# Trust Establishment Mechanisms for Distributed Service Environments

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

**Dissertation**

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

**Sheikh Mahbub Habib, M. Sc.**

geboren in *Chittagong, Bangladesch*

Referenten
Prof. Dr. Max Mühlhäuser (Technische Universität Darmstadt)
Prof. Dr. Vijay Varadharajan (Macquarie University)

Tag der Einreichung: 12.07.2013
Tag der mündlichen Prüfung: 27.08.2013

Darmstadt 2014
Hochschulkennziffer D17

# Dedication

To my mother for whom I exist!

# Acknowledgements

This thesis could not have been successfully completed without dedicated and generous support from my colleagues, friends and family members.

At first, I would like to convey my heartfelt gratitude and acknowledgement to my advisor Max Mühlhäuser for his faith on me and my work during PhD studies. His advice during the tough periods of doctoral studies as well as his critical assessment and appreciation of my work kept me always on the right track to achieve my goal. Next, I would like to thank my co-advisor Vijay Varadharajan for his guidance and valuable advice during my doctoral studies.

I would like to acknowledge intensive guidance and dedicated mentorship of Sebastian Ries during my PhD endeavour. During this endeavour, I got appreciation as well as criticism regarding my work from some of my brilliant colleagues, Sascha Hauke, Florian Volk, Leonardo Martucci, Stefan G. Weber, to name but a few. Thank you very much for providing your valuable feedback during my doctoral studies. Additionally, I would also like to thank Ryan Ko, Marlin Pohlman and Said Tabet for helping me to get the datasets, used in this thesis, from the Cloud Security Alliance (CSA).

Moreover, I am grateful to the Center for Advanced Security Research Darmstadt (CASED) and the Telecooperation Lab (TK) for providing necessary support and infrastructure during my doctoral studies at Technische Universität Darmstadt (TU Darmstadt). I would also like to thank the Advanced Cyber Security Research Center (ACSRC) of Macquarie University Australia for providing me the opportunity to work with their brilliant researchers during my research visit in 2012. Heartiest thanks goes to all of my dear colleagues at CASED, TK, and ACSRC for creating a supportive and friendly environment during my stay.

Furthermore, this thesis has been critically reviewed by a number of people. My generous thanks goes to Max Mühlhäuser, Stefan Schiffner, Mathias Fischer, and Vijay Varadharajan for providing valuable comments to improve the content of this thesis. Thanks goes to Cecilia Burns for proof-reading the thesis.

Finally, I would like to thank my parents, my brother, my sisters, and my dearest wife for encouraging me throughout the doctoral studies.

# Abstract

The aim and motivation of this dissertation can be best described in one of the most important application fields, the cloud computing. It has changed entire business model of service-oriented computing environments in the last decade. Cloud computing enables information technology related services in a more dynamic and scalable way than before – more cost-effective than before due to the economy of scale and of sharing resources. These opportunities are too attractive for consumers to ignore in today's highly competitive service environments. The way to realise these opportunities, however, is not free of obstacles. Services offered in cloud computing environments are often composed of multiple service components, which are hosted in distributed systems across the globe and managed by multiple parties. Potential consumers often feel that they lose the control over their data, due to the lack of transparent service specification and unclear security assurances in such environments. These issues encountered by the consumers boiled down to an unwillingness to depend on the service providers regarding the services they offer in the marketplaces. Therefore, consumers have to be put in a position where they can reliably assess the dependability of a service provider. At the same time, service providers have to be able to truthfully present the service-specific security capabilities. If both of these objectives can be achieved, consumers have a basis to make well-founded decisions about whether or not to depend on a particular service provider out of many alternatives.

In this thesis, *computational trust mechanisms* are leveraged to assess the capabilities and evaluate the dependability of service providers. These mechanisms, in the end, potentially support consumers to establish trust on service providers in distributed service environments, e.g., cloud computing. In such environments, acceptable quality of the services can be maintained if the providers possess required capabilities regarding different service-specific attributes, e.g., security, performance, compliance. As services in these environments are often composed of multiple services, subsystems and components, evaluating trustworthiness of the service providers based on the service-specific attributes is non-trivial.

In this vein, novel mechanisms are proposed for assessing and evaluating the trustworthiness of service providers considering the trustworthiness of composite services. The scientific contributions towards those novel mecha-

nisms are summarised as follows:

- Firstly, we introduce a list of service-specific attributes, *QoS+* [HRM10, HHRM12], based on a systematic and comprehensive analysis of existing literatures in the field of cloud computing security and trust.
- Secondly, a *formal framework* [SVRH11, RHMV11a, RHMV11b] is proposed to analyse the composite services along with their required service-specific attributes considering consumer requirements and represent them in simplified meaningful terms, i.e., Propositional Logic Terms (PLTs).
- Thirdly, a novel trust evaluation framework *CertainLogic* [RHMV11a, RHMV11b, HRHM12a, HRHM12b] is proposed to evaluate the PLTs, i.e., capabilities of service providers. The framework provides computational operators to evaluate the PLTs, considering that *uncertain* and *conflicting* information are associated with each of the PLTs and those information can be derived from multiple sources.
- Finally, harnessing these technical building blocks we present a novel *trust management architecture* [HRM11] for cloud computing marketplaces. The architecture is designed to support consumers in assessing and evaluating the *trustworthiness* of service providers based on the published information about their services.

The novel contributions of this thesis are evaluated using proof-of-concept-system, prototype implementations and formal proofs. The proof-of-concept-system [HRMV13, HVM13a, HVM13b] is a realisation of the proposed architecture for trust management in cloud marketplaces. The realisation of the system is implemented based on a self-assessment framework, proposed by the Cloud Security Alliance, where the formal framework and computational operators of CertainLogic are applied. The realisation of the system enables consumers to evaluate the *trustworthiness* of service providers based on their published datasets in the CSA STAR. A number of experiments are conducted in different cloud computing scenarios leveraging the datasets in order to demonstrate the technical feasibility of the contributions made in this thesis. Additionally, the prototype implementations of CertainLogic framework provide means to demonstrate the characteristics of the computational operators by means of various examples. The formal framework as well as computational operators of CertainLogic are validated against desirable mathematical properties, which are supported by formal algebraic proofs.

