Electronic Thesis and Dissertation Repository

11-21-2011 12:00 AM

# Trust-Based Service Selection

Zainab Aljazzaf, *The University of Western Ontario*

Supervisor: Mark Perry, *The University of Western Ontario*
Joint Supervisor: Miriam Capretz, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science
© Zainab Aljazzaf 2011

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the OS and Networks Commons

# Trust-Based Service Selection

by

## Zainab Mohammed Aljazzaf

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada
December, 2011

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

**CERTIFICATE OF EXAMINATION**

Supervisors

_____
Dr. Mark Perry

_____
Dr. Miriam Capretz

Examiners

_____
Dr. Qusay H. Mahmoud

_____
Dr. Abdallah Shami

_____
Dr. Michael A. Bauer

_____
Dr. Michael Katchabaw

The thesis by

**Zainab Mohammed Aljazzaf**

entitled

**TRUST-BASED SERVICE SELECTION**

is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

_____
Date

_____
Chair of the Thesis Examination Board

# Abstract

Service Oriented Architecture (SOA) is an architectural style that builds enterprise solutions based on services. In SOA, the lack of trust between different parties affects the adoption of such architecture. Trust is as significant a factor for successful online interactions as it is in real life communities, and consequently, it is an important factor that is used as a criterion for service selection. In the context of online services and SOA, the literature review shows that trust is not mature and does not reflect trust as it is in the online world in terms of trust definition and the consideration of the essentials trust aspects that reflects the nature of trust.

This thesis proposes a trust-based service selection solution, which requires establishing trust for services and supporting service selection based on trust. This work considers building trust for service providers besides rating services, an area that is neglected in the services trust literature. This work follows progressive steps to arrive at a solution. First, this work develops a trust definition and identifies trust principles, which cover different aspects of trust. Next, SOA is extended to build a trust-based SOA that supports trust-based service selection. In particular, a new component, the trust mediator, which is responsible for trust establishment is added to the architecture. Accordingly, a trust mediator framework is built according to the trust definition and principles to identify its main components. Subsequently, this work identifies the trust information, or metrics, for services and service providers. Accordingly, trust models are built to evaluate trust rates for the applicable metrics, services, and service providers.

Moreover, this work addresses the trust bootstrapping challenge in the literature. The proposed trust bootstrapping approach addresses different challenges in the literature such as whitewashing and cold start challenges. This approach is implemented through experiments, evaluations, and scenarios.

# Keywords

# Acknowledgements

*All Praise and Thanks to my Lord, ALLAH ..*

I would like to thank Kuwait University for the financial support throughout my study as well as the University of Western Ontario for the provision of facilities for completing my research.

I would like to express my sincere gratitude to my supervisors, Prof. Mark Perry and Dr. Miriam Capretz, for their great support during my studies. Their guidance, assistance, encouragement, and friendship have been instrumental in my fulfilment of this work.

I give great thanks to my husband, Abbas Khajah, his mother, Khair Al-Nisa Khajah, my children, Albatool, Athraa, Ahmed, and Hamza, my parents, Mohammed and Masooma, and my family for their great patience, understanding, prayer, and support. They have played an integral part in making this work complete and successful.

Finally, I greatly appreciate my colleagues, and give special thanks to David Allison, Raphael Mafita Bahati, Chern Har Yew, Diego Garcia, Vinson Wang, and Rizwan Tejpar, as well as the employees in the university, especially Maaike Froelich from the Student Development Centre and the staff in the Writing Support Centre and Teaching Support Centre.

"Whoever **Trusts** in Allah, will find Him sufficient.

Verily, Allah will accomplish his purpose."

[The Holy Quran, Surah Al Talaq 65:3]

# Abbreviations

BPEL            Business Process Execution Language
CA              Certificate Authority
OASIS           Organization for the Advancement of Structured Information Standards
OSTM            Objective Service Trust Metric
P2P             Peer to Peer network
PTM             Provider Trust Metric
QoBiz           Quality of Business
QoE             Quality of Experience
QoS             Quality of Service
SLA             Service Level Agreement
SOA             Service Oriented Architecture
SOAP            Simple Object Access Protocol
SOC             Service Oriented Computing
SSTM            Subjective Service Trust Metric
ST              Security Token
STM             Service Trust Metric
STS             Security Token Service
$T_{OSTM}$      Trust rate of Objective Service Trust Metric
$T_{pr}$        Trust rate of service provider
$T_{PTM}$       Trust rate of Provider Trust Metric
$T_s$           Trust rate of service
$T_{SSTM}$      Trust rate of Subjective Service Trust Metric
$T_{STM}$       Trust rate of Service Trust Metric
$T_{TM}$        Trust rate of Trust Metric
TM              Trust Metric
ToTEF           Total Trust Evaluator Framework
UDDI            Universal Description, Discovery, and Integration
W3C             World Wide Web Consortium
WS              Web Service
WSDL            Web Services Description Language
XML             Extensible Markup Language

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

*"It is impossible to go through life without trust."* Graham Greene

In human communities, there is uncertainty about the behaviour of strangers. Since people who do not trust others avoid interacting with them, trust plays a significant role in facilitating interactions in such uncertain environments. Trust is a property of relationships rather than individuals [96], and accordingly, it is usually specified as a relationship between a trustor and a trustee. A *trustor* is a subject that trusts a target entity known as a *trustee* [88]. In this case, an entity can include a person, store, bank, service, or product, and it can be identified by its properties, such as name, identification, picture, signature, store location, and stated policies [22, 80]. An entity's decision to interact with others represents an act of trust. In this case, the trustor relies on and places its trust in the trustee to accomplish the task as agreed upon [44]. Figure 1.1 depicts the trust relationship between the trustor and the trustee.

In the online world, trust affects the adoption of the Internet, which will increase e-commerce interactions and lead to economic growth [36]. For example, if customers trust a business and its services, they purchase additional products and services, and greater amount of trust leads to an increase in transactions. Accordingly, trust is an important feature for decision making between different entities on the Internet, as it is necessary for determining which services, markets, and sellers to select. A trust system can establish trust for entities, build reputations, and

1
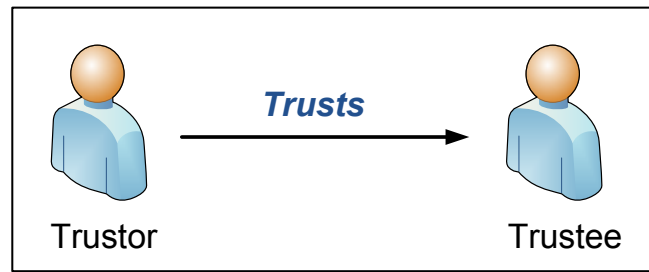
*Figure 1.1: Trust Relationship.*

perform trust management. Entities, such as sellers, buyers, items, and services in such systems are rated on the basis of their past interactions, which creates future expectations for the entities.

In the real or offline world, trust can be built based on obvious evidence, such as names and physical presences. However, trust establishment on the Internet contains additional issues. In such an online environment, entities are separated by physical distance and are likely to be complete strangers. Some entities on the Internet use real names and have physical location. However, this is not always the case, as entities are generally not physically identified, and there are many anonymous entities. Additionally, as in the offline world, the Internet is a diverse system of different domains, and each domain has different requirements. Hence, a domain's requirements need to be considered when establishing trust for entities joining that domain. For example, one domain may require an entity that can be trusted based on security criteria and another domain may require privacy specifications. Therefore, a trust solution needs to consider the diverse requirements of online organisations.

The development of a distributed software system requires the interaction of entities and the use of resources from diverse organisations throughout the Web. In such systems, various entities among different domains and organizations surpass the boundaries of a community, which has clear security and trust preferences. Service Oriented Computing (SOC) is "a computing paradigm that utilizes services as fundamental elements to support rapid, low-cost development of distributed applications in heterogeneous environments" [75]. Specifically, a distributed application may be composed of global services provided by different organizations and having different properties. In this environment, the development of trust challenging.

2

In order to realize the potential of SOC, Service Oriented Architecture (SOA) should be developed to overcome many enterprise challenges, including designing complex distributed services, managing business processes, ensuring transaction Quality of Service (QoS), complying with agreements, and leveraging different computing devices such as personal computers and cell phones [75].

Services around the Web are diverse with different properties. For example, one service may include response time, throughput, and privacy as its main properties and another service may consider reliability and security as its major properties. However, the identification of the development of trust information and trust for such diverse entities is more challenging. In addition, some services and service providers are malevolent in providing poor services or intentionally offering services that are not consistent with their promises [38]. Thus, it is necessary to determine the trustworthiness of services and the selection of a trustworthy service.

To build a service oriented application, the application developer, or service requestor, can select services from different providers on the Internet. Since there are many services with similar functionalities, service requestors need to differentiate between them. The only differentiating factor between similar services may be their non-functional properties, which can be considered as criteria for service selection. As a non-functional property, trust has been used as a criterion for service selection [21, 95, 37, 41]. Accordingly, Figure 1.2 shows the trust relationship between a service requestor and a service or service provider. A service requestor, or trustor, may select a service from a service provider, the trustees, based on their trustworthiness. Thus, trust can help requestors in their service selection decision, and it is a less expensive approach for service selection than monitoring or Service Level Agreements (SLA) [94].

The rest of this chapter is organized as follows: Section 1.1 discusses the thesis motivation and Section 1.2 presents the dissertation outline.

## 1.1 Motivation

Trust has been studied extensively in many areas such as e-commerce [32, 6, 18, 95, 53, 79] and Peer-to-Peer (P2P) environments [85, 26, 93, 92, 97]. However, trust research in services
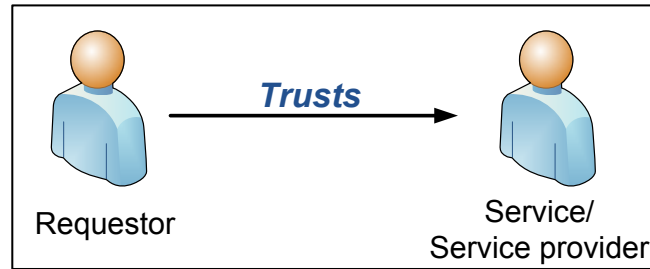
*Figure 1.2: Trust Relationship between a Service Requestor and a Service/Service Provider.*

and SOA is still relatively new and not mature [94, 39, 91]. Hence, there is a need for a solid solution to establish trust for services and select services based on trust criterion. The trust literature for online services and SOA does not reflect trust as it is in the online world in terms of defining trust and considering the essential trust aspects. In addition, there are different trust challenges that need to be addressed. The rest of this section identifies the gaps in the literature that motivate this work.

***What does trust mean?*** Trust is a complex subjective term. Defining trust has been thoroughly studied in the offline and online worlds [61, 42, 32, 17, 12]. There are dozens of trust definitions, mostly distinct while others overlap, and mostly related to a specific domain requirement such as security. However, defining trust in the services area is still new. The majority of studies of trust in Web Services and SOA do not define trust. Some studies refer to others definitions, do not define trust, or attempt to define trust based on a number of limited existing definitions. Moreover, some studies that address trust, define it as QoS, conduct QoS rating and scoring, or use a reputation approach. However, trust, QoS, and reputation are different terms. Reputation is "what is generally said or believed about a person's or thing's character or standing" [39]. There is a need to view trust in a more generic way and build a mature *'standardized definition of trust'*, which suits different domains and would allow researchers to compare their work and make progress in the field.

***What are the online trust principles?*** Trust has many aspects and requirements that reflect its core nature. In the online trust literature, researchers consider a limited number of trust aspects in their studies. Hence, there is a need to identify a list of trust requirements, or *'trust principles'* to establish trust. These principles incorporate trust aspects and identify the require-

4

ments for establishing a comprehensive and concrete solution for trust. Daignault et al. [18] defined several trust principles. However, the list is not complete or mature and it needs to be extended to add other important aspects and requirements.

***What information should be used to build trust?*** Trust is based on information [18, 94]. In the offline world, traditional forms of communication allow people to assess a wider range of cues related to trustworthiness than is currently possible through online communication. In fact, the Internet provides little evidence for the solidity of the entity behind it. As a result, it is challenging to find sufficient online substitutes for the traditional trust cues that are obvious in the physical world and to identify new information elements that are appropriate for deriving measures of trust [39].

The development of trust for an entity is based on the information provided by that entity. For example, a service can present its reliability as trust information and requestors will establish trust for the service based on its reliability. In the literature, there is no clear identification of trust information, or Trust Metrics, which should be considered for building trust for services and providers.

In SOA, a transaction may span a range of domains and organizations. In particular, services and service providers may encompass many domains with different properties and requirements, and services may have different functional and non-functional properties. Furthermore, a service requestor has many requirements and each requestor seeks for different functional and non-functional properties. Thus, the identification of trust metrics for such environment is a challenge. Some studies attempt to overcome this problem by defining a notion of community [52, 10] or addressing trust in specific domains [21, 44, 100]. Accordingly to the literature, a community is a container or a domain that groups related Web Services with specific areas of interest, such as auto makers or car dealers [52]. Communities may not be suitable in an SOA environment, which forces enterprises to move from private to public networks. Hence, a new approach is required for such environment that is not community-based.

Dragoni [21] states that relying on the existence of communities might leave consumers in a vulnerable position if they do not belong to any community or if the community does not provide a significant rating system. Figure 1.3 provides an example of building a composition of services using SOA. In this case, services are selected from different providers, communities,
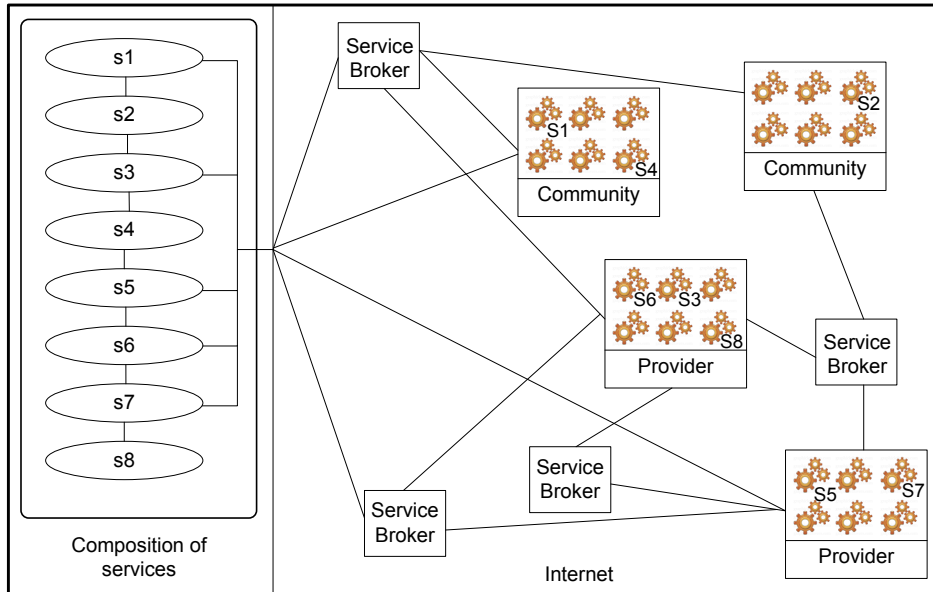
*Figure 1.3: SOA Environment.*

or service brokers, and thus from diverse domains. A requestor or an application should be able to select services in this diverse environment based on its trust preferences. In addition, services should suit this diverse domain and support different services' trust metrics, to satisfy the requirements of different domains and trust preferences.

Since there is a need to build a trust solution that is *'not community-based'* in order to suit a diverse environment such as SOA, a domain may need to support a range of trust metrics, which requires the identification of *'a unified trust metrics'* for an open environment.

QoS is considered as a non-functional property of a service and as a criterion for service selection and composition. Although QoS parameters are included in SLAs, there is no measure specifying how often the service has accurately delivered the agreed upon levels of QoSs. This absent measure involves the historical trustworthiness of the services [41]. Several researchers stated or used QoS as information for rating services [46, 58, 94, 41, 90, 57, 38]. However, QoS is insufficient as a criteria for building trust because some services are based on other information rather than just QoS. For example, trust may be built based on privacy as a criterion, which is not identified as a QoS in the literature [78, 48, 46, 98, 83].

6

***Is it effective to consider building trust for service providers?*** The rating of service providers is neglected in the services trust literature [94]. However, ratings of a service and its provider are related and interdependent. If a service provider offers highly rated services, its rating will also be high. Thus, the trustworthiness of a service provider can enhance the requestor's trust in its services [94], and a requestor can select a service from the most trustworthy providers [58].

It is important to consider and establish *'trust for service providers'* and to select a service based on the trustworthiness of the service and its provider. The rating of providers encourages them to behave honestly, increases the business opportunities of trustworthy providers, encourages competition between providers, influences the economic growth of the providers, increases the usage of the Internet technologies such as e-markets, and evolves online commerce.

***How to collect trust metrics and evaluate trust rates?*** In order to rate services and service providers, *'trust models'* need to be built. Services and service providers are rated based on their trust metrics. Therefore, trust models to rate trust metrics need to be built and different trust metrics require different trust models.

***How to extend SOA to support trust?*** It is essential to extend SOA to support trust and build *'a trust-based SOA'*, which supports the selection of services based on their trustworthiness. A trust mediator that is responsible to conduct the trust process, must be added to the trust-based SOA. Hence, it is important to build the trust mediator framework and identify its main components, which are founded on the trust definition and trust principles. The result is *'a unified trust framework'* that contains the important components for building a concrete trust solution. Specifically, trust framework components perform different tasks that require different techniques.

***Can the trust solution address different trust challenges?*** There are different trust challenges that need to be considered when building a trust solution:

- Trust bootstrapping: A mechanism to rate newcomers to a system that have no rating history [52]. Trust bootstrapping is important for reliable interaction with services and service providers who are new to the system. Most studies assume a system where trust and reputations already exist [57, 46, 49, 58, 59]. However, it is important to initialise trust rates for new services and service providers. Since the development of trust is a

crucial stage in any trust relationship, *trust bootstrapping* is the first step in trust building process. Thus, a trust solution needs to address the bootstrapping challenge.

- Whitewash/Zero-cost identity or Changing identity: In order to receive better ratings, an entity will change its identity if there has been a significant loss of reputation [39, 18, 80, 81, 26]. Whitewashers are entities that leave and rejoin the system with a new identity in an attempt to eradicate the poor reputation they have gathered under their previous identity [85]. Hence, a trust solution should address whitewashing challenge.

- Cold Start: A cold start describes a situation in which there is a lack of initial ratings. Specifically, it occurs when a new entity registers in the system and has no rating record [84]. A trust solution that addresses trust bootstrapping should also address the cold start challenge.

This thesis contributes to the literature by addressing the previous questions. In particular, it aims to provide a roadmap that builds a concrete solution of trust for services and service providers and a trust-based system for service selection. Specifically, it establishes trust for services and service providers and supports service selection based on trust. First, it will build a **trust definition**, identify **trust principles**, consider **rating service providers**, extend SOA and build a **trust-based SOA** to support trust-based service selection by adding a new component, the trust mediator, and additional link interactions. Subsequently, this work will build the **trust mediator framework** based on the trust definition and trust principles to identify its main components, identify the **trust metrics** for services and service providers, build **trust models** for the trust metrics, services, and service providers to evaluate their trust rates, and address different **trust challenges** in the literature, which include whitewashing, cold start, and trust bootstrapping. Since **trust bootstrapping** is the first step in any trust establishment process, it is the main trust challenge that this thesis address. However, the proposed technique for trust bootstrapping also addresses the whitewashing and cold start challenges. Finally, the trust solution needs to be implemented and evaluated.

## 1.2 Dissertation Outline

The rest of the dissertation is organized as follows:

- Chapter 2 presents the background on services, SOA, and trust. Section 2.1 presents the background on services, and Section 2.2 covers Service Oriented Architecture. Subsequently, Section 2.3 provides trust-based service selection approaches, and Section 2.4 discusses trust aspects. Reputation computation engines are examined in Section 2.5, and commercial and live reputation systems are discussed in Section 2.6. Section 2.7 discusses trust and security, and WS-Trust is presented in Section 2.8. Finally, Section 2.9 summarizes the chapter.

- Chapter 3 surveys the related work about trust. The chapter covers the trust definition in Section 3.1, trust information in Section 3.2, and trusting service providers in Section 3.3. Section 3.4 provides SOA extension, and Section 3.5 demonstrate community-based systems. Trust models are examined in Section 3.6, and trust bootstrapping is discussed in Section 3.7. Finally, Section 3.8 summarizes the chapter.

- Chapter 4 presents the proposed trust definition, trust principles, as well as the trust architecture and framework. Section 4.1 presents the trust definition and principles, while Section 4.2 introduces the trust-based SOA. Section 4.3 presents ToTEF, the trust mediator framework, and Section 4.4 summarizes the chapter.

- Chapter 5 discusses the trust metrics; specifically, Section 5.1 presents the trust metrics, and Section 5.2 discusses the trust metrics publication approaches. Section 5.3 summarizes the chapter.

- Chapter 6 presents the trust models for trust metrics. Section 6.1 presents trust rating scales. While Section 6.2 provides trust models for objective service trust metrics, Section 6.3 presents trust models for subjective service trust metrics. Trust models for service provider trust metrics is discussed in Section 6.4. lastly, Section 6.5 summarizes the chapter.

- Chapter 7 examines the trust models for services and service providers. Section 7.1 provides the service trust model, and Section 7.2 presents the service provider trust model. The trust matching model is discussed in Section 7.3, and Section 7.4 summarizes the chapter.

- Chapter 8 presents empirical studies that include the implementation, experiment, and evaluation of the trust bootstrapping solution. Section 8.1 examines the trust-based SOA prototype, and Section 8.2 presents the implementation of the prototype. The experiment is presented in Section 8.3. Section 8.4 presents the evaluation. Section 8.5 summarizes the chapter.

- Chapter 9 discusses the conclusion and future work. The chapter presents the contribution and discussion in Section 9.1, which includes the trust solution and trust principles in Section 9.1.1, and the trust solution and trust challenges in Section 9.1.2. Future work is presented in Section 9.2.

# Chapter 2

# Background

*"Trust yourself. You know more than you think you do."* Benjamin Spock

This chapter presents the background on services, SOA, and trust. Section 2.1 presents the background on services, and Section 2.2 covers Service Oriented Architecture. Subsequently, Section 2.3 provides trust-based service selection approaches, and Section 2.4 discusses trust aspects. Reputation computation engines are examined in Section 2.5, and commercial and live reputation systems are discussed in Section 2.6. Section 2.7 discusses trust and security, and WS-Trust is presented in Section 2.8. Finally, Section 2.9 summarizes the chapter.

## 2.1   Services

A service is "a discrete unit of business functionality that is made available through a service contract" [82]. The service contract includes a service interface, interface documents, service policies, QoS, and performance. The service interface defines the details of the interaction with a service, specifying the service operations, parameters, and protocols. However, a service interface is distinct from its implementation. For instance, a consumer perceive only the interface, or function, rather than how it is implemented. Therefore, the producers of services can alter the implementation of services as long as they do not change the interfaces or the behaviours of the services.

Services have many properties, as they are loosely-coupled, self-contained, platform-independent, dynamically discoverable, invokable, and composable [75]. Coupling refers to the extent of dependency between a consumer and a service. Hence, loosely coupled services have few dependencies, while tightly coupled services have more dependencies. A more tightly coupled system indicates that changes to one service will affect other services, and thus, the service is less flexible [82]. A self-contained service is one that maintains its own state independent of the application utilizing it. Furthermore, services are platform-independent, where requestors can invoke them using any hardware or software [75].

Services can be implemented using Web Service technology. Web Services are an emerging technology that enables applications running from different machines over the Web to integrate and exchange data regardless of their platform, hardware, operating system, and languages [74]. Moreover, Web Services are based on common standards, such as Extensible Markup Language (XML), and existing technologies, such as Hypertext Transfer Protocol (HTTP).

A Web Service is defined by the World Wide Web Consortium (W3C) as "a software application identified by a URI, whose interface and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols" [27]. Similarly, IBM defines a Web Service as "self-contained, self-described, dynamically discovered applications with Internet-based interfaces" [99]. Accordingly, a Web Service is a new breed of Web applications, which is modular and can be published, located, and invoked across the Web. Once a Web Service is deployed, other applications can discover and invoke it. Web Services are loosely coupled, platform-neutral, reusable, and distributed software components [99]. Papazoglou [74] defines a Web Service as "a platform-independent, loosely coupled, self-contained, programmable Web-enabled application that can be described, published, discovered, coordinated, and configured using XML artefacts (open standards) for the purpose of developing distributed interoperable applications".

Web Services perform functions from simple requests to complicated business processes. Web Services' main protocols include Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) [74]. The key to Web Services' success is the open standards that facilitate the interoperability
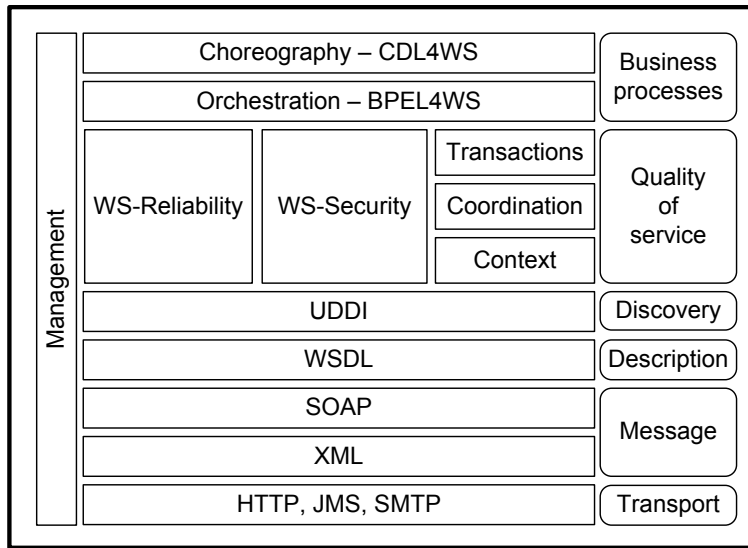
*Figure 2.1: Web Services Technology Stack [74].*

among different parties [99]. Additionally, there are many other Web Services standards, as shown in Figure 2.1 [74].

SOAP is an XML-based standard messaging protocol, using HTTP as a means of transport and circumventing the firewalls. WSDL is the service representation language used to describe the interface of and access to Web Services. This description includes the operations and parameters, location, and invocation protocol of the Web Services.

UDDI is a cross-industry directory standard for describing, publishing, and discovering of Web Services. This standard stores the Web Service interfaces described by WSDL, categorizes Web Service information, and allows searching the directory for Web Services. There are other alternatives to UDDI, such as electronic business XML (ebXML) and Directory Services Markup Language (DSML). Furthermore, Business Process Execution Language (BPEL) is an orchestration language for building a composition of Web Services [74].

Hoyle [34, 35] defines quality as "the degree to which a set of inherent characteristics fulfils a need or expectation that is stated, general implied or obligatory". The author elaborates that "quality is thought of as conformance to specification regardless of whether the specification actually meets the needs of the customer or society" [34]. The World Wide Web Consortium

13

(W3C) [48] describes the QoS requirements for Web Services as "the quality aspect of a Web Service". Since QoS is an important factor in SOC paradigms, many approaches have been proposed to examine QoS compliance by monitoring or collecting quality ratings from the users. QoS has been used as a non-functional property for selecting services and for establishing trust for services [76, 58, 21, 95, 37, 41, 57, 46, 49, 58].

## 2.2 Service Oriented Architecture

Software architecture is "a description of a software system in terms of its major components, their relationships, and the information that passes among them" [82]. Traditional software architecture is focused on building software applications. However, SOA is focused on the construction of cross-organizational enterprise applications that are based on the interactions between consumers or business processes with needs and services or service providers with aptitude [82].

The Organization for the Advancement of Structured Information Standards (OASIS) defines SOA as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" [56]. More specifically, SOA is "a framework for integrating business processes and supporting IT infrastructure as secure, standardized components - sevices - that can be reused and combined to address changing business priorities" [11], and it is "an architectural style for building enterprise solutions based on services" [82], which may be implemented within a single organization. However, most large organizations build an enterprise application of composable services, which are disseminated across the Internet. Therefore, SOA is responsible for creating the environment necessary to build and use composable services across the enterprise [82].

Because SOA is concerned with an enterprise scope beyond a single application [82], the design principles of SOA are independent of any technology [75]. SOA can be implemented using many distributed computing technologies such as Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), and Web Services. In particular, Web Services have gained more popularity as a technology for implementing SOA

14
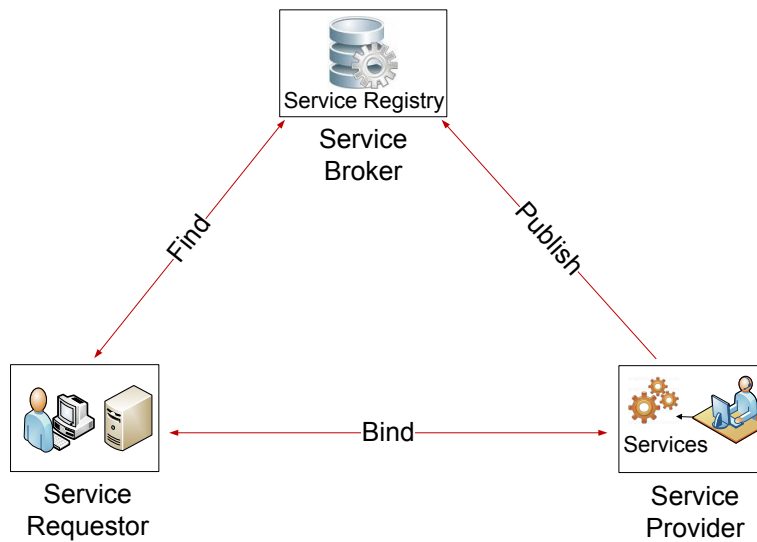
*Figure 2.2: Service Oriented Architecture.*

because of their important features, especially their interoperability and self description interfaces, as well as the fact that they base their development on existing ubiquitous infrastructures, such as HTTP and XML [75].

Four aspects are necessary for the successful development and deployment of a distributed SOA. With *service enablement*, each application is exposed as a service, while *service orchestration* orchestrates services in specified processes. Furthermore, *deployment* transmits services and processes from testing to the production environment and *management* ensures that services are monitored and that the service invocation and selection is adjusted to meet the application goals.

Figure 2.2 depicts SOA, illustrating the relationship between SOA roles and operations [74]. There are three interaction roles in SOA: the *service provider* is an organization or platform that owns, implements, and controls access to the services. Furthermore, a *service requestor* is an application, service, or client who is searching and invoking a service, and a *service broker* that groups all of the services together and maintains a registry of available services [75]. A service registry is a searchable directory where the description of services is published by the providers and searched by the requestors [74].

Moreover, there are three operations within SOA [74]: in the *publish operation*, service providers publish their services into the registry. This operation consists of two functions: describing the services in the WSDL file and registering the services by storing the service description in a categorized service registry for requestor access. With *find operation*, requestors search and find services from the service registry. It consists of two functions: discovering services and selecting the desired service from the result. Finally, in *bind operation*, requestors invoke services at run time using the technical information provided in the WSDL file to bind to the services. The invocation may occur directly between the requestors and services or it may occur through the service broker.

## 2.3 Trust-Based Service Selection Approaches

Figure 2.3 [21] presents the trust-based service selection approaches. In the *direct experience* approach, requestors build trust for services after utilizing them. However, since there is a need to trust services before executing them, the *Trusted Third Party (TTP)* approach enables consumers to consult a trusted third party for determining the trustworthiness of services. This approach relies on the underlying assumption that consumers trust the TTP they decided to consult. Specifically, there are two TTP approaches: social and matchmaking. In the *social TTP* approach, consumers evaluate the performance of services they have used, and the TTP gathers the evaluations and computes trust rates for the services. Accordingly, a consumer considering the use of a service consults the TTP for its trustworthiness. The social trust approach is further subdivided into three categories: reputation, recommendation, and referrals. *Reputation* is a public opinion about the characteristics of an entity; it represents a collective evaluation of that entity [94]. *Recommendation* systems [81][54] aggregate recommendations and match recommenders with others that search for recommendations. Finally, a *Referral* approach [96] is a decentralized approach based on software agents and communities, where agents can assist each other to find services by giving referrals to services that were useful for them. Therefore, each agent builds the trust of other agents based on the perceived quality of the services and the referrals that guide them to the services. In the *matchmaking TTP* approach [72, 28], a service description is matched with a requestor's request and trust preferences. The *hybrid* approach
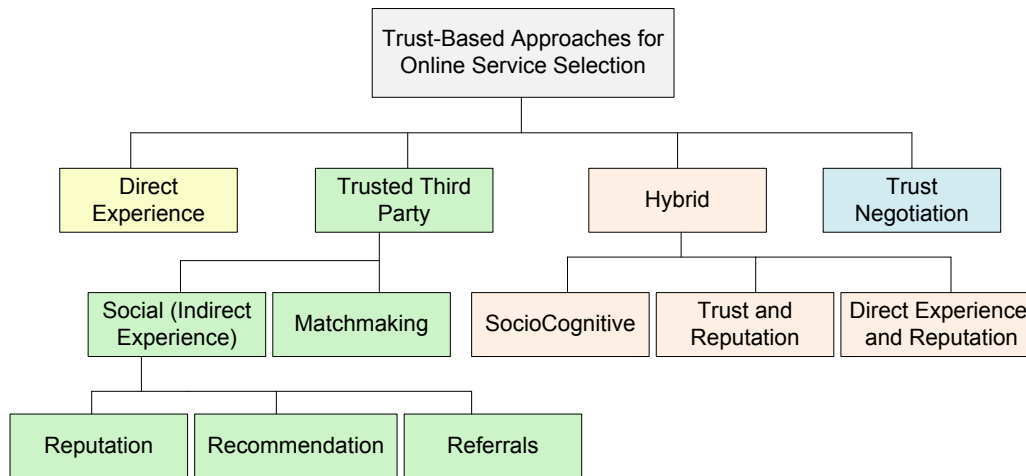
*Figure 2.3: Approaches for Trust-Based Service Selection [21].*

[38, 103] involves the combination of different approaches, aiming to improve the weaknesses of some approaches by combining with other approaches. The *automated trust negotiation* approach [21] builds a mutual trust between service requestors and service providers. In this approach, trust is assessed in two directions: a requestor trusts a service and the service trusts the requestor. The trust negotiation approach depends on the disclosure of digital credentials between the two parties.

In order to predict the trustworthiness of entities, Massa and Avesani [55] distinguish between two trust ratings: local and global ratings. Global ratings provide a unique trust ranking independently of the entity who performed the evaluation and define how the community as a whole trusts entities. In this system, an entity has a unique rating that is viewed by all other entities. In contrast, local ratings depend on the user performing the evaluation, and hence, it provides personalized ratings.

Accordingly, trust and reputation can be distinguished on the basis that trust is a local personalized rating whereas reputation is global rating. Trust is subjective and context-specific. The concept of subjectivity indicates that an entity's trust varies among different requestors, and context-specific indicates that trust is different in various situations. Massa and Avesani [55] argue that there should be a way to personalize trust ratings since global conformity about

17
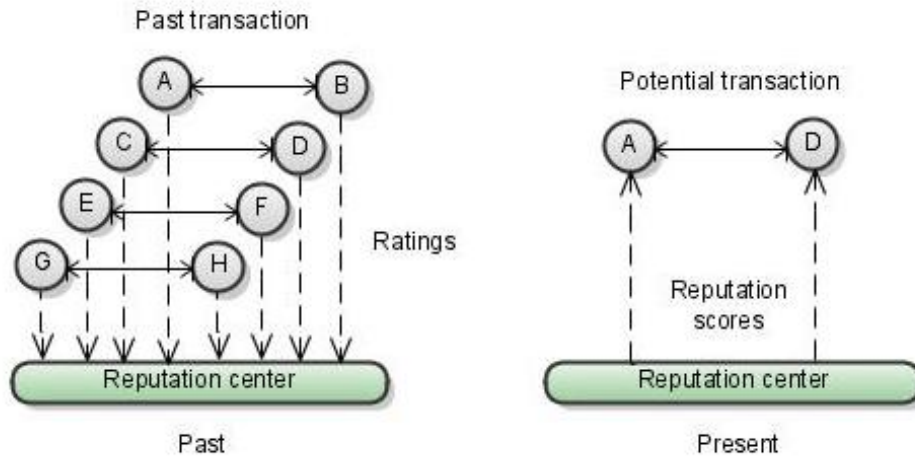
*Figure 2.4: Reputation System Architecture [39].*

user trustworthiness does not exist. Therefore, each user can build different trust rates for different entities.

