5	t_6		R_3	20

on Task t_2. Also, Task t_6 depends on Task t_1, then Tasks will be allocated over Resources as the following:

- R_1 : t_1^{FT} and $tm_1 = 8$ minutes, t_2^{FT} and $tm_2 = 3$ minutes
- R_2 : t_3^{FT} and $tm_3 = 6$ minutes, t_4 and $tm_4 = 4$ minutes
- R_3 : t_5 and $tm_5 = 7$ minutes, t_6 and $tm_6 = 5$ minutes

The implication here is that Task t_2 has higher Task Importance than Task t_1, then if the SSM lets Task t_2 run first it will cause a delay to run related Tasks. That will cause more delay to Task t_3 then to Tasks t_5 and t_6. The SSM runs Tasks t_2 then t_1, see Figure 4.7 for Tasks time line, and Tasks order on each Resource and the calculated time including waiting time (W) until Tasks start will be as follows:

- R_1: t_1^{FT} and tm_1= 8 minutes, t_2^{FT} and tm_2= 3 minutes, RT_1= 8+3 = 11 minutes
- R_2: t_3^{FT} and tm_3= 6 minutes, t_4 and tm_4= 4 minutes, RT_2= 3+6+4 = 13 minutes
- R_3: t_5 and tm_5= 7 minutes, t_6 and tm_6= 5 minutes, RT_3= 11+7+5 = 23 minutes

The reason of adding tm_2 as a waiting time (W) to RT_2 is that Task t_3 is waiting for Task t_2 to finish. Also, adding RT_1 to RT_3 because R_3 can not start running Tasks until R_1 is finished.

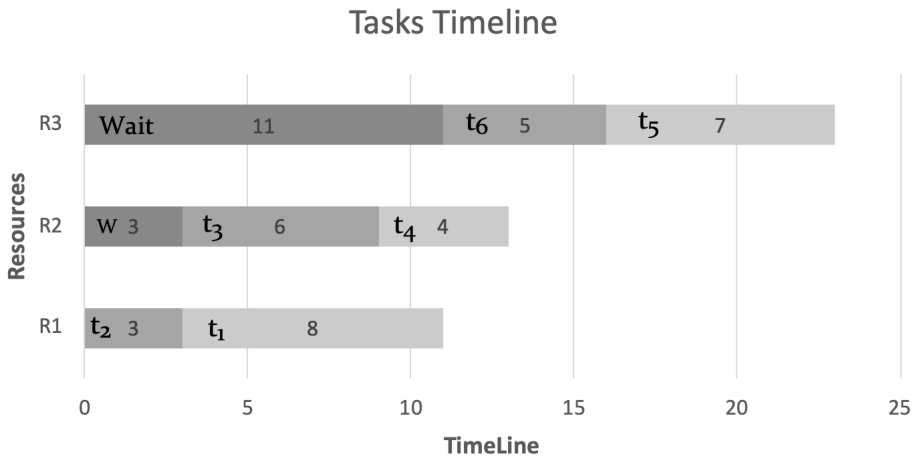

Figure 4.7: Tasks Time Line for Scenario 4.1

4.3 The SSM vs Others

Comparing the SSM against similar work will be useful to see similarities and differences and effects on cost.

For example, Tripathy and Patra [54] give an overview of scheduling Tasks (jobs) with priorities. Their aim is to run all high priority Tasks first over the number of Resources required. Because the Task priority is the main consideration it has been justified to have five level to order tasks to be executed over Resources where 1 is minimum priority and 5 is the highest. They indicated that a job can be executed on multiple Resources R_k. The following example explains how they execute jobs over Resources. As a start, the Task identified with the following attributes job$i(j,k,l)$: i for job or Task id, j number of Resources required, k for Task duration, and l for Task priority. Then the following service request given:

- job1(2,5,1)
- job2(6,10,5)
- job3(2,5,4)
- job4(2,5,2)

Table 4.5. simplifies this service request. The table shows the Task id, and the five Priority Level required for each Task, and there are no Task dependencies between Tasks. Also, it shows the number of Resources required for each Task.

In this service request, there are more than one Resource needed for each Task. That means the total number of Resources is 6 Resources.

Table 4.5: Service Request from Tripathy and Patra [54]

Jobs/Tasks	1	2	3	4	5	Resource
t_1	√					2 Resources
t_2					√	6 Resources
t_3				√		2 Resources
t_4		√				2 Resources

Figure 4.8 shows how Tasks are running over Resources. If the SSM is used for this service to calculate the cost assuming that the Security Level is the minimum and as same as the Quality of Service then the actual cost will be the total cost for all Resources. Also, all Resources have the same running time = 15 minutes. So, the cost for each Resource = (15*20)/60 = £5.
Therefore, the total cost for all Resources = 5 * 6 = £30.

4.4 Discussion

The worked examples show different cost that meet the customer requirements of cost and Quality of Service in the required time. For all examples, each Resource can run a single Task or a set of Tasks. Also, each example ends up with a different cost that is less than the initial cost of establishing the service and shows the different cost by calculating actual running time and calculating the elapsed time.

In Example 1, Scenario 1.1 shows a service request with two different Tasks at a different Security Level. So, the SSM allocated them to two different Resources,