# Zusammenfassung

Ziel und Motivation der vorliegenden Dissertation lassen sich am besten an einem ihrer wichtigsten Anwendungsgebiete beschreiben, dem Cloud Computing. Dessen zunehmende Bedeutung hat in letzten Jahrzehnt die Geschäftsmodelle für IT-Dienste grundlegend verändert. Cloud Computing ermöglicht dynamischere und flexiblere Dienstangebote; Skaleneffekte und gemeinsame Ressourcennutzung können erhebliche Kostenreduktion bewirken. Der allgegenwärtige Wettbewerb und Kostendruck führt zu großem Interesse vieler IT-Dienstanbieter an Cloud-Computing-Diensten; Einsatz und Nutzung entsprechender Angebote sind jedoch nicht frei von Hindernissen. In Cloud-Umgebungen angebotene Dienste sind zunehmend aus verschiedenen Teilkomponenten zusammengesetzt, welche ggf. auf über den ganzen Globus verteilten Systemen betrieben und von unterschiedlichen Anbietern verwaltet werden. Beispielsweise fürchten potentielle Kunden häufig, dass sie die Kontrolle über ihre eigenen Daten verlieren. Mangelnde Transparenz in Dienstspezifikationen und unklare Sicherheitsversprechen der Anbieter erhöhen die Unsicherheit potenzieller Nutzer von Cloud-Computing-(Mehrwert-)Diensten. Noch brisanter wird die Situation, wenn - bspw. über digitale Marktplätze - mehrere alternative Dienste angeboten werden: den potenziellen Kunden fehlt dann eine zuverlässige Vergleichsbasis, um sich zwischen den Alternativen fundiert entscheiden zu können.

Vor dem beschriebenen Hintergrund sind in diesem Beispiel also einerseits zuverlässige Hilfsmittel für Kunden d.h. potenzielle Dienstnehmer erforderlich, um Cloud-Dienste nach den für sie relevanten Kriterien bewerten zu können. Andererseits sollten auch Anbieter die Möglichkeit erhalten, die Qualität Ihrer Dienste hinsichtlich verschiedener Kriterien (möglichst nachprüfbar) beschreiben zu können. Diesen beiden Herausforderungen widmet sich die vorliegende Dissertation und versteht sie als zwei Facetten der Vertrauensbildung; konsequenterweise siedelt sich die Arbeit im Forschungsbereich *Computational Trust* an.

Dabei wird dem genannten Trend Rechnung getragen, dass zunehmend Cloud-Computing-Mehrwertdienste aus mehreren einfacheren Basisdiensten komponiert werden im Sinne einer Wertschöpfungskette. Unter diesem Blickwinkel sind Cloud-Computing-Dienste eine spezielle - und derzeit die populärste - Klasse verteilter Dienste-Netze. Die Dissertation leistet also wissenschaftliche Beiträge im Bereich *Vertrauensbildung für verteilte Dienste-Netze*

und verwendet Cloud-Computing-Dienste als konkretes Anwendungsgebiet, zur Konkretisierung der Forschungsansätze und für Evaluationszwecke.

Die vorliegende Dissertation stellt also neuartige Verfahren zum Vertrauensaufbau vor und fokussiert dabei die Einschätzung und Prüfung der Vertrauenswürdigkeit von zusammengesetzten Diensten und deren Anbietern. Inbesondere die folgenden wissenschaftlichen Beiträge werden in der Dissertation vorgestellt:

- Zunächst wird eine Liste von dienstspezifischen Eigenschaften eingeführt, genannt QoS+ [HRM10, HHRM12]. Diese Eigenschaften basieren auf einer systematischen und umfassenden Analyse existierender Literatur zu den Themenbereichen Cloud Computing, Sicherheit und Vertrauen.

- Darauffolgend wird ein formales Rahmenwerk [SVRH11, RHMV11a, RHMV11b] zur Analyse von zusammengesetzten Diensten definiert, zusammen mit deren erforderlichen dienstspezifischen Eigenschaften bezüglich Kundenanforderungen, welches diese Eigenschaften in leicht verständlichen Formeln ausdrücken kann: propositionallogische Ausdrücke (Propositional Logic Terms, PLTs).

- Anschließend wird ein Ansatz zur Vertrauensbewertung namens *CertainLogic* [RHMV11a, RHMV11b, HRHM12a, HRHM12b] vorgestellt, samt Operatoren zur Auswertung solcher PLTs, also der Fähigkeiten von Dienstanbietern, unter Berücksichtigung unsicherer und konfliktbehafteter Informationen aus verschiedenen Quellen.

- Die vor genannten Beiträge werden zu einer neuartigen Architektur [HRM11] für Vertrauensmanagement in Marktplätzen für Cloud Computing zusammengefügt, um Kunden die Einschätzung und Prüfung der Vertrauenswürdigkeit von Dienstanbietern zu ermöglichen. Diese Einschätzung und Prüfung basiert auf Veröffentlichungen der Dienstanbieter über ihre Dienste.

Die neuartigen Beiträge dieser Dissertation wurden mittels eines Proof-of-Concept-System, prototypischer Implementierungen und formaler Beweise evaluiert. Das Proof-of-Concept-System [HRMV13, HVM13a, HVM13b] setzt die vorgeschlagene Architektur für Vertrauensmanagement in Marktplätzen für Cloud Computing um. Die Umsetzung des Systems basiert auf umfangreichen Selbsteinschätzungen, die von der Organisation CSA (Cloud Security Alliance) als neutraler Instanz systematisch erfasst werden. Hierauf wurden das formale Analyserahmenwerk sowie die Operatoren aus CertainLogic angewandt. Diese Umsetzung ermöglicht Kunden die Prüfung der Vertrauenswürdigkeit von Cloud-Dienstanbietern. Um die technische Umsetzbarkeit der Beiträge dieser Dissertation zu demonstrieren, wurde eine Vielzahl von

Experimenten durchgeführt, welche unterschiedliche Szenarien des Cloud Computing mit verschiedenen Datensätzen berücksichtigten. Zusätzlich bietet die prototypische Implementierung des CertainLogic-Rahmenwerks die Möglichkeit, Charakteristiken der Operatoren an zahlreichen Beispielen aufzuzeigen. Das formale Rahmenwerk sowie die Operatoren von CertainLogic wurden auf geeignete mathematische Eigenschaften hin überprüft, gestützt durch algebraische Beweise.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Trust is a common phenomenon that determines our behaviour and actions in our daily life. In other words, it acts as a facilitator for decision making in environments, e.g., from ancient fish markets to electronic service provisioning to modern social interaction, where decisions are subject to risk and uncertainty. According to the Oxford online dictionary, trust is defined as firm belief in the reliability, truth, or ability of someone or something. In the real world one can use various facial or physical cues, e.g., rely on a document or a referral to known authorities to initiate the process of trust establishment [CDC08]. Today, distributed service environments, e.g., cloud computing, introduce new challenges for establishing trust on service providers, due to the fact that one faces the absence of these physical cues. Moreover, people in general struggle to trust online services than offline services [BKL07]. This thesis, thus, addresses the interesting challenges regarding trust establishment in distributed online service environments.