In reputation systems, requestors provide their feedback about services they have consumed. Figure 2.4 [39] presents a reputation system architecture that provides global ratings. In this system, entities evaluate each other and provide their feedback (ratings) to the reputation center, which acts as a TTP. The TTP gathers the feedback and computes the trust ratings for the entities. Subsequently, it stores the ratings and provides them globally. When an entity needs to interact with another entity, it consults the TTP about the entity's trustworthiness. For example, Entity A, which wants to interact with Entity D for the first time, consults the TTP. The TTP then will return the global rating or reputation of D to A. Thus, A trusts D based on the opinions of other entities. Figure 2.5 presents an example of personalized local ratings [55]. In this figure, Entity B has two ratings from the perspective of Entities, A and D. A has rated B as 0.9, and D has rated B as 0.3.

## 2.4  Trust Aspects

This section presents a number of online trust aspects.
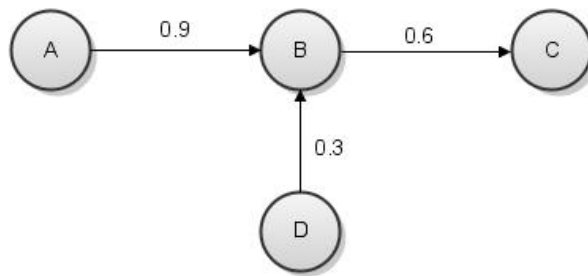
*Figure 2.5: Trust Network.*

**Trust development phases**

Trust is a dynamic concept that can be divided into three development phases [42]: trust building, trust stabilizing , and trust dissolution.

1. Trust building: In the trust building phase, trust is formed and initial trust formation is important. For example, an entity may build trust based on cognition rather than personal interactions. In cognition-based trust, the trustor relies on first impressions, such as vendor size, privacy, security, and convenience use of Web site. In this first development stage, online trust can also be based on rational calculation of possible costs and benefits, which is known as calculus-based trust.

2. Trust stabilizing : Trust stabilizing is where trust already exists. Since trust evolves over time, it is based on the interactions and observations between parties. In this phase, trust is based on the trustors' knowledge of trustees from past interactions, which is known as knowledge or experience-based trust. In identification-based trust, which represents a highest level of trust, trust is formed by joined values, tasks, and goals by creating a collective identity, such as a common team name.

3. Trust dissolution : Trust dissolution occurs when trust ends. Since trust can be declined, it is important to study the situations where trust declines and rebuilds after a decline.

**Properties of trust relationship**

There are a number of trust relationship properties [94, 33]. First of all, trust *can be transitive*; for example, if Alice trusts Bob, and Bob trusts Sam, then Alice can trust Sam. Furthermore, trust is *not symmetric*; for instance, even if Alice trusts Sam, this does not necessarily mean that Sam trusts Alice. Trust is *context-specific*; for example, someone may trust a person based on a specific context, such as in his/her role as a doctor, but not in another context, as a chef. Similarly, trust is *not absolute*, as a trustor trusts a trustee with respect to its ability to perform a specific action within a specific context; and it is *multi-faceted*, since differentiated trusts need to be developed for different aspects or properties of an entity. For example, a service may have different QoS properties, such as response time and scalability, and trust should be built for each QoS property. Hence, the overall trust depends on the combination of trusts for each property. Trust is *dynamic*; as it may decrease, increase, become less important or irrelevant, or decay with time. Trust relations can be one-to-one, between a trustor and a trustee, one-to-many, between a trustor and a group of entities, many-to-one, such as between the members of a department and the manager, and many-to-many, such as the mutual trust between members of a group [33].

**Principles of trust online**

Daignault et al. [18] identified ten principles surrounding trust online:

1. Trust depends on identity: The history or past experience of interactions are built and mapped to an identity. Identities allow a party's rating from the past to be associated with the party in the future [39]. While in the offline world, identities can be established by visual recognition or identification, in the online world, an identity may be established by using authentication and token [43].

2. Trust is based on information: To trust an entity, one must "get to know them" [18]. Information has many dimensions and each entity establishes its own information dimensions to build an information model. For example, many commercial sites use information such as performance, security, and privacy to help establish their trust.

3. Trust is the function of perception of risk: Trust and risk are related, since "risk is the core of trust in that trust is the degree to which a trustor holds a positive attitude toward the trustee's goodwill and reliability in a risky exchange situation" [18]. However, trust is not only about experiences with others, but it is also about the expectation of a trustee's behaviour in risky circumstances and the extent of their commitment to the rules.

4. Trust deepens over time and with increased reciprocity: "Trust is earned by meeting expectations" [18]. For example, as a consumer makes more transactions and has more experiences with the market, which has acted upon their obligation, their confidence grows with time and they will have high market expectations in the future. Even when the market initially pays a high cost for acquiring customers, the market will receive the customer's trust in subsequent years.

5. Trust is a matter of degree: Trust varies among individuals, organizations, and situations. For example, a customer may trust an organization more or less than its products. Thus, customers have to specify the degree level of trust information for different entities.

6. Culture affects trust: The Web is an open system that spans different countries, regulations, and cultures, which impact the trust-building process. For example, people from different cultures may respond differently to risk. Furthermore, the importance of trust information varies, as individuals in different cultures trust entities differently.

7. Third-party ratings are important in developing trust: An expert opinion from an authorized trusted third party, or TTP, is important. TTPs help users to obtain ratings of different entities without the user having direct experience with that entity.

8. Second-party opinions are important in developing trust: Trust feedback from some parties, such as friends can enhance trust between two entities.

9. First-party information is important in developing trust: First parties should provide their information to develop their trust. For example, an online business, a first party, should clearly specify their information, such as payment methods, insurance, delivery methods, product description, and pricing in their services, policies, and product so that the consumer is aware of their information.

10. Formal and social controls are important in developing trust: Formal controls employ rules to specify behaviours and penalties. On the other hand, social controls use organizational and cultural rules to encourage the desired behaviour.

**Trust classes**

Different classes of trust can be distinguished in the literature relating to Internet services [39, 33]. *Provision trust* is where users trust entities and seek protection from malicious entities. Furthermore, *resource access trust* describes trust as a principle for accessing resources, and *delegation trust* is where agents delegate or act on behalf of users. *Identity trust* describes the belief that an entity's identity is as claimed, and *context trust* describes trust in the presence of a system that has the ability to support transactions and perform remedies. Finally, the *certification of trustee* is based on a third party's certification of an entity's trustworthiness, where trust is founded on a criteria relating to a set of certificates, such as the Pretty Good Privacy (PGP) [6], VeriSign [5], and others [31] that are presented by the trustee to the trustor. For example, trustors may trust VeriSign to certify programs that can run on their machine.

**Categories of trust semantics**

Semantic characteristics of trust ratings are important for participants to interpret those measures [39]. Trust semantics can be described in terms of the"specificity-generality" dimension and the "subjectivity-objectivity" dimension. A *specific* measure is based on one characteristic, while a *general* measure represents the average of all characteristics. Furthermore, a *subjective* measure is based on judgement, while an *objective* measure is determined by assessing the entity against formal criteria, such as calculation. In total, there are four categories of trust semantics:

1. Subjective and specific: Focused on specific issues and based on judgement. For example, a survey questionnaire provides answers for questions on a scale of one to five.

2. Subjective and general: Based on the average of all characteristics of an entity. For example, users of eBay's reputation system rate their transactions and products providing an aggregate ratings.

3. Objective and specific: Based on the objective measure of a specific characteristic. For example, a product is tested according to energy consumption or noise and a company is tested according to earnings or profits.

4. Objective and general: Based on a set of characteristics and the derivation of an overall score. For example, the rating of a product is the average of all the product characteristic rates.

## 2.5   Reputation Computation Methods

It is necessary to assign a level of trust or trust rating for an entity, especially since some entities are trusted more than others. Trust values can be discrete or continuous [33]. Grandison and Sloman [33] mentioned that "it is not clear whether this level should be discrete or continuous. If discrete values are used, then a qualitative label such as high, medium or low may be sufficient. Some systems support arithmetic operations on trust recommendations so numeric quantification is more appropriate".

There are many trust and reputation computation methods for deriving scores. This section describes various principles for computing trust and reputation measures [39].

1. Simple summation or average ratings: This is the simplest way of computing scores. The total score can be computed as the total number of positive reputations minus the total number of negative reputations. For instance, this method of calculation is used in the eBay system. Another advanced way of computing trust scores entails averaging all ratings, which is used by Amazon and Epinions. Also additional factors, such as the rater's age and trustworthiness, can be considered. Simple summations or average ratings are advantageous because the trustor can understand the principle behind the score. However, this system is primitive and provides a poor perspective of the trustee's score [39].

2. Bayesian systems [66, 67]: The system computes scores by statistical updates of beta probability density functions (PDF). More specifically, it takes a binary input positive or negative, and computes reputation based on the previous ratings and new ratings. The result is represented as ($\propto$, $\beta$), the amount of positive and negative, respectively, and the probability value of beta PDF. This method is complex and difficult for the average person to understand [39].

3. Discrete trust models [53]: Humans are often able to rate more successfully in discrete verbal measures than in continuous measures. For example, an entity's trustworthiness can be in the high, medium, or low range.

4. Fuzzy models [53, 87]: Trust and reputation systems can be represented as fuzzy concepts; measuring the degree to which the entity can be trustworthy [39].

5. Flow models: In this model, systems evaluate entities' rates by iterative looping or long chains. For example, in Google's pageRank [73], any hyperlink to a Web page increases its pageRank, and any hyperlink from a Web page decreases its pageRank.

A good reputation computation or scoring system should be understandable, as users should understand what each score implies. The scores could be as simple as the sum of the rating, or they can involve more complicated functions, applying optimization techniques and using a neural network. However, complex rating systems will be difficult for the user to understand the implications of the score. [20]

## 2.6   Commercial and Live Reputation Systems

This section describes some popular applications of online reputation systems. All of the systems are centralized systems, and most of them use simple summation or average rating computation methods, while some systems use flow models [39, 80].

- eBay's feedback forum: eBay is an e-market, and hence its feedback forum is a centralized reputation system where sellers and buyers rate each other. It has a three-scale rating

system: positive, neutral, and negative, or 1, 0, and -1, respectively. To avoid repetitions in rating entities, eBay allows rating only after the transaction is completed. The observed ratings on eBay are mostly positive, accounting for 99% of ratings [39]. Also, eBay uses global trust ratings [55].

- Expert sites: There are a number of experts on different expert sites, including AllExperts, Advogato, and AskMe who answer questions in their area of expertise. Individuals who ask questions rate the experts based on the quality of the replies, which include metrics such as timeliness, politeness, knowledgeability, and clarity. For example, in the Allexpert reputation system, an individual rates the experts and provides ratings in the interval [1,10], and most experts receive a rating close to 10. On each site, the final score is the average of all received ratings.

- Product review sites: Reviewers provide information to assist consumers in making better purchase decisions. These reputation systems rate products and reviewers. For instance, Epinions is a review site established in 1999. It rates products, shops, reviews and reviewers, each with different metrics ranging from 1 to 5. In Epinions' Web of Trust, members can choose to trust or block other members. Epinions has a sophisticated reputation system because of its incentive mechanism based on revenue, where reviewers can earn money by helping customers to make or avoid certain purchase decisions. Also, there is a hierarchy of reviewers, which are Advisor, Top Reviewer, and Category Lead, where Category Leads will normally generate more revenue than Advisors and Top Reviewers.

- Amazon: An online store that allows members to rate products, such as books, and companies from 1 to 5. The total rating of a product or company is the average of its total ratings. Both members and non-members can vote on reviews, and Amazon rates the reviewers as top 1000, top 500, top 100, top 50, top 10, or #1 reviewer, which is the best reviewer. Amazon does not provide revenue incentives; however, other external sources, such as book publishers, pay people to review their goods.

- Google's Web page ranking system [73]: The ranking system in Google is a reputation system. The PageRank algorithm in Google achieves the best search result, which depends not only on search keywords on the page but also on the page rank. Specifically,

the page rank represents how many other pages are linked to it. The reputation of any page increases or decreases depending on the number of links referring to it or its referral to other pages. A single link to a page represents a positive rating of that page [39], so that pages are popular based on the number of links they are referenced. A page that is referenced from a popular page is given a lot of weight [20].

The main weakness of such systems is their vulnerability and unreliability. Anybody can sign up to become a member and vote on different products or reviewers. Some examples of dishonest ratings occur when members write many positive reviews to provide a high score or negatively rate products or reviewers to provide low ratings. Therefore, reputation systems should be vigilant and respond to any abuses that emerge from the system.

## 2.7   Trust and Security

In general, security mechanisms provide protection against malicious parties. However, such mechanisms are unable to protect against providers that act deceitfully, in which trust and reputation can provide protection against such threats. Accordingly, security can be subdivided into *hard security* and *soft security* [21]. Hard security involves traditional security mechanisms, such as encryption, authentication, authorization, confidentiality, and integrity [24], while soft security involves social control mechanisms, of which trust and reputation systems are examples [21]. Certification authorities (CA) and identity management systems assure the identity of an entity and provide 'identity trust' [39]. CAs are described as trust provider or identity trust systems [39]. However, users are also interested in provision trust, knowing the reliability of authenticated parties, and the quality of services they provide. Only trust and reputation systems are useful tools for deriving provision trust [39]. Therefore, soft security mechanisms are required to derive trust.

Figure 2.6 [1] shows the Web Services security standards addressed by IBM and Microsoft. These organizations cooperate to define a unified approach for managing message security exchange in a Web Services environment. As the figure shows, the proposed road map addresses many security concerns in Web Services. The foundational standard is WS-Security, which is
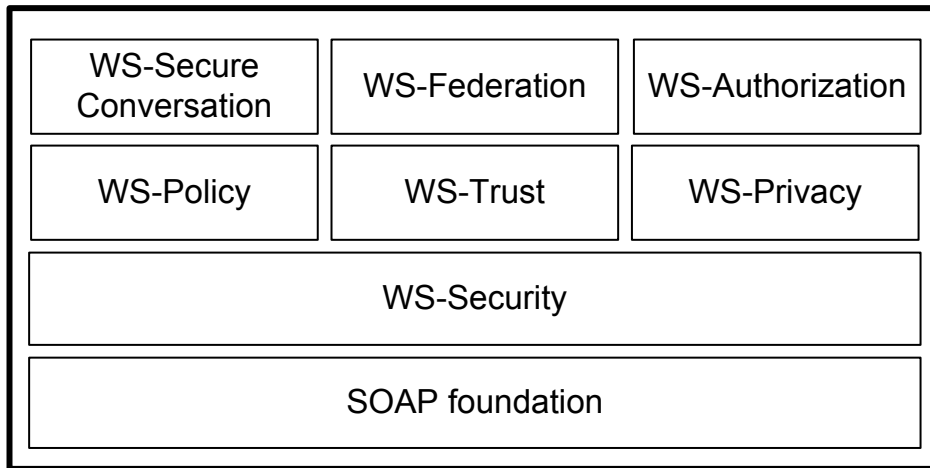
*Figure 2.6: WS-Security Standards BuilT on SOAP [1].*

built on XML Signature, XML Encryption and other security standards. The other standards, such as WS-Policy, WS-Trust, and WS-Privacy rely on WS-Security.

Although Web Services security technologies, such as secure socket layer, WS-Security, WS-Trust, and cryptography, increase the security in the Web Services environment, they cannot determine entities with hidden motives or other trust issues, such as detecting unfair ratings, determining the level of trustworthiness of entities, rewarding positive behaviour, punishing malicious behaviour, and reputation bias [38, 86].

Many researchers define levels of security [23] and privacy [8], and consequently, they define levels of trust based on security [60] and privacy [9]. Allison et al. [9, 8] define a meta-model for privacy policy creation and comparison. In this model, the privacy policies of a service consumer and provider can be compared to create an agreed upon privacy contract through independent trusted Privacy Services (PS), a privacy meta-model consisting of six elements that create a single privacy rule. One of these six elements is trust, which gives the consumer a degree of control over what PSs can be used to negotiate the privacy contract. There are four levels of trust that a consumer can select for a PS to reflect privacy levels: high, moderate, low, and not required. El Yamany [23] specifies four security levels: high, moderate, low, and guest. Each of these levels is related to the main aspects of SOA security, including authentication, authorization, and privacy. Mayer [60] presents two different environmental guidelines for determining

27

security 'levels of trust', which include two security requirements: features and assurances. The features requirement safeguards protected information and includes access control, authentication, user identification, and auditing. Alternatively, the assurance requirement determines the quality of a trusted system and includes architecture, verification, testing, and documentation requirements.

## 2.8   WS-Trust

WS-Trust [43] is an OASIS standard, that represents an extension built on to the WS-Security standard. Specifically, WS-Trust uses WS-Security mechanisms for providing secure messaging and adding extensions for security token exchange within different trust domains. Although different parties can secure the exchange of messages using credentials, they need to trust the asserted credentials. WS-Trust provides a framework for requesting, issuing, renewing, and validating security tokens as well as brokering trust relationships.

The WS-Trust specification document includes a list of the following terminologies:

- Claim: Statements made about clients, services, or other resources, such as name, identity, key, group, and privilege.

- Security Token (ST): Represents a collection of claims.

- Trust Engine: A conceptual component of a Web Service that evaluates the security related feature of the message.

- Security Token Service (STS): A service that issues security tokens.

- Direct Trust: When the service accepts all or some of the claims in the token sent by the requestor.

In the WS-Trust model [43, 74] shown in Figure 2.7, Web Services require the incoming messages that are sent by requestors to have a set of claims in the ST. A Web Service defines its required claims and related information in its policy, which is described by WS-Policy. A Web
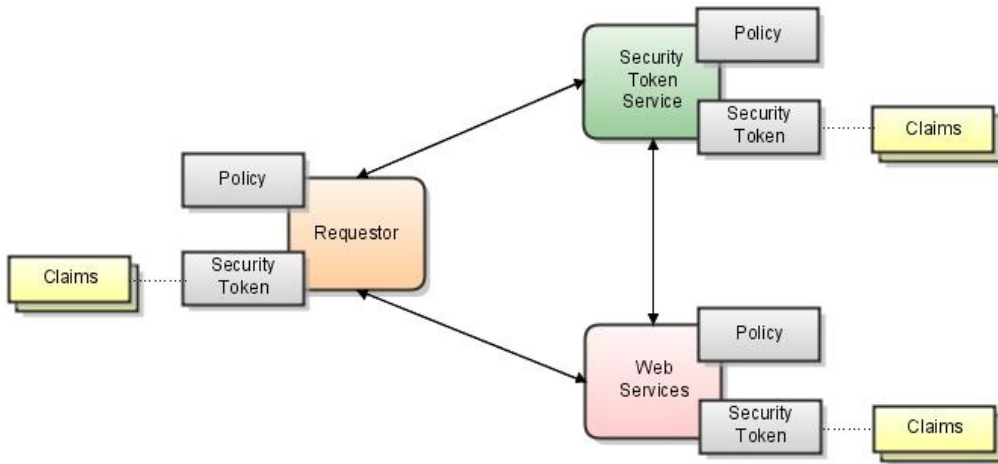
*Figure 2.7:  WS-Trust Model [43].*

Service should accept the claims in the ST to establish trust connection with the requestor. If the message comes without proof of claims, the message will be ignored or rejected by the Web Service. STS is a Web Service, and a requestor may also be a Web Service. Thus, STSs and requestors may require STs and express policies.

Technologies such as WS-Security can successfully meet users' security requirements. However, they can prevent entities with obvious malicious behaviours or purposes, but they cannot block entities with hidden motivations [38]. WS-Trust supports identity trust where the certification authority simply authenticates the owner's identity. However, the certification authority cannot vouch for the trustworthiness of the key owner [33]. WS-Trust provides 'hard trust' but there is still the need for deriving provision trust to give 'soft trust', which this work addresses.

Accordingly, WS-Trust is limited in addressing important trust aspects and requirements as well as various trust challenges. Although WS-Trust is the standard trust mechanism at the messaging level [102], STs in the SOAP messages of WS-Trust cannot provide evidence for some trust aspects or address certain challenges because they are irrelevant to these security tokens. For example, WS-Trust cannot identify entities with hidden motives, consider reputation, or determine levels of trustworthiness of entities. Moreover, WS-Trust have an overhead over requestors that have only basic knowledge and cannot understand complex trust processes or participate in a complex trust negotiation process, such as obtaining or validating a security

token. A requestor needs, simply, to obtain the trust ratings of different entities and to select among these entities.

## 2.9 Summary

This chapter presents background about services, SOA, and trust. Different trust-based service selection approaches are provided. Trust aspects are discussed, such as trust development phases, properties of the trust relationship, and trust classes. Moreover, the chapter examines reputation computation engines and commercial reputation systems. The relation between trust and security is presented, along with the WS-Trust standard, which is used to build trust between different parties.

# Chapter 3

# Related Work

*"You may be deceived if you trust too much, but you will*

*live in torment if you don't trust enough."* Frank Crane

There is a number of review papers and studies on trust and reputation systems [33, 39] and trust-based Web Services selection [94, 21]. Researchers study different approaches for building reputation, rating entities, and receiving feedback in different trust systems, such as the reputation and recommendation systems in centralized and decentralized architectures. This chapter will present literature related to this thesis. Specifically, Section 3.1 will present trust definition, and Section 3.2 will provide trust information. Section 3.3 discussing trust for service providers, and Section 3.4 examines SOA extension. A community-based system is presented in Section 3.5, and trust models are examined in Section 3.6. Finally, Section 3.7 discusses trust bootstrapping and Section 3.8 summarizes the chapter.

## 3.1 Trust Definition in the Literature

In the literature, there are dozens of trust definitions in different contexts and situations, and there is considerable variation in the meaning of trust. Trust is an important factor in many interactions involving uncertainty and dependency. However, the degree of uncertainty, dependency, and risk is higher in the online world than in the offline world [42]. This chapter provides

trust definitions in the offline and online worlds. Our approach in defining trust is to explore the trust literature for extracting the important components of a trust definition.

## Trust Definitions in the Offline World

The English dictionary defines trust as *reliability, reliance on integrity, confident expectations, obligations or responsibilities imposed on someone/something in whom confidence or authority is placed, a fiduciary relationship, being left in guardianship of another, belief, reliance, dependence, certainty, faith, no fear of consequences, commitment, and hope* [2, 3]. Furthermore, *companionship*, *friendship*, *love*, *agreement*, *relaxation*, and *comfort* are some emotions associated with trust [19].

According to David [19], "trust is both an emotional and logical act. Emotionally, it is where you expose your vulnerabilities to people, but believe they will not take advantage of your openness. Logically, it is where you have assessed the probabilities of gain and loss, calculating expected utility based on hard performance data, and concluded that the person in question will behave in a predictable manner" [19]. People trust others because they have experienced their trustworthiness and because they have faith in human nature [19]. Honesty is truthfulness, and an honest entity does not lie or cheat [2].

## Trust Definitions in the Online World

In the online world, trust has been defined in different ways by researchers, which often reflects the paradigms of the researchers' academic disciplines. The most frequently cited definition states that "trust is the willingness of a party to be vulnerable to the action of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective to the ability to monitor or control that other party" [61]. This definition suggests that there is something important to be lost by the trustors, which implies *vulnerability*, and that trustors have no control over trustees, which implies *lack of control*. Although trust is a very effective complexity reduction method, users cannot have control over the behaviour of others [42], which implies *lack of control*. Corritore et al. [17] define online trust as "an

32

attitude of confident expectation in an online situation of risk that one's vulnerabilities will not be exploited". The authors include some key concepts in their definition, which are *risk*, *vulnerability*, *expectation*, *confidence*, and *exploitation*. Alternatively, Chang et al. [15] define trust as "the belief that the Trusting Agent has in the Trusted Agent's willingness and capability to deliver a quality of service in a given context and in a given Timeslot". This definition implies the *context-specific* property of trust.

Buttyan and Hubaux maintain that "trust is about the ability to predict the behaviour of another party" [13]. Grandison and Sloman [33] define trust as "the firm belief in the competence of an entity to act dependably, securely, and reliably within the specified context". In contrast, these authors define distrust as "the lack of firm belief in the competence of an entity to act dependably, securely and reliably within a specified context". These definitions imply that trust is a composition of multiple attributes, such as reliability, honesty, dependability, security, timeliness, and competence, all of which must be considered in different environments where trust will be established.

There are two common definitions of trust in the literature: reliability trust and decision trust. Reliability trust [29] involves a context where A relies on and expects B to perform a given action on which A's welfare depends. This definition includes the concepts of *dependency* and *reliability*. On the other hand, decision trust [62] is "the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible". This definition implies the *context-specific* property of trust and includes the concepts of *dependency*, *reliability*, *utility*, *risk attitude*, *law enforcement*, *insurance and other remedies*.

In Web Services, trust is defined in the WS-Trust specification [43] as "the characteristic that one entity is willing to rely upon a second entity to execute a set of actions and/or to make assertions about a set of subjects and/or scopes". However, this definition fails to induce the concepts of risk and lack of control. Other studies define and specify trust or reputation as a QoS [95, 37, 21]. For instance, Dragoni [21] mentioned that the evaluation of trust is a key QoS aspect of Web Service selection. Kalepu et al. [41] add a "verity" metric to the QoS properties for Web Service selection and define it as "the ability to maintain the lowest difference between the projected and achieved levels of service metrics". Although QoS properties can be used

as information to establish trust, trust is not a QoS. Maximilian and Singh [58] distinguish between trust and QoS and maintain that the selection of Web Services is based on non-functional attributes, such as QoS and trust.

Often, trust is used synonymously with terms such as cooperation, faith, competence, reliance and credibility. However, *cooperation* is either a cause or a manifestation of trust, and trust includes reason, which is the opposite of *faith*. Moreover, trust goes beyond the belief in the *competence* of the trusted party. Trust in information means that the information is *credible* or believable [17]. Also, it is possible to *rely* on a person without trusting him/her [17, 12]. Mayer et al. [61] add confidence and predictability as terms that are considered synonymous with trust. However, with trust, risk is assumed, but with *confidence* it is not.

## 3.2   Trust Information

In the literature, there is no clear identification of trust information for services. Zhengping et al. [103] define domain-specific trust information. Their work monitors the behaviour of Web Services for bugs during operation, which will decrease the degree of trust placed in the Web Services. The authors identified some properties of services to establish their trust, such as services' functions and the run-time environment. Furthermore, they suggested properties for recommenders that recommends services, such as popularity and authenticity of the description. The system analyst defines the domain characteristics.

Kim and Doh [46] propose the selection of an optimal path for composing a number of Web Services based on QoS information and trust type. The trust type is the computed trust level based on aggregated ratings from the service consumer, thus indicating on estimation of the service provider's reliability. The authors assume that trust type is associated with each service, where the assignment of trust types is performed by the clients themselves or by a trust authority. Thus, trust information is not specified and trust is based on an assumed trust type.

Other researchers address trust as a QoS [21, 95, 41, 90], build trust based on a set of QoS parameters [41, 90, 57, 38], or build trust based on QoS parameters that are related to specific system, application, or domain [21, 90, 57, 38]. Dragoni [21] mentions that the evaluation

of trust is a key QoS aspect of Web Service selection. The author uses security features of services to establish trust that satisfy the provider's trust security requirements. Ying-Feng and Pei-Ji [95] specify trust or reputation as a QoS of Web Services. Similarly, Kalepu et al. [41] identify a new QoS attribute, *verity*, as an important contributor to the quality-driven selection and composition of Web Services, and they consider verity as a measure of trustworthiness for Web Services. Specifically, verity refers to the degree of variance in the compliance levels of the services and assesses the reputation of the provider based on local and global ratings. These authors identify verity for Web Services and for Web Service providers. Trust, however, has a different meaning than QoS.

Maximilien and Singh [57] model reputation as a vector of QoS attributes such as performance and reliability. Jin-Dian et al. [38] measure the possibilities of malicious behaviour and satisfaction values that measure how satisfied a user feels about a given interaction. Both measures are real numbers in the interval [0,1], where a high rate reflects a high interaction quality. Trust evaluation can include different QoS requirements, such as process time and access speed. Vu et al. [90] rank services according to their prospective level of satisfying users' QoS requirements. However, the development of trust should consider other properties besides QoS.

Some researchers attempt to distinguish between trust and QoS and consider or use QoS as trust information [58, 94]. Wang and Vassileva [94] state the importance of defining information required for a trust and reputation mechanism. Specifically, they stated the use of QoS to build trust where trust and reputation are built for each quality property of a service and where the overall trust and reputation depend on the combination of trust and reputation for each property. Maximilian and Singh [58] distinguish between trust and QoS, and present the selection of Web Services on the basis of non-functional attributes, such as QoS and trust.

Because QoS properties are considered as important information for establishing trust or reputation, there is a need to identify QoS properties and the literature on QoS, as presented in the following subsection.

## Quality of Service

There are many research efforts to define and categorize QoS as well as attempts to express, quantify, and model QoSs [48, 46, 78, 98, 83]. In [48, 41, 94, 78, 70, 65, 30, 77, 63, 34], including generic and business QoS requirements for services. Garcia and de Toledo [30] define a set of major Web Service QoS attributes. Menasce [63] examines QoS issues in Web Services, and Yu et al. [98] provide a list of QoS parameters and explain how to evaluate each. Although security is not a quantifiable QoS, these authors present a formula to test the security of Web Services based on the number of attack detections. Based on the most common QoS requirements in the literature, Rahman and Meziane [77] present five essential QoS requirements: readiness, transaction, reliability, speed, and security.

To identify a generic QoS, the QoS of services from the literature [48, 77, 78, 98, 46, 30, 70, 74, 24, 34, 63, 46, 20] are aggregated, including latency or network latency, execution time, response time, transaction time, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability, execution price, regulatory, timeliness, security, and transaction, which refers to ACID property: Atomicity, Consistency, Isolation, and Durability. Security includes: authentication, authorization, confidentiality, non-repudiation, accountability, traceability and auditability, data encryption, access control, and prevention of the Denial-of-Service attack (DoS). Table 3.1 presents the description of some QoS parameters that are used in this work.

Apart from these common characteristics, there are other QoS in the literature. O'Brien et al. [70] define other QoS requirements for SOA: Modifiability refers to "the ability to make changes to a system quickly and cost-effectively", and testability indicates "the degree to which a system or service aids the establishment of test criteria and the performance of tests to determine whether those criteria have been met". Moreover, usability is "a measure of the quality of a user's experience in interacting with information or services". In addition, the authors identify the required QoS for SOA and argue that QoS can be significantly affected by SOA. Ran [78] identifies other QoS, which include a supported standard, stability/change cycle, and completeness. In addition, there is a domain or application specific QoS.

*Table 3.1: QoS Parameters and Descriptions.*

| QoS parameters | Descriptions | References |
|---|---|---|
| Latency or Network Latency | "The time the SOAP message needs to reach its destination". | [83] |
| Execution Time | The time taken by the service to execute and process its sequence of activities. | [48] |
| Response Time | The time required to process and complete a service request; the response time includes the execution time and the latency. | [48, 77, 30, 70, 63, 78, 98, 46] |
| Throughput | The number of requests a service can process per unit of time. | [48, 30, 70, 63, 78] |
| Security | Offers mechanisms of authentication, authorization, confidentiality, non-repudiation, accountability, traceability, and auditability. | [48, 78, 34, 74, 20] |

Hoyle [34] identifies other quality characteristics for services, such as courtesy, comfort, competence, credibility, dependability, efficiency, effectiveness, flexibility, honesty, promptness, and responsiveness. Specifically, the author argues that people are either competent or incompetent without any varying degrees of competency. Individuals are competent if they have the ability to produce the desired results when required and demonstrate performance that meets all required standards. Competence, which can be assessed under close supervision, is "the ability to demonstrate the skills, behaviours, attributes, and qualifications to the level required for the job" and "a quality of individuals, groups, and organizations" [34]. "A competent entity is capable of performing the functions expected of it or services it is meant to provide correctly and within reasonable time scale" [88].

Grandison and Sloman [33] mention that trust is a complex subject relating to beliefs in honesty, truthfulness, competence, and reliability. In particular, a trustworthy service will tell the truth and be honest in interactions. Competence demonstrates a provider's ability to provide a service and perform the function expected from it. In fact, competence is a relevant term for the environment related to services and computing systems and it applies to entities that perform an action on behalf of the trustor. A customer's trust in the supplier's competence and honesty will influence their decision as to which supplier to use [33]. Corritore et al. [17] suggest that

37

competence is one of many cognitive cues for trust, and similarly, Mayer et al. [61] present competence and honesty as trust factors for trustees. "An honest entity is truthful and does not deceive or commit fraud" [88].

Some research work [51, 40, 64] evaluate honesty of raters through long-term interaction. Specifically, Malik and Bouguettaya [51] evaluate the credibility of raters in a reputation-based framework based on the evaluation of their honestly over time. This framework aims to protect the reputation system from malicious raters and to fairly assess the providers' reputations.

Larson [47] identifies serviceability and user satisfaction as performance measurements for service delivery, and Moorsel [65] discusses quantitative metrics and develops a framework for evaluating Internet services. This author defines three metrics that should be emerged to evaluate Business to Consumer (B2C), Business to Business (B2B), and service providers. The metrics include QoS, Quality of Experience (QoE), and Quality of Business (QoBiz). While QoE quantifies the user experience, QoBiz measures the business return.

## 3.3  Trusting Service Providers

In Web Services and SOA, the idea of building trust and reputation for service providers is neglected [94]. Jin-Dian et al. [38] present the idea of assigning provider trust rates to Web Services. Specifically, they mention that assigning trust rates for providers is an interesting research problem. Consequently, they suggest a registry to use past experiences with the Web Service's provider for initializing the rate of the new Web Services as equal to its provider's rate.

Maximilien and Singh [58] mention that if a service provider has been determined as trustworthy, then requestors will select a service from a provider that has the the highest trust level. The authors state that determining trust levels for providers is non-trivial, especially in an open environment. In particular, Maximilien and Singh [57] assess the trustworthiness of a Web Service provider by measuring its reputation on the basis of rates given by users.

Furthermore, in the services and SOA literature, there is no identified trust information for service providers, and the literature reveals no distinction between service trust information and

service provider trust information, which occurs in the Internet literature. The Internet literature has identified quality requirements for providers to enhance their trustworthiness and help users in their decision to use providers' services [42, 61]. Based on a trustee's attributes, Kautonen and Karjaluoto [42] have distinguished between two dimensions of online trust: a hard dimension and a soft dimension. The hard dimension contains a functionality-based nature, such as competence, ability, and predictability. Alternatively, the soft dimension includes characteristics, such as honesty, integrity, and credibility. Based on these dimensions, providers can provide important clues for trustors to assess their trustworthiness. For example, some merchants provide contact information, retail location, online contact on their Web site. In addition, the trustee can provide quality information on important issues such as order progress and delivery charges as well as policies on privacy, returns, and remedies. Mayer et al. [61] present a review of the proposed factors that lead to trust, such as availability, competence, consistency, integrity, loyalty, benevolence, honesty, group goals, openness, caring, goodwill, and promise. In particular, they examine three characteristics of a trustee that appear frequently: ability, benevolence, and integrity.

## 3.4 SOA Extension

This section will discuss the literature on extending the basic architecture of SOA based on QoS and trust. There are variations on the way in which SOA is extended to support trust. In addition, the three roles of SOA, requestor, service provider, and registry, and the new components addressing ranking or trust, such as service evaluation center and rating registry, are connected in different ways.