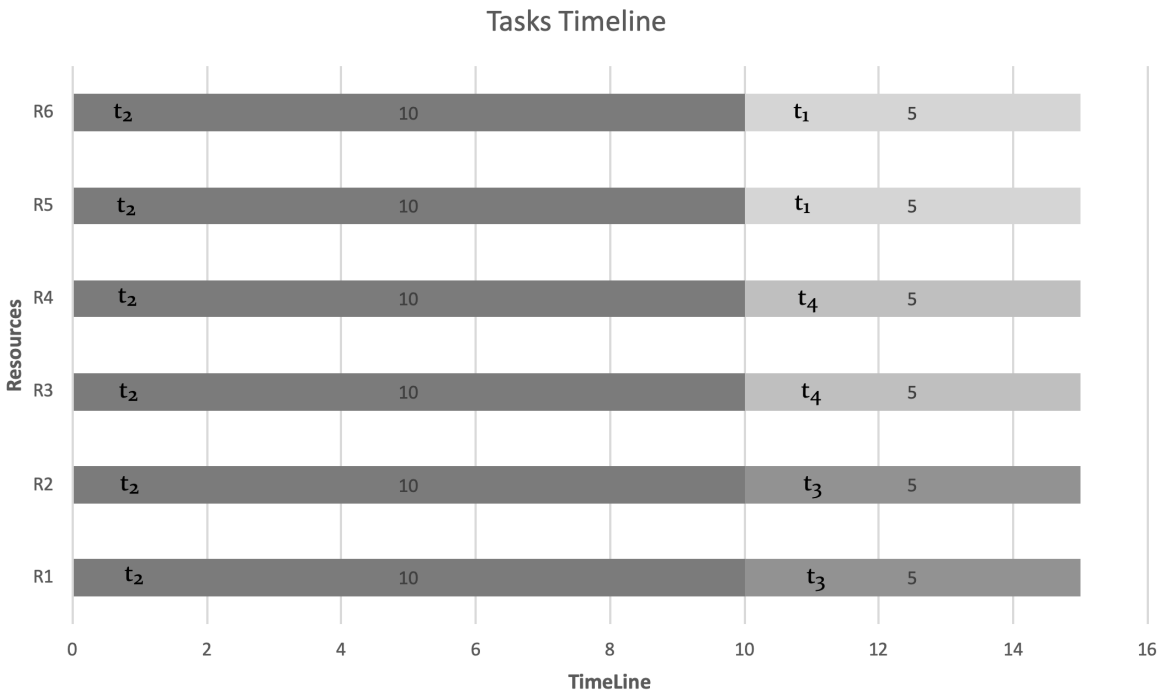

Figure 4.8: Tasks Time Line for Tripathy and Patra [54]

the basic cost was the cost per hour. In the Re-calculating step, the SSM calculates the actual running time of the Resources and as a result the AC becomes cheaper. Also, there is no big effect here in the cost because there is no difference between *AC* with elapsed time and *AC* with actual running time.

Example 2 shows a customer request of a service with more Tasks and different Security Levels. Then it presents different possible scenarios with Tasks dependencies to show the difference in cost and the differences that could affect it in the Re-Calculating step.

In Scenario 2.1, the SSM compares the time of the Tasks with the Fast Tracked Tasks t_1 and t_3 for the dependant Tasks t_5 and t_6. If the times $tm_3 > tm_1$, the SSM lets the Task with the less time run first. The effect is shown in Re-Calculating step the cost using either the elapsed time or the actual running time. As a result, the AC of the elapsed time is less than the AC of the running time.

Scenario 2.2 shows that the dependant Task t_1 has time tm_1 less than tm_3, but the SSM lets t_1 run first because the total delay time will be less than if the SSM lets t_3 run first. This is the only situation that the SSM lets a Task with less Task Importance run first to avoid any possible delay to the service.

In Scenario 2.3, the SSM considers the running time for each Task, and there will is a delay in executing Tasks t_6 and t_5 because of the dependencies.

Example 3 shows a customer request of a service with different Security Levels with Tasks that do not depend on any other Tasks with a higher Security Level. Then it presents different possible scenarios to show the difference in cost and how it affects it in the Re-Calculating step.

In Scenario 3.1, the SSM will start executing Task with higher Security Level but with different order depending on what dependant Tasks finish first. Here, the SSM compares times tm_7 with tm_1 and tm_3, and it found tm_7 is the smallest time. So, the SSM lets Task t_7 run first because the total waiting time will be less than if it waits for the other Tasks to finish.

In Scenario 3.2, the SSM found that one of the dependant Tasks t_1 with less time tm_1 finishes before the other related Tasks with higher Security Level. So, the SSM lets Task t_1 run first, and that lets the Tasks in the higher Security Resource start running after it finished. As a result, the effect is the waiting time becomes less and the cost is cheaper than the cost of establishing the service.

Example 4 shows a customer request of a service with different dependencies which is more complex. Then it shows how the SSM will respond to this kind of request in possible scenarios.

Scenario 4.1 shows how the SSM works with more complex dependencies. The SSM compares Task times to avoid any delay in the waiting time. Here, the SSM

found times $tm_3 > tm_1$. The main issue here is that Task t_2 has higher Task Importance than Task t_1. If the SSM lets Task t_2 run first it will cause a delay running related tasks. This will cause more delay to Task t_3 then to Task t_5 and Task t_6. So, the SSM lets Task t_2 run first to make Task t_3 run next then Task t_1 and then all tasks in R_3 to run after. The effect here is that the cost becomes cheaper and the SSM chooses the possible time without any extra waiting time.

Comparing SSM with Tripathy and Patra [54] gives different cost with different effects. For the example given by Tripathy and Patra the cost is 30. But if the SSM takes the service request and apply the model with same Tasks, the cost will be different. Because, there will be just one Resource to run all the Tasks for the same Security Level. Then will execute all Tasks in the order given with their times. The difference will be the Resource will be reserved for each Task at the time until it finishes, not like Fig 5.8 showing a delay and each Resource is on hold until the Task finishes. So, calculating the Actual Cost by the SSM will be the cost of total time for one Resource. The total time= 10 + 5 + 5 + 5 = 25 minutes, considering the Security Level is 1 and q is 0, then AC= 25*20/60 = £8.33.

Table 4.6 shows the comparison of the similarities and differences between SSM and the other work.

Table 4.6: Compare SSM with another Model

Model	Security	QoS	Priority	Time	Cost
SSM	✓	✓	✓		✓
Tripathy and Parta [54]			✓	✓	

In all the scenarios, the SSM has applied the following time calculating steps:

- For each Task ask if it has dependants .
  - If no dependants then add Task time to Resource time RT_k.
  - If dependants then go to dependant Task and ask again.

- There will be different scenarios for calculating the time if dependant Tasks are on different Resources R_k. So, time can be calculated from one of the following:

  - The dependant Task for the higher Security Task has less or equal time than the others. So, it will be run first then add its time to the correspondent Resource time RT_k.
  - The dependant Task for the higher Security Task requires more time than the others. So, the SSM will run the Task with less time and then return to it and add both times to RT_k.
  - In the case, where the dependant task t_1 with tm_1 also has a dependant Task t_2 with tm_2, the SSM will add tm_2 to the Resource time with tm_1. Then both times will be added to the correspondent Resource R_k.
  - In the case, where two or more Tasks with dependant Tasks, the SSM will check the time for the dependant Task with higher Security Level and then decide which dependant Task will run first then add its time to the correspondent Resource R_k.