## 1.1 Motivation

Emerging service computing [HS05] environments have generated new opportunities for individuals, organisations, and government authorities. In such environments, computing power, data storage, and software are modelled as services. Services of these types are widely adopted in distributed service environments, such as in cloud computing [AFG+09, HHRM12], where consumers are able to provision services in a cost-effective, dynamic and highly scalable manner. Moreover, cloud-based services are often composed of multiple service instances, which are hosted in distributed systems across

the globe and managed by multiple parties. Potential consumers often feel that they lose the control over their data due to the lack of transparent service specification and unclear security assurances in cloud computing environments [KM10]. These concerns act as a barrier for consumers to establish *trust* towards service providers in marketplaces, which is reflected in a survey [Fuj10] on 3000 consumers from 6 countries. According to this survey, 84% of the consumers are concerned about their data storage location and 88% of the consumers worry about who has access to their data. As a result, consumers are less reluctant to provision computing services available in cloud computing marketplaces. Hereby, trust mechanisms [UKJS10] can play a major role to establish consumers' confidence on the capability of service providers.

In order to motivate the importance of trust establishment in distributed service environments, we draw a typical service provisioning example in cloud computing. The example we consider here is the one of a healthcare provider who wants to outsource its in-house application that deals with medical records to a cloud-based service. As a potential consumer, the main goal of the healthcare provider is to minimise the IT expenditure as well as allow doctors, patients, and insurance companies to have seamless access to these medical records. In cloud marketplaces, there can be a number of service providers offering cost-effective medical record management services with the required functionality. As medical records consist of sensitive information, the healthcare provider wants to make sure that the service provider has the capability to offer assurances which are beyond the functional properties of a service. For the healthcare provider, assurances on compliance with regulatory acts, data protection, safe geographical location and high availability may be important. The healthcare provider considers a cloud provider "trustworthy", if the provider is able to fulfil the assurances on these attributes. Since the cloud service market for offering medical record services is competitive, the healthcare provider faces the challenge of assessing the *trustworthiness* of service providers that fulfil its requirements. This thesis aims to address the problem of assessing and evaluating the *trustworthiness* of service providers based on consumer requirements, i.e., service-specific attributes of a provider.

## 1.2 Research Challenges and Goals

The main objective of this thesis is to provide novel concepts and mechanisms for trust establishment in distributed service environments. In particular, attention is given for developing mechanisms in order to assess and evaluate the trustworthiness of service providers in such environments. In the course of establishing trust towards service providers, the consumers face the following challenges.

1. Which service-specific attributes are essential for trust establishment in distributed service environments?
2. How to assess those attributes and formally represent them in the context of trust establishment?
3. How to aggregate the attributes and present the aggregated result to the consumers?

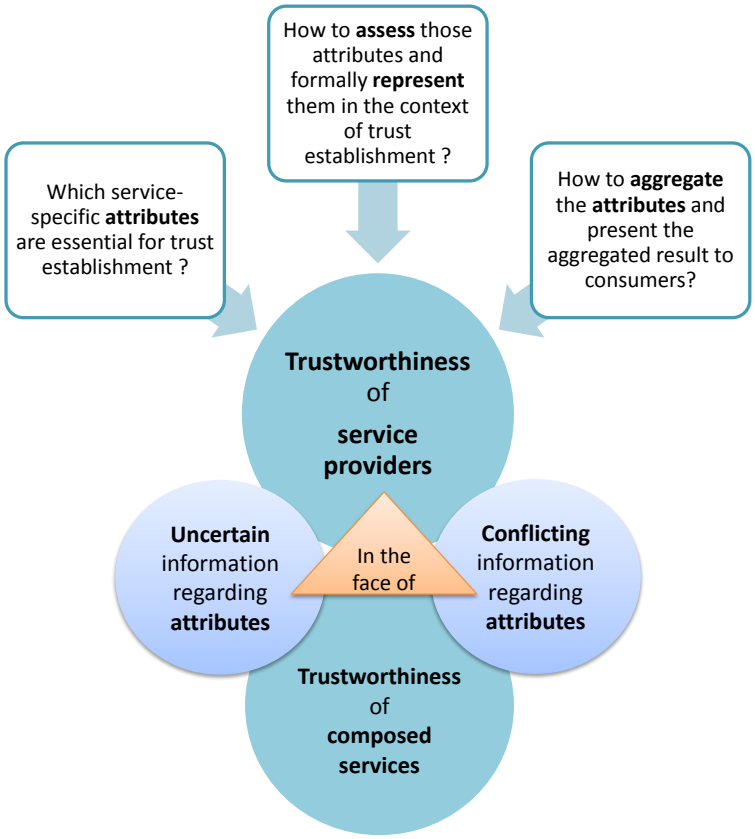



Figure 1.1: Research Challenges and Goals

Addressing the above-mentioned challenges is crucial to select trustworthy service providers in distributed service environments. Particularly, it is non-trivial when the offered services are composed of sub-services and these sub-services are managed by multiple parties. In order to achieve these goals, this thesis aims to provide novel trust establishment mechanisms that evaluate the trustworthiness of service providers considering:

i the trustworthiness of composed services regarding the service-specific attributes (independent from how the attributes are assessed),

ii information regarding different attributes (or requirements) under uncertainty (in the sense of incomplete or unreliable information), and

iii conflicting (in the sense of contradiction) information derived from multiple sources.