Figure 3.1 shows different extensions of SOA. Some studies, depicted in Figure 3.1 (a), use the regular SOA model where the ranking or trust process is conducted in the service broker by the service registry [16] or by additional component added to the service broker [46, 49]. Other studies [14, 78], as in Figure 3.1 (b), use the regular SOA model and add an auxiliary component outside of the service broker. The three roles of SOA are connected to the auxiliary component responsible for Web Service evaluation based on QoS and user preference [14] or QoS certifying and verifying [78]. Al-Masri and Mahmoud's model [7], shown in Figure 3.1
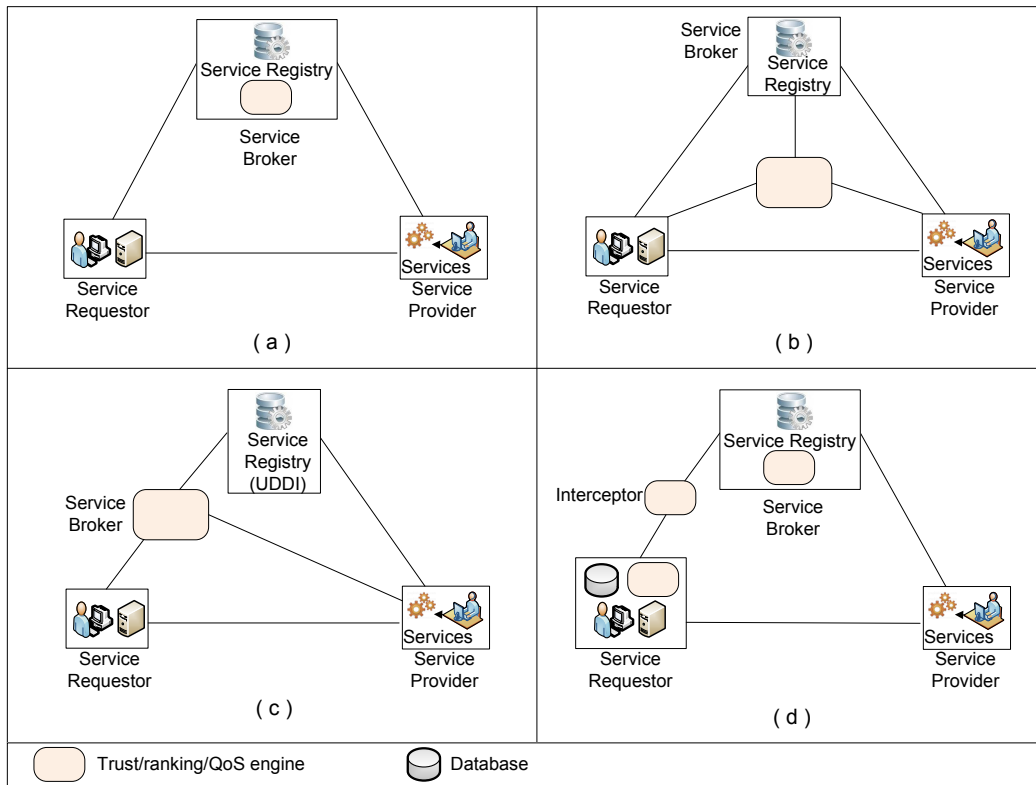
*Figure 3.1: SOA Extensions to Support QoS/Ranking/Trust.*

(c), use the regular SOA model, but they separate the service registry (UDDI) and the service broker that conducts QoS ranking. In this model, Web Service providers publish their Web Services in the service registry, such as UDDI. In Kalepu's extension [41], shown in Figure 3.1 (d), there are two verity calculators, one on the service broker side and one on the end user side, to calculate local and global rankings. In addition, there is an interceptor component between the service broker and the end user to measure the SLA parameter values delivered at the end of each service invocation and to send the values to both the end user and the service broker for verity calculations.

Based on these variations of SOA extensions, this work aims to derive a suitable way of extending SOA to support trust. Meihua et al. [16] provides an extension that modifies the basic roles of SOA, whereas the other extensions [46, 49, 14, 78, 7, 41] add new roles to SOA. Between these two extension approaches, the latter method is preferable because it does not

require modification of the basic roles of SOA, which facilitates the deployment of the component as a service to the SOA environment. However, the type of extension shown in Figure 3.1 (a), places less overhead on the requestors and service providers, where as the additional component can be used by requestors and providers in the same way that they use other services in the environment, such as the service registry. In addition, by having the trust process conducted through a third party, requestors can make a decision upon discovering the services, and the requestor and service can interact immediately after the decision.

## 3.5 Community-Based System

Malik and Bouguettaya [52] suggest a community model, where one member of the community is assigned to perform transaction for reputation bootstrapping the newcomers that satisfy the specified domain properties. Bataineh [10] stated that a set of Web Services with a similar functionality are grouped together into what is known as a community. Similarly, Shao-Jie et al. [86] group providers who provide identical or similar services into one domain.

However, SOA systems should be designed in a way that does not always group similar providers or services in one domain or community, especially since providers may publish their services in different service registries. In addition, providers may register their services in a different registry than one with similar services. Thus, a trust solution should not be community-based in order to suit an open computing environment, such as SOA environment.

## 3.6 Trust Models

In the trust literature, there are many research work that build models to evaluate trust. The trust models range from simple models, such as eBay, AllExperts, Epinions, and Amazon models, to complicated models [86, 100, 25]; they cover trust [46, 45, 103, 50], reputation [52, 100, 57, 89], or hybrid approach [38, 103]. While some models consider some trust aspects [52, 46, 45, 103, 41, 38, 50], most do not consider trust bootstrapping [46, 45, 103, 41, 38, 50, 86, 100]. Thus, a trust model should be simple, understandable, comprehensive in covering different trust

aspects, and extendible. In addition, it should consider trust bootstrapping and resolve various trust challenges.

Malik and Bouguettaya [52] adopt a model for reputation bootstrapping, however, their model does not consider different trust aspects or address trust challenges such as cold start and whitewashing. Kim [45] and Kim and Doh [46] build a model to select the optimal services composition execution path based on trust and QoS. The authors assume that a trust type is associated with each service, where the assignment of trust types is performed by the client or by a trust authority ( i.e. $TT(s_i) = a$, the assignment of a trust type 'a' to a component Web Service $s_i$). In this model, the execution plan with the maximum value of profit is the optimal one, where the profit is a function of trust weight and quality vector MQ, computed QoS attributes, of the execution plan. However, the model is based on undefined trust information and assumed trust types, and it does not consider bootstrapping.

Zhengping et al. [103] build a trust model where trust degrees are based on the properties of services and recommenders. In this model, there are known and unknown recommenders based on whether they are familiar to the trusted authority. The trust degree of a service is related to the service description $V_{kp}$, known party's recommendation $V_{up}$, and unknown party's recommendation $V_a$. Thus, $trustdegree(A, S) = f(V_{kp}, V_{up}, V_a)$. However, their model does not involve many trust aspects, such as the context-specific property of trust, and it does not consider trust bootstrapping.

Kalepu [41] calculates the compliance values for all of the services SLA parameters, which involves the difference between the projected and achieved parameter values. The result is expressed as positive, negative, or zero compliance. A positive compliance indicates that the agreed-upon values have been delivered without violations, while a negative compliance means that the provider failed to deliver the agreed-upon values. Finally, a zero compliance indicates an ideal value, where the delivered values are exactly equal to the agreed-upon values. The local and global rankings of the services and their providers are evaluated based on the compliance levels of SLA parameters. However, this model does not consider trust bootstrapping. Maximilien and Singh [59] derive a trust function based on QoS, where some QoS parameters are preferred by service customers if their values are high, such as availability, and others are preferred if their values are low, such as response time.

Jin-Dian et al. [38] propose a WSTrust model based on feedback and reputation, where requestors provide their feedback to the service broker about the services they have consumed. This trust model considers the following trust aspects and challenges: trust and reputation vary when the number of interactions increase, trust is based on direct experiences and recommendations, the prevention and punishment of repeated malicious behaviours by users, the addressing of users' unfair ratings, and trust is context-dependent and subjective, which involves dividing trust into three trust relationships: brokers trust services, requestors trust brokers, and requestors trust services. However, the authors cover a limited number of trust aspects and do not consider trust bootstrapping and other trust challenges. Similarly, Liu et al. [50] build a trust model that only considers three trust properties: specific, a matter of degree, and dynamic. The literature contains other complicated trust models that consider one or more trust aspects [86, 100, 25]. However, trust models should be simple, understandable, and comprehensive.

Some research work use the *monitoring* approach to check QoS compliance, ensure SLA, and rank services based on QoS attributes [83, 103, 101, 41, 90, 87, 69]. Rosenberg et al. [83] propose an evaluation approach for bootstrapping QoS attributes of Web Services that provide a set of up-to-date QoS attributes for Web service selection. Zhengping et al. [103] monitor the behaviour of services at run time, and Zhang et al. [101] present a Web Service search engine to find desired Web Services. Specifically, the engine ranks Web Services by monitoring the non-functional QoS characteristics of Web Services. Kalepu et al. [41] use a third party to monitor the transactions between requestors and services for detecting any SLA violations. The third party verifies the values of SLA parameters in the agreement against the obtained values by probing or intercepting the client invocation. Furthermore, Vu et al. [90] monitor QoS for Web Service ranking and selection and to detect and address false ratings. Sherchan et al. [87] measure the compliance of QoS attributes by comparing the projected values agreed-upon in the SLA and the delivered values obtained from the performance monitoring system. Nguyen et al. [69] build a trust and reputation model for Web Services based on feedback and QoS monitoring approaches.

## 3.7 Trust Bootstrapping

Trust has been used as a criteria for service selection [21, 95, 37, 41]. Since the majority of studies assume a system where trust and reputations already exist [57, 46, 49, 58, 59], little attention has been given to the trust bootstrapping problem of rating new services and providers. In addition, many weaknesses in the proposed bootstrapping solutions need to be addressed for such a solution to be effective.

Some bootstrapping approaches [100, 16] assign default reputation values, such as low, high, or no reputation, to newcomers. Jin-Dian et al. [38] mention that the trust rate for a Web Service can equal that of its provider; however, if the provider has no rating, a low reputation can be assigned to the provider's first service. Previous research attempted to solve trust/reputation bootstrapping problem without assigning a default value. For instance, Feldman and Chuang [25] collect all transaction information for first-time interactions with newcomers and use the aggregated information to calculate the probability that the next newcomer will cheat. Their approach, which is community-based, bases the rate of a new entity on the rates of other entities. However, the initial rate is not accurate or fair, and it does not reflect the actual rate of the new entity.

Swamynathan [89] proposes a proactive reputation for rating newcomers and addresses the cold start problem. This approach is based on a *peer-to-peer* reputation system, where the peers are allowed to proactively initiate interactions with other peers for generating reputation rates. Thus, peers generate rates for other peers who are relatively new.

Maximilien and Singh [57] propose an endorsement technique for Web Service selection that can be used to rate newcomers. In this system, newcomers gain reputation by endorsement from one of a trusted participant with high credibility. However, it is not easy for newcomers to be endorsed (rated) by a participant.

Malik and Bouguettaya [52] propose two reputation bootstrapping approaches. The first approach adapts according to the behaviour of most services, and it is based on the evaluated rate of maliciousness. The second approach assigns a *default initial reputation*, where a newcomer purchase an initial reputation from the community provider, or the community conducts an *initial reputation evaluation*, where services with high credibilities, known as elders, are asked

to evaluate the newcomer. These approaches have many limitations; they are based on the concept of community and different bootstrapping techniques suit different domains. Also, in the bootstrapping process, the contribution of requestors is high. Furthermore, both rational and malicious members can provide ratings, and the model does not consider the cold start and whitewashing problems.

Dragoni [21] proposes the concept of hard trust, which is based on contract, rather than soft trust, which is based on reputation and judgments. Specifically, the paper highlights a hard trust framework for Web Services publishing, selection, and monitoring, which is known as Trust-By-Contract. In this framework, a user can trust the service before its invocation without the service having been previously evaluated by the community. However, this approach has some limitations, as it is based only on the Web Services *security behaviour*. Moreover, the service requestor and provider should be able to engage in a complex negotiation, which sounds strong requirement for open large environment and requires the requestor to do a complex negotiation to select a service (i.e., requestor side overhead).

O'Hara et al. [71] identify five common strategies of trust and discuss their costs and benefits. The strategies are classified as *optimistic*, which assumes that all strangers are trustworthy unless proven otherwise. In contrast, a *pessimistic* strategy assumes all strangers are untrustworthy unless proven otherwise. A *centralized* strategy provides trust information from a third party, and an *investigating* strategy checks and evaluates the agents for their trustworthiness. Finally, a *transitivity* strategy analyses networks of agents to determine their trustworthiness.

However, none of these bootstrapping approaches is suitable for an open distributed environment, and hence, a new approach is required to address the weaknesses of the current approaches. In addition, the current bootstrapping approaches in the trust literature address *reputation* bootstrapping and not *trust* bootstrapping. To the best of our knowledge, there is no proposed approach for trust bootstrapping services or service providers. In each of the literature models, requestors play a big role in the bootstrapping process, which is not suitable because requestors may only have basic knowledge and cannot contribute in a complex trust process. Thus, the responsibility of requestors should be decreased to lower requestors' side overhead. In addition, the bootstrapping solution should resolve other challenges, such as community-based

models, default values, unfair reputation and feedback from users, negotiation and complex service selection and matching methods, cold start, and whitewashing.

## 3.8 Summary

This chapter presents the related work on how researchers define trust, identify trust information, consider the trustworthiness of service providers, extend SOA, build trust models, and address the bootstrapping challenge. In the current studies, there are many limitations for defining trust, identifying trust principles, considering trust rating service providers, identifying standardized trust information for services and service providers, and resolving different trust challenges such as trust bootstrapping, whitewashing, cold start, and requestor side overhead. Accordingly, there is a need to extend SOA to support trust and build trust models that are understandable, incorporate many trust aspects, can be easily extendible, consider trust bootstrapping, and resolve different trust challenges.

The following chapters present the proposed trust solution that incorporates a trust definition, trust principles, trust-based SOA, a trust framework, trust metrics, or trust information, for services and providers, trust models for trust metrics, and trust models for services and service providers. The subsequent chapter presents a trust definition, trust principles, a trust-based SOA, and a trust framework.

# Chapter 4

# Trust-Based SOA and Trust Framework

*"Trust no one unless you have eaten much salt with him."* Cicero

This work introduces a trust-based SOA for establishing trust for services and service providers to support trust-based service selection. In this work, trust is defined and trust principles are identified. Furthermore, SOA is extended and a new component is added, a trust mediator, whose framework is built and its components are identified on the basis of the trust definition and trust principles. The result is the Total Trust Evaluator Framework (ToTEF), which is a unified trust framework that handles trust issues in a comprehensive manner. ToTEF has the necessary components for trust establishment and management that address different trust challenges in the literature, such as trust bootstrapping, unfair feedback, and culture differences.

This chapter presents the proposed trust definition, trust principles, trust-based SOA, and trust mediator framework. Section 4.1 discusses the trust definition and trust principles, while Section 4.2 introduces the trust-based SOA. Section 4.3 presents ToTEF, the trust mediator framework, and Section 4.4 summarizes the chapter.

## 4.1 The Proposed Trust Definition and Trust Principles

This section presents the proposed trust definition and trust principles of this work.

### 4.1.1 The Trust Definition

Trust is a complex subjective term. In order to define trust it is necessary to explore the literature for extracting the important **components** of a trust definition and composing a standardized trust definition. To define trust, we analysed the trust definitions in the offline and online worlds, online services, Web Services, and SOA environments, as presented in Section 3.1. The main components of a trust definition are:

- Utility: A trustee needs to provide a utility, such as service's function, to a trustor. A trust definition assess trust by calculating the promised utility.

- Dependency and reliability: Trustors depend and rely on trustees to perform their promises.

- Risk attitude: Risk is the core of trust, as with trust, risk is assumed. However, when individuals display trust, they expect no harm.

- Vulnerability: Trustors should have something important to lose when they trust.

- Remedies: Since risk is implied in trust, the provision of remedies is an important part of trust. In risky situations, a trustee should provide remedies if they are unable to fulfil their promise. Thus, remedies are *implicitly* included in the trust term. However, in the online world, different remedies should be explicitly identified for supporting trust.

- Confidence expectation: Trust and confidence are different, since the possession of confidence makes trust unnecessary. With trust, trustors assume risk, but with confidence, they do not. For example, individuals have confidence that they can arrive on time for a meeting by using their car, but they need to trust a taxi driver to drive them to the meeting on time. However, a trustor *implicitly* has confidence in the trustee although there is an assumption of risk. If you have confidence, you do not need to trust.

- Context-specific: Trust is context-specific, which indicates placing trust in a trustee to perform a *specific action* within a *specific context/situation*. Therefore, trust is different in various situations.

- Subjective: Trust is subjective, which means that trust is experienced differently for different trustees. For example, one requestor may trust a service, but other requestors may not trust the same service, since every requestor builds trust based on his/her trust preferences, such as reliability and security.

- Lack of control: A trustor may have no control over the trustee. The more control a trustor has over the trustee, the less need there is for trust.

- Complexity-reduction method: There are different approaches used to ensure that a service provides what it promises, such as SLA and monitoring approaches. If a requestor trusts a service, he/she does not need to monitor the service and the system does not need to check SLA, perform monitoring, or apply policies. However, trust establishment initially requires *long-term interactions* to ensure the trust rates of the entities and overcomes various challenges, such as whitewashing and malicious behaviour. For example, a requestor may use Google or Amazon services with confidence because he/she has been using their services for a long time, trusts their services, and has built a confident relationship with the providers. However, a long-term interaction requires monitoring and verifying SLA compliance. Once a trustee is highly trusted, the system can stop monitoring their services. Thus, trust can be a complexity-reduction method if a trust relationship is developed on the basis of a robust trust technique that overcomes the trust challenges and if a trustor builds a confident relationship with the trustee based on long-term interactions.

- Trust should not be used synonymously with certain terms, such as trustworthiness, cooperation, faith, confidence, or QoS.

- A trust definition need to have an inclusive view and not to be related solely to a specific domain requirement. Accordingly, trust information should not be included in a trust definition.

- Trust emotions: There are some emotions associated with trust, such as comfort, agreement, no fair of consequences, and relaxation. These emotions are *implicitly* included in the trust term.

Based on this analysis and the identified trust definition components, this work proposes the following definition of trust:

*Trust is the willingness of the trustor to rely on a trustee to do what is promised in a given context, irrespective of the ability to monitor or control the trustee, and even though negative consequences may occur.*

The trust definition explicitly includes the concepts of dependency, reliability, and vulnerability (the willingness to rely), utility (to do), subjectivity and context-specificity (in a given context), lack of control (irrespective of the ability to monitor or control the trustee), and risk attitude (even though negative consequences may occur). On the other hand, the definition implicitly includes confidence expectation, safety and comfort, and insurance and other remedies.

It is important to trust an entity before selecting it. Trust bootstrapping is *a mechanism that assigns a trust rating for a new entity that has unknown trustworthiness and has not interacted with other entities.* Bootstrapping is important because a new entity that joins a network will have a trust value. Thus, the trustworthiness of a new entity is known and a trustor can select and interact with it based on its trust rate.

## 4.1.2   The Trust Principles

As discussed in Section 2.4, trust has many aspects, such as properties of the trust relationship, categories of trust semantics, and trust classes. Researchers attempt to consider a few aspects of trust in their trust solutions. However, it is important to consider many aspects of trust in order to build a concrete trust solution. Therefore, the trust aspects and trust requirements should be analysed and grouped in order to identify trust principles that present the nature of trust and underpin the concept of trust, and thus develop a concrete solution of trust. This work proposes comprehensive trust principles based on the exploration and analysis of trust in the literature. Fifteen trust principles are presented below:

1. Trust and risk are related. It is important to consider risk in the trust establishment process and to include penalties, rewards, insurance, and other risk remedies to support trust intention.

50

2. Trust is dynamic. Trust establishment should consider the dynamic nature of trust, which requires a continuous evaluation of entities' trustworthiness. Some approaches used to evaluate, test, personalise, evolve, and ensure trust include the trust broker opinion, reputation, recommendation, and referral methods.

3. The trust development phases should be considered. Researchers need to regard the dynamic nature of trust and take different development phases of trust into account when establishing trust. Most studies assume a system where trust and reputations already exist, as in the trust stabilising phase. However, it is important to consider the trust building phase and initialise trust rates for new entities. Moreover, the trust dissolution phase should be considered for addressing trust failure and reconstruction.

4. Trust depends on identity. Entities should be identified to establish their trust and to be distinguished from others. The possession of identity enables the history or past experience of interactions to be built and mapped to that identity [18].

5. Consider categories of trust semantics. Semantics categories of trust rates are important for interpreting trust measures.

6. Properties of the trust relationship should be considered. Trust relationship properties, such as context specificity, transitivity, and asymmetry, should be considered when establishing trust.

7. Consider global and local ratings. Global trust rates shows how the community as a whole trusts a specific trustee whereas local trust rates are personalized scores. Global rates are advantageous because they communicate experiences, but the information is potentially unreliable and comes from unknown or anonymous second parties [18]. Local rates are more reliable, as they are based on user preferences. A Consideration of both global and local rates helps the trustor to make a better decision about an entity.

8. Trust is based on information. An individual needs to know information about entities to establish their trust. It is required to identify what information should be used to build trust for entities.

9. First party information is important to establish trust. First parties, or trustees, should provide their information to develop their trust. For services, QoS properties and other information, such as delivery methods, insurance, privacy, security, pricing, and availability is important information on which to build trust.

10. Third party ratings are important to establish trust. An expert opinion from a TTP, such as a certification authority, service broker, or trust broker, plays an important role in trust establishment, evolution, and self-adjustment.

11. Consider various trust approaches. The combination of different trust approaches, or the hybrid approach, improves the weaknesses of other approaches and results in a more robust approach for establishing trust. Thus, it is essential to consider various trust approaches for building a concrete trust solution.

12. Trust should be distinguished from QoS. There is a clear distinction between trust and QoS definitions. Trust is not a QoS; however, QoS can be used as information for establishing trust.

13. Consider trustworthiness of providers. Trust ratings of services and their providers are related, and each rating affects the other. In general, a trustworthy service provider provides trustworthy services. A consideration of providers' ratings will encourage providers to offer high quality services, because the trust rates of the provider and each of their services are related.

14. Trustor preferences should be considered. Trust is context-specific, multi-faceted, not absolute, matter of degree, unsymmetrical, and not necessarily transitive. Therefore, a trust-based system should support trustors' preferences. Since different trustors have various preferences, the same trustee will receive different trust rates, known as personalized or local ratings. Trustors should be able to select a trustee based on their trust preferences in various contexts, setting different degrees of preferences. For example, in service selection, a requestor may request a service based on its availability and execution time as preferences, providing a degree of 60% for the availability and 100% for the execution time.

Moreover, the Web is an open environment that spans different countries, regulations, and cultures with various requirements and properties. These differences affect the trust building process, as trustors of different countries and culture rate trustees differently. A consideration of trustors' trust preferences mitigates the division of trust between different countries and cultures.

15. Consider trust classes. In order to establish trust, it is effective to support different trust classes, such as provision, delegation, context, and certification of trustees. In provision trust, a trustor should be able to rely on a TTP to rate entities and seek protection from malicious entities. Furthermore, delegation plays a major role in trust establishment; a trustor can delegate a TTP or an agent to act on his/her behalf in order to identify an entity's trustworthiness. In context trust, a system should support trust by having the ability to support transactions and perform remedies, such as law enforcement and insurance in case something goes wrong. Finally, the certification of a trustee can help to build trust. In particular, certification from a third party can enhance and ensure the trustee's trustworthiness, since a trustor should trust a certificate given to the trustee by a TTP.

## 4.2   Trust-Based SOA

This section presents the SOA extension for supporting trust. Figure 4.1 shows the proposed trust-based SOA; the architecture adheres to that of SOA model and it has the same SOA roles and operations. The service broker is considered as a TTP that establish trust for services and service providers.

The SOA extension supports trust by including a trust mediator, interfaces, and additional link interactions. The *trust mediator* is added into the service broker as a service, and it is responsible for conducting the trust process. The trust mediator has a rating registry. The trust-based SOA needs to have a *provider interface* and *requestor interface*. The provider interface allows service providers to publish their services, provide their trust information, and view their ratings and service ratings to improve their services and build their QoBiz (Section 3.2).
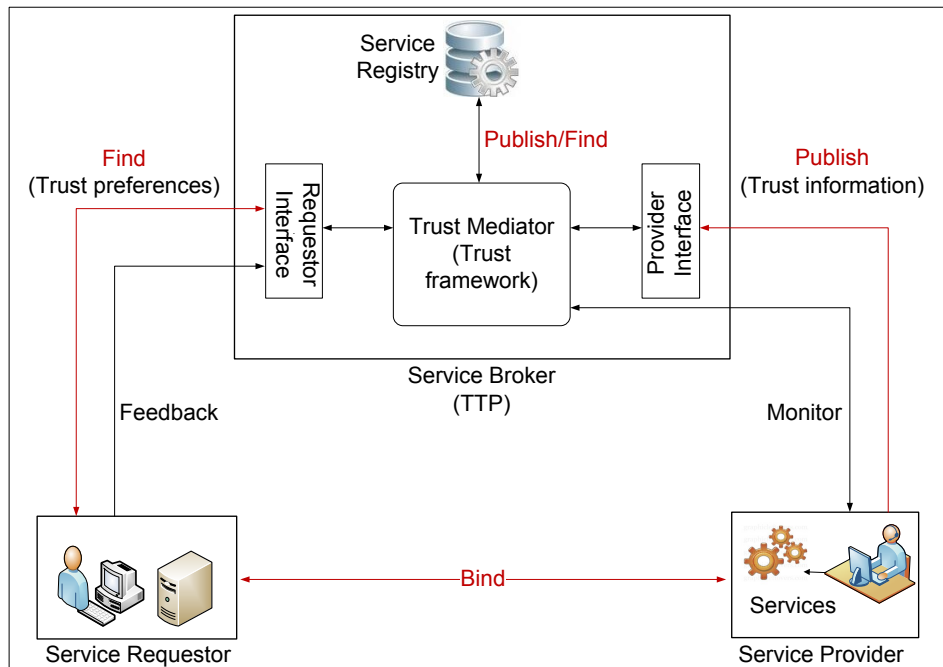
*Figure 4.1: Trust-Based SOA.*

Moreover, the requestor interface allows requestors to search for services and enables service consumers to provide feedback on the services they have consumed.

The additional *link interactions* include a monitor link and feedback link; the monitor link between the trust mediator and service provider monitors the registered services. Alternatively, the feedback link is for requestors to provide their feedback about services and service providers.

Service providers need to publish their services along with the trust information. In the publish operation, the trust mediator obtains the publish request from the provider interface. Subsequently, the mediator obtains the service description and trust information, and then publishes the service in the service registry and stores the trust information in the rating registry. Prior to publishing the services in the service registry, the trust mediator identifies the service and provider from the service description.

The find operation should support the selection of services based on their functional properties and trustworthiness on a set of requestors' trust preferences. In this operation, the trust

mediator obtains the find request from the requestor interface and discovers the service registry for services that match the functional properties. Then, the trust mediator selects services that match the non-functional property, trustworthiness, of services from the rating registry. Consequently, the services that satisfy the requestor's functional and trust preferences are returned to the requestor.

Trust establishment is performed at the TTP rather than the requestor side for the following reasons:

- Trust bootstrapping new services. The trust mediator in the TTP evaluates trust for new services before requestors interact with them.

- The trust for all advertised trust information of a service can be built at registration time rather than only specific trust information being selected by a requestor at searching time of a service.

- The TTP provides support for the requestors, where they do not need to understand the complex trust process. Thus, this lowers the overhead on the requestor side.

- The challenge of providing motivation for rating services by service consumers is overcome, since mediator runs the trust process and evaluates trust rates before getting feedback from the service consumers.


## 4.3  ToTEF: Total Trust Evaluator Framework

This section presents the Total Trust Evaluator Framework (ToTEF), which is the trust mediator framework. Figure 4.2 shows the proposed framework; ToTEF consists of three main phases, including the pre-processing phase, the processing and evaluation phase, and the post-processing phase. Each phase consists of several components that performs different functions. In addition, ToTEF has a rating registry that stores trust information, known as Trust Metrics (TMs), and trust rates that are evaluated by the trust mediator. It supports searching, matching, and selecting services based on requestors' trust preferences. All the framework components are
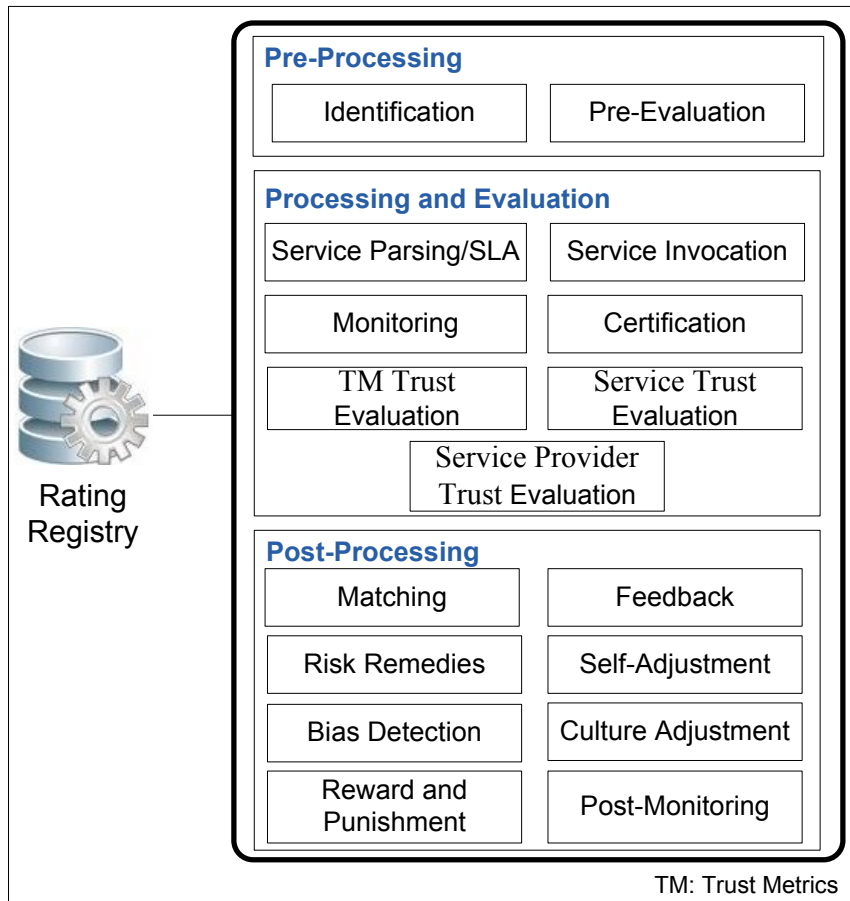
*Figure 4.2: ToTEF: Total Trust Evaluator Framework.*

connected to the rating registry, and each component can use information evaluated and stored by the other components. In the following subsections, ToTEF phases, components, functions, and requirements are explored.

### 4.3.1 Pre-Processing Phase

Service providers publish services and TM with the service broker. The pre-processing phase identifies and registers new services and service providers into the service registry, stores the TMs into the rating registry, and pre-evaluates services and providers. As depicted in Figure 4.2, the pre-processing phase has two components: identification and pre-evaluation.

1-<u>Identification</u>

Trust depends on identity; each service and service provider needs to have ID. The identification component identifies the services and service providers and assigns IDs to them as well as publishes the services into the service registry. Moreover, this component stores the services and service providers' IDs, along with their TMs, into the rating registry. The identification component assigns IDs to the new service provider and its service. If a service provider is already identified, the identification component will assign an ID to only their new services.

2-<u>Pre-Evaluation</u>

The next phase, the processing and evaluation phase, is responsible for trust bootstrapping and dynamic trust evaluation for TMs, services, and service providers, which may have high overhead on the service broker. However, this work reduces the overhead by *rating services based on their providers' rates*. If a provider is trustworthy, its services can also be considered as trustworthy. In this case, the pre-evaluation component will assign the trust rates of the provider's new services as equal to providers' trust rates rather than running the next phase, the processing and evaluation phase. The trust mediator will determine the extent to which a service provider should be trustworthy for assigning its rate to its newly-registered services, as proposed by this work in Section 6.4.

## 4.3.2  Processing and Evaluation Phase

This phase is responsible for trust bootstrapping and for the dynamic trust evaluation of the TMs, services, and service providers. This work proposes a number of dynamic approaches for trust bootstrapping, evaluation, and evolution. These approaches include monitoring, certification, and feedback from service consumers. This phase parses services descriptions, invokes services, monitors and certifies TMs, and evaluates the trust rates of the TMs, services, and service providers. This phase can involve seven components as shown in Figure 4.2: service parsing/SLA, service invocation, monitoring, certification, TM trust evaluation, service trust evaluation, and service provider trust evaluation, as follows.

1- <u>Service Parsing/SLA</u>

The trust bootstrapping process necessitates a dynamic evaluation of services, which requires

obtaining information necessary for service evaluation. Specifically, this component obtains the information about a service that is necessary for each of the subsequent components, invocation, monitoring, and certification, to perform their processes. The required service information can include the service operation, input and output parameters of the operation and their data types, and binding information as well as policies. Moreover, TMs can be obtained from service description, which requires extension, or from SLA. In this work, the TMs are obtained from the provider interface.

### 2- Service Invocation

The service invocation component invokes services; specifically, it requires services' operations, input and output parameters, data types, and binding information, which is obtained by the previous component, the service parsing/SLA.

### 3- Monitoring

In this work, the monitoring approach is proposed as a method for trust bootstrapping and dynamic trust evaluation. In particular, the monitoring component bootstraps TMs that can be measured, such as execution time. Monitoring a TM can be conducted either once or several times and obtain the average for the final value. For some TMs, such as throughput, the monitoring component needs to monitor the metric several times to obtain the final value. The collected information is stored in the rating registry and used by the subsequent components for trust evaluation.

### 4- Certification

This component is responsible for certifying some TMs, such as security and privacy metrics, based on the services' policies about such TMs.

### 5- TM Trust Evaluation

TM trust evaluation component evaluates trust rates for TMs. TMs ratings are based on the TMs published by a service provider and collected by the monitoring and certification components of the processing and evaluation phase. The TM trust evaluation models are presented in Chapter 6.

### 6- Service Trust Evaluation

Service trust evaluation component evaluates trust rates for services. The trust rates of services

are based on the trust rates of their published TMs, the information from the TM trust evaluation component. The service trust evaluation model is presented in Section 7.1.

   7- Service Provider Trust Evaluation

Service provider trust evaluation component evaluates trust rates for service providers. The trust rates of service providers are based on the trust rates of their services, the information from the service trust evaluation component. The service provider trust evaluation model is presented in Section 7.2.

## 4.3.3   Post-Processing Phase

The post-processing phase contains various components that play significant roles in service discovery based on trust and trust management. In addition, it addresses different trust challenges in the literature, such as culture differences, unfair feedback, and bias detection. As shown in Figure 4.2, the post-processing phase can include eight components: matching, feedback, risk remedies, self-adjustment, bias detection, culture adjustment, reward and punishment, and post-monitoring. However, this work addresses only the matching component as the other components are beyond the scope of this thesis. Nevertheless, this work will present the importance of each component in the ToTEF. The following describes the post-processing phase components.

   1- Matching

The matching component supports service selecting based on the trustworthiness of services and requestors' trust preferences. In particular, it returns services that match requestor's trust preferences by matching the preferences with services' TMs. As a result, the service broker returns a number of services with a similar functionality and different trust ratings. The trust matching model is presented in Section 7.3.

   2- Feedback

Service consumers may provide their feedback about the services and service providers to represent their satisfaction or dissatisfaction. Since consumers may provide unfair feedback, it is essential to impede such feedback. Hence, the feedback component is responsible for addressing the unfair feedback problem.

### 3- Risk Remedies

Since trust and risk are related, it is important to provide remedies in case an unexpected event occurs. In addition to trusting services and providers, requestors need to trust service brokers, which are TTPs. Hence, this component supports risk remedies for service brokers.

### 4- Self-Adjustment

The self-adjustment component considers the dynamic nature of trust and is responsible for trust degradation, trust declining, and trust re-building. Trust rates should be continuously evaluated to reflect recent interactions. The trust mediator should be able to decline and rebuild trust, and the service providers should be able, through the service broker, to review their trust rates and consumer feedback. This can provide a good opportunity for service providers to improve their services, understand consumer needs, and build their QoBiz.

### 5- Bias Detection

The bias detection component is responsible for detecting trust biases, which may occur if there is a significant decline from the stored rate to the evaluated one. The trust mediator can monitor services to detect biases.

### 6- Culture Adjustment

The Web is an open environment that spans different countries, regulations, and cultures. It is important to consider cultural differences when establishing trust and selecting services. Thus, culture adjustment component is responsible for mitigating cultural differences.

### 7- Reward and Punishment

This component is responsible for punishments and rewards. Service brokers can motivate providers to contribute positively to the network and punish other providers who act negatively and try to disrupt the system.