4.5 Summary

This chapter examined the Scheduling Security Model (SSM) through worked examples of customer submitted service requests. Each example is presented with some possible scenarios. The SSM showed the Actual Cost (AC), then it applied the time calculating steps in the Re-Calculating stage to present the cost of both elapsed time and actual running time. Furthermore, it discussed the cost and effect of the service in each scenario. The main effect is that both the AC of the elapsed time and AC of actual running time are cheaper than the cost of establishing the service. Also, by applying the time in the calculating steps the service will run with less waiting time. Finally, as a result of these worked examples and scenarios, it is suggested that comparing the SSM against similar scheduling models in same area would identify more implications and clarify other cases related to cost and effects on the service.

Chapter 5

Evaluation

5.1 Introduction

This Chapter will discuss and evaluate the SSM and how it developed incorporating Security and Scheduling. Then it discusses the examples from Chapter 4 to evaluate the SSM and to see what is the impact on the service cost before and after establishing the service. Finally, it discusses the SSM and its benefits compared to other approaches that have been found depending on the literature survey that has been done in Chapter 2.

5.2 Evaluation Discussion

5.2.1 Main Questions

1. Can a Scheduling Security Model (SSM) be developed or adapted from existing models to incorporate security and scheduling?
   Chapter 3 Section 3.2 has defined the SSM incorporating with scheduling. As the scheduling process can serve a specific type of Tasks and order them depends on the Task Priority, the SSM added Security feature to be the main driver for scheduling the Tasks over the allocated Resources. Then Task Importance will be used to put the Task in the right order in each Resource.

2. What are the barriers to scheduling in terms of security and how they affect

scheduling?

Tasks dependencies affects scheduling and Categorising Tasks by Security can make the scheduling more effective it discussed in Chapter 4 Section 4.2.

3. Can any barriers identified to the use of the SSM be overcome?

   The SSM tries to avoid unnecessary waiting time especially when Tasks dependencies existed for a service request discussed in Chapter 4 Section 4.4.

5.2.2 Background Questions

The following questions are background questions that have been identified which will be a part of the literature review process in Chapter 2:

1. What previous research has been done in terms of Scheduling in the cloud in the presence of security constraints?

   Most of previous research focused on Tasks Priority and Scheduling performance and Security has been addressed in other models for different cloud services such as Database services and Data storing.

2. What constraints other than security need to be considered by each cloud stakeholder in terms of scheduling in the cloud?

   As it discussed, security is a shared responsibility for all parties. Everyone should follow all security procedure to make the service in top security.

3. What different types of security constraints are identified and what do these security constraints defend against?

   Security is a terms that include securing all assets from any threats, even if the asset is physical or digital.

4. What evaluation metrics have been used to help to evaluate recent research into Scheduling in the cloud?

   The SSM takes time, cost, security, priority, and QoS in consideration to be addressed in the scheduling process. All examples in Chapter 4 Section 4.4. discussed how the service time and cost become better after the SSM establishing the service.

5. Based on the analysed research identified, which forms of security aware scheduling merits further research, and why is more research needed and what issues should the further research be addressing?
   There is a need to investigate having more than one Resources in each security level. Does it bring more security implications especially when there are Tasks dependencies as it allows transactions between Resources.

5.2.3 Evaluation Questions

The following questions are the evaluation questions that will be used to see how this research achieved its aims:

1. How does the SSM improve the security aspects of the cloud service?
   The SSM applies security to the tasks level to make the customer to be more specific on the service requirement. After that the SSM applies security to the resource level to help running the cloud service on a trusted level (See Chapter 3 Section 3.3 and Section 3.4).

2. How does the SSM impact Resource scheduling and performance and security?
   Performance is a big issue in regards to cloud-based services and currently, the SSM allows for each resource to run a single task or a set of tasks. However, there is a need to investigate how it impacts the service from the performance perspective.

3. How well does the SSM help to achieve QoS?
   The SSM applies different levels of QoS, but there is still a need to clarify and inspect these levels of QoS, as well as explore how QoS affects the factors that were considered in these levels (See Chapter 3 Section 3.4).

Figure 5.1. shows the SSM features for a cloud service request. These features are Security, Priority, QoS, Time, and Cost per Resource. QoS contributes to the overall service cost and any change in QoS levels will affect the actual service cost. Security applied to the service cost as well and used in the scheduling process by Categorising the Tasks by the Task Security Level. Priority will be used in the

scheduling process to put the Tasks in the right order for each Resource. Time will be calculated initially before establishing the service and after to show the different between the elapsed time and the actual running time. Cost per Resource will be calculated before establishing the service and after receiving customer confirmation it will be Re-Calculating to have the Actual Cost.

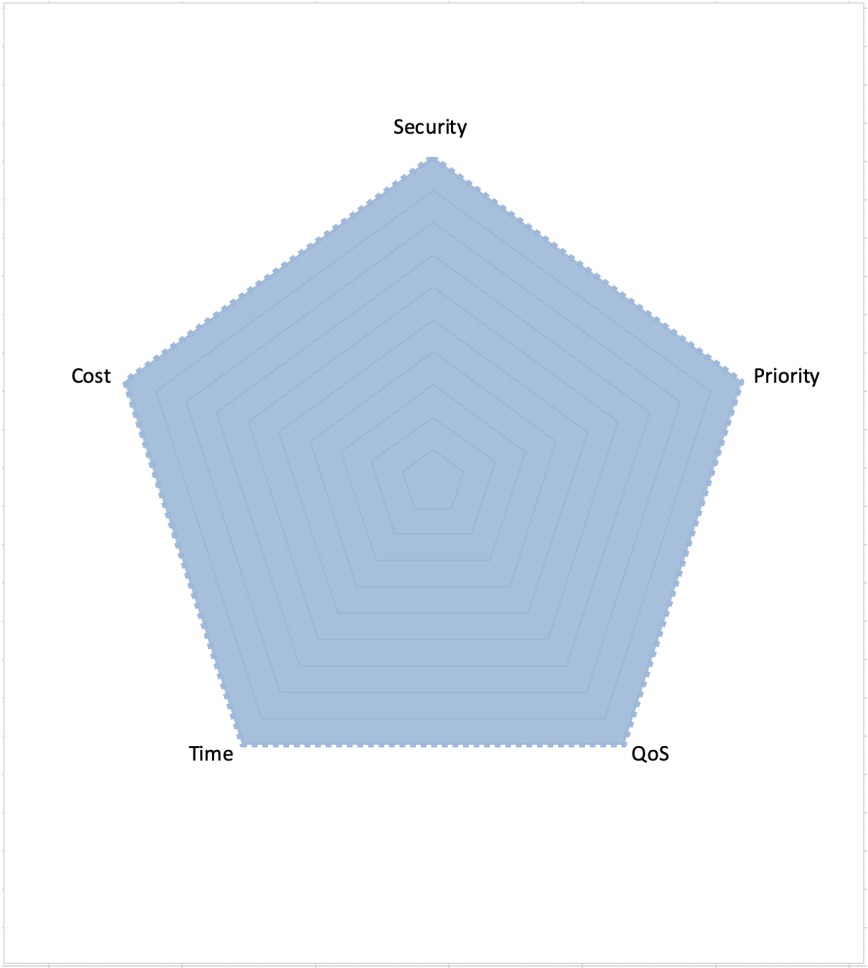

Figure 5.1: Service Features of Applying the SSM

5.3 Examples and Scenarios

For all examples and scenarios will discuss aspects that important to be addressed by the service which are Security, Time, and Cost.