An overview of the research challenges and the objectives of this thesis are illustrated in Figure 1.1.

## 1.3 Scientific Contributions

This thesis derives from the introduced research problems following the reference scenario of cloud computing, which is an ideal example of distributed service environment. In particular, we instantiated our research contributions in the challenging domain of cloud computing marketplaces. However, our contributions are not restricted to this particular domain.

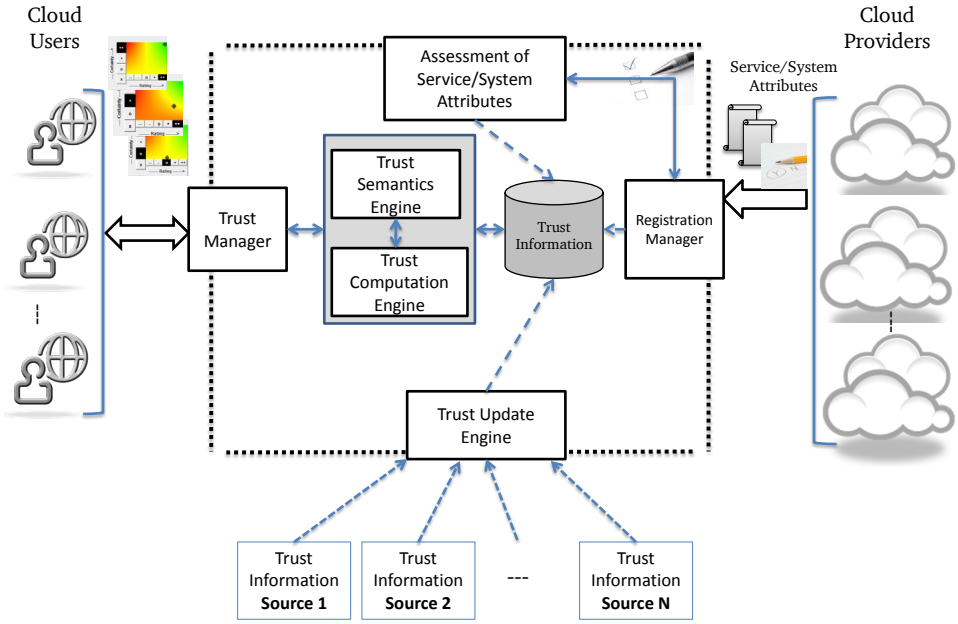



Figure 1.2: Overview of the Trust Management System Architecture

In cloud computing marketplaces, it is essential to assess the *trustworthiness* of service providers based on their published service-specific attributes. The number of service providers in cloud marketplaces is growing rapidly with new providers entering the market. Hence, the providers will increasingly compete for customers by providing services with similar functionality. In traditional service oriented environments, functional attributes serves as a basis for matching services according to user requirements. The functional

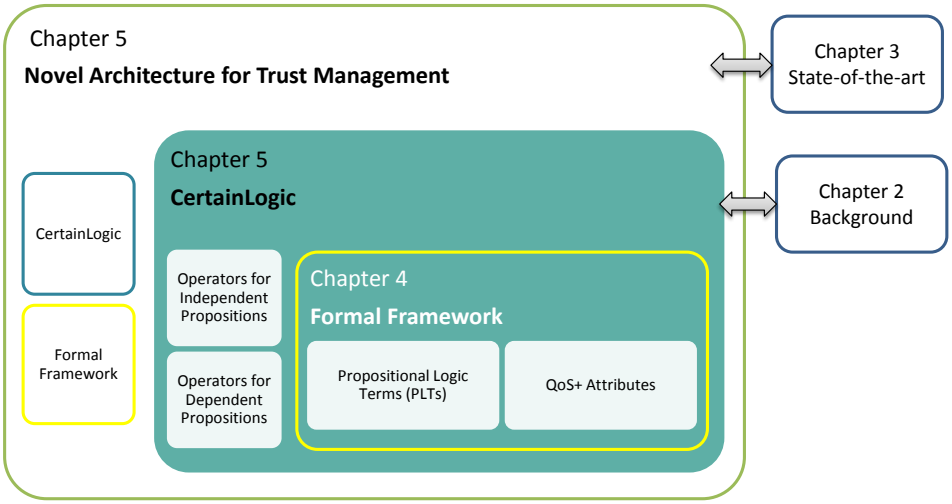



Figure 1.3: Thesis Contributions and Contents

attributes are usually referred to service types, e.g., software as a service or platform as a service. In a real-world setting, a consumer does not only depend on the service types, but also on the non-functional attributes of the service and their underlying instances.

In order to support the consumers for assessing and evaluating the trustworthiness of service providers regarding non-functional attributes, we propose a novel architecture [HRM11, HRMV13] for trust management in cloud computing marketplaces. On an abstract level, we contribute to this architecture novel trust mechanisms considering i) consumer requirements, ii) non-functional attributes regarding the services, iii) *composite* structure of the services as well as the attributes, and iv) multiple sources that provide information about the fulfilment of the attributes published by service providers. An overview of the architecture is provided in Figure 1.2.

On a technical level, we contribute new trust mechanisms that are essential for mechanising the proposed trust management system architecture (cf. Figure 1.2). Within this context, the main contributions are as follows:

- *QoS+* [HRM10, HHRM12], a list of non-functional attributes (or trust attributes) that potentially contribute to the trustworthiness of a service in cloud computing marketplaces. These attributes go beyond the usual Quality of Service (QoS) parameters introduced in the context of trust-aware web service selection by Wang et al. [WV07b]. *QoS+* attributes are identified by a thorough analysis of existing literatures in the field of cloud computing security and trust.

- *Formal framework* [SVRH11, RHMV11a, RHMV11b], an approach to formally analyse the composite services along with their attributes

considering consumer requirements and represent them by means of Propositional Logic Terms (PLTs). The proposed framework is generalised and flexible so that it is able to represent distributed composition of services into *PLT*s. These *PLT*s can be customised by consumers according to their requirements.

- *CertainLogic* [RHMV11a, RHMV11b, HRHM12a, HRHM12b] is a novel framework to evaluate the trustworthiness of composite services in the face of uncertain and conflicting information. The proposed framework relies on the *PLT*s that represent composite services and their attributes in a simplified manner. The framework considers that information regarding each proposition in *PLT*s are derived from multiple sources and are represented by the CertainTrust representational model proposed by Ries [Rie09b]. The CertainLogic framework contains computational trust operators for evaluating *PLT*s and is a novel extension of the established CertainTrust representational model.