### 8- Post-Monitoring

The monitoring component in the processing and evaluation phase is dedicated to the trust bootstrapping process. This component plays an important role in trust management, since the self-adjustment and bias detection components need to monitor services for dynamically detecting changes in their behaviour that may affect their rates as well as the rates of their providers. The post-monitoring component can periodically monitor services to test their trust

rates and their providers' trust rates. In addition, post-monitoring is important for monitoring the interactions between requestors and services to measure the TM of the consumed services.

## 4.4   Summary

This chapter presents the trust definition and principles, the trust-based SOA, and a trust mediator framework (ToTEF) for trust establishment and trust-based service selection. ToTEF considers the identified trust definition and principles to include necessary components for building a concrete solution of trust. Consequently, the unified trust framework contains the necessary components for trust bootstrapping, trust evaluation, trust management, and trust challenges, such as unfair feedback, trust bias, and culture differences. The trust bootstrapping process requires the components of the pre-processing phase and the processing and evaluation phase as well as the matching component of the post-processing phase of the ToTEF. The service broker conducts the trust bootstrapping process by acting as a requestor of the services. This work do not consider mutual trust between service requestors and services or service providers.

The subsequent chapters present the trust establishment process for services and service providers. Figure 4.3 depicts the required steps for trust establishment. First, the trust information, or trust metrics, for services and service providers is identified in Chapter 5. Next, trust models for bootstrapping and evaluating the trust rates of trust metrics are developed in Chapter 6. Finally in Chapter 7, the trust models for bootstrapping and evaluating the trust rates of services and service providers are built. Since trust models for services and providers are based on the trust metrics models, which are based on the trust metrics, the next chapter presents the trust metrics for services and service providers.

| Trust Metrics<br>*Chapter 5* | Trust Models for<br>Trust Metrics<br>*Chapter 6* | Trust Models for<br>Services and<br>Service<br>Providers<br>*Chapter 7* |
|---|---|---|
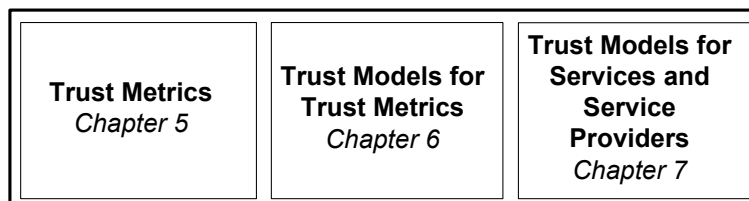
*Figure 4.3:  Trust Establishment Process for Trust Metrics, Services, and Service Providers.*

# Chapter 5

# Trust Metrics

*"Trust him as much as you would a rattlesnake with a silencer on its rattle."* Dean Acheson

Since SOA has been extended to support trust and a trust mediator framework has been built, the next step involves establishing trust for services and service providers. Because trust is based on information, services and service providers need to provide their information to establish trust. This chapter presents Trust Metrics (TM), which are defined as *the information of an entity that is required and used to evaluate the trustworthiness of that entity.* An entity, in this work, can be a service or a service provider. Trust metrics are information about the first party, the service or service provider, where a service provider offers information about itself and its own services for the evaluation of their trustworthiness. For example, a service provider can present its reliability as a TM, and requestors can trust the service based on the service's reliability TM. The term 'metric' represents the need to quantify the information and define it as a set of measures necessary for evaluating trust.

To identify TMs, the information required to build trust for services and service providers should be explored. Services and providers may support different TMs. In SOA, transactions may span a range of domains and organizations. In particular, services and service providers may traverse many domains with different properties and requirements. Also, requestors have many requirements and seek different services' properties. Hence, a domain may need to support a range of TMs, requiring identification of a unified set of TMs.

Since QoS can be used as important information for establishing trust, this work defines QoS parameters as TMs. To identify TMs for an open environment, it is important to generalize a list of TMs applicable for most services and service providers. Specifically, QoS should be generalized for diverse services and service providers. To define a unified TM categorization, various QoS parameters need to be extracted from the literature. Some QoS are easily measured by using a mathematical formula while others are more difficult to measure and require other approaches to quantify their use in a trust rating algorithm and calculation. For example, although security is not a quantifiable element, there are different degrees of security that a system can provide based on the system's support level of security technologies.

Based on the aggregated QoS in the literature, as presented in Section 3.2, this work categorizes QoS into two categories: objective QoS and subjective QoS, as shown in Figure 5.1. Objective QoS can be measured and contain formulas for measurement. Subjective QoS are difficult to measure and require other quantification approaches.
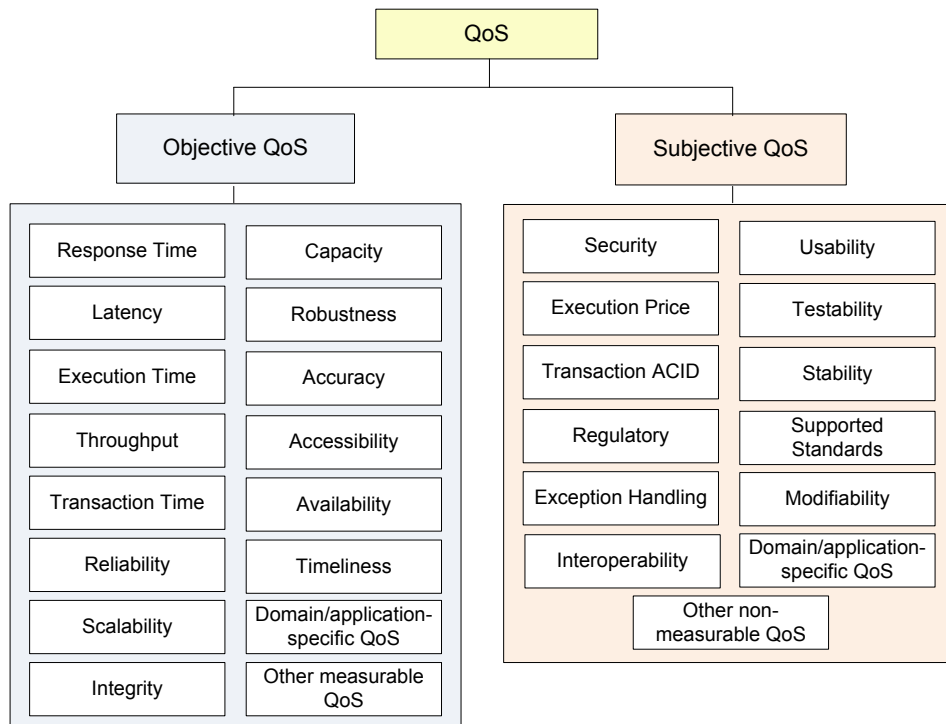


*Figure 5.1: QoS Properties for Services.*

In this chapter, Section 5.1 presents the proposed trust metrics, Section 5.2 discusses the trust metrics publication approaches, and Section 5.3 summarizes the chapter.

# 5.1 The Proposed Trust Metrics (TM)

In this work, TMs consist of a unified set of metrics that incorporate information from diverse domains, such as governments, marketing companies, and banks, QoS, service properties, and provider properties. Some TMs may not be applied to all services, and it is possible to use other TMs not included in the proposed set.

Figure 5.2 presents the TMs for services and service providers, which are categorized into Service Trust Metrics (STM) and Service Provider Trust Metrics (PTM). QoS properties are part of the TMs. As QoS categorization, STMs are categorized into Objective Service Trust Metrics (OSTM) and Subjective Service Trust Metrics (SSTM). PTMs include service provider properties, services properties, and important clues to support providers' trustworthiness. Figure 5.3 depicts the overlap between QoS, OSTM, SSTM, and PTM. TMs need to be published by service providers when they publish their services. As QoS, it is essential to quantify TMs. In the following subsections, STM and PTM are presented.

## 5.1.1 Service Trust Metrics (STM)

STMs include the trust information about services and their properties. STMs are categorized into OSTM and SSTM; OSTMs are TMs that have a formula for measurement and can use a monitoring approach, whereas SSTMs require a different approach for quantification, such as feedback approach. The following subsections present the OSTM and SSTM:

### 5.1.1.1 Objective Service Trust Metrics (OSTM)

OSTMs for services include *objective QoS properties*, such as response time, latency, execution time, throughput, transaction time, reliability, scalability, integrity, capacity, robustness, accuracy, accessibility, availability, and timeliness as well as domain/application-specific properties.
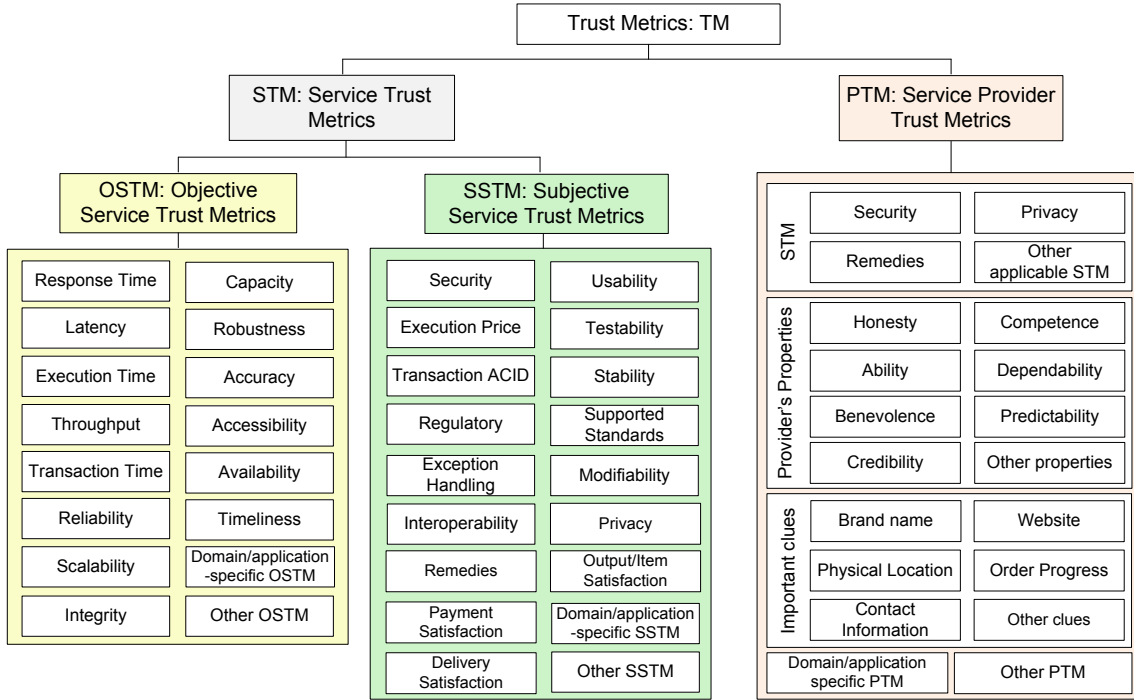
*Figure 5.2: Trust Metrics.*

The abbreviations for some of the OSTMs are as follows: execution time OSTM ($OSTM_e$), latency OSTM ($OSTM_l$), response time OSTM ($OSTM_r$), and throughput OSTM ($OSTM_{thp}$). The definitions for these metrics are presented in Table 3.1.

### 5.1.1.2 Subjective Service Trust Metrics (SSTM)

SSTMs include *subjective QoS properties*, such as security, execution price, transaction ACID (Atomicity, Consistency, Isolation, and Durability), regulatory, exception handling, interoperability, usability, testability, stability, supported standards, and modifiability; *other services properties*, such as remedies, payment satisfaction, delivery satisfaction, privacy, and output/item satisfaction; and *domain/application-specific properties*. The following presents some SSTMs:

- Remedies SSTM ($SSTM_{rem}$): Services should provide remedies in case of unexpected event. Each service has different remedies; for example, if a shipment service experiences a delay, the service can offer a reduction in the shipment price as a remedy. Also, if
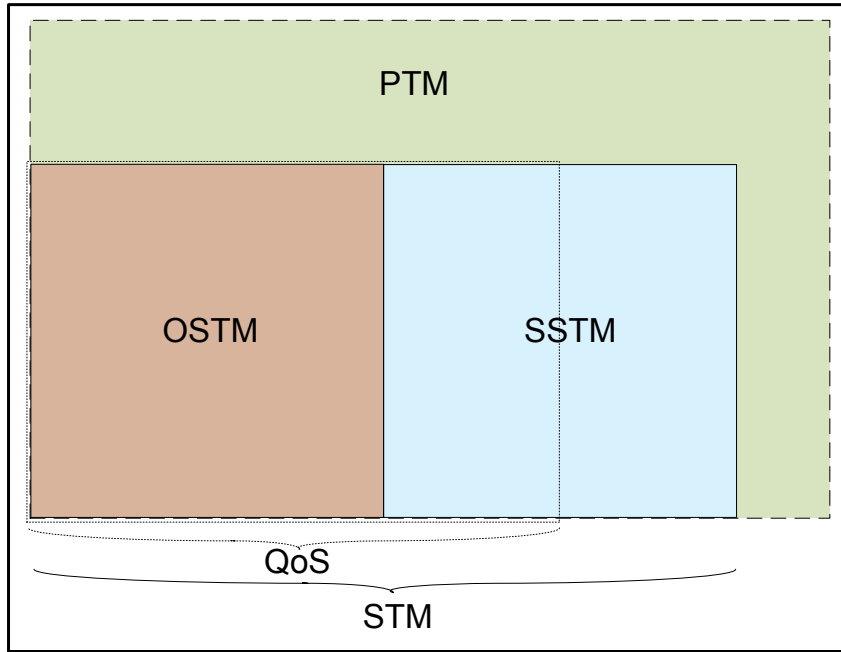
*Figure 5.3: The Overlap between the Trust Metrics.*

a service provides a video that plays slowly on a customer's subscribed network, the service should increase the bandwidth for the customer.

- Security SSTM ($SSTM_{sec}$): A requestor can trust a service or provider based on security, which is an important factor in trust establishment process. For example, a requestor may require an online banking service, for which security is necessary in protecting the customer's money.

- Privacy SSTM ($SSTM_{prv}$): A requestor can trust a service or provider based on privacy, which is an important factor in the trust establishment process. For example, requestors may make online purchases and use a place order service, where they enter their information, such as name and address. In this case, requestors need to trust the service to maintain their privacy and refrain from disclosing their information.

- Payment Satisfaction SSTM ($SSTM_{pym}$): Payment satisfaction refers to the degree of user satisfaction for the offered service on the basis of the payment. For example, this

metric measures whether the service charges the users the stated amount or whether users pay unexpected fees.

- Output/Item satisfaction SSTM ($SSTM_{out}$): Output/item satisfaction refers to the user's degree of satisfaction for the offered services based on the output/item provided. For example, this metric assesses whether users receive the same output/item they ordered or expected, their degree of satisfaction with the output/item, and the quality of the output/item they received.

- Delivery satisfaction SSTM ($SSTM_{delv}$): This metric refers to the degree of user satisfaction for the delivery of the service. For example, it assesses whether the item is delivered on time and if the customer can return the item in case of dissatisfaction.

### 5.1.2 Provider Trust Metrics (PTM)

The process of rating service providers can enhance the requestors' trust in the providers and their services. If requestors have the alternative of choosing among services from different providers, they can select a service from a provider with the highest trust rating.

The trustworthiness of a service provider is based on the *trustworthiness of its services* and the rates of its TMs. A trustworthy service provider should behave on the basis of its advertised properties, its PTM, and the advertised properties of its services, STM. Thus, STMs are included as PTMs. Service providers have many properties that can be considered as useful PTM for building trust. In addition, providers can offer important clues that can support their trustworthiness and assess requestors on their service selection. The following subsections presents the PTMs.

**Providers' Services Properties**

The trustworthiness of service providers is based on the trustworthiness of its services. Therefore, STMs are implicitly included as metrics to evaluate the trust of service providers. Any STM can be explicitly identified as a PTM to emphasize its importance, and it can be used as

a PTM, such as security PTM ($PTM_{sec}$), privacy PTM ($PTM_{prv}$), remedies PTM ($PTM_{rem}$), and other STMs, such as reliability, availability, payment satisfaction, and delivery satisfaction.

**Providers' Properties**

Service provider properties that can be considered as PTMs include honesty, ability, benevolence, credibility, competence, dependability, predictability as well as other properties, such as courtesy, comfort, efficiency, effectiveness, flexibility, promptness, and responsiveness. For example, the competence PTM ($PTM_{comp}$) and honesty PTM ($PTM_{hons}$) of a service provider will influence the requestor's decision for using the provider's services. Competence and honesty are identified in Section 3.2.

**Important Clues**

Service providers can provide important clues to support their trustworthiness and enhance their opportunity for being chosen by requestors. For example, a service provider who has contact information may encourage requestors to select its services because they can contact the service provider in case of a mishap. The following presents some clue PTMs:

- Provider brand name PTM ($PTM_{brand}$): A brand name is a popular name that can be established through long-term interaction with the provider. A trustworthy service provider can associate its behaviour to a name that will become popular. Service providers who have brand names may encourage requestors to use their services, which will positively influence the economic growth of the service providers. Trust-based systems can play an important role in the establishment of brand names for service providers. A service provider can provide a name and then the service broker can build a brand name for the service provider based on the provider's trustworthiness.

- Provider physical location PTM ($PTM_{loc}$): A physical location, such as a store, may increase a provider's trustworthiness. In this case, the requestor can select a service from some locations but not others.

- Provider contact information PTM ($PTM_{inf}$): Contact information, such as telephone numbers and e-mails have a great impact in the assessment of a service provider's trustworthiness. Providers with contact information allow requestors to contact them for resolving issues.

- Provider website PTM ($PTM_{wsite}$): A service provider who has a website may encourage the requestor to trust the provider and use their services. Moreover, websites may contain information, such as security certification, that can assess the trustworthiness of service providers.

- Provider order progress PTM ($PTM_{ord}$): While order progress is more easily obtained in the physical world, it should also be provided online in order to support requestors' trust of providers.

## 5.2   TM Publication Approaches

Providers should advertise their services and publish the TMs when they publish their services. There is a range of different TMs identified in this work. However, different TMs require various approaches for publishing and rating. Therefore, a set of TMs from Figure 5.2 will be selected and their publishing and rating approaches will be identified. Figure 5.4 shows the selected TMs, which are chosen from different TM categories. Therefore, the TMs within the same category may have similar approaches for publishing and rating.

This section presents the publication approaches for the selected TMs. Table 5.1 shows the selected TMs and indicates whether they can be published by service providers. At publishing time, service providers publish the TMs that they support and provide the TMs values. The TMs that are not supported or published are not considered in the trust establishment process. However, some PTMs should not be published because their ratings are based on other TMs. For example, the rate of security PTM is based on the rate of the security SSTM. The following TMs are the selected TMs as presented in Table 5.1:

- OSTMs, which include execution time ($OSTM_e$), response time ($OSTM_r$), latency ($OSTM_l$), and throughput ($OSTM_{thr}$). Providers may publish the OSTMs of their services in a
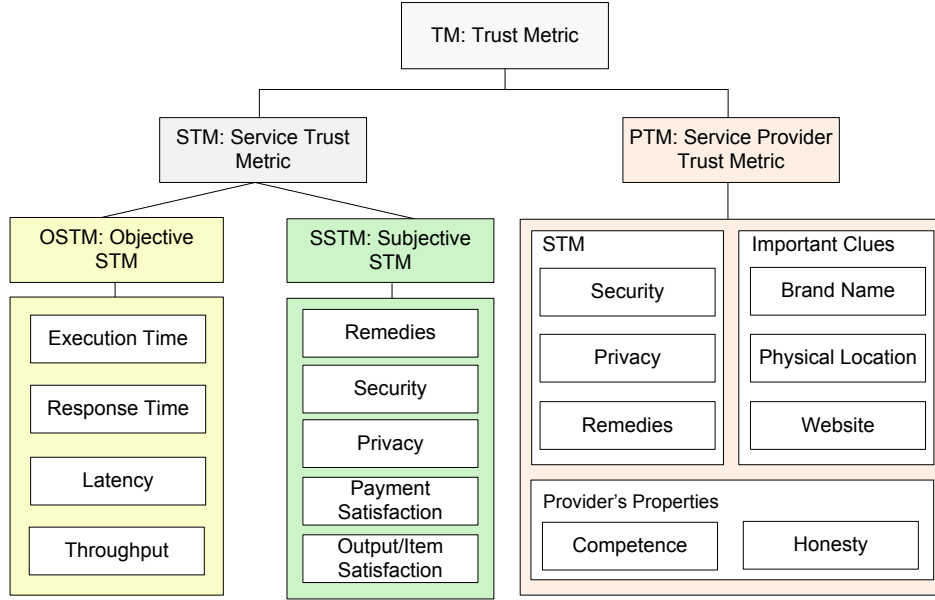
*Figure 5.4: The Selected Service and Service Provider Trust Metrics.*

range of values between the minimum and maximum. For example, a service provider may publish a service, s, with $OSTM_r(s) = [15 - 35]$ ms, a range of values with minimum and maximum values of 15 ms and 35 ms respectively. Accordingly, $OSTM_r(s) = [15-35]$ ms means that the supported range of service response time is in a range between 15 ms and 35 ms.

- SSTMs, which include remedies ($SSTM_{rem}$), security ($SSTM_{sec}$), privacy ($SSTM_{prv}$), payment satisfaction ($SSTM_{pym}$), and output satisfaction ($SSTM_{out}$). SSTMs have a value of 1 if a service publishes and support the SSTM. For example, if the service provider support the $SSTM_{out}$, then $SSTM_{out} = 1$. Providers publish the $SSTM_{rem}$, $SSTM_{sec}$, or $SSTM_{prv}$ and provide their policies in security, privacy and remedies.

- PTMs, which are: remedies ($PTM_{rem}$), security ($PTM_{sec}$), privacy ($PTM_{prv}$), brand name ($PTM_{brand}$), competence ($PTM_{comp}$), honesty ($PTM_{hons}$), website ($PTM_{wsite}$), and physical location ($PTM_{loc}$). $PTM_{brand}$, $PTM_{loc}$, and $PTM_{wsite}$ values are published by providers. $PTM_{sec}$, $PTM_{prv}$, and $PTM_{rem}$ are based on the $SSTM_{sec}$, $SSTM_{prv}$, and $SSTM_{rem}$ of the providers' services. $PTM_{comp}$ and $PTM_{hons}$ are evaluated for all

Table 5.1: Trust Metrics and their Published Values.

| | | TM | Can be published | Published values |
|---|---|---|---|---|
| STM | OSTM | $OSTM_e$: Execution time | Yes | Execution time |
| | | $OSTM_r$: Response time | Yes | Response time |
| | | $OSTM_l$: Latency | Yes | Latency time |
| | | $OSTM_{thr}$: Throughput | Yes | Throughput |
| | SSTM | $SSTM_{rem}$: Remedies | Yes | Policies |
| | | $SSTM_{sec}$: Security | Yes | Policies |
| | | $SSTM_{prv}$: Privacy | Yes | Policies |
| | | $SSTM_{pym}$: Payment satisfaction | No | 1 |
| | | $SSTM_{out}$: Output satisfaction | No | 1 |
| PTM | STM | $PTM_{rem}$: Remedies | No | - |
| | | $PTM_{sec}$: Security | No | - |
| | | $PTM_{prv}$: Privacy | No | - |
| | Pr properties | $PTM_{brand}$: Brand name | Yes | Brand name |
| | | $PTM_{comp}$: Competence | No | - |
| | | $PTM_{hons}$: Honesty | No | - |
| | Clues | $PTM_{wsite}$: Website | Yes | Website |
| | | $PTM_{loc}$: Physical location | Yes | Physical location |
| TM: Trust metric, STM : Service TM, PTM : provider TM, Pr: Provider | | | | |

providers based on other TMs through long-term interaction. Thus, providers do not need to publish the $PTM_{sec}$, $PTM_{prv}$, $PTM_{rem}$, $PTM_{comp}$, and $PTM_{hons}$.

## The published STM

The STMs published by service providers are stored in the rating registry to be used for rating the STMs. The following presents the collected STMs in a matrix format. Each row represents a service, and each column represents one of the STMs (m:s and n:STM). For example, $STM_{21}$ is the first STM of the service $s_2$.

$$\mathbf{STM} = \begin{bmatrix} STM_{11} & STM_{12} & \dots & STM_{1n} \\ STM_{21} & STM_{22} & \dots & STM_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ STM_{m1} & STM_{m2} & \dots & STM_{mn} \end{bmatrix}$$

**The published PTM**

The PTMs published for each service provider are stored in the rating registry to be used for rating purposes and to support the trustworthiness of service providers. The following presents the collected PTMs in a matrix format. Each row represents a service provider and each column represents one of the PTMs (m:provider and n:PTM). For example, $PTM_{12}$ is the second PTM of the service provider $pr_1$.

$$\mathbf{PTM} = \begin{bmatrix} PTM_{11} & PTM_{12} & \dots & PTM_{1n} \\ PTM_{21} & PTM_{22} & \dots & PTM_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ PTM_{m1} & PTM_{m2} & \dots & PTM_{mn} \end{bmatrix}$$

## 5.3   Summary

This chapter presents trust metrics for services and service providers. Trust metrics include the properties and other information about services and providers. These metrics are extendable and support domain-specific properties. Service providers publish the trust metrics when they publish their services. The trust mediator of the service broker stores the published trust metrics values in the rating registry. Based on the identified trust metrics in this chapter, the next step involves building trust models to rate the trust metrics, which are presented next in Chapter 6.

# Chapter 6

# Trust Models for Trust Metrics

*"Where large sums of money are concerned, it is advisable to trust nobody."* Agatha Christie

The Trust Metrics (TM) of services (STM) and service providers (PTM), which were introduced in Chapter 5, need to be rated. When services are being published, the supported TMs by services and providers are provided. Accordingly, the trust mediator begins rating the TMs to initialize their trust rates through bootstrapping. Thus, the bootstrapped trust rates of TMs ($T_{TM}$) will be used to bootstrap the trust rates of services ($T_s$) and the trust rates of service providers ($T_{pr}$).

To rate an entity, it is necessary to build a trust model, which represents the trust evaluation steps that are followed for rating an entity. For example, a trust model can be represented as an algorithm with steps and mathematical equations for computing trust measures. Specifically, this work considers building trust models for TMs, services, and service providers, and this chapter presents trust models for TMs. Subsequently, the next chapter provides the trust models for services and service providers.

The trust mediator rates the TMs using different trust models for various TMs. Accordingly, this work builds trust models for the selected TMs, which are presented in Figure 5.4. The selected TMs and their trust rates ($T_{TM}$) are shown in Table 6.1, which includes different trust bootstrapping and evaluation approaches as well as trust rating scales. $T_{TM}$ is evaluated by the TM trust evaluation component of the trust mediator framework and stored in the rating registry.

Table 6.1: Trust Ratings of the TM ($T_{TM}$) and the Evaluation Approaches.

| | $T_{TM}$ | Can TM be trust bootstrapped ? | Trust bootstrapping and evaluation approach | Rating scale | Rating values |
|---|---|---|---|---|---|
| $T_{STM}$ | $T_{OSTM_e}$ | Yes | Monitoring (Algorithm 1, Section 6.2) | Continuous | [1-10] |
| | $T_{OSTM_r}$ | Yes | Monitoring (Algorithm 1, Section 6.2) | Continuous | [1-10] |
| | $T_{OSTM_l}$ | Yes | Monitoring (Algorithm 1, Section 6.2) | Continuous | [1-10] |
| | $T_{OSTM_{thr}}$ | Yes | Monitoring (Algorithm 2, Section 6.2) | Continuous | [1-10] |
| | $T_{SSTM_{rem}}$ | Yes | Certification | Discrete | 1-10 |
| | $T_{SSTM_{sec}}$ | Yes | Certification | Discrete | 1-10 |
| | $T_{SSTM_{prv}}$ | Yes | Certification | Discrete | 1-10 |
| | $T_{SSTM_{pym}}$ | No | Feedback | Continuous | [1-10] |
| | $T_{SSTM_{out}}$ | No | Feedback | Continuous | [1-10] |
| $T_{PTM}$ | $T_{PTM_{rem}}$ | Yes | Algorithm 3, Section 6.4 | Discrete | 1-10 |
| | $T_{PTM_{sec}}$ | Yes | Algorithm 3, Section 6.4 | Discrete | 1-10 |
| | $T_{PTM_{prv}}$ | Yes | Algorithm 3, Section 6.4 | Discrete | 1-10 |
| | $T_{PTM_{brand}}$ | No | Algorithm 4, Section 6.4 | Two-scale | 0/1 |
| | $T_{PTM_{comp}}$ | No | Algorithm 4, Section 6.4 | Two-scale | 0/1 |
| | $T_{PTM_{hons}}$ | No | Algorithm 4, Section 6.4 | Two-scale | 0/1 |
| | $T_{PTM_{wsite}}$ | No | Feedback | Two-scale | 0/1 |
| | $T_{PTM_{loc}}$ | No | Feedback | Two-scale | 0/1 |
| $T_{TM}$: Trust rate of TM, $T_{STM}$: Trust rate of STM, $T_{PTM}$: Trust rate of PTM | | | | | |

The trust mediator is incapable of trust bootstrapping some TMs because they are based totally in the consumer feedback; thus, they are rated on the basis of feedback approach after consuming the services. For example, the payment satisfaction TM ($SSTM_{pym}$) of a service is rated and its trust rate ($T_{SSTM_{pym}}$) is evaluated by the feedback approach. However, feedback can provide useful information on customer satisfaction for all TMs and update the service's trust rate accordingly. Furthermore, using the feedback approach, the trust mediator can directly obtain opinions from the service consumer. In addition, feedback helps to identify violations, which lower the rate of the services and their providers.

The trust bootstrapping and evaluation approaches as well as the rating scales will be discussed throughout the chapter. This chapter presents the models for evaluating the trust rates of the TMs ($T_{TM}$). In particular, the trust models for $T_{OSTM}$, $T_{SSTM}$, and $T_{PTM}$ are examined.

The trust measures should be understandable by the requestor; therefore, a trust model should be simple as well as efficient through a consideration of the trust principles. This chapter is organized as follows. Section 6.1 presents trust rating scales, and Section 6.2 presents OSTM trust models. While Section 6.3 presents SSTM trust models, PTM trust models are discussed in Section 6.4. Finally, Section 6.5 summarizes the chapter.

## 6.1 Trust Rating Scales

A trust rate is measurable and represents the degree of trust. The trust rate can have discrete or continuous values. Since service registries have a large number of services that have the same functionality published by different providers, a two-scale rating, which rates a service as either trustworthy or untrustworthy, and a discrete scale rating, which ranks a service as high, medium, or low, would be inadequate. Specifically, these rating scales increase the chance that services with similar functionalities will have the same trust rates, and service requestors will not be able to distinguish between them. A continuous scale, which involves values within the interval of [0,10], provides a range of data points that helps to distinguish between services. In particular, a continuous scale helps requestors to make comparisons and select between services with the same functionality.

This work uses the continuous rating scale with a range of real data numbers in the interval of [0,10] to rate services and service providers. While a rating of ten indicates that a service or provider is highly trusted, a rating of zero means that it is most distrusted. For example, if $T_{s1} = 9.8$ and $T_{s2} = 7.2$, then service s1 is more trusted than service s2. Although fuzzy logic models can be used to obtain these values, such models are beyond the scope of this thesis. Also, service requestors may consider other criteria for service selection, such as the service providers and PTMs rates.

However, this work uses different rating scales to rate various TMs, as shown in Table 6.1. For instance, the monitoring approach is used to rate OSTM, and the $T_{OSTM}$ scale is continuous. The SSTMs and PTMs of security, privacy, and remedies have a discrete number of levels, and so their rating scale is discrete with ten levels (i.e., 1-10 levels). Payment and output satisfaction SSTMs are based on consumer feedback along a continuous scale. The provider's brand name,

```
┌─────────────────────────────────────────────────────────────────────┐
│ TM Publication          OSTM Trust Models                            │
│ approaches                  Section 6.2                              │
│ Section 5.2                                                          │
│                                                                      │
│  Publishing           Collecting          Trust evaluating           │
│ ┌──────────┐      ┌──────────────┐      ┌──────────────┐            │
│ │ Publish  │─────▶│ Monitor the  │─────▶│ Trust rate   │            │
│ │ OSTM     │      │ published    │      │ OSTM         │            │
│ │          │      │ OSTM         │      │              │            │
│ └──────────┘      └──────────────┘      └──────────────┘            │
└─────────────────────────────────────────────────────────────────────┘
```
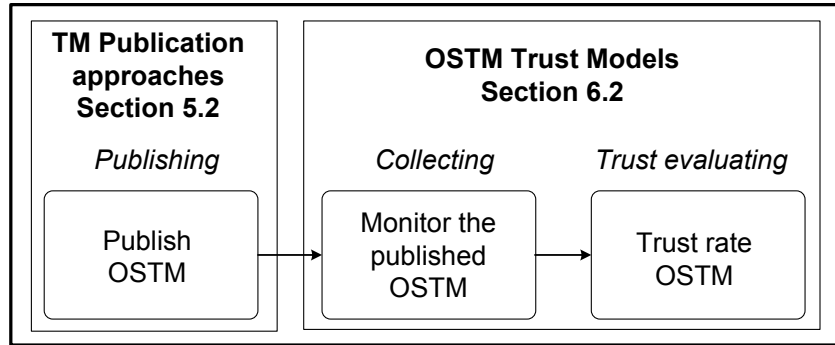
*Figure 6.1: OSTM Trust Models: The Steps and an Approach to Rating OSTM.*

competence, honesty, website, and physical location metrics values are either "yes" or "no", thus having a two-scale rating of 0 or 1.

## 6.2 OSTM Trust Models

To evaluate $T_{OSTM}$, it is necessary to build OSTM trust models. The trust models for OSTM follow different steps, and they are based on the monitoring approach, as depicted in Figure 6.1. After OSTMs are publishing, they need to be collected using the monitoring approach. Then, the monitored OSTM and the published OSTM are compared (compliance with the stated OSTM) and used to evaluate the trust rate of the OSTM ($T_{OSTM}$) to ensure that services provide what they promised.

While some OSTMs are better if the monitored values are lower than the published values, other OSTMs are preferable if the monitored values are higher than the published values. This variation should be considered when building a trust model for OSTM to evaluate the $T_{OSTM}$. Therefore, two different trust models are defined to evaluate $T_{OSTM}$, one where lower values are better and one where higher values are preferable.

76

# The "Lower OSTM is Better" Trust Model

Some OSTMs, such as response time ($OSTM_r$), will have higher trust rates if their monitored values are lower than their published values. For example, if the published value of $OSTM_r = 15ms$, and the monitored value of $OSTM_r = 12ms$, the monitored value is lower than the published value and the $T_{OSTM_r}$ will be high.

Algorithm 1 shows the trust model for evaluating the $T_{OSTM}$ for such OSTM. At the first step in the model, if the provider publishes a service with a range of values, then the provided value will be assigned the maximum value within the range; for example, if $OSTM_r = [10 - 20]ms$ , then $OSTM_{r_{provided}} = Max[10 - 20]ms = 20ms$ . The difference (diff) is 10 times the difference between the provided and the collected or monitored values divided by the provided value. If diff is positive, the collected value of OSTM is lower than the provided value and OSTM will receive a trust rate of 10. If diff is negative, the collected value of OSTM is higher than the provided value and OSTM will receive a trust rate in the range of [0-10), depending on the collected value. A higher collected value indicates a lower trust rate for the OSTM, which means less trustworthy.