First, in all examples and scenarios Tasks Security is identified by the customer and the value that added to the service request is to know all service requirements. Which is better than just requesting a service in a certain level of security with Resources then requesting a different Resource with different security level.

Next, one of the main important issues for any cloud services is to be in a convenience time for any customer requests. Also, the service provider should manage to finish the cloud service on requested time or less.

Last, it is very important that the customer identified all service requirements for any service request, because it would help the service provider to analyse the requirements and establish the service without any issues or asking the customer for more details.

Table 5.1: Example 1.1.1

SSM Features	**Example 1**	
	Before	**After**
Security	Applied to Categorise Tasks	✓
QoS	Calculated	✓
Priority	Applied to Order Tasks	✓
Time	Initial Time Calculated for 1 hour or 60 minutes	18 minutes
Cost	Initial Cost Calculated £55	£14

Table 5.1 shows the changes to the service requested on the SSM features in Example 1.1.1. The Features are Security, QoS, Priority, Time, and Cost. In this example, there are no change in the Security, Priority, and QoS. The change means

there are no values reduced or increased that could cause any effects to the service. Also, this the reason the starting points for all of this three features is in the centre of the graph.

As shown in Figure 5.2, the calculated time is less than the initial time after establishing the service. The Cost calculated in £Pounds as a cost unit for this example. As seen in Figure 5.3 the initial cost before establishing the service is £55. Then it reduced after running the service to £14. Service time is calculated initially for 1 hour or 60 minutes, but the SSM will Re-calculating the service time after getting the Customer Confirmation to establish the service.

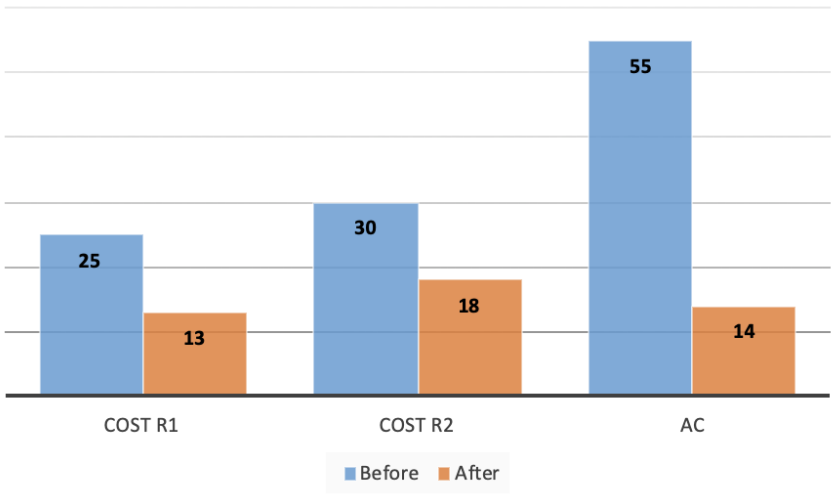

Figure 5.2: Example 1 Scenario 1: Change in Service Cost Before and After SSM Applied

So, in this example the service time calculated for each Resource and for the first Resource the time is 13 minutes and for the second Resource is 18 minutes. As a result, both Resources did not take more than the higher Time which is the time for the second Resource, and it reflected on the total or the Actual Cost (AC).

Table 5.2. Example 2 shows that in the Re-Calculating step the cost using either the elapsed time or the actual running time are less than the initial cost. Figure 5.4. shows the change in service time for each Resource before and after applying the

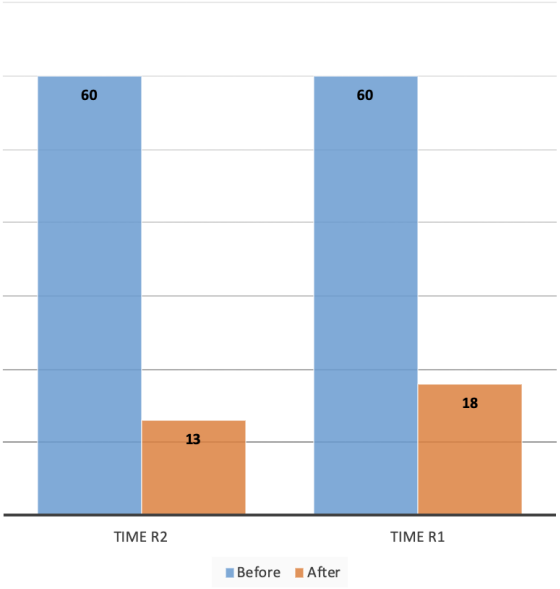

Figure 5.3: Example 1 Scenario 1: Changes in Service Time Before and After SSM Applied

SSM. As a result, the AC of the elapsed time is less than the AC of the running time.

Table 5.2: Example 2

SSM Features	**Example 1**	
	Before	**After**
Security	Applied to Categorise Tasks	✓
QoS	Calculated	✓
Priority	Applied to Order Tasks	✓
Time	Initial Time Calculated	✓
Cost	Initial Calculated	✓

The change in cost for each Resource and then the different between the initial cost and AC shown in Figure 5.5. Also, this example has shown how the SSM works with the Tasks dependencies by explaining the Fast-Track technique with its benefits for scheduling the dependent Tasks with high Security Level.