An overview of the thesis contributions and the coherent connection among the contents are illustrated in Figure 1.3.

## 1.4 Evaluation

The novel contributions of this thesis are evaluated using proof-of-concept and prototype implementations along with formal proofs. The main goal of the evaluation is to demonstrate the applicability and technical feasibility of our contributions in the domain of distributed service environments.

For that, in the context of this thesis a realisation of the novel architecture (cf. Figure 1.2) for trust management has been implemented. The implemented system considers a self-assessment framework proposed by the Cloud Security Alliance (CSA) as a trust information source. *CSA*'s framework enables cloud providers to publish detailed information about the capabilities of the services they offer. The proof-of-concept system integrates the formal framework to assess the capabilities and represent them in terms of *PLT*s. The computational operators of CertainLogic are used to evaluate the *PLT*s. Each of the propositions in the PLTs are associated with information derived from publicly available registry, i.e., STAR (Security, Trust & Assurance Registry [CSAd], about cloud providers' capabilities. The *STAR* datasets are leveraged to conduct experiments in different cloud computing scenarios. The goal of the conducted experiments is to demonstrate the technical feasibility of trust establishment mechanisms in assessing and evaluating the trustworthiness of service providers. Additionally, the prototype implementations of CertainLogic framework provide means to demonstrate the characteristics of the computational operators. The mathematical validity of the formal

framework as well as computational operators of CertainLogic is supported by formal algebraic proofs.

## 1.5 Publications

The contributions in this thesis are published in a number of peer-reviewed journals, conferences, workshop proceedings and technical reports. The results as well as the proof-of-concept system are also presented and discussed at conferences and demo sessions.

The research challenges in cloud computing regarding trust establishment and the importance of introducing QoS+ attributes in this domain have been addressed in [HRM10, HHRM12]. The mechanisms for representing and evaluating distributed composite services and systems in the form of PLTs and how the evaluation of PLTs relates to the evaluation of the composite services' trustworthiness, are published in [SVRH11, RHMV11a, RHMV11b]. The mechanism for evaluating the trustworthiness of service providers based on the information derived from multiple sources is published in [HRHM12a, HRHM12b]. The novel trust management system architecture comprising the computational trust operators and the formal framework has been published in [HRM11]. The experimental evaluation of the proposed architecture as well as the computational trust operators are published in [HRMV13, HVM13a, HVM13b].

## 1.6 Thesis Outline

The rest of this thesis comprises of six chapters, which are as follows:

**Chapter 2** provides background information related to trust establishment mechanisms discussed in this thesis. The chapter is divided into two parts, where the first part provides the basic understanding of trust from *social*, as well as from *security* point of view. The second part pins up the basic mechanisms of trust and focus on the essential technical apparatus of trust establishment mechanisms.

**Chapter 3** first specifies requirements that are essential to design trust systems for distributed service environments. Then, state-of-the-art trust systems and their mechanisms are discussed with respect to the requirements. Finally, the gaps in the state-of-the-art are identified and the pointers to address those gaps to fulfil the objectives of this thesis are provided.

**Chapter 4** introduces the formal framework that models and formalises trustworthiness in the context of composite services. The framework serves as a building block for *trustworthiness* assessment of service providers in distributed service environments.

**Chapter 5** provides novel mechanisms to evaluate the trustworthiness of service providers in the face of uncertain and conflicting information derived

from multiple sources. The chapter also introduces a novel architecture for trust management in cloud marketplaces, in which the formal framework and the trustworthiness evaluation mechanisms are integrated.

**Chapter 6** provides the implementation details of the realised system for trust management, which is leveraged to conduct the experiments in different scenarios. Moreover, experimental evaluation of the CertainLogic operators is provided under different test cases.

**Chapter 7** concludes the thesis and the chapter ends with directions for future research.

# 2
# Background

This chapter provides the necessary background information related to the contents of this thesis. Section 2.1 provides basic concepts and definitions of trust. Section 2.2 discusses the basis of trust mechanisms available in the field of security and related areas.

## 2.1 Trust Concepts

Trust is a complex notion that has been studied in various fields such as sociology, psychology, and even economics. It is a common phenomenon that determines our behaviour and actions in our everyday social life. Hereby, trust serve us as a basis to interact with unknown participants in uncertain environments, be the environment *physical* or *virtual*. For example, Alice prefers to eat pizza in restaurant 'X' instead of restaurant 'Y', because she trust 'X' for making tasty pizzas but not 'Y' for the same purpose or in a virtual environment, Alice buys a notebook from provider 'XYZ' whom she trust for delivering an authentic notebook in time. Likewise, trust also plays a major role in the field of information security. For example, Alice provisions a cloud service if it is hosted in a trusted platform or in other words, she trusts the service provider if the service is hosted in trusted platform. Thus, we discuss the concept of trust from two perspectives. Though we draw a basic example related to trust concept from *physical* environment, this thesis focuses on trust concepts in *virtual* environments only.

### 2.1.1 Trust as a social concept

Trust is usually reasoned in terms of a relationship within a specific context between a *trustor* and a *trustee*, where *trustor* is the subject that trusts a target entity, which is referred to as *trustee*. In a social environment, trust facilitates a person (i.e., trustor) to delegate tasks and responsibilities to another person (i.e., trustee). In order to delegate tasks to the trustee, a trustor requires evidence about the trustee's behaviour in the past. Evidence about a trustee can be derived from *direct experience* or asking another trustor about their own experience, i.e., *indirect experience*. This is termed as social concept of *trust* and has been widely used in the field of computer science [JIB07, GS00]. More detailed discussion on the definitions in the field of philosophy, sociology, psychology, and economics are provided in [MC96, Gra07].

Although researchers agree on the social concept of trust, it is not easy to get a single definition of trust based on universal consensus. A definition, that is adopted by many researchers in the field of computer science, is the definition provided by the sociologist Diego Gambetta [Gam90, Gam00]:

**Definition 2.1.1** *Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.*

In the context of this thesis, *an agent or a group of agents* refers to service providers, who provide the requested service by contemporaneously meeting service-specific requirement $r$, e.g., confidentiality, availability, etc. In another setting, a service provider may publish $r$ by means of service attributes, e.g., compliance, information security, data governance. The assurance of $r$ corresponds to what is referred to as *perform a particular action*.