---

**Algorithm 1** Lower OSTM is Better.

---

**if** TM is a range of values **then**
    $OSTM_{provided} = Max\{range\ of\ OSTM\ values\}$
**end if**
Lets $diff = \frac{OSTM_{provided} - OSTM_{collected}}{OSTM_{provided}} * 10$
**if** $diff < 0$ **then**
    $T_{OSTM} = 10 + diff$
**else**
    $T_{OSTM} = 10$
**end if**

---

Figure 6.2 depicts the relationship between diff and the possible trust rate values for a range of lower OSTM values. For example, if a service publishes $OSTM_e = [10 - 20]$ ms and the monitor obtains $OSTM_e = 16$ ms (i.e., values in [10-20] ms), then $diff = 2$ and $T_{OSTM_e} = 10$; if $OSTM_e = 6$ ms, then $diff = 7$ and $T_{OSTM_e} = 10$. Finally, if $OSTM_e = 28$ ms, then $diff = -4$ and $T_{OSTM_e} = 6$. Moreover, the $OSTM$ can be monitored several times and the

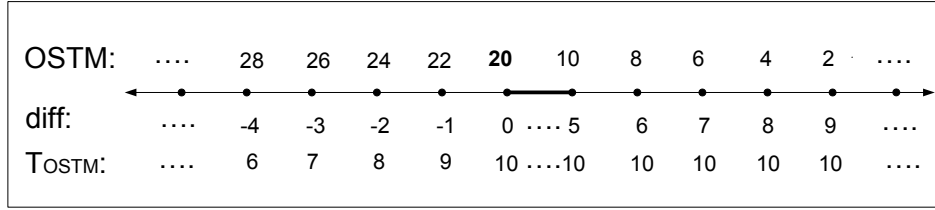| OSTM: | .... | 28 | 26 | 24 | 22 | **20** | 10 | 8 | 6 | 4 | 2 | .... |
| diff: | .... | -4 | -3 | -2 | -1 | 0 .... 5 | | 6 | 7 | 8 | 9 | .... |
| T$_{OSTM}$: | .... | 6 | 7 | 8 | 9 | 10 ....10 | | 10 | 10 | 10 | 10 | .... |

*Figure 6.2: Low OSTM Possible Range of Values, diff Values, and Trust Rates.*

final $OSTM$, which will be used in the evaluation, will equal the average of the monitored values.

## The "Higher OSTM is Better" Trust Model

Some OSTMs, such as throughput ($OSTM_{thp}$), will have higher trust rates if their monitored value are higher than their published values. For example, if the published value of $OSTM_{thp} = 0.7$ and the monitored value of $OSTM_{thp} = 0.9$, the monitored value is higher than the published value and the $T_{OSTM_{thp}}$ will be high.

Algorithm 2 presents the trust model for evaluating the $T_{OSTM}$ for such OSTM. If the provider publishes a service with a range of values, then the provided value will be assigned the minimum value within the range; for example, if $OSTM_{thp} = [1-1.1]$, then $OSTM_{thp_{provided}} = Min[1 - 1.1] = 1$. The difference (diff) is 10 times the difference between the provided and the collected or monitored values divided by the provided value. If diff is negative, the collected value of OSTM is higher than the provided value and OSTM will receive a trust rate of 10. If diff is positive, the collected value of OSTM is lower than the provided value and OSTM will receive a trust rate in the range of [0-10), depending on the collected value. A lower collected value indicates a lower trust rate for the OSTM, which means less trustworthy.

Figure 6.3 illustrates the relationship between diff and the possible trust rate values for a range of higher OSTM values. For example, if a service publishes $OSTM_{thp} = [1-1.1]$ and the monitor obtains $OSTM_{thp} = 1.05$ (i.e., values in [1-1.1]), then $diff = -0.5$ and $T_{OSTM_{thp}} = 10$; if $OSTM_{thp} = 1.4$, then $diff = -4$ and $T_{OSTM_{thp}} = 10$. Finally, if $OSTM_{thp} = 0.7$, then $diff = 3$ and $T_{OSTM_{thp}} = 7$.

**Algorithm 2** Higher OSTM is Better.

**if** TM is a range of values **then**
    $OSTM_{provided} = Min\{range\ of\ OSTM\ values\}$
**end if**
Lets $diff = \frac{OSTM_{provided} - OSTM_{collected}}{OSTM_{provided}} * 10$
**if** $diff > 0$ **then**
    $T_{OSTM} = 10 - diff$
**else**
    $T_{OSTM} = 10$
**end if**

| OSTM: | …. | 1.5 | 1.4 | 1.3 | 1.2 | 1.1 | **1** | 0.9 | 0.8 | 0.7 | 0.6 | …. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| diff: | …. | -5 | -4 | -3 | -2 | -1 …. 0 | | 1 | 2 | 3 | 4 | …. |
| T$_{OSTM}$: | …. | 10 | 10 | 10 | 10 | 10 …. 10 | | 9 | 8 | 7 | 6 | …. |
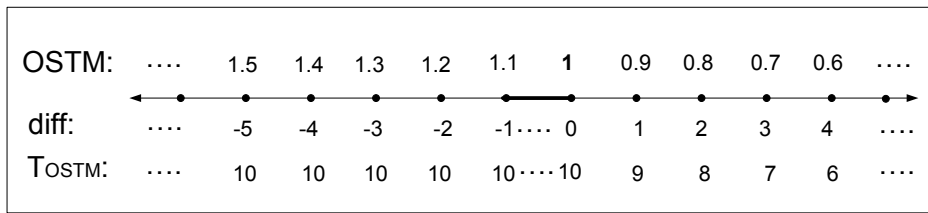
*Figure 6.3: High OSTM Possible Range of Values, diff Values, and Trust Rates.*

## $T_{OSTM}$ **Collection Values**

The measured $T_{OSTM}$ values for each service are stored in the rating registry to support a re-questor's preferences, thus allowing the requestor to select a service based on the service's trustworthiness among a set of preferred OSTMs. The following represents the collected $T_{OSTM}$ values in a matrix format, where each row represents a service and each column represents one of the $T_{OSTM}$ values (m:s and n:$T_{OSTM}$).

$$\mathbf{T_{OSTM}} = \begin{bmatrix} T_{OSTM_{11}} & T_{OSTM_{12}} & \dots & T_{OSTM_{1n}} \\ T_{OSTM_{21}} & T_{OSTM_{22}} & \dots & T_{OSTM_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ T_{OSTM_{m1}} & T_{OSTM_{m2}} & \dots & T_{OSTM_{mn}} \end{bmatrix}$$
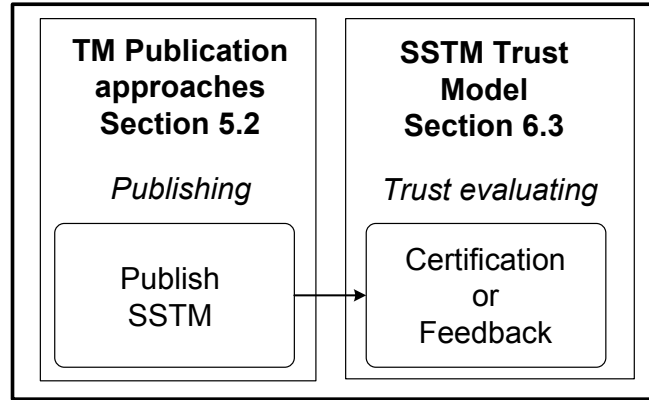
*Figure 6.4: SSTM Trust Models: The Steps and Approaches to Rating SSTM.*

## 6.3 SSTM Trust Models

Different SSTMs are rated using various steps and approaches, i.e., different SSTM trust models. In general, the trust models for SSTM use the certification approach or the feedback approach. SSTMs include remedies ($SSTM_{rem}$), security ($SSTM_{sec}$), privacy ($SSTM_{prv}$), payment satisfaction ($SSTM_{pym}$), and output/item satisfaction ($SSTM_{out}$). Figure 6.4 shows the steps and approaches used to rate SSTMs.

- The trust models for $SSTM_{pym}$ and $SSTM_{out}$ are based on the feedback approach, and they will have initial rates after receiving feedback from the service consumer.

- The trust models for $SSTM_{rem}$, $SSTM_{sec}$, and $SSTM_{rem}$ are based on the certification approach. After the SSTM policies are published, the degree of the $SSTM_{sec}$, $SSTM_{prv}$, and $SSTM_{rem}$ will be evaluated. The evaluated degree will be assigned as trust rate for the SSTM and will be certified. For example, the degree of security SSTM ($SSTM_{sec}$) of a service *s* is evaluated based on the provided $SSTM_{sec}$ policies. If the evaluated degree of $SSTM_{sec}$ is 4, then $T_{SSTM_{sec}}(s) = degree = 4$. Subsequently, the $SSTM_{sec}$ will be certified. $SSTM_{prv}$ providers can present their privacy policies in the collected information of their system, and $T_{SSTM_{prv}}$ will be assigned a level of trust for privacy. For $SSTM_{rem}$, policies may include a list of possible risk and risk remedies, and $T_{SSTM_{rem}}$

will be assigned a level of trust for risk remedies. However, the certification process for assigning trust levels to SSTMs is beyond the scope of this thesis.

### $T_{SSTM}$ **Collection Values**

The measured $T_{SSTM}$ values for each service are stored in the rating registry to support a requestor's trust preferences, thus allowing the requestor to select a service based on the service's trustworthiness among a set of preferred SSTMs. The following represents the collected $T_{SSTM}$ values in a matrix format, where each row represents a service and each column represents one of the $T_{SSTM}$ values (m:s and n:$T_{SSTM}$).

$$
\mathbf{T_{SSTM}} = \begin{bmatrix}
T_{SSTM_{11}} & T_{SSTM_{12}} & \cdots & T_{SSTM_{1n}} \\
T_{SSTM_{21}} & T_{SSTM_{22}} & \cdots & T_{SSTM_{2n}} \\
\vdots & \vdots & \ddots & \vdots \\
T_{SSTM_{m1}} & T_{SSTM_{m2}} & \cdots & T_{SSTM_{mn}}
\end{bmatrix}
$$

## 6.4 PTM Trust Models

Building PTMs trust models to evaluate $T_{PTM}$ is based on various steps and involves different approaches that includes the certification approach, a long-term interaction approach, or feedback approach. PTMs include remedies ($PTM_{rem}$), security ($PTM_{sec}$), privacy ($PTM_{prv}$), brand name ($PTM_{brand}$), competence ($PTM_{comp}$), honesty ($PTM_{hons}$), website ($PTM_{wsite}$), and physical location ($PTM_{loc}$). After publishing the PTMs, the trust mediator bootstraps and evaluates the PTMs. Figure 6.5 shows the steps and approaches used to rate PTMs, as follows:

- Certification approach based on STMs: $T_{PTM_{sec}}$, $T_{PTM_{prv}}$, and $T_{PTM_{rem}}$ are rated based on the $T_{SSTM_{sec}}$, $T_{SSTM_{prv}}$, and $T_{SSTM_{rem}}$ of their services. Algorithm 3 presents the trust model for the $T_{PTM_{sec}}$ and the same algorithm can be used for $T_{PTM_{prv}}$ and $T_{PTM_{rem}}$. Finally, the PTMs are certified.

  A service provider, $pr_j$, with a service, $s_{ij}$, supports security by publishing $SSTM_{sec}(s_{ij})$. The trust mediator evaluates the $T_{SSTM_{sec}(s_{ij})}$ based on the $SSTM_{sec}(s_{ij})$. Subsequently,
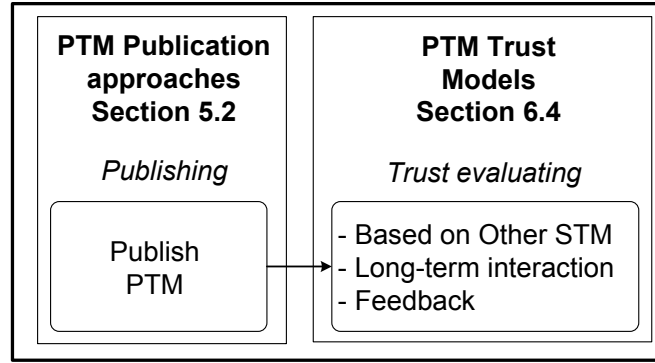
*Figure 6.5: PTM Trust Models: The Steps and Approaches to Rating PTM.*

---

**Algorithm 3** Trust Model for $T_{PTM_{sec}}$.

---

$pr_j$ publishes a service $s_{ij}$
**if** $SSTM_{sec}(s_{ij}) = 1$ **then**
   $\{s_{ij}$ *support security*$\}$
   Evaluate $T_{SSTM_{sec}(s_{ij})}$
   **if** $pr_j$ is a new provider **then**
      $T_{PTM_{sec}(pr_j)} = T_{SSTM_{sec}(s_{ij})}$
   **else**
      Update $T_{PTM_{sec}(pr_j)}$:
      $T_{PTM_{sec}(pr_j)} = avg[T_{PTM_{sec}(pr_j)}, T_{SSTM_{sec}(s_{ij})}]$
   **end if**
**end if**

---

the trust mediator evaluates $T_{PTM_{sec}(pr_j)}$ based on the $T_{SSTM_{sec}(s_{ij})}$ of the provider's services. If the provider is new, then $T_{PTM_{sec}(pr_j)} = T_{SSTM_{sec}(s_{ij})}$ of its new service $s_{ij}$; otherwise, the value of $T_{PTM_{sec}(pr_j)}$ is updated, and $T_{PTM_{sec}(pr_j)} = avg[T_{PTM_{sec}(pr_j)}, T_{SSTM_{sec}(s_{ij})}]$. The same process is conducted for the all published $SSTM_{sec}(s_{ij})$ of the provider's services. The evaluation of $T_{PTM_{prv}}$ and $T_{PTM_{rem}}$ follows the same steps.

- The long-term interaction approach: This approach involves the trust models for $T_{PTM_{comp}}$, $T_{PTM_{hons}}$, and $T_{PTM_{brand}}$.

   1. For $T_{PTM_{comp}}$: To evaluate the competency of providers, it is important to identify the necessary requirements for demonstrating the providers' competencies. In this work, a service provider is considered as competent if it is a trustworthy provider,

where its trust rate is higher than a certain threshold, d. Then, $T_{PTM_{comp}} = 1$ if the providers are showed as competent for c iterations, a constant that reflects the competency of service providers.

2. For $T_{PTM_{hons}}$: This work considers the service provider to be honest if it is competent over time. If $T_{PTM_{comp}} = 1$ for h iterations, then $T_{PTM_{hons}} = 1$. Thus, the constant h reflects the honesty of service providers.

3. For $T_{PTM_{brand}}$: The provided name of the service provider will be branded if the provider become honest; thus, $T_{PTM_{brand}} = 1$ if $T_{PTM_{hons}} = 1$.

Algorithm 4 presents the trust model for $T_{PTM_{comp}}$, $T_{PTM_{hons}}$, and $T_{PTM_{brand}}$. For any provider, $pr_j$, $c_j = c$ and $h_j = h$. Each time the trust rate of a service provider is updated, the trust mediator checks whether the provider is competent and honest. For example, if c=2, h=4, and d=7, then a service provider whose its trust rate is more than or equal to 7 will be competent, $T_{PTM_{comp}} = 1$ after c=2 iterations, $T_{PTM_{hons}} = 1$ when $T_{PTM_{comp}} = 1$ for h=4 iterations, and then $T_{PTM_{brand}} = 1$ if $T_{PTM_{hons}} = 1$. Thus, trust evaluation for the PTM undergoes long-term interactions through c iterations to become competent and cXh iterations to become honest and for its name to be branded. When a service provider is considered honest, the certifier component of the trust mediator can certify the provider's honesty and brand name.

The service broker administrator assigns the values of c, h, and d. Higher values for c, h, and d indicate that the service broker is more restricted in applying trust. For example, if a service broker assigns low values for c and h, a provider can build its competency and honesty more quickly with fewer registered services and fewer interactions with its services. On the other hand, high c and h values mean that the competence and honesty of providers will be based on more of its published services and more interactions with its services. Thus, the assignment of high values for c and h helps to overcome the whitewashing problem.

- Feedback approach: The trust rates of the clues PTMs such as payment satisfaction $T_{PTM_{pym}}$ and website $T_{PTM_{wsite}}$ are rated by service requestors, who can provide their feedback for $PTM_{pym}$ and $PTM_{wsite}$ after they consume the services. Trust rates for

**Algorithm 4** Check Provider's Competence and Honesty.
___

Initialization: For $pr_j$, $c_j = c$ and $h_j = h$
**if** $h_j \neq 0$ **then**
    {*the provider is not honest*}
    **if** A provider is competent, $(T_{pr} \geq d)$ **then**
        $c_j = c_j - 1$
        **if** $c_j \leq 0$ **then**
            {*c competence iterations*}
            $T_{PTM_{comp_j}} = 1$
            $h_j = h_j - 1$
            $c_j = c$ {*to start a new competence iteration*}
            **if** $h_j \leq 0$ **then**
                {*h iterations of $T_{PTM_{comp_j}} = 1$*}
                $T_{PTM_{hons_j}} = 1$ and $T_{PTM_{brand_j}} = 1$
            **end if**
        **end if**
    **end if**
**end if**
___

such PTMs can strength the trustworthiness of services and providers and support requestors in their selection decision. For example, if a service provider publishes $PTM_{wsite}$, a clue indicating that they have a website, then requestors can check if the provider has a website and provide their feedback about the $PTM_{wsite}$. If requestors discover that a provider has a website, then they provide their feedback and $T_{PTM_{wsite}} = 1$; otherwise, $T_{PTM_{wsite}} = 0$.

## $T_{PTM}$ **Collection Values**

The measured $T_{PTM}$ values for each service provider are stored in the rating registry to support a requestor's trust preferences, hence allowing the requestor to select a service based on the provider's trustworthiness among a set of requestor's preferred PTMs. The following represents the collected $T_{PTM}$ values in a matrix format. Each row represents a service provider and each column represents one of the $T_{PTM}$ values (m:pr and n:$T_{PTM}$).

$$\mathbf{T_{PTM}} = \begin{bmatrix} T_{PTM_{11}} & T_{PTM_{12}} & \ldots & T_{PTM_{1n}} \\ T_{PTM_{21}} & T_{PTM_{22}} & \ldots & T_{PTM_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ T_{PTM_{m1}} & T_{PTM_{m2}} & \ldots & T_{PTM_{mn}} \end{bmatrix}$$

## 6.5   Summary

This chapter presents trust models for the trust metrics, which are used to bootstrap and evaluate the trust rates of the trust metrics. Different trust models are built for various trust metrics. The identification and rating of trust metrics helps to bootstrap services and service providers, which will be rated on the basis of their rated trust metrics. In the next step of this work, the trust models for services and providers are built, which are presented next in Chapter 7.

# Chapter 7

# Trust Models for Services and Service Providers

*"One must be fond of people and trust them if one is not to make a mess of life."* E.M. Forster

As trust models are used to evaluate trust rates, Chapter 6 presents trust models for the Trust Metrics (TMs). Subsequently, trust models for evaluating the trust rates for services ($T_s$) and service providers ($T_{pr}$) need to be built. Since $T_s$ and $T_{pr}$ are based on the rates of their TMs, the trust models for services and providers are based on the trust models of TMs. The identification of TMs facilitates the trust bootstrapping process. After a service provider publishes a service and the TMs, the trust rates for TMs ($T_{TM}$) are bootstrapped for initial trust ratings. Then, $T_s$ is bootstrapped from the initial $T_{TM}$. Accordingly, $T_{pr}$, which is based on the $T_s$, is bootstrapped from the initial $T_s$.

Only the published TMs are considered in rating services and service providers. Trust is context-specific and built from information given by the provider. If a service provider does not publish specific TMs, then the provider trustworthiness can not be evaluated based on such TMs, and the TMs are excluded from the evaluation of $T_s$ and $T_{pr}$. However, a provider that wishes to be selected by different requestors should support and publish different TMs.

Based on the TMs used in the trust evaluation, this work defines two types of trust ratings: *general trust* and *trust preference*. The *general trust* rating is the evaluated trust rating based

on all of the service's published TMs. Specifically, it is the average rate of all published TMs. For example, the general $T_s$ is the average of $T_{STM}$ of the all published services' STMs. On the other hand, the *trust preference* rating is the evaluated trust rating based on the requestor's trust preferences, thus, it is the average rate of the selected TMs by the requestors. For example, the trust preference $T_s$ is the average of $T_{STM}$ of the selected STMs by the requestors based on their trust preferences. A consideration of trust preference rates supports the subjective and context-specific properties of trust. The following example demonstrates the difference between general and preference trust rates:

Service s publishes TM1, TM2, TM3, TM4

General $T_s = avg(T_{TM1}, T_{TM2}, T_{TM3}, T_{TM4})$

Preference $T_s = avg(T_{TM1}, T_{TM4})$, where TM1 and TM4 are the requestor trust preferences.

During the publishing of services, the trust mediator builds general trust ratings for services and service providers. When services are discovered and searched by requestors, the trust mediator builds trust preference rates.

The general trust rate can be calculated by using two averaging methods: simple averaging and exponential averaging [68]. Specifically, the trust model in this work uses simple averaging for trust bootstrapping and then uses exponential averaging for trust evolving. Exponential averaging provides weights to the most recent observations, thus disregarding past observations and mitigating trust variation and trust bias. In addition, exponential averaging reflects the true value of the current rating; thus using this rating the trust mediator can detect malicious behaviour. The following formulas depict the proposed simple trust averaging and exponential trust averaging:

- Simple averaging:

$avg(T) = \sum_{i=1}^{n} T_{STM(s_n)}/n$

- For any time period, t, exponential averaging:

$avg(T_t) = \propto \sum_{i=1}^{t-2}(1- \propto)^{i-1}T_{t-i} + (1- \propto)^{t-2}T_1 \qquad t \geq 3 \,, \propto= (0,1]$

Exponential averaging example:

$T_4 = \propto (1- \propto)^0 T_3 + \propto (1- \propto)^1 T_2 + (1- \propto)^2 T_1$

Where 'n' is the number of averaged elements and $\propto$ is the 'smoothing constant', which determines the weights given to the most recent past observations. A smaller $\propto$ indicates a slower the past observations are forgotten. The simple trust averaging and exponential trust averaging have similar results if a service behaves in a consistent manner. Specifically, Table 7.1 and Figure 7.1 show the simple and exponential trust averages with two possible $\propto$ values for a service that is behaving 'well'.

*Table 7.1: Trust Rating: Simple vs. Exponential Averaging (Behaving Well).*

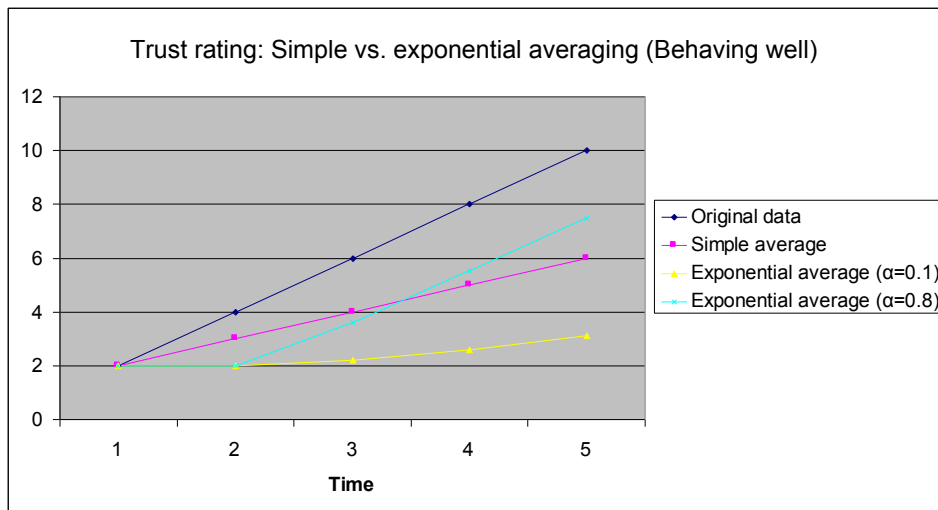| Time | Trust values | Simple averaging | Exponential averaging | |
|------|--------------|------------------|-----------------------|--------------|
|      |              |                  | $\propto = 0.1$ | $\propto = 0.8$ |
| 1 | 2 | 2 | 2 | 2 |
| 2 | 4 | 3 | 2 | 2 |
| 3 | 6 | 4 | 2.2 | 3.6 |
| 4 | 8 | 5 | 2.58 | 5.52 |
| 5 | 10 | 6 | 3.122 | 7.504 |



*Figure 7.1:  Trust Rating:  Simple vs. Exponential Averaging (Behaving Well).*

This chapter presents the trust models of services and service providers, where the service trust model is used to evaluate $T_s$, and the service provider trust model is used to evaluate $T_{pr}$. In

this chapter, Section 7.1 presents the service trust model, while Section 7.2 presents the service provider trust model. Finally, the trust matching model is discussed in Section 7.3 and Section 7.4 summarizes the chapter.

## 7.1 Service Trust Model

The trust rate of services $(T_s)$ are evaluated based on their evaluated $T_{STM}$. $T_s$ is the average of its $T_{STM}$. $T_s$ is evaluated by the service trust evaluation component of the trust mediator framework, in Section 4.3, and stored in the rating registry. The trust service model calculates the *general trust* rate of services, which is weighted as the average $T_{STM}$ of all of the service's published STMs.

When services are published, the trust mediator evaluates the general trust rates for the newly published services. Subsequently, the mediator stores the evaluated $T_{STM}$ in addition to the $T_s$, for supporting trust preference rates. During the discovery of services, requestors provide their trust preferences by selecting a set of STMs. Then, the trust mediator will obtain the $T_{STM}$ of the requestors' preferred STMs. Finally, the system averages the $T_{STM}$ of the selected STMs to calculate a new $T_s$, or *trust preference* rate, which is different from the trust general rate, as it is based on the requestor's trust preferences. The result, trust preference $T_s$, is returned to the requestor.

The trust service model supports the dynamic nature of trust through feedback from the service consumers. Specifically, service consumers provide their feedback about the services' TMs rather than about the services, which supports the context-specific property of trust. However, getting feedback about a service build reputation for the service. Thus, a service broker can build reputation for services and providers by getting feedback about the services and providers. For example, if a service, s, published TM1, TM2, and TM3, the requestor of the service may select service s based on TM1 and TM3 as their trust preferences. After consuming the service, the requestor provides his/her feedback on only TM1 and TM3, which will satisfy the context-specific property of trust. If the requestor sends feedback on s, this service will gain reputation.
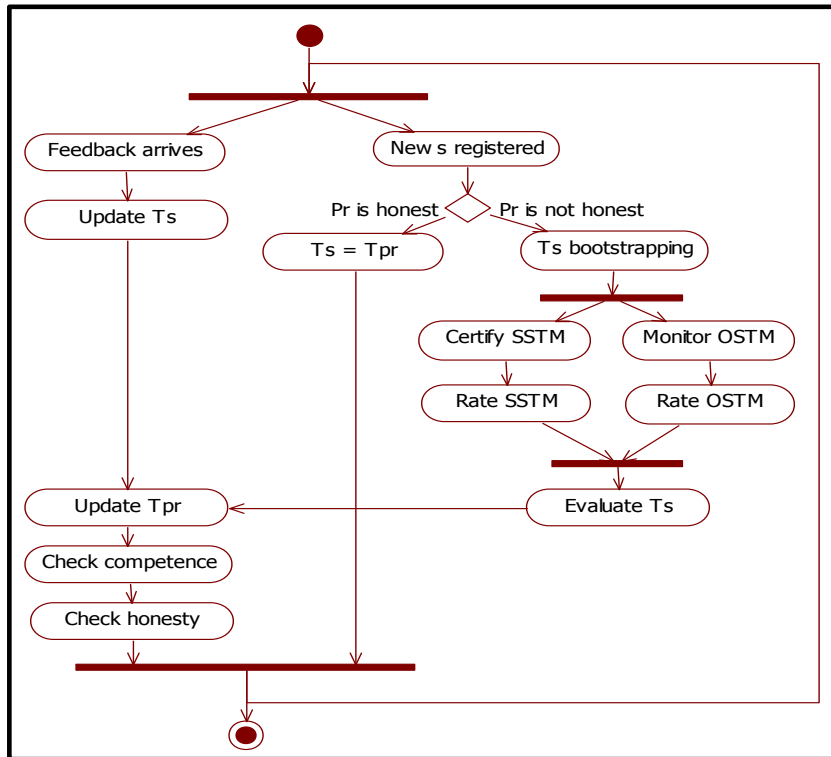
*Figure 7.2: Trust Bootstrapping and Rating Services: Activity Diagram.*

The trust rating of service providers is another dynamic approach that is used to support the trust bootstrapping and rating of services. One of the trust bootstrapping techniques is to assign the trust rate of an *honest* service provider to its newly registered services. The system evaluates the honesty of service providers (Section 6.4); if a service provider is honest, the service broker will trust his/her new services and will assign its rate to its new services. Thus, the service broker will not conduct the trust bootstrapping process for the provider's new services. By assigning a trust rate of provider to its services, this work *lowers the overhead of rating new services* by the trust mediator.

In the UML activity diagram depicted in Figure 7.2, the service trust evaluation process starts with the trust bootstrapping process. Within this diagram, the service broker evaluates and updates $T_s$, as shown in Algorithm 5.

**Algorithm 5** Service Trust Model: Trust Bootstrapping and Rating Services.

---

**while** 1 **do**
  **if** a new service, $s_{ij}$ is registered **then**
    **if** $T_{hons_j} = 1$ **then**
      {*The service provider is honest*}
      $T_{s_{ij}} = T_{pr_j}$
    **else**
      {*The service provider is not honest, thus trust bootstrap $T_{s_{ij}}$*}
      Monitor $OSTM(s_{ij})$
      Evaluate $T_{OSTM(s_{ij})}$ {*Algorithm 1 and 2, Section 6.2*}
      Identify the levels of $SSTM_{sec}(s_{ij}), SSTM_{prv}(s_{ij}), SSTM_{rem}(s_{ij})$
      Evaluate and certify $T_{SSTM_{sec}(s_{ij})}, T_{SSTM_{prv}(s_{ij})}, T_{SSTM_{rem}(s_{ij})}$
      $T_{s_{ij}} = Avg[Avg(T_{OSTM(s_{ij})}), Avg((T_{SSTM(s_{ij})})]$ {Simple averaging}
      Update $T_{pr_j}$: $Tpr_j \leftarrow Avg(T_{pr_j}, T_{s_{ij}})$ {Exponential averaging}
      Check $pr_j$ competence and honesty (Algorithm 4, Section 6.4)
    **end if**
  **end if**
  **if** Feedback arrived on $STM(s_{ij})$ **then**
    Update $T_{STM_j(s_{ij})}$
    Update $T_{s_{ij}}$: $T_{s_{ij}} \leftarrow Avg[T_{s_{ij}}, T_{STM_{(s_{ij})}}]$ {Exponential averaging}
    Update $T_{pr_j}$ {Exponential averaging}
    Check $pr_j$ competence and honesty (Algorithm 4, Section 6.4)
  **end if**
**end while**

---

- If a service is new, the trust mediator starts the bootstrapping process for the new service. There are two trust bootstrapping techniques based on the honesty of the service provider, as shown in the sequence diagram in Figure 7.3.

    - A provider is not honest (Figure 7.3, a): If the service provider is not honest, the trust mediator will start the trust bootstrapping process for services. The mediator will monitor and rate OSTM, certify and rate SSTM, and then evaluate $T_s$ by averaging its $T_{OSTM}$ and $T_{SSTM}$ with simple averaging.

    - A provider is honest (Figure 7.3, b): If the provider is honest, the rates of its new services will equal the rating of their provider.

- If feedback is returned on one or more of the service's STMs, then $T_{STM}(s)$ and $T_s$ are updated accordingly. Although the provision of feedback on $T_{OSTM}(s)$ is possible, it
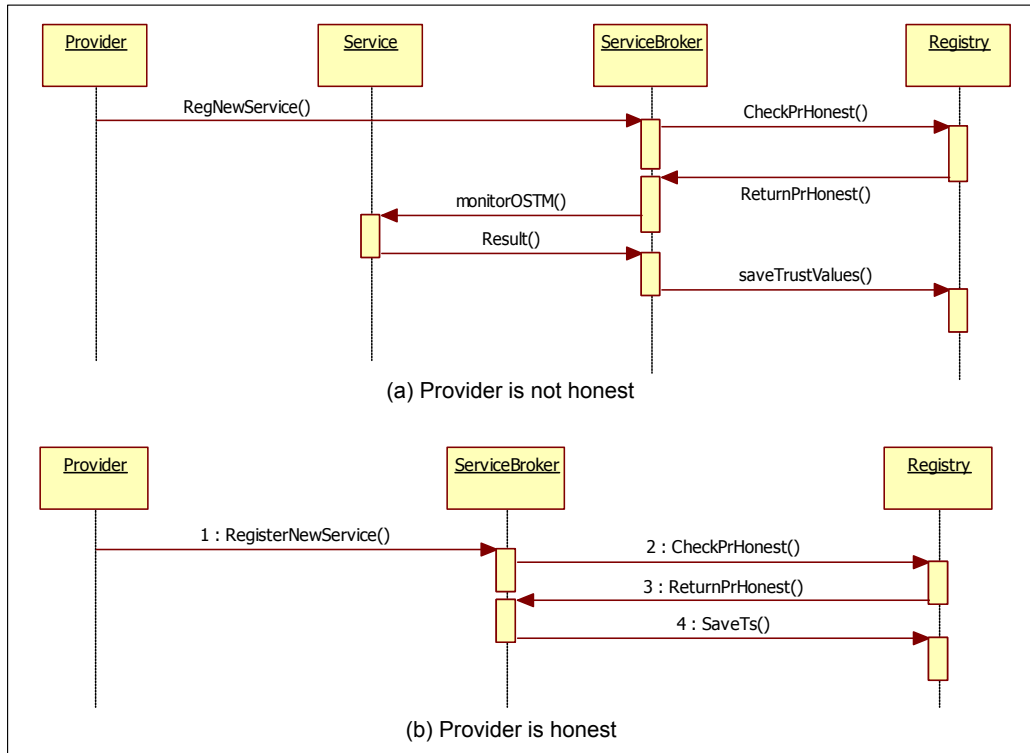
*Figure 7.3: Trust Bootstrapping and Rating Services: Sequence Diagrams.*

requires monitoring the OSTM. The monitoring can be performed on the TTP side or the requestor side.

- Every time the $T_s$ is updated, the $T_{pr}$ will also be updated and the trust mediator will check the competence and honesty of the provider.

The evaluated general $T_s$ is stored in the rating registry. The following matrix depicts the collected general $T_s$ values, where each row represents a service and a column represents the $T_s$ value (m:s).

$$\mathbf{T_s} = \begin{bmatrix} T_{s_1} \\ T_{s_2} \\ \vdots \\ T_{s_m} \end{bmatrix}$$

## 7.2 Service Provider Trust Model

The trust rates of service providers, $T_{pr}$, are evaluated based on the trust rate of their services, $T_s$. $T_{pr}$, the average of its services' $T_s$, is evaluated by the service provider trust evaluation component of the trust mediator framework and stored in the rating registry.

Since the service and provider trust models are related, the evaluated trust rates of services affect the trustworthiness of their providers. With the exception of clue PTMs, $T_{PTM}$ are evaluated based on the $T_{STM}$ and $T_s$ of the provider's services. The service provider trust model uses different approaches for supporting the dynamic nature of trust. The dynamic approaches include the following:

- Services' ratings: trust rates of service providers are affected by the trust rates of its services.

- Implicitly support monitoring, certification, and feedback approaches supported by the TM and trust service models.

- Support feedback on clue PTMs.

Algorithm 6 presents the service provider's trust model and shows how the trust mediator evaluates and updates $T_{pr}$, which is further depicted in the UML activity diagram in Figure 7.4. If a provider is new, the mediator starts the trust bootstrapping process for the provider and its new service. First, $T_s$ is bootstrapped, and then the bootstrap value of $T_{pr}$ equal the bootstrapped $T_s$. If the provider is not a new provider, the mediator will trust bootstrap the new service, evaluate the $T_s$, and then update $T_{pr}$ ($T_{pr} \leftarrow Avg(T_{pr}, T_s)$). $T_{PTM}$ is updated using feedback on the PTM. In addition, the trust mediator checks the competence and honesty of the provider, as shown in Algorithm 4 from Section 6.4.

The evaluated general $T_{pr}$ is stored in the rating registry. The following matrix represents the collected $T_{pr}$ values, where each row represents a service provider and the column represents a $T_{pr}$ value (m:pr).