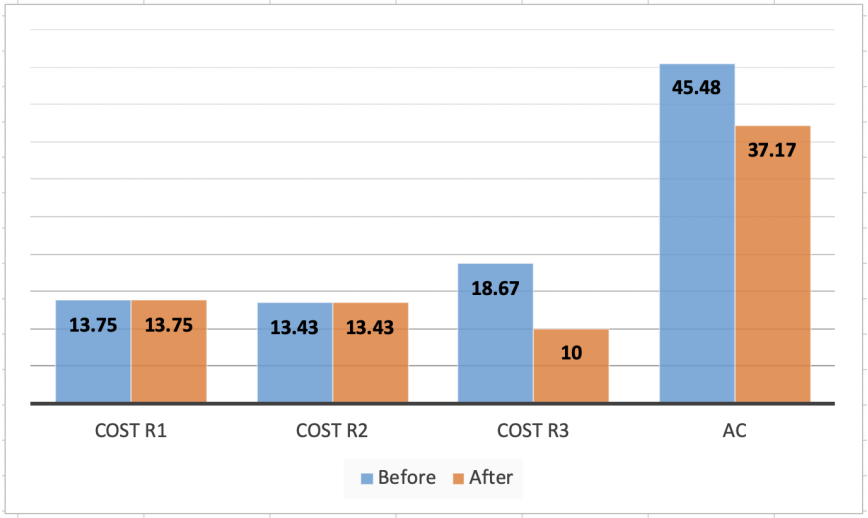

Figure 5.4: Example 2 Scenario 1: Change in Service Cost Before and After SSM Applied

For Example 3, Table 5.3. has shown that the SSM very effective and it works to reduce the waiting time until the dependent Tasks run. Moreover, this example has been extended in Example 4. to have more complex Tasks dependencies. It shows how the SSM select the dependent Tasks to avoid unnecessary waiting time to execute the Task with higher Security Level.

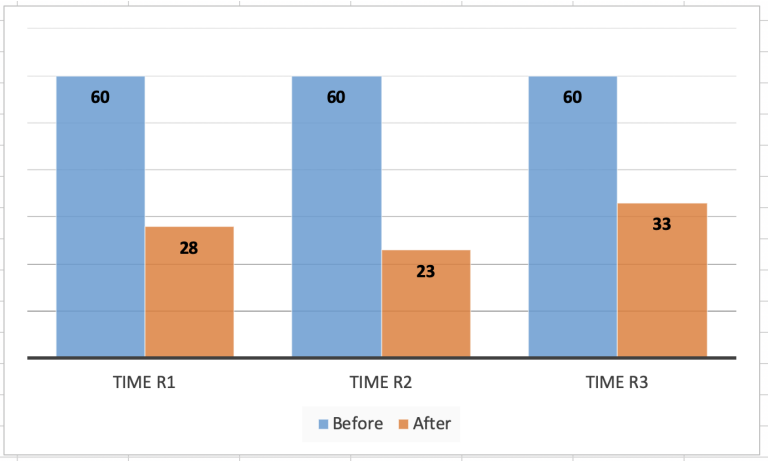

Figure 5.5: Example 2 Scenario 1: Change in Service Time Before and After SSM Applied

Table 5.3: Example 3

SSM Features	Example 1	
	Before	**After**
Security	Applied to Categorise Tasks	✓
QoS	Calculated	✓
Priority	Applied to Order Tasks	✓
Time	Initial Time Calculated	✓
Cost	Initial Calculated	✓

5.4 Comparison with Other Approaches

In order to identify more implications and to clarify issues in the SSM. As a result, the comparison showed that the SSM and other model have different cost and with different effects on service time. The SSM has more features than other models, but one of the implications that have been founded is the SSM does not allow more than one Resource in the same Security Level. This can affect the total service time and it might cause a delay to execute other Tasks in the same Resource.

However, there is a lack of similar work with security as main feature for scheduling Tasks over allocated Resources. Table 5.4. shows that what is the SSM providing against other Cloud Models. Also, it shows that there are shared features but there are for different purpose. For example, the Cloud Trust Model [42] is including security but it serves DaaS field. So, The SSM adds to Scheduling and QoS the features Security, Cost, Service Availability, and IaaS all together to improve the cloud service.

Table 5.4: The SSM and Other Cloud Models

Ref	Category				Main Focus												
	DaaS	SaaS	IaaS	Cloud storage	Availability	Confidentiality	Reliability	Intrusion	Integrity	Fault tolerance	Recovery fail	Cost	Scalability	Performance	Accountability	Latency	Security	Authentication
DepSky [7]	√				√	√												
Bluesky [57]	√																	
SafeStore [25]	√					√												
NetDB2-MS [2]	√				√			√	√									
NCCloud [21]				√						√	√	√						
HAIL [8]				√	√				√									
ICStore [11]				√				√										
SPORC [16]				√	√													
Depot [29]				√	√						√					√		
Logging Solutions [61]			√										√	√	√			
Venus [49]				√					√									
TCCP [41]			√		√				√									
CCM [13]																	√	
Hexagon Model [13]																	√	
MTCM [13]																	√	
CSA [13]																	√	
Mapping model [13]																	√	
Separation Model [65]										√								
Migration Model [65]									√								√	
Availability Model [65]					√												√	
Tunnel Model [65]									√								√	
Cryptography Model [65]						√											√	
NDSM [3]	√			√													√	
Cloud Trust Model [42]	√																√	
DSM [62]	√					√											√	
DSSM [34]	√			√													√	√
SC [54]														√				
SSM			√		√							√					√	

5.5 Summary

This Chapter provided a discussion of the results from Chapter 4, which helped to discuss the evaluation questions to give better understanding to the SSM. It presented how the SSM improve the security aspect of the Cloud service by implementing the security as a main feature for executing Tasks and the effects on the Cost.

Chapter 6

Conclusions

6.1 Introduction

The rapid development of the field of Information Technology (IT) has introduced different methods for services such as storing and saving data and running tasks from a remote location. One of these methods is cloud computing, and it has many benefits as it can provide convenient services with the availability of accessing resources such as networks, data storage, and applications anytime from anywhere.

It becomes widely used by different sectors and individuals for different purpose. Also, it provides many facilities and services that do not required any involvement from the service providers to start the services or to adjust the service settings for the customers. In general, customers can start the cloud services just by purchasing what they need online via the Internet.

Cloud services brings many benefits for all users, they can access to software and applications without installing to their devices. This can reduce costs of buying the full software or buying more computing resources such as more storage or more advanced units.

As there are benefits of using cloud computing services there are many risks. Security and Privacy are the major concerns for the cloud environment, that can

affect trusting the service for all parties. Another concern is losing the business as their no quality of the service provided.

So, the consideration of having better Quality of Service (QoS) includes applying aspects such as service time, performance, cost, and security successfully. As a result of this, the cloud services will be improved and will meet customer expectations.