Following Gambetta's definition, we consider the subjective notion of trust in this thesis. Mui et al. [MMA+01] also defined trust as a *subjective expectation* that an agent has about another's *future behaviour* based on the history of their encounters. The definition particularly demonstrates the need to learn from past interactions or experiences.

By analysing the definitions in existing literature, Grandison [Gra07] found the following notable characteristics of trust. Summarising the definitions, *trust* is measurable, subjective belief about a particular action, a belief that expresses an expectation about the trustee [Jon99] and a belief that has implications on the features, properties, attributes of a service or system [KC98]. However, these definitions do not consider the needs of electronic service or distributed service environments.

Grandison [Gra07] defined trust by merging the important aspects of the state-of-the-art definitions previously discussed and that reflect the needs of distributed e-service environments.

**Definition 2.1.2** *Trust is the quantified belief by a trustor with respect to competence, honesty, security, and dependability of a trustee within a specified context.*

The central theme of trust is well-integrated in this definition, namely subjectivity, contextual belief, expectation, the implications of trust on system attributes, and measurable trust. A set of specific trust requirements in the context of distributed service environments is discussed in Section 3.1.

### 2.1.2 Trust concept in security

Security technologies, services, and primitives aim to provide safe and tamper-proof computing environments and network. Hereby, trust is placed on service platforms or systems based on the existence of *provable* security primitives. In this sense, trust is synonymous to security. A detailed discussion is given in the following to clarify the understanding of trust in the security field.

**TCSEC and Common Criteria:** Trust has been playing a foundational role in the field of security since last four decades. Trusted Computer System Evaluation Criteria (TCSEC) [oD85], often referred to as the *Orange Book*, is one of the earliest standards that involved the notion of trust in the field of computer system security. In this standard, trust was reasoned through the process of convincing the observers that a system's model, design, and implementation was correct and secure and the system behaved as intended. The TCSEC defines four divisions, $D$,$C$,$B$, and $A$, where each of the divisions represents a significant difference in trust an organisation or an indvidual could place on the evaluated system. Systems in $D$ division had minimal protection, it means that the systems in this division failed to meet the security requirements upon evaluation for the higher divisions whereas systems in division $A$ had highest protection that the security requirements of the systems were formally verified. Thus, the systems in higher division corresponded to more secure system than the system that are in lower divisions. These differentiation also leads to higher or lower level of trust that an organisation or individual could place on a evaluated system. The TCSEC was replaced by the Common Criteria for Information Technology Security Evaluation [Nat12b, Nat12c, Nat12d, Nat12a] standard in 2005.

**Trust Management:** In the mid-1980s and 1990s, with the rise of distributed systems, trust played an implicit role in distributed system security. The notion of trust was reasoned in the form of *trusted* authorities responsible

for the management of security services. A number of trusted authorities such as certification authority, authentication and key management authority, access control authority and trusted third party were introduced to collectively assure trust on the security services. A typical access control service in today's world can be referred to as a security service that managed by several trusted authorities. For instance, Alice wants to access a resource of Bob; Bob will only grant the access right to Alice, if Alice can provide the required credentials. Policies regarding the required credentials are stated by the access control authority. The credentials are issued and signed by the certification authority in the form of certificates. The certificates may state information about the identity of the owner [ITU97] or information about the rights of the owner [BFL96]. In [BFL96], the main idea of the proposed approach is to establish trust using necessary credentials that are defined using policies. This is referred to as *policy-based trust management*.

**Soft security and Hard security:** In the late 1990s, the social concept of trust became popular in the field of information system security. Rasmusson et al. coined the need for trust derived from social control mechanisms to provide security in the context of e-commerce. They refer this type of security as *soft security* that can be derived from intangible information such as past experiences, reputation and coalition. They also coined the term *hard security* that refers to security derived from traditional mechanisms such as passwords and certificates. These are validated using concrete security techniques and can be characterised by certainty. They argued that systems become vulnerable once the hard security mechanisms are bypassed. However, soft security mechanisms provides persistent security as they only accept good behaviours. Few years later, Jøsang analysed trust (i.e., soft security) and security (e.g., hard security) in a similar manner. In his words, security represents the idealistic side, including formal modelling, verification and development, i.e., how the system should be in theory. Trust on the other hand assumes that no hard security mechanisms are perfect and errors exist no matter how rigid the design procedures are. Chapter 3 provides a detailed survey on the trust systems that support soft and hard security.

**Trusted Computing:** The notion of trusted platform was introduced in the early 2000s by Trusted Computing Platform Alliance (*TCPA*), currently known as Trusted Computing Group (*TCG*). This is considered as a significant development towards promoting trustworthy computing in the field of security. A 'trusted platform' is one that contains hardware based subsystem, i.e., Trusted Platform Module (*TPM*), devoted to maintain trust and security between machines. The *TPM* has a mechanism by which it can collect and provide evidence on the state of the hardware, software and firmware that are installed on the platform. It has another special mechanism 'attestation'

which enables a trusted platform to disclose the state of its components to a third party. The default attestation mechanism, proposed by TCG, is based on hash values and the mechanism is defined as binary attestation [TCG11]. However, hash values have the disadvantage of having frequent changes even for a trivial update in the system. Furthermore, hash values are cumbersome to use as policies, as it is difficult to interpret them to be meaningful system states. These shortcomings led researchers to propose the property based attestation [PSHW04, SS04] as an extension of the binary attestation. The property based attestation leverages the binary attestation to abstract the low level hash values to high level meaningful security properties of the platforms. Using this mechanism, it is possible to prove that the availability of a certain hash measurement guarantees the availability of certain security property.