**Algorithm 6** Service Provider Trust Model: Trust Bootstrapping and Rating Service Providers.

---

**while** 1 **do**

    **if** A new provider $(pr_j)$ arrives **then**

        {*Trust bootstrap $T_{pr_j}$*}

        Initialization:

        $c_j \leftarrow c, h_j \leftarrow h, T_{PTM_{comp_j}} \leftarrow 0, T_{PTM_{hons_j}} \leftarrow 0$

        {*A provider provide a new service*}

        Trust bootstrapping the service (Algorithm 5, Section 7.1)

        $T_{pr_j} \leftarrow T_{s_{ij}}$

        Check $pr_j$ competence and honesty (Algorithm 4, Section 6.4)

    **else**

        {*The provider is not new, thus trust bootstrap $T_{s_{ij}}$*}

        Update $T_{pr_j}$: $Tpr_j \leftarrow Avg(T_{pr_j}, T_{s_{ij}})$ {Exponential averaging}

        Check $pr_j$ competence and honesty (Algorithm 4, Section 6.4)

    **end if**

    **if** Feedback arrives on $PTM_j$ **then**

        Update $T_{PTM_j}$

    **end if**

**end while**

---

$$\mathbf{T_{pr}} = \begin{bmatrix} T_{pr_1} \\ T_{pr_2} \\ \vdots \\ T_{pr_m} \end{bmatrix}$$

## 7.3 Trust Matching Model

The bootstrapped $T_{TM}$ are stored in the rating registry, from where a requestor can select a service based on a set of TMs. This work proposes a matching model that evaluates the *trust preference* rate. The matching model is run by the matching component of the trust mediator framework. Specifically, this model evaluates trust preference $T_s$ based on the set of requestors' preferred TMs, where the requestor selects a list of TMs, and the matching model evaluates the $T_s$ for the services based on the selected TMs, as follows:

Preference $T_s = Avg(T_{STM})$

Preference $T_s = Avg[Avg(T_{OSTM}) + Avg(T_{SSTM})]$
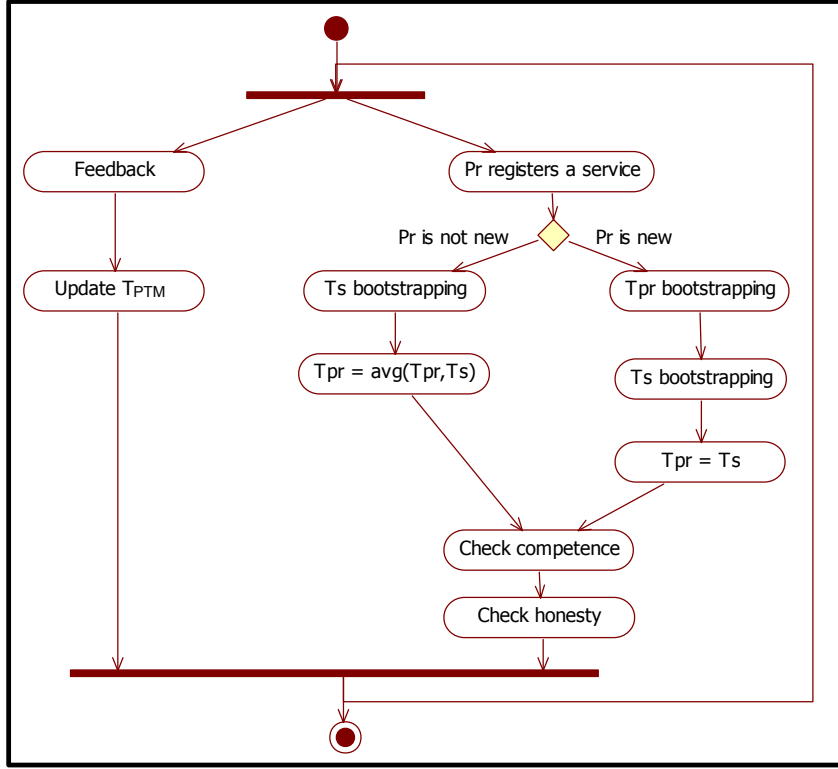
*Figure 7.4: Trust Bootstrapping and Rating Service Providers.*

For example, if a requestor requests a service based on $OSTM_e(s), OSTM_l(s), OSTM_{thr}(s)$, and $SSTM_{sec}(s)$ in addition to the service's functional property, then the trust mediator will evaluate the $T_s$ based only on the $OSTM_e(s)$, $OSTM_l(s)$, $OSTM_{thr}(s)$, and $SSTM_{sec}(s)$. Thus, if $T_{OSTM_e}(s) = 8.1, T_{OSTM_l}(s) = 6.5, T_{OSTM_{thr}}(s) = 7.2$, and $T_{SSTM_{sec}}(s) = 6$, then:

Preference $T_s = Avg[T_{OSTM_e}(s) + T_{OSTM_l}(s) + T_{OSTM_{thr}}(s) + T_{SSTM_{sec}}(s)]$
Preference $T_s = Avg[8.1 + 6.5 + 7.2 + 6] = 6.95$

The trust matching model supports the multi-degree of trust by allowing requestors to specify degrees for the selected TMs. For example, a requestor may provide a degree of 60% for the $OSTM_e(s)$, 70% for the $OSTM_l(s)$, 100% for the $OSTM_{thr}(s)$, and 100% for the $SSTM_{sec}(s)$, as follows:

Preference $T_s = Avg[0.6 \times T_{OSTM_e}(s) + 0.7 \times T_{OSTM_l}(s) + 1 \times T_{OSTM_{thr}}(s) + 1 \times T_{SSTM_{sec}}(s)]$
Preference $T_s = Avg[0.6 \times 8.1 + 0.7 \times 6.5 + 1 \times 7.2 + 1 \times 6] = 5.65$

In addition, a requestor can select a service based on the trust rate of its provider and choose a service from a provider based on a set of PTMs, such as $PTM_{sec}$, $PTM_{comp}$, and $PTM_{wsite}$. Subsequently, the system will return the $T_{PTM_{sec}}$, $T_{PTM_{comp}}$, and $T_{PTM_{wsite}}$, which will support the requestors in their service selection decision.

The system can support the Quality of Experience (QoE) and Quality of Business (QoBiz), as discussed in Section 3.2. Specifically, the QoE can be gathered by feedback from the service consumers and QoBiz can be obtained by enabling the providers to see their rates and their services rates for the purpose of improvement.

## 7.4   Summary

This chapter presents the trust models for services and service providers, which are used to bootstrap and evaluate trust rates for services and their providers. Two types of trust ratings are presented: the general trust rating and the trust preference rating. The services and service providers are bootstrapped and their trust rates are initialized using the general trust rating.

Moreover, the trust matching model for service selection is presented, which uses the trust preference rating. The trust matching model supports the context-specific property of trust and allows requestors to select services based on the requestors' trust preferences. The next chapter presents the implementation prototype for the trust mediator framework as well as experiments and evaluations of the trust bootstrapping approach.

# Chapter 8

# Implementation and Experiment

*"Trust only movement. Life happens at the level of events, not of words."* Alfred Adler

Analytical and empirical studies are the basis of any modelling effort [51]. Accordingly, the analytical trust framework and models discussed in this work require empirical analyses to evaluate the trust solution, show its practical uses, and obtain the optimal outcome. This chapter presents the empirical studies that include the implementation, experimentation, and evaluation of the **trust bootstrapping** solution. Section 8.1 presents the trust-based SOA prototype, and Section 8.2 presents the implementation of the prototype. The experiment is presented in Section 8.3, and Section 8.4 presents the evaluation. Finally, Section 8.5 summarizes the chapter.

## 8.1 Trust-Based SOA Prototype

This section presents the trust-based SOA prototype, as shown in Figure 8.1. The prototype presents the elements and their relations as well as the software tools, and it consists of service providers, service requestors, and a service broker. The service broker contains a service registry and the trust mediator, which conducts the bootstrapping process and includes the components of the pre-processing phase and the processing and evaluation phase as well as the matching component of the post-processing phase of the trust framework (ToTEF), which is

presented in Section 4.3. Other post-processing components are responsible for trust management; however, these components are beyond the cope of this work. Moreover, the implementation does not incorporate the services publication and the discovery operations with the service registry.
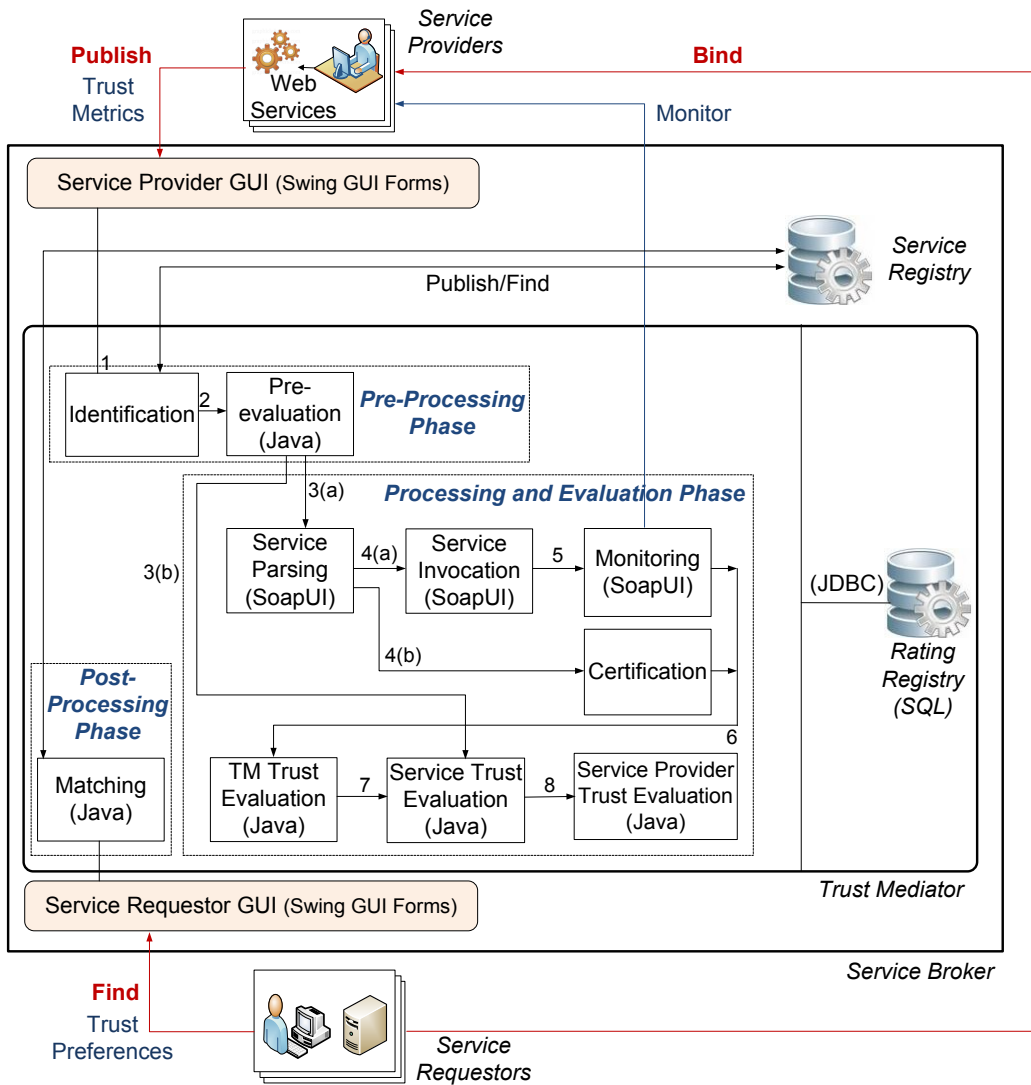


*Figure 8.1: Trust-Based SOA Prototype.*

When a service provider publishes a service, the identification component identifies the service and service provider (1). Then, the pre-evaluation component decides whether to assign

the provider's trust rate to its new service (2). If the provider is honest, the trust mediator assigns the trust rate of the provider to its new service (3(b)). Otherwise, the mediator starts trust bootstrapping the service and its provider by parsing the WSDL (3(a)), invoking the service and monitoring the OSTMs (4(a) and 5), and certifying the SSTMs (4(b)). Next, the collected information is used to evaluate the trust rates of the TMs (6), service (7), and service provider (8). When a service requestor requests a service, the matching component returns services based on services' functional properties, which is discovered from the service registry, and trust criteria based on the requestor's preferences from the rating registry.

## 8.2 Prototype Implementation

This section presents the prototype implementation, which includes the prototype design and the Graphical User Interfaces (GUI).

### 8.2.1 Prototype Design

In the current implementation, services are deployed on Windows Vista Home Premium, which features a 2.1 GHz Processor, 4 GB of RAM, and a 220 GB Hard Drive. As shown in Figure 8.1, Java programming language is used to implement different trust mediator components, such as pre-evaluation, TM trust evaluation, and matching parts. Various software tools are used to implement other components of the trust mediator and the roles of the trust-based SOA prototype, as follows.

**Services and Service Providers**

Services are implemented using *Web Services* technology. Specifically, *WSDL* is used to describe the services and *SOAP* is used as a messaging standard. Using *Java* and *NetBeans IDE 6.9.1*, the service providers are implemented as *Enterprise Java Bean (EJB)*. The Web Services are deployed into *GlassFish Server 3*.

99

Figure 8.2 shows a snapshot list of the created providers and their services on the left. Moreover, WSDLs are created and Web Services are implemented from the WSDLs. The EJB projects represent Web Service providers and include the providers' Web Services and the Web Services' WSDLs. EJB 1-6 presents the six providers; for example, EJB3 is Provider 3, which has three services: p3Converter, a temperature converter, p3FindMax, which is designed to find a maximum, and p3Mean, which calculates a mean. The services' WSDL files are WS-DLp3Converter, WSDLp3FindMax, and WSDLp3Mean. Specifically, the find maximum Web Service (p3FindMax) from Provider 3 is presented in Appendix A.1 and its created WSDL is presented in Appendix A.2.
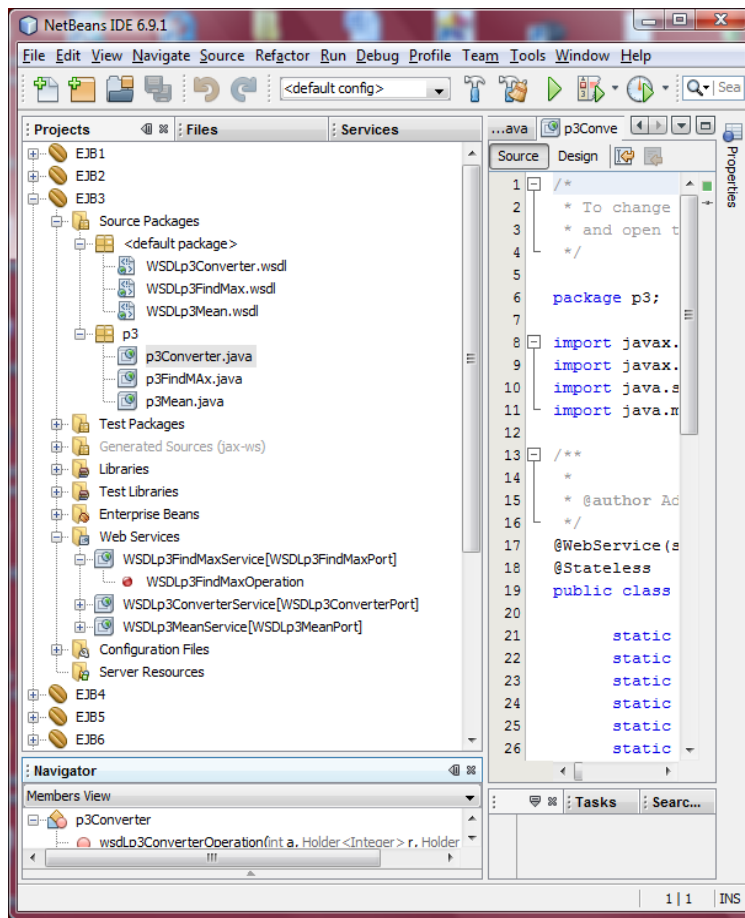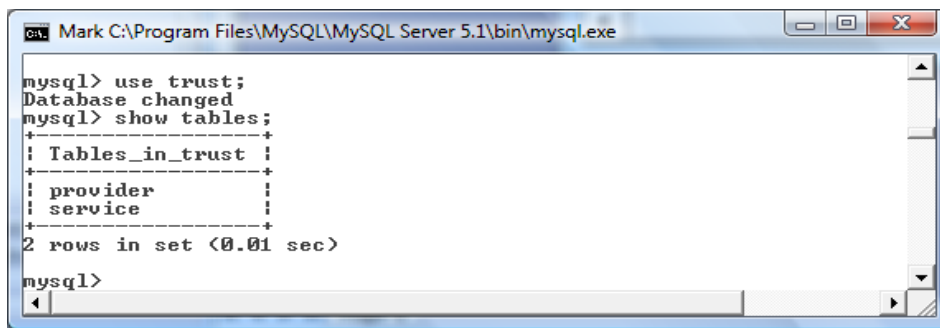


*Figure 8.2: A Snapshot of the List of the Created Web Service Providers (EJB): Providers' Web Services, and Services' WSDLs.*

**Rating Registry**

The rating registry database is implemented as a *Structured Query Language (SQL) database* using *MySQL Server 5.1*. The implemented rating registry SQL database, known as 'trust', is shown in Figure 8.3. The trust database has two tables: a service table and a provider table. The columns of the tables are presented in Appendix A.3. *Java Database Connectivity (JDBC)* API is used to connect the trust SQL database and the trust mediator components. Appendix A.4 presents the JDBC for reading and writing from the service table.



*Figure 8.3: Rating Registry: Trust Database.*

**Monitoring**

*SoapUI* is a functional testing tool for testing and monitoring Web Services. The trust mediator uses SoapUI to parse WSDL, invoke Web Services, and monitor Web Services, which are performed by parsing/SLA, invocation, and monitoring components, respectively. The SoapUI tool uses *XML*, *XPath*, *Groovy*, and JDBC to write different scripts, including collecting TMs, monitoring services, and connecting to the trust database. Figure 8.4 shows a snapshot of the SoapUI software tool. The left side depicts the services' WSDL while the right side demonstrates the Groovy code for monitoring the execution time ($OSTM_e$) and response time ($OSTM_r$) of one of the Web Services.
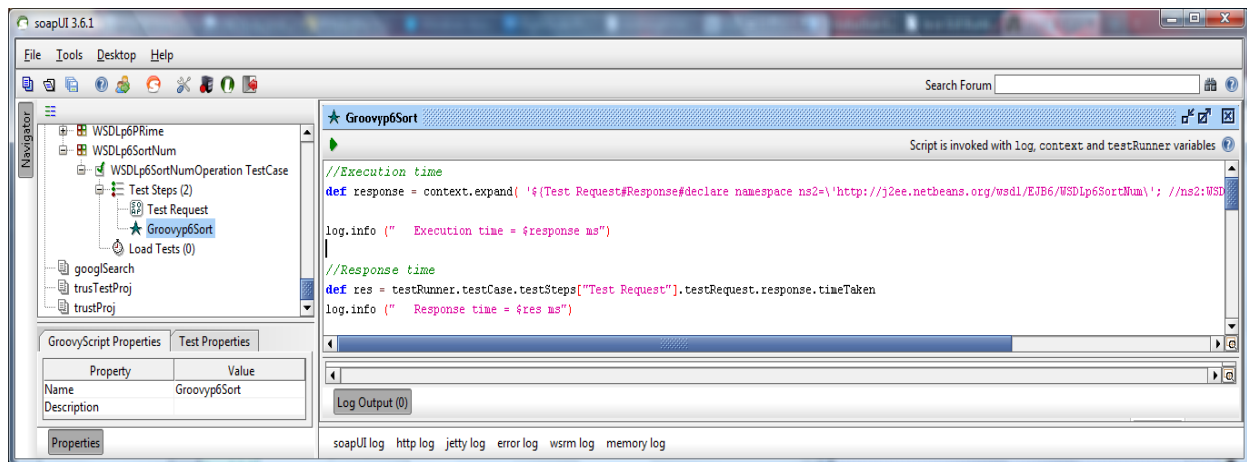
*Figure 8.4: SoapUI Project Snapshot.*

## 8.2.2 Prototype GUIs

The service provider and the requestor GUIs are implemented using the *Swing GUI Forms* and the *JFrame Form* within the NetBeans IDE. Specifically, swing is the "built-in" GUI component technology of the Java platform.

### Service Provider GUI

Figure 8.5 shows a snapshot of the provider trust GUI. Within this framework, a provider registers its services by entering the provider and service IDs, the service's function, and the list of TMs, which may vary for different services. The provider enters the minimum and maximum values of the OSTMs for execution time ($OSTM_e$), response time ($OSTM_r$), throughput ($OSTM_{thr}$), and latency ($OSTM_l$), selects SSTMs, including remedies ($SSTM_{rem}$), security ($SSTM_{sec}$), and privacy ($SSTM_{prv}$), and enters PTMs, such as brand ($PTM_{brand}$), website ($PTM_{wsite}$), and physical location ($PTM_{loc}$). Subsequently, the provider presses the "Publish" button to publish the service. The "Clear" button can be used to clear the fields for re-entering new information. Appendix A.5 presents a portion of the code for publishing some TMs into the SQL database using the provider GUI.
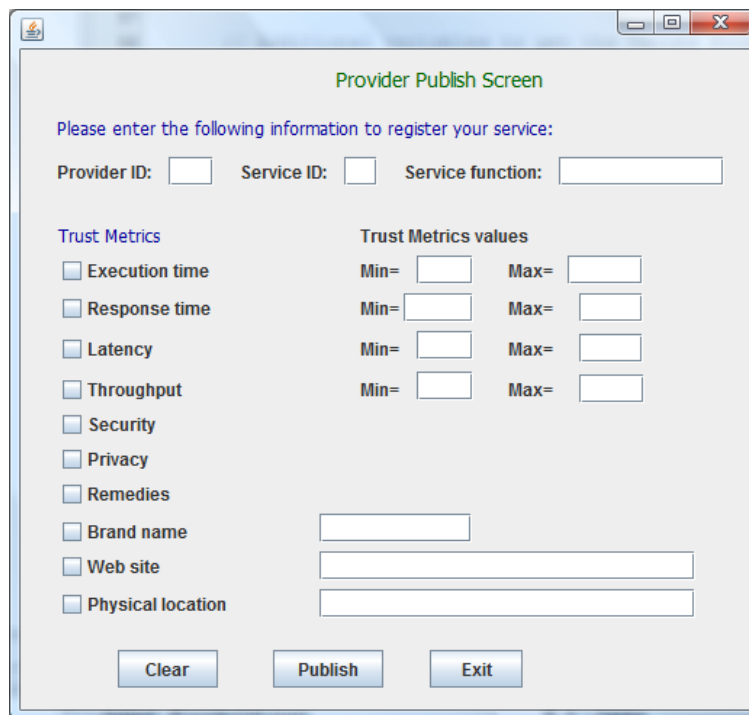
*Figure 8.5: A Snapshot of the Provider Trust GUI.*

**Service Requestor GUI**

Figure 8.6 shows the snapshot of the requestor trust GUI, which allows requesters to enter the name of the required service, or its functional property, and their trust preferences by selecting among various TMs. After entering the required service's functional property and preferred TMs, the requestor presses the "Find" button, which will return a list of services based on the selected properties and preferences. The result can be a service with the highest rate, a list of services based on a threshold provided by requestors, or a list of all matched services from which requestors can select based on the highest trust rate. The "Clear" button allows requestors to re-enter the information and make different choices. Appendix A.6 presents a portion of the code for finding a service based on the service's functional property and requestor trust preferences.
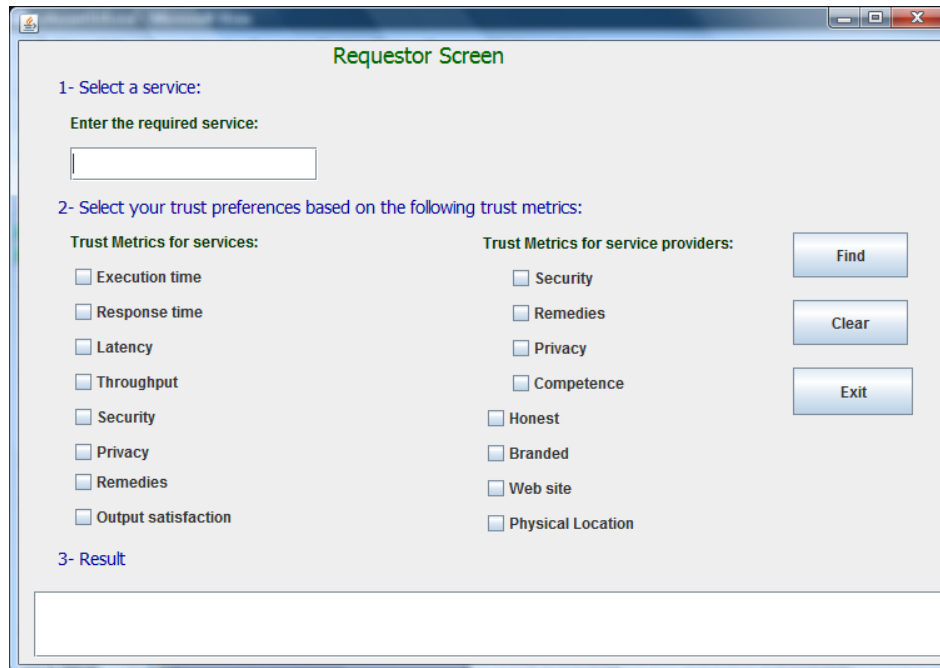
*Figure 8.6: A Snapshot of the Requestor Trust GUI.*

## 8.3 Experiments

As part of the empirical analysis, a number of experiments are conducted to provide evidence for assessing the robustness of the proposed trust bootstrapping solution. Specifically, the trust bootstrapping evaluation requires an assessment of the proposed trust metrics and trust models. Accordingly, a set of TMs for different services are selected and collected, and their trust rates are evaluated. Then, the trust rates of services and providers are evaluated using the service and provider trust models.

Two scenarios are examined and discussed that cover different requestors searching for services on the basis of their trust preferences. A requestor can be a person or an application, which may be a composition of Web Services. For example, a user may need to select a 'payment' service, or a composition of Web Service application may require a 'search' service. Since there are many 'payment' and 'search' services, the requestors need to select a service using trust criteria. Because requestors require services that they can trust, the trust mediator in the

service broker establishes trust for services and providers. Subsequently, it returns trust rates for services and their providers so that requestors can make their selection decision.

### 8.3.1 A Case Study: Electronic-Market

This section presents a case study in SOA environment, where requestors search for Web Services from a service registry. One example of an online service registry is Seekda, a free Web-based search engine for Web Services' APIs and their providers, allowing requestors to locate public Web Services. Specifically, the Seekda Web Services search engine helps requestors to find Web Services based on services' functional properties. The services listed at Seekda incorporate a wide range of functionalities. UDDI is another example of such universal Web Service registries.

Figure 8.7 presents an electronic-market, or e-market, case study. The e-market is constructed as a composition of many Web Services, such as 'search' for items, 'sort items' based on different criteria, such as price, 'calculate' the final price, and 'check credit' for the buyers. Moreover, there are many Web Services that have the same functionality, such as searching, provided by different service providers. To build the e-market enterprise application, the developer needs to select Web Services that he/she can trust. The service broker acts as a TTP and supports trust-based service selection. Specifically, the e-market developer communicates with the service broker to select Web Services based on their functional properties and trust criteria that meet the functional and trust preferences of the application.

### 8.3.2 Experimental Methodology

Table 8.1 shows the steps that are followed to conduct the experiments, as well as related information including the thesis chapter related to the step and the framework component responsible for conducting the process. There are eight steps, as described below:
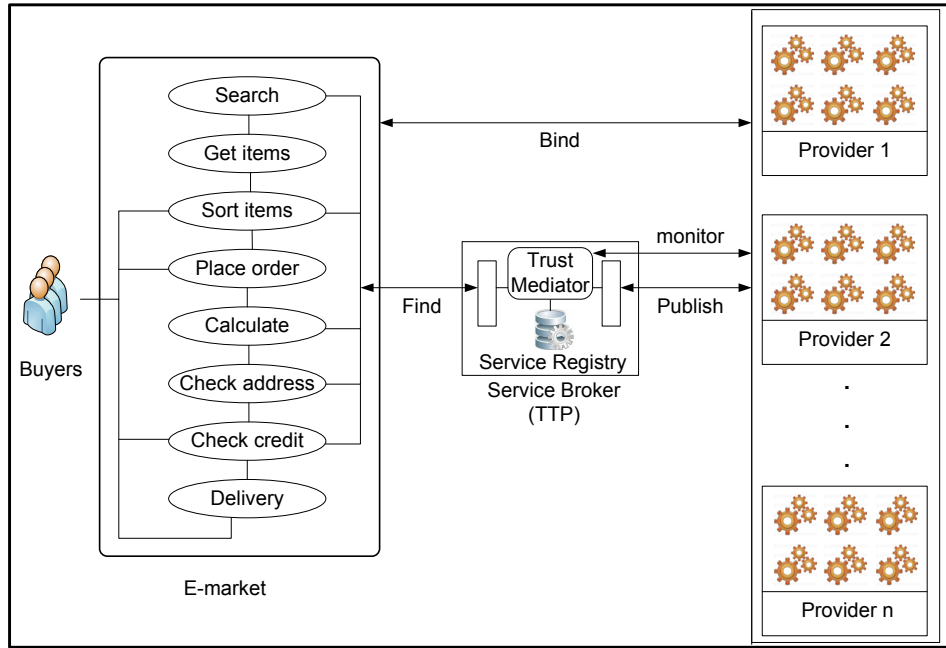
*Figure 8.7: Trust-Based SOA, E-market Case Study.*

**1- We selected a set of Trust Metrics (TM) for the experiments**

When providers publish their services, they publish their TMs (PTMs) and their services' TM (STMs). In this work, the selected TMs presented in Figure 5.4 and Table 5.1 are used in the experiment. The TMs are as follows:

- Service Trust Metrics (STMs), including:

    - Objective Service Trust Metrics (OSTMs), including: Execution time ($OSTM_e$), response time ($OSTM_r$), latency ($OSTM_l$), and throughput ($OSTM_{thr}$).

    - Subjective Service Trust Metrics (SSTMs), including: remedies ($SSTM_{rem}$), security ($SSTM_{sec}$), privacy ($SSTM_{prv}$), payment satisfaction ($SSTM_{pym}$), and output satisfaction ($SSTM_{out}$).

- Service Provider Trust Metrics (PTMs), including: remedies ($PTM_{rem}$), security ($PTM_{sec}$), privacy ($PTM_{prv}$), brand name ($PTM_{brand}$), competence ($PTM_{comp}$), honesty ($PTM_{hons}$), website ($PTM_{wsite}$), and physical location ($PTM_{loc}$).

*Table 8.1: Experimental Methodology.*

| | Steps | Chapter | Trust framework component responsible for the step |
|---|---|---|---|
| 1 | We selected a set of Trust Metrics (TM) for the experiments | Chapter 5 | |
| 2 | We created services and service providers | | |
| 3 | We evaluated the TM values (to be published by the service providers) | | |
| 4 | Service providers started registering their services | Chapter 5 (Section 5.2, TM publication approaches) | Pre-processing phase (identification and pre-evaluation components) |
| 5 | $T_{TM}$ were evaluated | Chapter 6 (TM trust models) | Processing and evaluation phase (Service parsing, invocation, monitoring, certification, and TM trust evaluation components) |
| 6 | $T_s$ were evaluated | Chapter 7 (Section 7.1, Service trust model) | Processing and evaluation phase (Service trust evaluation component) |
| 7 | $T_{pr}$ were evaluated | Chapter 7 (Section 7.2, Service provider trust model) | Processing and evaluation phase (Service provider trust evaluation component) |
| 8 | Requestors selected services based on their trust preferences | Chapter 7 (Section 7.3, Trust matching model) | Post-Processing phase (Matching component) |

## 2- We created services and service providers

The experiment requires a set of providers, each of whom offer a set of services. Different providers may provide services with the same functionality but with a different set of TMs. For example, a provider may offer a service with functionality f1 and publish a set of TMs such as

$OSTM_r$, $OSTM_l$, and $SSTM_{sec}$, while another provider may offer a service with the same functionality (f1) but with a set of different TMs such as $OSTM_{thr}$, $SSTM_{sec}$, and $SSTM_{rem}$. In addition, the services may have the same TMs, but each TM and service may have different trust rates evaluated by the trust mediator. Therefore, services with similar functional properties may have different trust rates, and the service with, for example, the highest rate will be selected by the requestor.

Subsequently, a number of services are created, each of which are provided by a number of providers. Table 8.2 shows the list of providers and the services offered by each provider. There are six providers that provide different services, including a calculator service, which does calculation; a find maximum service, which finds the maximum number among a list of numbers; a capital to small service, which converts capital letters to small letters; a temperature converter service, which converts temperature from Celsius to Fahrenheit; a mean service, which finds the mean of a list of numbers; a prime service, which determines whether or not a number is prime; and a sort numbers service, which sorts a list of numbers. For example, Provider 5 has published three services, which include find maximum, temperature converter, and sort numbers services.

*Table 8.2: Services and Service Providers Used in the Experiment.*

| | Services | | | | | | |
|---|---|---|---|---|---|---|---|
| Providers | Calculator | Find Maximum | Capital to Small | Temperature Converter | Mean | Prime | Sort Number |
| Provider 1 | √ | √ | √ | √ | √ | √ | |
| Provider 2 | √ | | √ | √ | | | √ |
| Provider 3 | | √ | | √ | √ | | |
| Provider 4 | | √ | | | √ | √ | √ |
| Provider 5 | | √ | | √ | | | √ |
| Provider 6 | √ | | | √ | | √ | √ |

**3- We evaluated the TMs values (to be published by the service providers)**

When providers publish services, they need to provide the TMs, which are determined before publication. In our experiment, the TMs values are determined in the following ways:

- OSTM: The OSTM values for each service are monitored for evaluation. According to the experiments, the OSTMs may have a range of convergent values every time the OSTMs are monitored. Thus, the minimum and maximum OSTM values are determined and the providers publish the OSTMs of their services in a range of values between a minimum and maximum. For example, a provider publishes the response time ($OSTM_r$) for one of its services in a range of [24.5,31.6] ms, where 24.5 ms and 31.6 ms are the minimum and maximum response times respectively.

- SSTM: SSTMs will have a value of 1 if they are supported by the services.

- PTM: Providers provide their brand name ($PTM_{brand}$), website ($PTM_{wsite}$), and physical location ($PTM_{loc}$). The other PTMs are not required to be published, including $PTM_{rem}$, $PTM_{sec}$, $PTM_{prv}$, $PTM_{comp}$, and $PTM_{hons}$.

**4- Service providers started registering their services**

After determining the TM values, a provider publishes its services along with the STMs and PTMs. Figure 8.8 shows a snapshot of the provider trust GUI, where a provider registers its services by entering the provider and the service IDs, the service's function, and a list of TMs. Specifically, the figure shows a provider with ID=2 registering a service with ID=1, which is a calculator service with a list of TMs. The values of the OSTMs are submitted; for instance, $OSTM_{r_{max}}(s1) = 48$ ms. Moreover, the SSTM are selected, where $SSTM_{sec}(s1) = 1$, and the PTM are entered, where $PTM_{loc}(pr2) = Address2$. However, the calculator service does not support privacy ($SSTM_{prv}$). Subsequently, the provider presses the "Publish" button to publish the service.

*Figure 8.8: A Snapshot of a Provider Publishing a Service.*

**5- $T_{TM}$ were evaluated**

After publishing the services, the trust mediator starts trust bootstrapping the TMs and evaluates the $T_{TM}$, which are stored in the rating registry. There are different approaches for evaluating $T_{TM}$ of different TMs, as presented in Table 6.1. These methods are described below:

- For $T_{OSTM}$: The values of the $T_{OSTM}$ for each service are trust bootstrapped using the monitoring approach. When a provider registers a service and provides the OSTM, the trust mediator monitors the OSTM. The monitored OSTM and the provided OSTM, described in Step 4, are used to evaluate the $T_{OSTM}$ with the OSTM Algorithm 1 and Algorithm 2 in Section 6.2.

- For $T_{SSTM}$: Provided SSTMs are certified based on the certification process of evaluating trust levels. For the purpose of the prototype, this work assumes that the security and

110

privacy SSTM ratings are obtained from security and privacy rating systems [60, 23, 9] and, for simplicity, uses two-scale rating of either 1 or 10.

- For $T_{PTM}$: Some $T_{PTM}$, such as $T_{PTM_{rem}}$, $T_{PTM_{sec}}$, and $T_{PTM_{prv}}$, are trust bootstrapped from the $T_{SSTM}$ of its services, using Algorithm 3 in Section 6.4. For example, $T_{PTM_{sec}}$ will be evaluated based on the $T_{SSTM_{sec}}$ of all of its services. $PTM_{wsite}$ and $PTM_{loc}$ will be rated from user feedback. Finally, $T_{PTM_{comp}}$, $T_{PTM_{hons}}$), and $T_{PTM_{hons}}$ are evaluated by using Algorithm 4 from Section 6.4.