Moreover, cloud security (and the potential breaching of this security) is significant issue that needs to be addressed within the cloud environment, because interaction with cloud services is all done through the Internet. So, that brings concerns and shared same responsibilities to all parties to keep the security in the top level.

Interacting with cloud services such as running tasks on resources requires scheduling technique that can handle requests in right order without any delays. Scheduling is a very complex process and it should be precise to make the service more reliable.

In addition, there are different type of scheduling but the most important is to include security in the scheduling process. That will help in providing secure services to very complicating requests with high security requirements.

In this thesis, investigation of the current situation described and then discussed the requirement of having a cloud security model based on cost that can manage requests by identifying security as a main feature associated with QoS aspects to meet customer requirements. Then the model runs the scheduled tasks over allocated resources.

Also, this thesis defined and developed a Scheduling Security Model (SSM) for a Cloud Environment that uses Security as a main feature and QoS to handle customers requests with different security levels.

6.2 Research Criteria for Success

Research criteria for success were introduced and outlined in Chapter 1 to assess this thesis in achieving the identified objectives. The review for each one will be as the following:

6.2.1 Identification of Security Issues in Cloud Computing

It is very important to identify security issues in cloud computing as it is a major concern to all parties in the cloud environment. This will help to propose the new service model. Chapter 2 Section 2.7, has identified what are the security factors in cloud computing that could affect the services.

6.2.2 Development of a Scheduling Security Model

There are several models have been reviewed in Chapter 2 Section 2.8 and Section 2.9, to identify the direction of this thesis. As a result, this thesis proposed and defined a new Scheduling Security Model (SSM), that serves security as a main feature to schedule tasks and execute them over Resources and this model successfully developed to be Scheduling Security Model (SSM) introduced in Chapter 3 Section 3.2.

In Chapter 4 Section 4.4, examples with scenarios used to examine the SSM and discussed the cost and the effect of the service in each scenario. As a result, each time at the Re-Calculating process the AC is cheaper than AC of establishing the service.

6.2.3 Evaluating the Model

In order to achieve the aims of the SSM, it has been evaluated in Chapter 5. It shows that how is applying the SSM can keep the security for a service requested in the required level. As a result, it present that how it can improve the security for the cloud service. On the other hand, the service performance needs to be more investigated by applying the SSM as currently does not show any implications about having a high load work with a huge number of tasks. Also, the SSM has not been

investigated with QoS levels and all examples have been set up with the minimum QoS level as there many affecting factors have been pointed in Chapter 2 Section 2.3. This will help for more investigation to understand the impact on service cost of applying different QoS levels.

6.2.4 Comparison Against Other Approaches

In order to identify more implications and to clarify issues in the SSM. A comparison has been suggested in Chapter 4 Section 4.3 and Section 4.4 to compare the SSM and other Models. As a result, the comparison showed that the SSM and other model have different cost with different effects. The SSM has more features than other models, but one of the implications that have been founded is the SSM does not allow more than one Resource in the same level of security and it affects the total service time and it might cause a delay execute other tasks in same Resource. However, this comparison needs to be extended against different approaches that have same security feature but currently there is a lack of similar work with security as main feature for scheduling tasks over allocated resources.

6.3 Future Work

This thesis propose a number of future directions this section discusses these future work as follows:

6.3.1 Developing the SSM: Security Level with Multi Resources

This section discusses possible developing to the SSM and trying to avoid the overall waiting time. The proposed development is to allow creating more resources in each security level to avoid waiting time.

- **How?**

  This technique can allow to have more than one Resource for each Security Level. To do that, the SSM will create more Resource in same security level required.

- **Mini Resources:**

  The SSM will create mini Resources with same security level that have a Fast-Track list into mini Resources. The first one includes the Fast-Track list and the others as required include other tasks with the normal scheduling order. All mini Resources will be running in the same time.

- **Benefits of proposed technique:**

  This technique will try to avoid any waiting time that cause delay to the overall running time for all Resources. Also, it will help to avoid waiting time to other tasks in Resources especially tasks are not in fast-track list.

- **Examples:**

  Chapter 4 has presented examples with scenarios to examine the SSM. The proposed technique will use the same examples and scenarios to see how it can be applied.

6.3.2 Developing the SSM with GUI

The current system is implemented using command line. But, for more usability it would be better to develop the system using Graphic User Interface (GUI) to make it more clear for any customers to deal with the system. That will help to make the Submit process easier for the customer. Identifying the service inputs in better way will make the system result more understandable.

6.3.3 Investigate Different Security Levels

It suggested that to investigate the SSM with different security levels more or less. That could to see more impact on the cloud service and service cost.

6.3.4 Investigate Service Performance for the SSM

The SSM allows for a cloud service request a single task or a set of tasks to be executing over Resources. But there is a need for more investigation of the impact on service performance and to know what implications can be result in of using the SSM.

6.4 Summary

This Thesis discussed and presented the overview of features involved to develop the new service model. These features are Scheduling, Quality of Service (QoS), Security, and Cloud Environment. Then it explained what part every feature it gives to the cloud services. First of all, it discussed the Scheduling and why it is important to have an efficient use of cloud resources. Then it explained that to have a better service quality there is a need to consider the QoS affecting factors. After that, it discussed the Security as a shared responsibility for all cloud parties and how can classify different assets to multiple security levels. Next, Cloud Environment has been presented and discussed by defining the cloud characteristics and architecture.

Also, it explained the security issues in the service models and the deployment models. Then it explained the relation between all the features that involved to develop the new service model within the cloud. After it investigated the related work to identify the assumptions that used to develop and define the new model. As a result, it explained why there is a need to have the Scheduling Security Model (SSM) based on cost for a cloud environment as a new approach. The results showed that the SSM is cost effective by applying the SSM for a service request and by keeping the security level as it required. Finally, it evaluated and compared different possible scenarios to find any implications that could occur by applying the SSM on a cloud service request.

References

[1] A. Abdelmaboud, D. Jawawi, I. Ghani, A. Elsafi and B. Kitchenham, ***Quality of service approaches in cloud computing: A systematic mapping study***, Journal of Systems and Software, Elsevier, Vol 101, pp 159–179, 2015.

[2] M. Alzain, B. Soh and E. Pardede, ***A new model to ensure security in cloud computing services***, Journal of Service Science Research, Springer, Vol. 4, No. 1, pp. 49–70, 2012.

[3] Sh. Ajoudanian and Mr. Ahmadi, ***A novel data security model for cloud computing***, International Journal of Engineering and Technology, IACSIT Press, Vol. 4, No. 3, pp. 326–329, 2012.