According to the discussion above, we see that trust concepts has played an important role in the landscape of security. Though the relationship between security and trust is debated by Nissenbaum [Nis99] arguing that security primitives can only mitigate risk, a major portion of the security community believes that trust and security are closely related to each other. They argue that changes in the security levels influence the levels of trust. For example, a consumer is willing to provision a service from a service provider if their offered services are hosted in trusted platforms. Similarly, changes in trust levels also influences the security levels [Nag10]. This is particularly relevant to emerging distributed service environments [HHRM12, UKJS10] where entities may initiate interaction with each other without having had prior contacts. For example, consumers are willing to provision services from cloud providers that transparently publish their security-specific capabilities, i.e., a service provider is believed to be less transparent (hence, less trusted) than the provider who is transparent about security-specific capabilities. This thesis views trust and security as a complementary technologies and demonstrates trust as a augmenting concept to security.

## 2.2 Trust Establishment

According to the trust concepts discussed in the previous sections, it is certain that a trustor can establish trust on a trustee using two approaches in the context of distributed service environments. In the existing literature [AG07, Rie09b], the approaches are termed as *policy-based* and *evidence-based* trust management.

### 2.2.1 Policy-based trust mechanisms

The approaches behind the policy-based trust management basically rely on hard security mechanisms, e.g., policy orchestration and credentials. Trusted third parties ensure that the credentials, defined in the policy base, owned by

an entity is genuine based on the information about the identity of the entity. Thus, the entity is considered *trustworthy*. Blaze et al. [BFL96, BFIK99] demonstrate that credentials may include the information about specific rights of the owner. According to Blaze et al. [BFL96, BFIK99],

**Definition 2.2.1** *Trust management is an unified approach to specifying and interpreting security policies, credentials and relationships that allow direct authorization of security-critical actions.*

This trust management approach treats the concept of trust implicitly while the process of trust establishment is external. These issues are further confirmed by Grandison [Gra03] and Cahill et al. [CSG+03]. Moreover, trust based on the credentials issued by the public key infrastructure [ITU97] requires additional means for key distribution, verification and revocation. Further discussion on the research trends of trust management systems are provided in Section 3.4.

### 2.2.2 Evidence-based trust mechanisms

Trust mechanisms, which usually rely on pieces of evidence derived from past interactions or experiences, are referred to as *evidence-based trust mechanisms*. Evidence can be derived from direct interactions between a trustor and a trustee. Direct interactions, however, may be rare in certain cases, e.g., newcomers in service marketplaces. Thus, evidence-based mechanisms also consider evidence derived from indirect interactions, i.e., an entity provides another entity with pieces of evidence about its past interactions. This is usually referred to the *exchange of recommendations*. Apart from the direct and indirect interactions, pieces of evidence are also derived from various virtual cues, e.g., certifications. Deriving evidence based on these cues are shown practical and essential in the context of distributed service environments [Nag10, HVHM12]. Often these pieces of evidence are not only based on explicit interactions with the trustee, but also based on past interactions with certification authorities that issue certificates to a trustee or that are based on security assessments of a trustee.

Unlike policy-based trust mechanisms, evidence-based trust mechanisms establish trust between entities based on their previous interactions or experiences. In the policy-based mechanisms, a trustor directly evaluates a trustee based on its presented credential(s) and access rules but not based on their previous interactions or experiences. Recently, researchers have demonstrated the integration of both mechanisms in the context of distributed authorisation in trusted platforms [KV11].

The existing evidence-based trust mechanisms [BLB04, TPJL06, JI02, WJI05, HJS04, RH08, HWS09] usually leverage Bayesian probabilities [Bol04] to estimate the future behaviour of the trustee based on the available pieces of evidence from the past interactions. These are referred to as *Bayesian*

*trust mechanisms*. These mechanisms leverage the Beta probability density function to estimate the future behaviour. As the approach considers evidence from the past, it is subject to *uncertainty*.

**Uncertainty** For making our understanding of *uncertainty* more explicit, we refer to a simple example with *dice*. For a *Laplace* die, it is not possible to say whether it will show a 6 when thrown the next time, but it is known that the probability for showing a 6 is 1/6. Although, in this case, the outcome of the next throw is uncertain, there is no uncertainty associated to the probability. In contrast, for a *real* die the latter must not be true. When given a *real* die, one could assume that the probability for showing a 6 is 1/6 based on a subject's prior knowledge about dice, however, this statement is still subject to uncertainty, as the die could have been manipulated. In order to reduce the uncertainty, one could throw the die a number of times, e.g., 5 times, 10 times, 100 times, based on the assumption that this leads to more representative estimates about the probability for showing a 6. Formally, this could be modelled using the Beta probability density function. On the other hand, instead of throwing the die one could examine it and say based on one's expert knowledge that it is quite certain that the probability for a 6 is 1/6, or a non-expert could say, "I guess the probability is 1/6, but I am not really certain about this guess". In this thesis, we are focusing on the latter type of uncertainty, where uncertainty is associated to the probabilities under evaluation and relates to the question whether the past pieces of evidence are representative for the future behaviour.

Apart from the Bayesian approach, there are well-known approaches for modelling *uncertainty* outside the trust field. At first, there is the standard probabilistic approach. However, this approach only allows to deal with the uncertainty of the outcome of the next event, but probabilities are assumed to be known.

Fuzzy logic [Zad75] seems to be related, however, it models another type of uncertainty, which could be typed as linguistic uncertainty. For example, if it is *hot* in a room with a degree of 0.8, it does not mean that the probability that it is hot in this room is 80% (assuming that being hot means temp $> 30$ degree celcius); but, it means that one cannot agree on a specific threshold when it is hot (we assume 30 degrees), and thus a degree of 80% states that it is closer to hot than to cold.

**Beta probability distribution** The Beta distribution [Bol04] is a commonly used distribution for a continuous random variable $0 \leq p \leq 1$. The Beta probability density function $f(p \mid \alpha, \beta)$ can be given as:

$$f(p \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \quad \text{where } 0 \leq p \leq 1,\ \alpha > 0,\ \beta > 0\ . \tag{2.1}$$

By defining $\alpha = r+r_0$ and $\beta = s+r_0$, it is possible to relate the probability function directly to the collected evidence (or observed outcomes). Here, $r$ and $s$ represent the number of *positive* and *negative* pieces of evidence, respectively, and $r_0$ and $s_0$ define the prior knowledge ($r_0+s_0 \neq 0$) (cf. [Rie09a, JHF03]). The expectation value is defined as $E = \frac{\alpha}{\alpha+\beta}$. The mathematical foundations of the Bayesian approach are described in [Bol04].