## 6- $T_s$ were evaluated

The rate of services, $T_s$, are evaluated based on their evaluated $T_{STM}$. The trust mediator initiates the trust bootstrapping and rating services by using Algorithm 5 from Section 7.1. $T_s$ values are stored in the rating registry.

## 7- $T_{pr}$ were evaluated

The trust rates of service providers, $T_{pr}$, are bootstrapped based on the bootstrapped $T_s$ of their services. In order to evaluate $T_{pr}$, the trust mediator starts trust bootstrapping and rating service providers with Algorithm 6 in Section 7.2. $T_{pr}$ values are stored in the rating registry. The constants $c$ and $h$ reflect the competence and honesty of providers, as shown in Section 6.4. For simplicity, the value of these constants are initialized with a value of 3 and 2, respectively, and the threshold $d$ is assigned a value of 8. Therefore, a service provider whose trust rate is greater than or equal to 8 will be competent; as Algorithm 4 in Section 6.4 shows, $T_{PTM_{comp}} = 1$ after $c$=3 iterations, and $T_{PTM_{hons}} = 1$ and $T_{PTM_{brand}} = 1$ when $T_{PTM_{comp}} = 1$ for $h$=2 iterations.

## 8- Requestors selected services based on their trust preferences

At this point, all of the published TMs, services, and service providers have initial bootstrapped trust rates stored in the rating registry prior to interaction with any requestors. The bootstrapped $T_s$ that are stored in the rating registry are the *general trust* rates, which are based on all of the published TMs for a service.

This prototype acknowledges the various trust preferences of requestors by storing the $T_{TM}$ in the rating registry. Accordingly, requestors can select a service from a set of TMs. During service discovery time, the trust mediator evaluates the *trust preference* rates based on requestor's sets of selected TMs, as shown in the trust matching model in Section 7.3. In this case, the newly evaluated $T_s$, which is based on the selected TMs, the trust preference rate, is different from the stored $T_s$, the general trust rate, in the service registry.

## 8.3.3    Experimental Results

Experiments are conducted on the basis of the experimental methodology in Section 8.3.2. This section presents the experimental results, including the trust rates of the TMs, services, and service providers.

**1- Trust rates for STMs and services**

Table 8.3 shows a small portion of the 'services table' for the services published by Provider 2; the complete table that includes all services and STMs are presented in Appendix B. This table only presents two published TMs, $OSTM_r$ and $SSTM_{sec}$; however, $T_s$ is based on all of the published TMs for the services. In addition, for each service, there is an 'snum' that represents the number of times $T_s$ is evaluated (Appendix B). The term 'sid' refers to the service's ID number; for example, the service where $sid = 7$ has a general $T_s$ of 8.71.

*Table 8.3: Portion of the Service Table Presents Services Published by Provider 2.*

| sid | Service's function | $OSTM_r$ provided | | $OSTM_r$ collected | $T_{OSTM_r}$ | $SSTM_{sec}$ supported? | $T_{SSTM_{sec}}$ | ... | $T_s$ |
|---|---|---|---|---|---|---|---|---|---|
| | | $OSTM_r min$ | $OSTM_r max$ | | | | | | |
| 7 | Calculator | 33 | 48 | 34 | 10 | 1 | 10 | ... | 8.71 |
| 8 | Capital to small | 27 | 36 | 44 | 7.78 | 1 | 10 | ... | 7.11 |
| 9 | Temperature converter | 38 | 47 | 49 | 9.57 | 1 | 10 | ... | 8.35 |
| 10 | Sort numbers | 64 | 72 | 74 | 9.72 | 1 | 10 | ... | 8.39 |

## 2- Trust rates for PTMs and service providers

Table 8.4 shows a service provider table that presents PTMs, $T_{PTM}$, and $T_{pr}$. Providers may have different trust rates, which can assist the requestors in their selection decision. While 'pid' is the provider ID number, 'pnum' represents the number of times $T_{pr}$ is evaluated. For example, the service provider that has $pid = 1$ is competent and honest because its trust rate was more than $d=8$ for pnum $= c$ X $h = 3$ X $2 = 6$ iterations, has a brand name of "Star", and its $T_{pr} = 9.787$. The multiple evaluations of $T_{pr}$ addresses the *cold start* problem, where service providers may have many initial rates.

The trust rates of service providers are affected by the trust rates of their services. Figure 8.9 shows the trust rates of the services for each provider, and Figure 8.10 indicates the trust rates of the service providers. These figures demonstrate that the trust rates of service providers are based on the trust rates of their services. If a provider offers trusted services, it will also be trusted, as a higher service provider rate indicates that its services are also highly rated. For example, since Provider 1 has highly trusted services, then its trust rate is high, at $T_{pr} = 9.787$. However, Provider 4 offers services with around average trust rates, and thus, its trust rate is around an average value, at $T_{pr} = 7.036$.
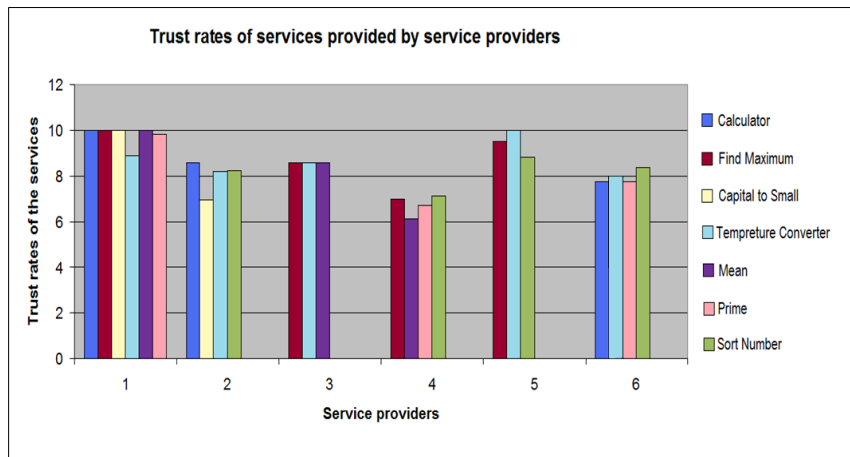


*Figure 8.9: Trust Rates of Services Provided by the Service Providers.*

Figure 8.11 shows trust rates for a set of services with the same functionality provided by different providers. The services have different trust rates because they are provided by different

*Table 8.4: Service Provider TM and Trust Ratings.*

| pid | $T_{PTM_{sec}}$ | $T_{PTM_{rem}}$ | $T_{PTM_{prv}}$ | pnum | $T_{pr}$ | $PTM_{brand}$ | $PTM_{wsite}$ | $PTM_{loc}$ | $T_{PTM_{comp}}$ | compN | $T_{PTM_{hons}}$ | $T_{PTM_{brand}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 10 | 10 | 6 | 9.787 | Star | http://www.Star.com | Address1 | 1 | 2 | 1 | 1 |
| 2 | 10 | 10 | 1 | 4 | 8.141 | Moon | http://www.Moon.com | Address2 | 1 | 1 | 0 | 0 |
| 3 | 10 | 1 | 10 | 3 | 8.571 | Sun | http://www.Sun.com | Address3 | 1 | 1 | 0 | 0 |
| 4 | 1 | 10 | 1 | 4 | 7.036 | Earth | http://www.Earth.com | Address4 | 0 | 0 | 0 | 0 |
| 5 | 10 | 10 | 10 | 3 | 9.449 | Ocean | http://www.Ocean.com | Address5 | 1 | 1 | 0 | 0 |
| 6 | 10 | 10 | 1 | 4 | 8.087 | Gulf | http://www.Gulf.com | Address6 | 0 | 0 | 0 | 0 |
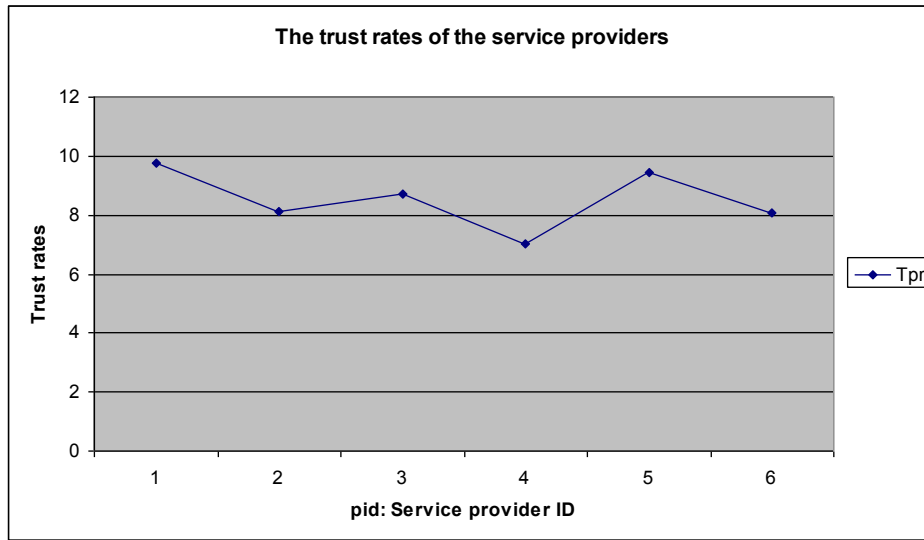
*Figure 8.10:  Trust Rates of the Service Providers.*

providers. For example, a requestor that requests a temperature converter service will receive five temperature converter services, each with a different trust rate. The trust mediator can return a service with the highest rate. In this case, the highest trust rate is the service with $T_s = 10$, which is provided by Provider 5. The trust rates shown in the figure are the *general trust* rates, which are based on all of the published TMs.

## 8.4   Evaluation

This section presents different scenarios for an e-market application developer requesting services using the system requestor GUI. The e-market application developer may require different trust preferences for various services. Accordingly, two scenarios are presented that show the trust bootstrapping process, the requestor's trust preferences, and the *trust preference $T_s$* versus *general $T_s$* as well as the importance of establishing trust for service providers to support service selecting.
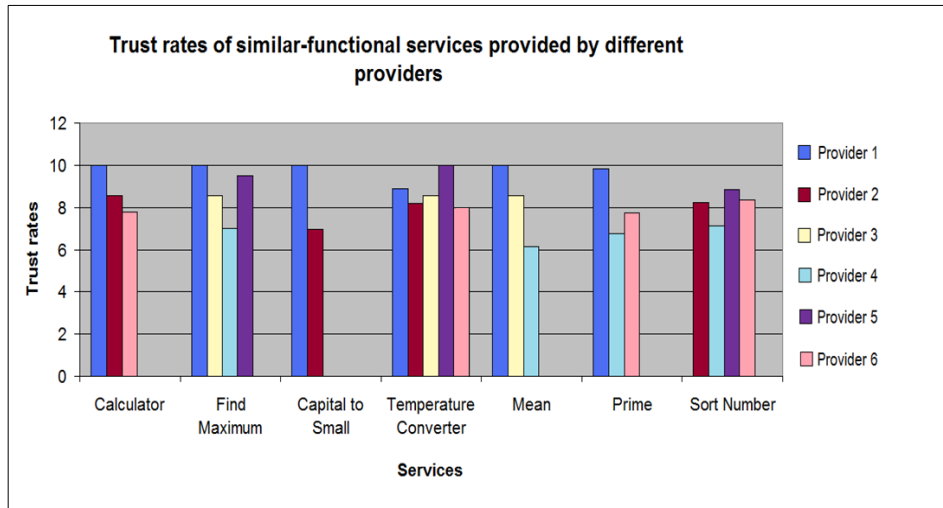
*Figure 8.11: Trust Rates of the Similar-Functional Services Provided by Different Providers.*

## 8.4.1 Scenario 1: Service Selection Based on Requestor Trust Preferences and Providers Rates

This scenario shows how requestors can select a service based on their trust preferences and service providers' trust rates. In this situation, an e-market application developer wants to select a calculator service to build the composition of services. Since there are many calculator services, the developer should select a service that he/she can trust based on his/her trust preferences, which include execution time, throughput, privacy, competence, and honesty. Subsequently, the trust mediator finds different services based on the developer's preferred TMs and displays the results for the developer, as presented in Table 8.5.

The developer can select a service with the highest trust rate, which in this case, is Service 1, with a rating of 10. The PTM supports the developer's decision; for example, the competence and honesty of a provider will encourage the developer to select its services. Therefore, the developer would select Service 1, which is provided by a competent and honest provider.

116

*Table 8.5: Scenario 1: Selecting a 'Calculator' Service Based on Requestor's Trust Preferences.*

| sid | $T_{OSTM_e}$ | $T_{OSTM_{thr}}$ | $T_{SSTM_{prv}}$ | Preference $T_s$ | General $T_s$ | $T_{PTM_{comp}}$ | $T_{PTM_{hons}}$ | pid | pnum |
|-----|------|------|------|------|------|------|------|-----|------|
| 1 | 10 | 10 | 10 | 10 | 10 | 1 | 1 | 1 | 6 |
| 7 | 10 | 10 | 1 | 7 | 8.714 | 1 | 0 | 2 | 4 |
| 21 | 10 | 9.52 | 1 | 6.840 | 7.908 | 0 | 0 | 6 | 4 |

### 8.4.2   Scenario 2: Trust Preference $T_s$ versus General $T_s$

This scenario shows how requestors can select a service based on their trust preferences, and it compares the *trust preference* $T_s$ to the *general trust* $T_s$. In this case, the e-market application developer wants to select a sort numbers service. Since there are many sort numbers services, the developer should be able to select a service that he/she can trust based on his/her trust preferences from a set of TMs. Specifically, the developer selects response time, latency, remedies, competence, and honesty as his/her trust preferences. Based on these preferences, the trust mediator finds different services with different trust rates and displays the result, as presented in Table 8.6.

Subsequently, the developer can select a service with a high trust rate, such as Service 20 or 24. The general trust rates are different from the calculated trust rate based on the developer's trust preferences, as shown for the sort numbers service in Figure 8.12. If the developer selects a service based on the general $T_s$, he/she will be selecting Service 20, which has the highest general rating of $T_s = 8.8313$. However, based on the developer trust preferences, he/she will select Service 24, which has the highest preference rating of $T_s = 9.532$.

## 8.5   Summary

Trust bootstrapping, an important process for any trust-based system, helps requestors to select services. This chapter presents the trust-based SOA prototype as well as the implementation, experiment, and evaluation of the trust bootstrapping process. Specifically, the experiment

*Table 8.6: Scenario 2: Selecting a 'Sort Numbers' Service Based on Requestor's Trust Preferences.*

| sid | $T_{OSTM_r}$ | $T_{OSTM_l}$ | $T_{SSTM_{rem}}$ | Preference $T_s$ | General $T_s$ | $T_{PTM_{comp}}$ | $T_{PTM_{hons}}$ | pid | pnum |
|-----|--------------|--------------|------------------|------------------|---------------|------------------|------------------|-----|------|
| 10 | 9.7223 | 8 | 10 | 9.241 | 8.389 | 1 | 0 | 2 | 4 |
| 17 | 10 | 10 | 1 | 7 | 7.429 | 0 | 0 | 4 | 4 |
| 20 | 9.0667 | 3 | 10 | 7.356 | 8.831 | 1 | 0 | 5 | 3 |
| 24 | 9.846 | 8.75 | 10 | 9.532 | 8.514 | 0 | 0 | 6 | 4 |
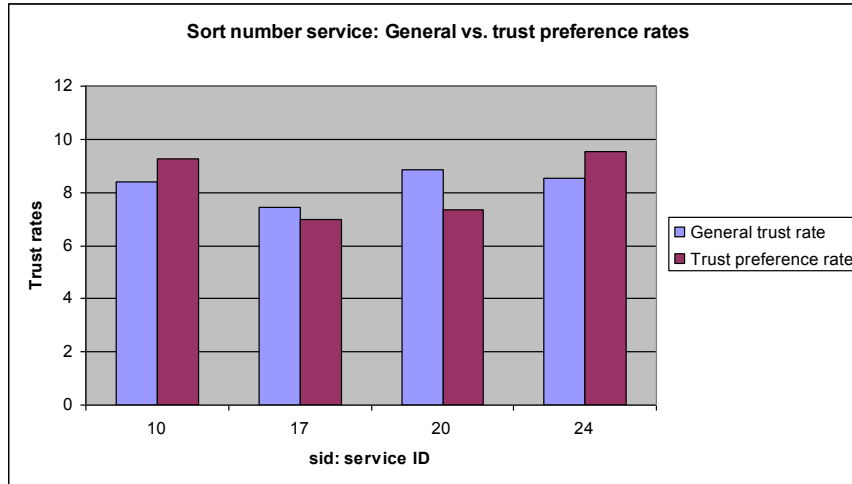


*Figure 8.12: General vs. Trust Preference Rate of the Sort Numbers Service.*

shows the trust bootstrapping output where services and providers have initial trust rates that are available at the discovery time of services to support service selection. Moreover, the trust experiment also demonstrates that the trust rates of services and their providers are related, and that services can be selected based on requestors' trust preferences. The next chapter presents the discussion and conclusion.

# Chapter 9

# Conclusion and Future Work

*"Trust GOD."* Zainab Aljazzaf

The main contribution of this thesis is to introduce a trust-based solution that includes the establishment of trust for services and providers and the selection of services based on trust criteria. In order to achieve this objective, this work builds a trust definition, identifies trust principles, develops a trust-based SOA by adding a trust mediator and additional link interactions, creates the trust mediator framework according to the trust definition and principles, and considers the rating of service providers. Subsequently, this thesis identifies trust metrics for services and providers, builds trust models for the metrics, services, and service providers to evaluate their trust ratings, and addresses different trust challenges, such as trust bootstrapping, whitewashing, and cold start.

Furthermore, a prototype of the trust-based SOA is developed, experiments are conducted, and the results are analysed to study the trust bootstrapping solution. The experiment shows that the registered services and service providers have initial trust rates through bootstrapping. Additionally, the results demonstrate that trust rates are available at the search time of services, the trust ratings of services and their providers are related, and services can be selected based on the requestors' trust preferences.

This chapter presents the contribution and future work. Specifically, the contribution and discussion are examined in Section 9.1, including the trust solution and trust principles in Sec-

tion 9.1.1 and the trust solution and trust challenges in Section 9.1.2. Future work in presented in Section 9.2.

# 9.1 Contributions and Discussion

A system that is not trust-based may return many services with similar functionalities. However, service requestors need to distinguish between similar services or select a service randomly. In a trust-based system, the system returns similar services with different trust rates, which helps service requestors make decisions about service selection.

The trust bootstrapping process assigns initial trust rates for new services and service providers. In a trust-based system that does not bootstrap new services, the system may return services that have no trust rates. Thus, the trust bootstrapping enhances the difficulty of selecting among competing services. This work addresses trust bootstrapping for services and service providers. To the best of our knowledge, there is no trust solution that establishes trust and addresses the trust bootstrapping problem as our solution.

This section summarizes and discusses the overall achievements of the thesis. Our solution is comprehensive and unique in building a trust solution based on three main aspects. First, the trust solution is built according to a defined trust term, as explicated in Section 4.1.1. Many researchers [21, 41, 38] do not define trust, provide immature trust definitions, or address trust as reputation or ranking. Second, while other trust solutions only cover some aspects of trust [21, 41, 38], our trust solution identifies and follows various trust principles incorporating many trust aspects, as presented in Section 4.1.2. Other trust models cover . Third, our trust solution addresses different trust challenges. The following two subsections present the trust solution with the trust principles and the trust challenges respectively.

## 9.1.1 Trust Solution and Trust Principles

Our trust solution is based on the trust principles, as follows:

- The trust solution considers the relation between trust and risk, and accordingly, the trust mediator includes risk remedies in the TMs, including $SSTM_{rem}$ and $PTM_{rem}$, to support this principle for trusting services and service providers. Furthermore, the trust mediator supports risk remedies for service broker, which is a TTP, by adding a remedies component in the trust mediator framework.

- The trust solution considers the trust development phases. Specifically, it addresses trust building and bootstrapping and the stabilization of trust through continuous dynamic evaluation. The trust framework supports the dissolution trust phase by adding the self-adjustment component to the trust framework.

- Trust establishment process considers the dynamic nature of trust. The trust mediator continuously evaluates the trustworthiness of the services and providers through different dynamic approaches, such as feedback and rating service providers based on their services.

- Trust depends on identity. In this work, services and service providers are identified by assigning IDs, as sid and pid.

- The trust solution considers trust semantic categories. It includes both specific and general measures based on one or more TMs. Trust measurement is based on judgement such as feedback or the calculation of trust rates using trust models to assess the trustworthiness of services and service providers.

- The trust solution considers the context specific, absolute, subjective, multi-degree, and multi-faceted relationship properties of trust by selecting services based on requestors' trust preferences. In addition, the trust mediator supports the development of trust relationships, such as the one-to-one bond between a requestor and a service and the one-to-many relation between a requestor and a service provider that offers many services.

- Global and local rates are both considered. The trust mediator builds both *general trust* rates based on all TMs and *trust preferences* rates, which are **local rates** based on trust preferences. The trust mediator can support **global rates**, or reputation, by allowing the

service consumers to send their feedback on the services and providers besides providing feedback on the TMs.

- Trust is based on information. The trust solution incorporates identified trust metrics as explained in Chapter 5. Specifically, the trust mediator tests services and service providers based on a list of identified TMs. Other researchers build trust based on QoS [7, 59, 87, 101] or one trust criterion, such as security [21] or privacy [4].

- First-party information is important for establishing trust. In particular, this information is supported where services and providers publish their information, or TMs, to build trust based on the provided information. For example, if the privacy TM is not published for a service, the trust mediator will not consider the privacy TM in the evaluation of the service's trust rate.

- Third-party ratings are important for establishing trust. The service broker act as a TTP and has a trust mediator for conducting the trust process and evaluating trust rates for services and service providers. In addition, the service broker plays an important role in trust management.

- This work clearly distinguishes between trust and QoS, and uses QoS properties as TMs.

- The trust mediator considers rating service provider.

- The trust solution supports requestors' trust preferences by identifying the TMs. The trust mediator rates the published TMs and stores the evaluated $T_{TM}$ into the rating registry, thus support services selection based on the requestors' trust preferences. When a requestor searches for a service, he/she selects a set of preferred TMs. Subsequently, the trust mediator calculates the trust rates of services based on the $T_{TM}$ of the selected TMs, or preference $T_s$. Accordingly, subjective and context-specific properties of trust are satisfied.

- The trust mediator considers different trust classes. In provision trust, requestors rely on the service broker and its trust mediator to rate services and providers and seek protection from malicious services and providers by overcoming whitewashers and malicious

122

behaviour. In delegation trust, requestors delegate the service broker to act on their behalf for identifying the trustworthiness of services and providers. In context trust, the trust solution takes into account remedies for services, service providers, and trust brokers. Finally, in certification trust, some of the service provider TMs, such as security, privacy, and honesty, can be certified.

## 9.1.2 Trust Solution and Trust Challenges

The trust mediator addresses different trust challenges, as follows:

- Community-based challenge (domain-specific information): The trust mediator is not community-based, because the trust solution utilizes general TMs, which are categorized in different categories and suit different domains. Requestors have many requirements, or trust preferences, and services have many properties, which are published as TMs. For example, if a requestor requires a service based on response time and security, he/she is able to search from different service brokers in different domains and communities for services that meet his/her requirement, because services support different TMs that can satisfy his/her trust preferences.

- Rating providers challenge: The trust solution rates service providers as well as services. The rating of providers helps to impede the problems of whitewashing, cold start, and malicious behaviour as well as reducing the overhead of trust bootstrapping new services in the trust mediator.

- Trust bootstrapping challenge: The trust solution includes an approach for trust bootstrapping services and providers. While current solutions in the trust literature address *reputation* bootstrapping, our approach is unique in addressing *trust* bootstrapping. In addition, the trust mediator bootstraps service providers beside bootstrapping services, and there are no previous works that consider the trust bootstrapping of service providers. Furthermore, the proposed trust bootstrapping approach addresses other limitations on the current bootstrapping approaches in the literature. For instance, our trust bootstrapping approach does not assign default values, is not based entirely on feedback, limits

the overhead on the requestor side, is not based on communities, thus it is a universal approach, and addresses cold start and whitewashing challenges.

- All services will be trust bootstrapped at publication time and have initial trust rates. The $T_{TM}$, $T_s$, and $T_{pr}$ are stored in the rating registry ready to be returned to the requestors. In particular, the initial $T_{TM}$ support the bootstrapping of the trust preference rates. Therefore, trust rates are available and the trust-based search can be conducted immediately at the search time, which results in shorter searching time.

  Other approaches in the literature may evaluate services' rates on the basis of information that is not supported by the services as trust information. For example, a service may have a low response time, but it does not publish the response time as a TM. Our approach will not consider the response time TM in the trust evaluation of the service, because the service does not provide that TM as a trust information; however, other approaches will consider the response time on the service trust evaluation. The other approaches rank all services that meet the non-functional QoS properties of a requestor regardless of whether or not the services publish the selected information as a TM.

- The trust bootstrapping of services and providers is initially conducted, and then the trust mediator continues the evaluation of the $T_{TM}$, $T_s$, and $T_{pr}$. This continuous and dynamic evaluation, based on different approaches, such as rating service providers, allows for additional evaluations of the trust rates and helps to impede the whitewashing and cold start. In contrast, other approaches are based on a one-time evaluation [7, 52].

- Change identity challenge or whitewashing: The trust mediator addresses the whitewashing challenge and establishes trust in a way that discourages providers from changing identities for the following reasons:

  1. Trust establishment based on a fair trust bootstrapping approach. The rejected providers cannot enter the service broker with initial default ratings values, such as average or high ratings.

  2. Trust establishment occurs through a long-term interactions with many of the provider's services. Thus, service providers who have changed their identity will undergo

a long process of trust validation to rebuild their trust, competence, and honesty. Moreover, the service broker can develop a more robust trust rate by considering the time and the value of the transactions.

3. Trust establishment is based on the providers' services. Thus, the trust rate of a service affects the rate of the provider. A provider who attempts to change a service's identity will not benefit because the provider's rate has already been affected by that malicious service. As a result, providers will be encouraged to behave in a trustworthy and in consistent manner.

- Cold start: The trust solution addresses the cold start issue by trust bootstrapping and conducting continuous evaluations of trust rates. Because rating is conducted at registration time, the trust mediator may perform a number of evaluations of the $T_{TM}$, $T_s$, and $T_{pr}$. The number of trust evaluations for $T_s$ and $T_{pr}$ is presented as snum and pnum, respectively, as presented in Section 8.3.3. Thus, services and service providers may have enough initial rates at searching time. $T_s$ and $T_{pr}$, along with snum and pnum, enable requestors to make better selection decisions.

- Services and service providers may accumulate high trust ratings and then attack the consumers or systems. The trust models use exponential averaging, which assigns more weight to the most recent interactions, thus minimizing the importance of past observations and mitigating variation and bias. In addition, exponential averaging reflects the true value of the current rating, and therefore, it is more capable of detecting malicious behaviour.

- The trust bootstrapping technique minimize the overhead on the requestor side. Specifically, requestors do not play a significant role in the trust bootstrapping process. Their contribution is relatively low, as they have the option of providing feedback after consuming services.

- The trust solution eliminates the need for matchmaking approach and the disclosing of requestors policies. Specifically, the solution identifies the TMs, from which requestors can select a service based on their trust preferences.

## 9.2 Future Work

Trust studies in Web Services and SOA are not mature, and further studies need to be conducted. This thesis represents a solid starting point for addressing trust for services and service providers in the SOA environment. The work started with the trust definition, trust principles, service provider ratings, trust-based SOA, trust framework, trust metrics, trust models, and trust bootstrapping, whitewashing, and cold start challenges. However, there are other issues as well as the trust mediator framework has many components that need to be further addressed in order to complete the trust solution, which is depicted as shaded areas in Figure 9.1. The following presents some future works.
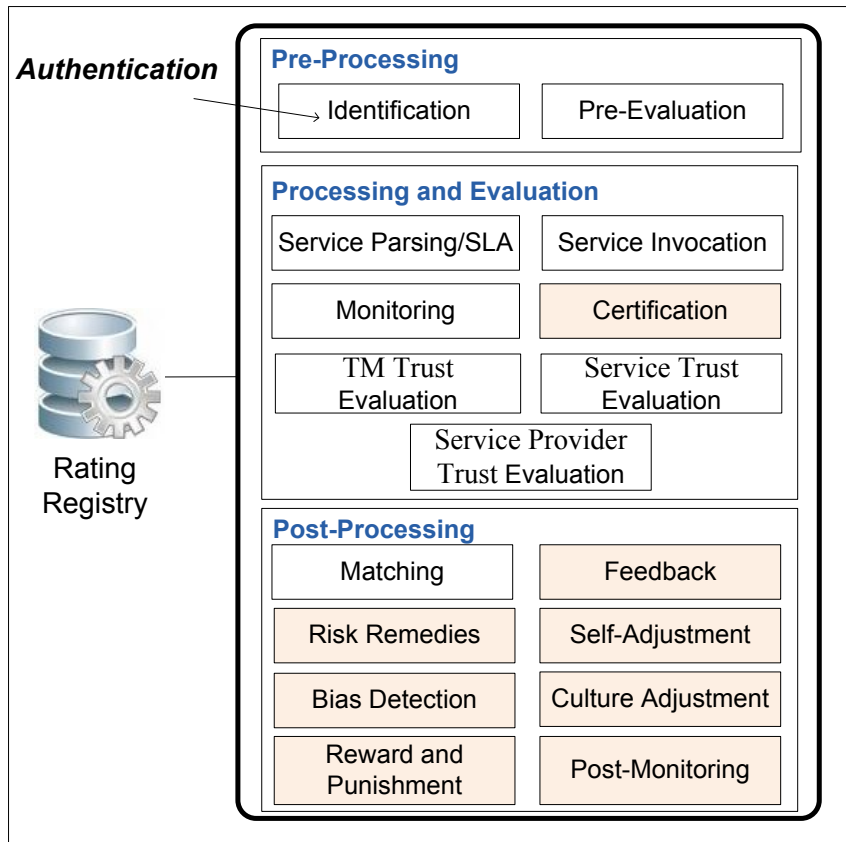


*Figure 9.1: Trust Mediator Framework - Future Work.*

- The identification component identifies services and service providers. However, authentication is required to insure their identity using authentication techniques such as X.509 [31]. Thus, there is a need to address the authentication for services and service providers.

- The certification component of the trust mediator needs to be addressed. Although this work uses a two-scale rating and the experiment assumes that trust rates are obtained from security and privacy rating systems, the certification component needs to conduct the certification process to assign trust levels for the security, privacy, and remedies SSTMs and provide a discrete scale rating for such SSTMs.

- The feedback component is necessary for addressing the unfair feedback and rating problem, and a technique is required to impede unjust ratings.

- The risk remedies component supports risk remedies for service brokers, which are TTPs. However, a technique is required to address the challenge of providing risk remedies for TTPs.

- The self-adjustment component considers the dynamic nature of trust, and so it is important to address trust degradation, trust declining, and trust re-building. Accordingly, a question that arises is: How to prove that it is theoretically possible for a service and service provider to get better after they started off with a bad rating?

- The bias detection component is responsible for detecting trust bias. Research is required to detect bias, ensure bias, and to deal with the services and service providers that are identified as having one or more biases in their trust rating.

- The culture adjustment component addresses the culture differences on the Internet by mitigating cultural differences when establishing trust. However, research is required to address cultural differences.

- The reward and punishment component is responsible for punishing and rewarding services and service providers. Accordingly, it requires developing different reward and punishment mechanisms.

- The post-monitoring component plays an important role in trust management. In particular, the self-adjustment and bias detection components require monitoring services for dynamically detecting any changes in their behaviour that may affect their overall trust rates and their providers' competence and honesty. In addition, a monitor component is important for the interactions between requestors and services to measure the OSTM of the consumed services. However, a monitor at the requestor side is important for monitoring the interactions between requestors and services to measure the TMs in practice of the consumed services by the service requestor.

- This work addresses and identifies trust models for a subset of the TMs, as presented in Figure 5.4, from different TMs categories from Figure 5.2. However, it is necessary to identify trust models for the other TMs, such as reliability, accuracy, usability, and ability, to be included in the trust evaluation of services and providers.

- This work assumes that the service broker is trusted, and most trust studies assume that TTPs are trusted. However, there is a need to address how to trust the TTPs, such as service brokers.

- The process of trust bootstrapping a composition of services needs to be addressed.

- Apply the trust solution into cloud computing to build a trust-based cloud computing. In particular, the trust mediator can be incorporated into a TTP to build trust for services from different clouds, which can be considered as service providers.

- Experiments need to be conducted to obtain feedback from requestors and study the effect of feedback in evaluating services and providers. In particular, the challenge of unfair feedback should be examined.

- The requestor GUI can be more simplified by adding a pull down menu with different options, such as "I want a really good service" or "I want a service from an honest provider". Moreover, trust metrics can be grouped into a more easy-to-understand terms, such as performance. Performance represents how fast a service request can be completed, and it includes response time, latency, execution time, transaction time, and throughput [78].

- Considering to address both the whitewashing and cold start challenges may raise some other challenges that need to be addressed further. For instance, a challenge arises if a provider publishes many good services to accumulate rates (i.e., addressing cold start) and become honest, and then defeat the system.

- Considering all the information provided by service providers as trust information will enable the trust mediator to build trust based on the all provided information. Accordingly, this will allow the solution to be more applicable for online use as well as it will encourage providers to behave well in all aspects rather than only a limited number of published trust information. Accordingly, the mediator needs to inform providers about considering their provided information as a trust information. Moreover, this allows requestors to provide feedback on any trust information and not only the ones selected as preferred trust information by the requestors.

- There are other important issues that needs to be addressed, such as the complexity of the trust bootstrapping process; the scalability of the trust mediator to handle growth; considering a decentralized trust-based architecture to overcome a single point of failure or single point of attack; and building a more comprehensive solution by considering asymmetric property of trust and build a mutual trust between the roles of SOA.

- The trust solution should be deployed in a real SOA application in order to determine its accuracy and performance as well as to optimize its features and functionalities. Systems using SOA need to support the trust metrics and incorporate the trust mediator in service brokers. Also, the services and service providers need to support the trust metrics.

Trust research in the services and SOA environment is not mature. Consequently, intensive research efforts need to be conducted for trust studies to mature and reach applicability. This work derives provision trust to provide soft trust, which can be built on WS-Trust that provides hard trust. The work in this thesis provides a foundation and a solid starting point by building a generic view definition of trust and identifying trust principles for the development of a robust trust solution. The ultimate objective is to obtain a standardized trust definition that facilitates the comparison of research works and achieves progress in the field. After trust is defined and its

principles are identified, the framework is built. Our proposed trust framework is comprehensive in establishing trust for services and providers starting with trust bootstrapping, a challenge that has been given a little attention in the literature. Our trust bootstrapping approach is unique in that it addresses *trust* bootstrapping rather than *reputation* bootstrapping. It considers trust bootstrapping service providers, it does not assign default values, thus it addresses the white-washing challenge, it addresses cold start challenge where services and service providers may accumulate a number of initial evaluated rates. Furthermore, it is not community-based because the trust solution utilizes general TMs, thus it is a universal approach, and it lowers the overhead at the requestor side without complex options. Moreover, trust rates are available at publishing time and the trust-based search can be conducted immediately at the search time by service requestors. In addition, the proposed trust solution considers different trust metrics, and builds trust models to rate services and service providers. Moreover, the designed framework can be deployed at the service broker of SOA and it can be deployed at other TTPs to establish trust for services and service providers.