[4] J. Anupa and KC. Sekaran, ***Cloud workflow and security: A survey***, International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, pp. 1598–1607, 2014.

[5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinsky, L. Andrew, P. Gunho, A. David, A. Rabkin and I. Stoica, ***Above the clouds: a Berkeley view of cloud computing***, Tech. Report UCB/EECS-2009-28, EECS Dept., University of California, Berkeley, 2009.

[6] X. Bai, M. Li, B. Chen, WT. Tsai and J. Gao, ***Cloud testing tools***, International Symposium on Service Oriented System Engineering (SOSE), IEEE, pp. 1–12, 2011.

[7] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, ***DepSky: dependable and secure storage in a cloud-of-clouds***, ACM Transactions on Storage (TOS), ACM , Vol. 9, No. 4, pp. 31-46, 2013.

[8] K. Bowers, A. Juels and A. Oprea, ***HAIL: a high-availability and integrity layer for cloud storage***, Proceedings of the 16th ACM conference on Computer and communications security, ACM, pp. 187–198, 2009.

[9] G. Brunette and R. Mogull, ***Security guidance for critical areas of focus in cloud computing***, Cloud Security Alliance, Vol. 2, No. 1, pp. 1–76, 2009.

[10] D. Budgen, ***Protocol for a Systematic Literature Review on Empirical Studies of Software Visualisation***, Unpublished paper, 2011.

[11] C. Cachin, R. Haas and M. Vukolic, ***Dependable storage in the intercloud***, IBM Research, Vol. 3783, pp. 1–6, 2010.

[12] RN. Calheiros, R. Ranjan, A. Beloglazov, A. De Rose, A. César and R. Buyya, ***CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms***, Software Practice and Experience, Wiley Online Library, Vol. 41, No. 1, pp. 23–50, 2011.

[13] V. Chang, D. Bacigalupo, G. Wills and D. De Roure, ***A categorisation of cloud computing business models***, IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 509–512, 2010.

[14] J. Che, Y. Duan, T. Zhang and J. Fan, ***Study on the security models and strategies of cloud computing***, Procedia Engineering, Elsevier, Vol 23, pp. 586–593, 2011.

[15] T. Dillon, C. Wu, and E. Chang, ***Cloud computing: issues and challenges***, International Conference on Advanced Information Networking and Applications, IEEE, pp. 27–33, 2010.

[16] AJ. Feldman, WP. Zeller, MJ. Freedman and EW. Felten, ***SPORC: Group Collaboration using Untrusted Cloud Resources***, Proceedings of the

9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), USENIX, Vol. 10, pp. 337–350, 2010.

[17] I. Foster, Y. Zhao, I. Raicu and S. Lu, ***Cloud computing and grid computing 360-degree compared***, Grid Computing Environments Workshop, IEEE, pp. 1–10, 2008.

[18] Wu. Fuhui, Wu. Qingbo and T. Yusong, ***Workflow scheduling in cloud: a survey***, The Journal of Supercomputing, Springer, Vol. 71, No. 9, pp. 3373–3418, 2015.

[19] H. Goudarzi and M. Pedram, ***Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems***, International Conference on Cloud Computing (CLOUD), IEEE, pp. 324–331, 2011.

[20] D. Goutam, A. Verma and N. Agrawal, ***The Performance Evaluation of Proactive Fault Tolerant Scheme over cloud using CloudSim Simulator***, International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), IEEE, Vol. 5, pp. 171–176, 2014.

[21] Tu. Hu, HCH. Chen, P. Lee and Y. Tang, ***NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds***, 10th USENIX Conference on File and Storage Technologies (FAST), USENIX, pp. 265-272, 2012.

[22] A. Josh, ***Understanding the Linux 2.6. 8.1 CPU scheduler***, `http://www.inf.ed.ac.uk/teaching/courses/os/prac/linux_cpu_scheduler.pdf`[Access in October 2017].

[23] C. Kankanamge, ***Web services testing with soapUI***, Packt Publishing Ltd, 2012.

[24] R. Kaur and NS. Ghumman, ***A Survey and Comparison of Various Cloud Simulators Available for Cloud Environment***, International Journal of Advanced Computing and Communication Engineering (IJARCCE), 2015.

[25] R. Kotla, L. Alvisi and M. Dahlin, ***SafeStore: a durable and practical storage system***, USENIX Annual Technical Conference, USENIX , pp. 129–142, 2007.

[26] J. Li, M. Qiu, JW. Niu, Y. Chen and Z. Ming, ***Adaptive Resource Allocation for Preemptable Jobs in Cloud Systems***, International Conference on Intelligent Systems Design and Applications, IEEE, Vol. 10, pp. 31–36, 2010.

[27] Q. Li, Q. Hao, L. Xiao and Z. Li, ***Adaptive Management of Virtualized Resources in Cloud Computing using Feedback Control***, First International Conference on Information Science and Engineering, IEEE, Vol. 1, pp. 99–102, 2009.

[28] M. Mathirajan and AI. Sivakumar, ***A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor***, The International Journal of Advanced Manufacturing Technology, Springer, Vol. 29, No. 9, pp. 990–1001, 2006.

[29] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin and M. Walfish, ***Depot: Cloud storage with minimal trust***, Transactions on Computer Systems (TOCS), ACM, Vol. 29, No. 4, pp. 307-322, 2011.

[30] P. Mell and T. Grance, ***The NIST definition of cloud computing***, Computer Security Division, Information Technology, Laboratory, National Institute of Standards and Technology, Gaithersburg, 2011, `http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf`[Accessed in July 2016].

[31] V. Nallur and R. Bahsoon, ***A decentralized self-adaptation mechanism for service-based applications in the cloud***, Transactions on Software Engineering, IEEE, Vol. 39, No. 5, pp. 591–612, 2013.

[32] CP. Pfleeger and SL. Pfleeger, ***Security in computing***, Prentice Hall Professional Technical Reference, 2002.

[33] F. Panzieri, O. Babaoglu, S. Ferretti, V. Ghini and M. Marzolla, ***Distributed Computing in The 21st Century: Some Aspects of Cloud Computing***, Dependable and Historic Computing, Springer, pp. 393–412, 2011.

[34] HB. Patel, DR. Patel, B. Borisaniya and A. Patel, ***Data Storage Security Model for Cloud Computing***, International Conference on Advances in Communication, Network, and Computing, Springer, pp. 37–45, 2012.