# Appendix A

### A.1 The find maximum Web Service

```
package  p3 ;
import  javax . ejb . Stateless ;
import  javax . jws . WebService ;
import  java . sql . Timestamp ;
import  java . math . BigInteger ;



@WebService ( serviceName  =  ”WSDLp3FindMaxService ” ,
portName  =  ”WSDLp3FindMaxPort ” ,  endpointInterface  =
”org . netbeans . j2ee . wsdl . ejb3 . wsdlp3findmax . WSDLp3FindMaxPortType ” ,
targetNamespace  =  ”http :// j2ee . netbeans . org / wsdl /EJB3/WSDLp3FindMax” ,
wsdlLocation  =  ”META–INF/ wsdl /p3FindMAx/WSDLp3FindMaxWrapper . wsdl ”)
@Stateless
public  class  p3FindMAx {

        static  Timestamp  t1 ;
        static  Timestamp  t2 ;
        static  BigInteger  firstTime ;
        static  BigInteger  secondTime ;
        static  BigInteger  diffTime ;
        static  float  totalTime =0;
```

```
public void wsdLp3FindMaxOperation(int a, int b, int c, int d,
javax.xml.ws.Holder<Integer> r, javax.xml.ws.Holder<Float> te)
throws InterruptedException {

        BigInteger ONE_BILLION = new BigInteger ("1000000000");
        t1 = new Timestamp(System.currentTimeMillis());
        Thread.sleep(65);
        int[] num = new int[10];
        num[0]=a;
        num[1]=b;
        num[2]=c;
        num[3]=d;
        int Max = a;
        for ( int i = 1; i <= 3; i++)
            if ( num[i] > Max )
                Max = num[i];
        r.value = Max;

        t2 = new Timestamp(System.currentTimeMillis());

        firstTime  = BigInteger.valueOf(t1.getTime() / 1000 * 1000).
        multiply(ONE_BILLION).add(BigInteger.valueOf(t1.getNanos()));
        secondTime = BigInteger.valueOf(t2.getTime() / 1000 * 1000).
        multiply(ONE_BILLION).add(BigInteger.valueOf(t2.getNanos()));
        diffTime   = secondTime.subtract(firstTime);

        te.value = diffTime.floatValue()/1000000;  // response time
    }
}
```

**A.2 The WSDL file for the find maximum Web Service**

132

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="WSDLp3FindMax"
targetNamespace="http://j2ee.netbeans.org/wsdl/EJB3/WSDLp3FindMax"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://j2ee.netbeans.org/wsdl/EJB3/WSDLp3FindMax"
xmlns:plnk="http://docs.oasisopen.org/wsbpel/2.0/plnktype">
<types/>
<message name="WSDLp3FindMaxOperationRequest">
   <part name="a" type="xsd:int"/>
   <part name="b" type="xsd:int"/>
   <part name="c" type="xsd:int"/>
   <part name="d" type="xsd:int"/>
</message>
<message name="WSDLp3FindMaxOperationResponse">
   <part name="r" type="xsd:int"/>
   <part name="te" type="xsd:float"/>
</message>
   <portType name="WSDLp3FindMaxPortType">
      <operation name="WSDLp3FindMaxOperation">
         <input name="input1"
                message="tns:WSDLp3FindMaxOperationRequest"/>
         <output name="output1"
                message="tns:WSDLp3FindMaxOperationResponse"/>
      </operation>
   </portType>
<plnk:partnerLinkType name="WSDLp3FindMax">
   <plnk:role name="WSDLp3FindMaxPortTypeRole"
                portType="tns:WSDLp3FindMaxPortType"/>
```

```
</plnk:partnerLinkType>
</definitions>
```

**A.3 The service and provider tables**



**A.4 JDBC for reading and writing from the service table**

```
String dbtime;
String dbUrl = "jdbc:mysql://localhost:3306/trust";
String dbClass = "com.mysql.jdbc.Driver";
```

```
try {
try {
    Class . forName ("com. mysql . jdbc . Driver "). newInstance ();
} catch ( InstantiationException ex) {
    Logger . getLogger ( TrustMain . class . getName ()). log ( Level .SEVERE, null , ex );
} catch ( IllegalAccessException ex) {
    Logger . getLogger ( TrustMain . class . getName ()). log ( Level .SEVERE, null , ex );
}


Connection con = DriverManager . getConnection ( dbUrl , "root", "******");


// Now read and write to the service and the provider
// service :
// service table : For reading − get
Statement stmt = con . createStatement ();
ResultSet rs = stmt . executeQuery (" Select * FROM service ");
// service table : For writing − update
Statement stmtu = con . createStatement ( ResultSet . TYPE_SCROLL_SENSITIVE ,
                                          ResultSet . CONCUR_UPDATABLE );
ResultSet srs = stmtu . executeQuery ("SELECT * FROM service ");
```

**A.5 Provider GUI: publishing some TMs into the 'trust' SQL database**

```
PreparedStatement stmtp = con . prepareStatement ("INSERT INTO provider
                              VALUES ( ? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,?)");


// Publish latency OSTM
if ( vl ==1) {
    l_min= Float . parseFloat ( jTextField7 . getText ());
    l_max= Float . parseFloat ( jTextField9 . getText ());
    stmt . setFloat (11, l_min );
    stmt . setFloat (12, l_max );
```

```
} else {
   stmt.setFloat(11, l_min);
   stmt.setFloat(12, l_max);
}


// Publish throughput OSTM
if (vthr==1) {
   thr_min= Float.parseFloat(jTextField8.getText());
   thr_max= Float.parseFloat(jTextField10.getText());
   stmt.setFloat(15, thr_min);
   stmt.setFloat(16, thr_max);
}else {
   stmt.setFloat(15, thr_min);
   stmt.setFloat(16, thr_max);
}


// Publish remedies SSTM
if (vrem==1)
   rem = 1;
stmt.setInt(23, rem);


// Publish security SSTM
if (vsec==1)
   sec = 1;
stmt.setInt(19, sec);
```

**A.6 Requestor GUI: Find services based on the services' functional property and re-questor trust preferences**

```
if (!cal.equals("")){

 rss = con.prepareStatement("Select * FROM service where fun=?");
```

```java
rss.setString(1, cal);
ResultSet rs = rss.executeQuery();

while (rs.next()){
  // initialization
  int i=0; int o =0; int oo =0; int ooo =0;

  // Get the trust rates of the selected TMs
  if (te ==1){
     t[i]=rs.getFloat("T_te");
     i++;
  }
  if (tr ==1){
     t[i]=rs.getFloat("T_tr");
        i++;
  }
  if (l==1){
     t[i]=rs.getFloat("T_l");
     i++;
  }
  if (thr ==1){
     t[i]=rs.getFloat("T_thr");
     i++;
  }
  if (rem==1){
     t[i]=rs.getFloat("T_rem");
     i++;
  }
  if (sec ==1){
     t[i]=rs.getFloat("T_sec");
```

```
    i++;
}
if (prv==1){
    t[i]=rs.getFloat("T_prv");
    i++;
}

// array t[] has the values to calculate Ts
for (int j = 0; j < i; j++) {
    Tr=Tr+t[j];
}

// The same process to get the PTM and Tpr
......

// Print the output: list of services that satisfies the functional
// property and trust preferences along with trust rates..
String s = null;
s = "1-Service information:\nsid    Ts    Ts general    pid\n" +
Float.toString(T[0][0]) + "    " + Float.toString(T[0][1]) + "
" + Float.toString(T[0][2]) + "    " + Float.toString(T[0][3])
        + "\nProvider information:\nremedies   security
privacy   Tp   location   web site   brand name   comp?   #comp
honest   branded\n"
        + "  "+ p[0][0] + "    "+ p[0][1] + "
"+ p[0][2] + "    "+p[0][3]+ "    "+p1[0][0]+ "    "+p1[0][1]+ "
"+p1[0][2]+ "    "+ p2[0][0]+ "    "+p2[0][1]+ "    "+p2[0][2]+"
"+p2[0][3]+"\n";
        jTextField1.setText(s); // Send the output to the requestor GUI.
```

# Appendix B

Table 9.1, the Services Trust Table, presents the following information:

- sid: Represents the services' identifications.

- Function: Represents a function that a service provides.

- Provided OSTM: The provided OSTM is divided into the minimum and the maximum OSTM: OSTM min and OSTM max. For example, $OSTM_r min$ and $OSTM_r max$ are the provided response time ($OSTM_r$) by the provider.

- OSTM collected: OSTM collected are the monitored values of the OSTM. For example, the $OSTM_r$ collected is the monitored $OSTM_r$.

- $T_{OSTM}$: $T_{OSTM}$ are the trust rates of the OSTM. For example, $T_{OSTM_r}$ is the trust rate of the $OSTM_r$.

- SSTM: SSTM are the provided SSTM by the provider. For example, $OSTM_{sec}$ provided is the provided $OSTM_{sec}$.

- $T_{SSTM}$: $T_{SSTM}$ are the trust rates of the SSTM. For example, $T_{SSTM_{sec}}$ is the trust rate of the $OSTM_{sec}$.

- $T_s$: $T_s$ is the trust rate of the service.

- snum: Represents the number of times the trust rate of a service has been evaluated. Snum can be available for the requestors to support their decision in selecting services.

- pid: Represents the identification of the provider that provides a service.

*Table 9.1: Service Trust Table*

| pid | sid | Service's function | OSTM$_e$ provided | | OSTM$_e$ collected | T$_{OSTM_e}$ | OSTM$_r$ provided | | OSTM$_r$ collected | T$_{OSTM_r}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | OSTM$_e$min | OSTM$_e$max | | | OSTM$_r$min | OSTM$_r$max | | |
| 1 | 1 | Calculator | 25 | 27 | 25 | 10 | 26 | 36 | 28 | 10 |
| 1 | 2 | Find Maximum | 74 | 78 | 74 | 10 | 76 | 84 | 78 | 10 |
| 1 | 3 | Capital to Small | 34 | 35 | 34 | 10 | 35 | 43 | 38 | 10 |
| 1 | 4 | Temperature Converter | 42 | 44 | 42 | 10 | 44 | 52 | 58 | 8.8462 |
| 1 | 5 | Mean | 17 | 21 | 17 | 10 | 19 | 29 | 26 | 10 |
| 1 | 6 | Prime | 55 | 56 | 55 | 10 | 56 | 65 | 66 | 6.8462 |
| 2 | 7 | Calculator | 32 | 35 | 32 | 10 | 33 | 48 | 34 | 10 |
| 2 | 8 | Capital to Small | 25 | 26 | 25 | 10 | 27 | 36 | 44 | 7.7778 |
| 2 | 9 | Temperature Converter | 37 | 38 | 37 | 10 | 38 | 47 | 49 | 9.5745 |
| 2 | 10 | Sort Numbers | 62 | 63 | 62 | 10 | 64 | 72 | 74 | 9.7222 |
| 3 | 11 | Find Maximum | 65 | 67 | 66 | 10 | 67 | 76 | 70 | 10 |
| 3 | 12 | Temperature Converter | 27 | 28 | 27 | 10 | 29 | 36 | 36 | 10 |
| 3 | 13 | Mean | 20 | 21 | 20 | 10 | 22 | 30 | 22 | 10 |
| 4 | 14 | Find Maximum | 63 | 63 | 63 | 10 | 65 | 74 | 75 | 9.8649 |
| 4 | 15 | Mean | 23 | 23 | 23 | 10 | 25 | 32 | 73 | 8.4375 |
| 4 | 16 | Prime | 52 | 53 | 52 | 10 | 54 | 60 | 62 | 9.6667 |
| 4 | 17 | Sort Numbers | 60 | 60 | 60 | 10 | 61 | 67 | 63 | 10 |
| 5 | 18 | Find Maximum | 70 | 73 | 70 | 10 | 72 | 80 | 83 | 9.625 |
| 5 | 19 | Temperature Converter | 54 | 56 | 54 | 10 | 47 | 55 | 48 | 10 |
| 5 | 20 | Sort Numbers | 65 | 66 | 65 | 10 | 67 | 75 | 82 | 9.0667 |
| 6 | 21 | Calculator | 27 | 30 | 27 | 10 | 29 | 34 | 37 | 9.1176 |
| 6 | 22 | Temperature Converter | 43 | 44 | 43 | 10 | 45 | 48 | 50 | 9.5833 |
| 6 | 23 | Prime | 50 | 51 | 50 | 10 | 52 | 59 | 63 | 9.322 |
| 6 | 24 | Sort Numbers | 57 | 58 | 57 | 10 | 59 | 65 | 66 | 9.8462 |

*Table 9.1: continued*

| pid | sid | OSTM$_l$ provided | | OSTM$_l$ collected | T$_{OSTM_l}$ | OSTM$_{thr}$ provided | | OSTM$_{thr}$ collected | T$_{OSTM_{thr}}$ |
|---|---|---|---|---|---|---|---|---|---|
| | | OSTM$_l$min | OSTM$_l$max | | | OSTM$_{thr}$min | OSTM$_{thr}$max | | |
| 1 | 1 | 1 | 11 | 3 | 10 | 27.78 | 38.46 | 36 | 10 |
| 1 | 2 | 1 | 10 | 4 | 10 | 11.9 | 13.16 | 13 | 10 |
| 1 | 3 | 1 | 9 | 4 | 10 | 23.26 | 28.57 | 27 | 10 |
| 1 | 4 | 0 | 10 | 16 | 4 | 19.23 | 22.72 | 18 | 9.3604 |
| 1 | 5 | 2 | 12 | 9 | 10 | 34.48 | 52.63 | 39 | 10 |
| 1 | 6 | 1 | 10 | 11 | 9 | 15.38 | 17.86 | 16 | 10 |
| 2 | 7 | 1 | 11 | 2 | 10 | 23.25 | 30.3 | 30 | 10 |
| 2 | 8 | 1 | 11 | 19 | 2.727 | 27.78 | 37.03 | 23 | 8.2793 |
| 2 | 9 | 1 | 10 | 12 | 8 | 21.28 | 26.32 | 21 | 9.8684 |
| 2 | 10 | 2 | 10 | 12 | 8 | 13.89 | 15.53 | 14 | 10 |
| 3 | 11 | 2 | 11 | 4 | 10 | 13.16 | 14.92 | 15 | 10 |
| 3 | 12 | 2 | 9 | 9 | 10 | 27.78 | 34.48 | 28 | 10 |
| 3 | 13 | 2 | 10 | 2 | 10 | 33.33 | 45.45 | 46 | 10 |
| 4 | 14 | 2 | 11 | 12 | 9.091 | 13.5 | 15.38 | 14 | 10 |
| 4 | 15 | 2 | 9 | 14 | 4.444 | 27.78 | 40 | 28 | 10 |
| 4 | 16 | 2 | 8 | 10 | 7.5 | 16.67 | 18.51 | 17 | 10 |
| 4 | 17 | 1 | 7 | 3 | 10 | 14.92 | 16.39 | 16 | 10 |
| 5 | 18 | 2 | 10 | 13 | 7 | 12.5 | 13.89 | 13 | 10 |
| 5 | 19 | 2 | 10 | 3 | 10 | 18.18 | 21.28 | 21 | 10 |
| 5 | 20 | 2 | 10 | 17 | 3 | 13.33 | 14.92 | 13 | 9.7524 |
| 6 | 21 | 2 | 7 | 10 | 5.714 | 29.41 | 34.48 | 28 | 9.5206 |
| 6 | 22 | 2 | 5 | 7 | 6 | 20.83 | 22.22 | 20 | 9.6015 |
| 6 | 23 | 2 | 9 | 13 | 5.556 | 16.94 | 19.23 | 16 | 9.4451 |
| 6 | 24 | 2 | 8 | 9 | 8.75 | 15.38 | 16.95 | 16 | 10 |

Table 9.1: continued

| pid | sid | $SSTM_{sec}$ supported? | $T_{SSTM_{sec}}$ | $SSTM_{prv}$ supported? | $T_{SSTM_{prv}}$ | $SSTM_{rem}$ supported? | $T_{SSTM_{rem}}$ | $T_s$ | snum |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 10 | 1 | 10 | 1 | 10 | 10 | 1 |
| 1 | 2 | 1 | 10 | 1 | 10 | 1 | 10 | 10 | 1 |
| 1 | 3 | 1 | 10 | 1 | 10 | 1 | 10 | 10 | 1 |
| 1 | 4 | 1 | 10 | 1 | 10 | 1 | 10 | 8.8866 | 1 |
| 1 | 5 | 1 | 10 | 1 | 10 | 1 | 10 | 10 | 1 |
| 1 | 6 | 1 | 10 | 1 | 10 | 1 | 10 | 9.8352 | 1 |
| 2 | 7 | 1 | 10 | 1 | 1 | 1 | 10 | 8.714 | 1 |
| 2 | 8 | 1 | 10 | 1 | 1 | 1 | 10 | 7.112 | 1 |
| 2 | 9 | 1 | 10 | 1 | 1 | 1 | 10 | 8.349 | 1 |
| 2 | 10 | 1 | 10 | 1 | 1 | 1 | 10 | 8.389 | 1 |
| 3 | 11 | 1 | 1 | 1 | 10 | 1 | 10 | 8.714 | 1 |
| 3 | 12 | 1 | 1 | 1 | 10 | 1 | 10 | 8.714 | 1 |
| 3 | 13 | 1 | 1 | 1 | 10 | 1 | 10 | 8.714 | 1 |
| 4 | 14 | 1 | 10 | 1 | 1 | 1 | 1 | 7.279 | 1 |
| 4 | 15 | 1 | 10 | 1 | 1 | 1 | 1 | 6.412 | 1 |
| 4 | 16 | 1 | 10 | 1 | 1 | 1 | 1 | 7.024 | 1 |
| 4 | 17 | 1 | 10 | 1 | 1 | 1 | 1 | 7.429 | 1 |
| 5 | 18 | 1 | 10 | 1 | 10 | 1 | 10 | 9.518 | 1 |
| 5 | 19 | 1 | 10 | 1 | 10 | 1 | 10 | 10 | 1 |
| 5 | 20 | 1 | 10 | 1 | 10 | 1 | 10 | 8.831 | 1 |
| 6 | 21 | 1 | 10 | 1 | 1 | 1 | 10 | 7.907 | 1 |
| 6 | 22 | 1 | 10 | 1 | 1 | 1 | 10 | 8.026 | 1 |
| 6 | 23 | 1 | 10 | 1 | 1 | 1 | 10 | 7.903 | 1 |
| 6 | 24 | 1 | 10 | 1 | 1 | 1 | 10 | 8.514 | 1 |

# Bibliography

[1] Security in a web services world: a proposed architecture and roadmap. Technical report, IBM corporation and Microsoft corporation, April 2002.

[2] Dictionary.com. http://dictionary.reference.com/browse/trust - last accessed April, 2011.

[3] *Merriam-Webster, 2011.* , http://www.merriam-webster.com/dictionary/trust - last accessed April, 2011.

[4] Truste. http://www.truste.com/ - last accessed April, 2011.

[5] Verisign certification practice statement. Version 3.8.1, Effective Date: February 01, 2009. [online] https://www.verisign.com/ - last accessed April, 2011.

[6] A. Abdul-Rahman. The PGP trust model. *Journal of Electronic Commerce*, 1997.

[7] E. Al-Masri and Q.H. Mahmoud. Discovering the best web service. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1257–1258, New York, NY, USA, 2007. ACM.

[8] D.S. Allison, H.F. El Yamany, and M.A.M. Capretz. Metamodel for privacy policies within SOA. *ICSE Workshop on SESS '09: Software Engineering for Secure Systems,*, pages 40 –46, may 2009.

[9] D.S. Allison, H.F. El Yamany, and M.A.M. Capretz. Privacy and trust policies within SOA. *ICITST '09: International Conference for Internet Technology and Secured Transactions, 2009.*, pages 1–6, Nov. 2009.

[10] E. Bataineh, Z. Maamar, D. Benslimane, and C. Ghedira. Web service = E + F + I. *WET-ICE '06: 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 41 –46, june 2006.

[11] N. Bieberstein, S. Bose, M. Fiammante, K. Jones, and R. Shah. *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

[12] K. J. Blois. Trust in business to business relationships: An evaluation of its status. *Journal of Management Studies*, 36(2):197–215, 03 1999.

[13] L. Buttyan and J. Hubaux. *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press, New York, NY, USA, 2007.

[14] J. Cao, J. Huang, G. Wang, and J. Gu. QoS and preference based web service evaluation approach. *GCC '09: Eighth International Conference on Grid and Cooperative Computing,*, pages 420 – 6, 2009.

[15] E. Chang, T.S. Dillon, and F.K. Hussain. Trust and reputation relationships in service-oriented environments. *ICITA 2005: Third International Conference on Information Technology and Applications*, 1:4 – 14 vol.1, July 2005.

[16] M. Chen, L. He, X. Cai, and W. Xia. Trust evaluation model for composite service based on subjective logic. *IIHMSP '08: International Conference on Intelligent Information Hiding and Multimedia Signal Processing.*, pages 1482 –1485, Aug. 2008.

[17] C. L. Corritore, B. Kracher, and S. Wiedenbeck. On-line trust: concepts, evolving themes, a model. *International Journal of Human-Computer Studies*, 58(6):737–758, 2003.

[18] M. Daignault, M. Shepherd, S. Marche, and C. Watters. Enabling trust online. In *ISEC '02: Proceedings of the Third International Symposium on Electronic Commerce*, page 3. IEEE Computer Society, 2002.

[19] S. David. *Changing Minds: in Detail*. Syque Press, 2008.

[20] R. Dingledine, M.J. Freedman, and D. Molnar. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter 16: accountability. O'Reilly, 2000.

[21] N. Dragoni. Toward trustworthy web services - approaches, weaknesses and trust-by-contract framework. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 3:599–606, 2009.

[22] J. V. Dyke. Establishing federated trust networks among web services. Bachelor thesis of Science in Computer Engineering, University of Virginia, 2004.

[23] H. F. El Yamany. *A fine-grained intelligent security framework for service-oriented architecture*. PhD thesis, Ontario, Canada, 2009.

[24] Entrust. Web services trust and XML security standards. Version: 1.0, April 2001. http://download.entrust.com/resources/download.cfm/21165/ - last accessed November, 2011.

[25] M. Feldman and Chuang J. The evolution of cooperation under cheap pseudonyms. *Proceedings of the Seventh IEEE International Conference on E-Commerce Technology*, pages 284–291, 2005.

[26] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111, New York, NY, USA, 2004. ACM.

[27] C. Ferris, A. Barbir, S. Garg, and D. Austin. Web services architecture requirements. W3C note, W3C, Feb 2004. http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211 - last accessed April, 2011.

[28] S. Galizia, A. Gugliotta, and J. Domingue. A trust based methodology for web service selection. In *ICSC 2007 - International Conference on Semantic Computing, 2007.*, pages 193 –200, sept. 2007.

[29] D. Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[30] D. Z. G. Garcia and M. B. F. de Toledo. Semantics-enriched qos policies for web service interactions. *Proceedings of the 12th Brazilian Symposium on Multimedia and the web*, pages 35–44, 2006.

[31] E. Gerck and E. Gerck. Overview of certification systems: X.509, CA, PGP and SKIP. *Journal of Computer Science and Technology*, first published on April 17, 1997.

[32] S. Grabner-Kräuter and E. A. Kaluscha. Empirical research in on-line trust: a review and critical assessment. *International Journal of Human-Computer Studies*, 58(6):783–812, June 2003.

[33] T. Grandison and S. Sloman. A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials*, 3(4), 2000.

[34] D. Hoyle. *Automotive Quality Systems Handbook*. Elsevier Ltd, second edition edition, 2005.

[35] D. Hoyle. *ISO 9000 Quality Systems Handbook*. Elsevier Ltd, fifth edition edition, 2006.

[36] H. Huang, C. Keser, J. Leland, and J. Shachat. Trust, the Internet, and the digital divide. *IBM system journal*, 42(3):507–518, 2003.

[37] M. N. Huhns and M. P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9:75–81, 2005.

[38] S. Jin-Dian, G. He-Qing, and G. Yin. An adaptive trust model of web services. *Journal Wuhan University Journal of Natural Sciences*, 2005.

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

[40] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1026–1027, 2003.

[41] S. Kalepu, S. Krishnaswamy, and S.W. Loke. Verity: a QoS metric for selecting web services and providers. *WISEW '03: Proceedings Fourth International Conference on Web Information Systems Engineering Workshops*, pages 131 – 139, Dec. 2003.

[42] T. Kautonen and H. Karjaluoto, editors. *Trust and New Technologies: Marketing and Management on the Internet and Mobile Media.* Edward Elgar, 2008.

[43] L. Kelvin, K. Chris, N. Anthony, G. Marc, G. Martin, B. Abbie, and G. Hans. Ws-trust 1.4. Technical report, OASIS Standard, [online] http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html, February 2009.

[44] T. A. Khopkar. *Provision, Interpretation and Effects of Feedback in Reputation Systems.* PhD thesis, School of Information, The University of Michigan, 2008.

[45] Y. Kim. QoS-aware web services discovery with trust management. *JCIT '08: Journal of Convergence Information Technology*, 3(2):67–73, 2008.

[46] Y. Kim and D. Doh. A trust type based model for managing QoS in web services composition. *International Conference on Convergence Information Technology*, 0:438–443, 2007.

[47] K.D. Larson. The role of service level agreements in IT service delivery. *Information Management and amp; Computer Security*, 6(3):128 – 32, 1998.

[48] K. Lee, J. Jeon, W. Lee, S. Jeong, and S. Park. QoS for web services: Requirements and possible approaches. Technical report, W3C, Web Services Architecture Working Group, November 2003.

[49] Y. Liu, A. Ngu, and L. Zeng. QoS computation and policing in dynamic web service selection. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73, New York, NY, USA, 2004. ACM.

[50] Z. Liu, S.S. Yau, D. Peng, and Y. Yin. A flexible trust model for distributed service infrastructures. In *(ISORC '11: 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing*, pages 108 –115, 5-7 2008.

[51] Z. Malik and A. Bouguettaya. Rater credibility assessment in web services interactions. *World Wide Web journal*, 12(1):3–25, 2009.

[52] Z. Malik and A. Bouguettaya. Reputation bootstrapping for trust establishment among web services. *IEEE Internet Computing*, 13(1):40–47, 2009.

[53] D. W. Manchala. Trust metrics, models and protocols for electronic commerce transactions. In *ICDCS '98: Proceedings of the The $18^{th}$ International Conference on Distributed Computing Systems*, page 312, Washington, DC, USA, 1998. IEEE Computer Society.

[54] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, New York, NY, USA, 2007. ACM.

[55] P. Massa and P. Avesani. Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. *International Journal on Semantic web and Information Systems*, 2007.

[56] C. M. Matthew, L. Ken, M. Francis, B. Peter, and M. Rebekah. Reference model for service oriented architecture 1.0. Technical report.

[57] E. Maximilien and M. Singh. Reputation and endorsement for web services. *In Proceedings of SIGecom Exchanges*, 3(1), 2002.

[58] E. Maximilien and M. Singh. Toward autonomic web services trust and selection. In Marco Aiello, Mikio Aoyama, Francisco Curbera, and Mike P. Papazoglou, editors, *IC-SOC '04: In Proceedings of Second International Conference in Service-Oriented Computing*, pages 212–221, New York, NY, USA, 2004. ACM.

[59] E. Maximilien and M. Singh. Multiagent system for dynamic web services selection. In *In Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS)*, pages 25–29, 2005.

[60] F.L. Mayer. A brief comparison of two different environmental guidelines for determining 'levels of trust' [computer security]. *Proceedings of the Sixth Annual Conference on Computer Security Applications , 1990.*, pages 244 –250, Dec 1990.

[61] R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *The Academy of Management Review*, 20(3):709–734, 1995.

[62] D. H. McKnight and N. L. Chervany. The meanings of trust. Technical report MISRC Working Paper, University of Minnesota, 1996.

[63] D.A. Menasce. QoS issues in web services. *IEEE Internet Computing*, 6(6):72 – 75, Nov/Dec 2002.

[64] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting honest feedback in electronic markets. Working Paper Series rwp02-039, Harvard University, John F. Kennedy School of Government, September 2002.

[65] A. Moorsel. Metrics for the internet age: Quality of experience and quality of business. 5th Performability Workshop, 2001.

[66] L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt. Ratings in distributed systems: A bayesian approach. *WITS '01: Proceedings of the Workshop on Information Technologies and Systems*, 2001.

[67] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 2431 – 2439, 7-10 2002.

[68] M. Natrella. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH, July 2010.

[69] H. T. Nguyen, W. Zhao, and J. Yang. A trust and reputation model based on bayesian network for web services. *ICWS '10: IEEE International Conference on Web Services*, pages 251 –258, jul. 2010.

[70] L. O'Brien, P. Merson, and L. Bass. Quality attributes for service-oriented architectures. In *SDSOA '07: Proceedings of the International Workshop on Systems Development in SOA Environments*, page 3, Washington, DC, USA, 2007. IEEE Computer Society.

[71] K. O'Hara, H. Alani, Y. Kalfoglou, and N. Shadbolt. Trust strategies for the semantic web. In *ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, 2004.

[72] D. Olmedilla, R. Lara, A. Polleres, and H. Lausen. Trust negotiation for semantic web services. In *Proc. of the first International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*, San Diego, California, USA, July 2004.

[73] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[74] M. Papazoglou. *Web Services: Principles and Technology*. Prentice Hall, 2008.

[75] M. P. Papazoglou and D. Georgakopoulos, editors. *Service-Oriented Computing*. The MIT Press, Cambridge, MA, 2008.

[76] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, and B. J. Krmer. Service-oriented computing research roadmap. In *Dagstuhl Seminar Proceedings 05462*, pages 1–29, April 2006.

[77] W. Rahman and F. Meziane. Challenges to describe QoS requirements for web services quality prediction to support web services interoperability in electronic commerce. In *Proceedings of the 10th IBIMA Conference on Innovation and Knowledge Management in Business Globalization, Kuala Lumpur, Malaysia, 30 June - 2 July 2008, 4 (6) , pp. 50-58*.

[78] S. Ran. A model for web services discovery with QoS. *ACM SIGecom Exchanges*, 4(1):1–10, 2003.

[79] L. Rasmusson and S. Jansson. Simulated social control for secure internet commerce. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 18–25, New York, NY, USA, 1996. ACM.

[80] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[81] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[82] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer. *Applied SOA: Service-Oriented Architecture and Design Strategies*. Wiley Publishing, 2008.

[83] F. Rosenberg, C. Platzer, and S. Dustdar. Bootstrapping performance and dependability attributes of web services. pages 205–212. IEEE Computer Society, 2006.

[84] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. The adaptive web. chapter Collaborative filtering recommender systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[85] M. Sergio and G. Hector. Taxonomy of trust: Categorizing P2P reputation systems. Working Paper 2005-11, Stanford InfoLab, 2005.

[86] W. Shao-Jie, S. Gui-Cheng, Z. Xue-Feng, C. Li-Jun, and Y. Zhen. A trust model of web services based on individual experience. *WiCom 2007: International Conference on Wireless Communications, Networking and Mobile Computing.*, pages 3205 –3208, sept. 2007.

[87] W. Sherchan, S. W. Loke, and S. Krishnaswamy. A fuzzy model for reasoning about reputation in web services. In Hisham Haddad, editor, *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1886–1892. ACM, 2006.

[88] R. Song, L. Korba, and G. Yee. *Trust in E-services: Technologies, Practices and Challenges*. IGI Global, 2007.

[89] G. Swamynathan. *Towards Reliable Reputations For Distributed Applications*. PhD thesis, Computer Science, University of California, Santa Barbara, 2008.

[90] L. Vu, M. Hauswirth, and K.l Aberer. Qos-based service selection and ranking with trust and reputation management. In *Lecture Notes in Computer Science (including subseries*

*Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3760 LNCS, pages 466 – 483, Agia Napa, Cyprus, 2005.

[91] S. Wang, L. Zhang, S. Wang, and X. Qiu. A cloud-based trust model for evaluating quality of web services. *Journal of Computer Science and Technology*, 25(6):1130–1142, 2010.

[92] Y. Wang and J. Vassileva. Bayesian network trust model in peer-to-peer networks. In Gianluca Moro, Claudio Sartori, and Munindar P. Singh, editors, *AP2PC '03: Agents and Peer-to-Peer Computing, Second International Workshop*, volume 2872 of *Lecture Notes in Computer Science*, pages 23–34. Springer, 2003.

[93] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *P2P '03: Proceedings Third International Conference on Peer-to-Peer Computing*, pages 150 – 157, 1-3 2003.

[94] Y. Wang and J. Vassileva. A review on trust and reputation for web service selection. In *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, page 25, Washington, DC, USA, 2007. IEEE Computer Society.

[95] Z. Ying-Feng and S. Pei-Ji. The model for consumer trust in C2C online auction. *ICMSE '06: International Conference on Management Science and Engineering*, pages 125 – 129, Oct. 2006.

[96] P. Yolum and M. P. Singh. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man, and Cybernetics. Part A*, 35(3):396–407, 2005.

[97] Bin Yu, M.P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *2004 IEEE First Symposium on Multi-Agent Security and Survivability*, pages 1 – 10, 30-31 2004.

[98] W. D. Yu, R. B. Radhakrishna, S. Pingali, and V. Kolluri. Modeling the measurements of QoS requirements in web service systems. *Simulation journal*, 83(1):75–91, 2007.

[99] M. Yuan and J. Long. Securing wireless j2me. Technical report, IBM, 2002.

[100] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29(4):371–388, 2000.

[101] Y. Zhang, Z. Zheng, and M.R. Lyu. Wsexpress: A qos-aware search engine for web services. *ICWS '10: IEEE International Conference on Web Services*, pages 91 –98, jul. 2010.

[102] W. Zhao and V. Varadharajan. Trust management for web services. In *ICWS '08: Proceedings of the 2008 IEEE International Conference on Web Services*, pages 818–821, Washington, DC, USA, 2008. IEEE Computer Society.

[103] L. Zhengping, L. Xiaoli, W. Guoqing, Y. Min, and Z. Fan. A formal framework for trust management of service-oriented systems. In *SOCA '07: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications*, pages 241–248, Washington, DC, USA, 2007. IEEE Computer Society.

# Curriculum Vitae

**Name:**

Zainab Mohammed Hussain Aljazzaf

**Post-Secondary Education and Degrees:**

- Bachelor, Electrical and Computer Engineering
  College of Engineering and Petroleum
  Kuwait University, Kuwait
  $1993 - 1997$

- Master, Electrical and Computer Engineering
  College of Engineering and Petroleum
  Kuwait University, Kuwait
  $2003 - 2005$

- Doctorate, Computer Science
  Faculty of Science
  The University of Western Ontario, Canada
  $2006 - 2011$

**Honours and Awards:**

- Student advisory committee member of the top student
  in Electrical and Computer Engineering Department
  College of Engineering and Petroleum

Kuwait University, Kuwait

1996

- Student advisory committee member of the top student
  in Electrical and Computer Engineering Department
  College of Engineering and Petroleum
  Kuwait University, Kuwait
  1997

- Kuwait University award for the top graduates
  2005

- Kuwait University Scholarship
  $2006 - 2011$

- Western Graduate Thesis Research Award
  Faculty of Science
  The University of Western Ontario, Canada
  $2010 - 2011$

- First prize awards in UWO Resarch in Computer Science Conference, UWORCS 2011
  Faculty of Science
  The University of Western Ontario, Canada
  April, 2011

**Related Work Experiences:**

- Computer Engineer
  Kuwait Ministry of Affairs
  $1997 - 2003$

- Teaching Assistant
  Electrical and Computer Engineering Department
  Kuwait University, Kuwait
  2005

- Teaching Assistant
  College for Women
  Kuwait University, Kuwait
  $2005 - 2006$

**Publications:**

1- **Z. M. Aljazzaf**, M. Perry, and M. A.M. Capretz. Trust in Web Services. IEEE 6th World Congress on Services 2010, SERVICES 2010, pp. 189-190, Miami, FL, USA, 2010. IEEE Computer Society.

2- **Z. M. Aljazzaf**, M. Perry, and M. A.M. Capretz. Online Trust: Definition and principles. ICCGI 2010: The Fifth International Multi-Conference on Computing in the Global Information Technology, September 20-25, Valencia, Spain, 2010. IEEE Computer Society.

3- **Z. M. Aljazzaf**, M. Perry, and M. A.M. Capretz. Trust Metrics for Services and Service Providers. ICIW 2011: The Sixth International Conference on Internet and Web Applications and Services, March 20-25, St. Maarten, The Netherlands Antilles, 2011.

4- **Z. M. Aljazzaf**, M. Perry, and M. A.M. Capretz. Toward a Unified Trust Framework for Trust Establishment and Trust Based Service Selection. CCECE 2011: $24^{th}$ Canadian Conference on Electrical and Computer Engineering, May 8-11, Niagara Falls, Ontario, Canada, 2011. IEEE Canada.

5- **Z. M. Aljazzaf**, M. A.M. Capretz, and M. Perry. Trust Bootstrapping Services and Service Providers. PST 2011: The Ninth Annual International Conference on Privacy, Security and Trust, July 19-21, Montreal, Quebec, Canada, 2011. IEEE Computer Society.