[35] P. Patel, AH. Ranabahu and AP. Sheth, ***Service Level Agreement in Cloud Computing***, 2009, `https://corescholar.libraries.wright.edu/knoesis/78/`[Accessed in September 2016].

[36] R. Patil and RK. Singh, ***Scaling in Cloud Computing***, International Journal of Advance Research, IJOAR, Vol. 1, pp. 21–27, ISSN:2320-9194, 2013.

[37] M. Petticrew and H. Roberts, ***Systematic reviews in the social sciences: A practical guide***, John Wiley & Sons, 2008.

[38] M. Pinedo, ***Scheduling: Theory, Algorithms, and Systems***, Springer, pp. 1–10, 2008.

[39] H. Ramadan, and D. Kashyap, ***Quality of Service (QoS) in Cloud Computing***, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 8, No. 3, pp. 318–320, 2017.

[40] S. Ramgovind, M. Eloff, and E. Smith, ***The management of security in cloud computing***, Information Security for South Africa, IEEE, pp. 1–7, 2010.

[41] N. Santos, KP. Gummadi and R. Rodrigues, ***Towards Trusted Cloud Computing***, HotCloud, Vol. 9, No. 9, pp. 1–5, 2009.

[42] H. Sato, A. Kanai and S. Tanimoto, ***A cloud trust model in a security aware cloud***, International Symposium on Applications and the Internet (SAINT), IEEE, Vol. 10, pp. 121–124, 2010.

[43] A. Shamir, ***How to Share a Secret***, Communications of the ACM, ACM, Vol. 22, No. 11, pp. 612–613, 1979.

[44] B. Sharma, R. Thulasiram, P. Thulasiraman, S. Garg and R. Buyya, ***Pricing Cloud Compute Commodities: a Novel Financial Economic Model***, 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), IEEE, pp. 451–457, 2012.

[45] A. Sheikh, M. Munro and D. Budgen, ***SSM: Scheduling Security Model for a Cloud Environment***, International Conference on Cloud and Big Data Computing (ICCBDC 2018) ISBN 978-1-4503-6474-4, ACM, Vol. 2, No. 1, pp. 11–15, 2018.

[46] A. Sheikh, M. Munro and D. Budgen, ***Cost and Effect of Using Scheduling Security Model in a Cloud Environment***, International Conference on Cyber Security and Cloud Computing and Edge Computing and Scalable Cloud (CSCloud/EdgeCom), IEEE, pp. 95–101, 2019.

[47] A. Sheikh, M. Munro and D. Budgen, ***Systematic Literature Review (SLR) of Resource Scheduling and Security in Cloud Computing***, International Journal of Advanced Computer Science and Applications (IJCSA), IEEE, Vol. 10, No. 4, pp. 35–44, 2019.

[48] R. Shelke and R. Rajani, ***Dynamic Resource Allocation in Cloud Computing***, International Journal of Engineering Research and Technology, ESRSA, Vol. 2, No. 10, 2013.

[49] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky and D. Shaket, ***Venus: Verification for Untrusted Cloud Storage***, ACM workshop on Cloud Computing Security Workshop, ACM, pp. 19–30, 2010,

[50] PC. Singh, ***Completely fair scheduler***, Linux Journal, ACM, Vol. 1, No. 184, pp. 4, 2009.

[51] S. Sing and I. Chana, ***A survey on resource scheduling in cloud computing: Issues and challenges***, Journal of Grid Computing, Springer, Vol. 14, No. 2, pp. 217–264, 2016.

[52] J. Slegers, I. Mitrani and N. Thomas, ***Static and Dynamic Server Allocation in Systems with on/off Sources***, Annals of Operations Research, Springer, Vol. 170, No. 1, pp. 251–263, 2009.

[53] S. Subashini and V. Kavitha, ***A survey on security issues in service delivery models of cloud computing***, Journal of Network and Computer Applications, Elsevier, Vol. 34, No. 1, pp. 1–11, 2011.

[54] L. Tripathy and R.R. Patra, ***Scheduling in cloud computing***, International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol. 4, No. 5, pp. 21-7, 2014.

[55] Tutorials Point, ***Cloud Computing - Quick Guide***, `http://www.tutorialspoint.com/cloud_computing/cloud_computing_quick_guide.htm`[Access in May 2019].

[56] WT. Tsai, Q. Shao, and W. Li, ***Oic: Ontology-Based Intelligent Customization Framework for SaaS***, International Conference on Service-Oriented Computing and Applications (SOCA), IEEE, pp. 1–8, 2010.

[57] M. Vrable, S. Savage, GM. Voelker, ***BlueSky: a Cloud-Backed File System for the Enterprise***, USENIX Conference on File and Storage Technologies, USENIX Association, pp. 19–19, 2012.

[58] J. Viega, ***Cloud computing and the common man***, Institute of Electrical and Electronics Engineers, Computer, Vol. 42, No. 8, pp. 106–108, 2009.

[59] WE. Walsh, G. Tesauro, JO. Kephart, and R. Das, ***Utility Functions in Autonomic Systems***, International Conference of Autonomic Computing, IEEE , pp. 70–77, 2004.

[60] P. Watson, ***A multi-level security model for partitioning workflows over federated clouds***, Journal of Cloud Computing, Springer, Vol. 1, No. 1, pp. 1-15, 2012.

[61] W. Wongthai, F. Rocha, and A. Van Moorsel, ***Logging Solutions to Mitigate Risks Associated with Threats in Infrastructure as a Service Cloud***, International Conference on Cloud Computing and Big Data (CloudCom-Asia), IEEE, pp. 163–170, 2013.

[62] Z. Xin, L. Song-qing and L. Nai-wen, ***Research on cloud computing data security model based on multi-dimension***, International Symposium on Information Technology in Medicine and Education (ITME), IEEE, Vol. 2, pp. 897–900, 2012.

[63] K. Yang G, W. Yu, P. ByoungSeob and C. Hyo Hyun, ***A Heuristic Resource Scheduling Scheme in Time-Constrained Networks***, Computers & Electrical Engineering, Elsevier, Vol. 54, pp. 1–15, 2016.

[64] YO. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti and Y. Coady, ***Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis***, International Conference on Cloud Computing, IEEE, Vol. 3, pp. 91–98, 2010.

[65] G. Zhao, C. Rong, MG. Jaatun and FE. Sandnes, ***Deployment models: Towards eliminating security concerns from cloud computing***, International Conference on High Performance Computing and Simulation (HPCS), IEEE, pp. 189–195, 2010.