The Bayesian models are extended [Rie09a] to integrate context-dependent parameters, e.g., dispositional trust, as well as support human users with an intuitive representation of trust, *CertainTrust*. The Bayesian model [JI02], based on *Subjective logic*, also includes a quantitative representation of trust. A detailed discussion on the *Subjective Logic* and *CertainTrust* representational models are given in Section 2.2.2.1 and Section 2.2.2.2 respectively. This is important to understand the novel contributions of this thesis. Note that our discussion is limited to binomial representation of both approaches.

#### 2.2.2.1 Subjective Logic (SL) Opinion Model

Jøsang proposed a belief-based trust model [Jøs99, Jøs01] for decision making in electronic transactions (e.g., eCommerce). It is based upon the Dempster-Shafer belief theory [Sha76], a mathematical theory of evidence. Belief theory posits that the sum of the degrees of belief assigned to different alternatives in the decision process is 1. Contrary to this mathematical theory, the sum of degrees of beliefs over all possible outcomes in Jøsang's approach does not necessarily add up to 1. Rather, the remaining probability mass, i.e., the difference between 1 and the sum of the degrees of belief over all outcomes, is interpreted as *uncertainty* [Jøs01]. In this model, an *opinion* is denoted as $\omega_x^A = (b, d, u, a)$, which expresses the relying party A's belief in the truth of statement $x$. Here, $b$, $d$, and $u$ represent *belief*, *disbelief* and *uncertainty* respectively where $b$, $d$, $u \in [0, 1]$ and $b + d + u = 1$. Thus, three parameters are dependent to each other and one parameter is redundant. The parameter $a \in [0, 1]$ represents the relative *atomicity* that is used in absence of evidence for computing an opinion's probability expectation value $E(\omega_x^a) = b + au$, meaning that $a$ determines how uncertainty shall contribute to $E(\omega_x^a)$.

A graphical representation, the opinion triangle, is proposed to map an opinion to a point in an equal-sided triangle. According to [Jøs01, Jøs07], the horizontal line between the *belief* and *disbelief* points or corners in Fig. 2.1 is the *probability axis*. The relative *atomicity* is graphically represented as a point on the probability axis. The line joining the top corner of the triangle and the atomicty ($a_x$) point is *director* in Fig. 2.1. As an example, the

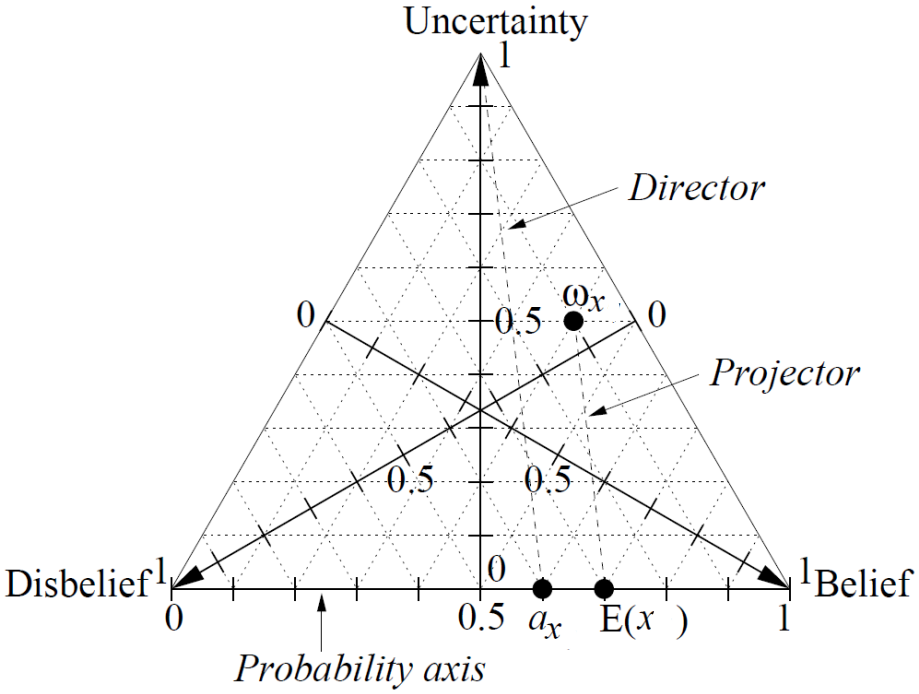



Figure 2.1: Opinion triangle

position of the opinion, $\omega_x = (0.40, 0.10, 0.50, 0.60)$, is indicated as a point in the triangle. The positions of the probability expectation value, $E(x) = 0.70$ as well as the relative atomicity, $a_x = 0.60$, are shown in Fig. 2.1.

**Operators:** *SL* consists of a set of operators that operates on *subjective beliefs* about an entity. The representation of subjective belief is denoted as the term *opinion* (i.e., $\omega_x = (b, d, u, a)$). *SL* operators operates on opinions using standard logical operators as well as non-standard operators. For example, *consensus* and *discounting* operators are for combining opinions from multiple observers and weighting opinions from the recommenders respectively; these are non-standard operators. The standard logical operators are *conjunction*, *disjunction* and *negation*, which are the special case of binary logic operators, *AND*, *OR* and *NOT* respectively. The interpretation and justification of *SL* operators are provided in [Jøs01, Jøs07].

*Subjective Logic* provides mapping [Jøs01] to the Bayesian probabilities, i.e., evidence space. This allows the elements of evidence space to be combined with the *belief* space in order to model trust based on pieces of evidence under uncertainty.

#### 2.2.2.2 CertainTrust (CT) Opinion Model

Ries [Rie09b] proposed *CertainTrust* as a representational trust model. This model is able to represent trust under uncertain probabilities. The truth of a statement can also be expressed by a construct called *opinion*. By design, this opinion construction addresses evidence under uncertainty and user's initial expectation about the truth of a statement. CertainTrust opinion is defined as follows.

**Definition 2.2.2 (Representation CertainTrust)**
*An opinion $o_A$ about the truth of a proposition A is given as $o_A = (t, c, f)$ where the parameters are called* average rating $t \in [0, 1]$, certainty $c \in [0, 1]$, *and* initial expectation value $f \in [0, 1]$. *If it holds $c = 0$ (complete uncertainty), the expectation value (see Def. 2.2.3) depends only on $f$, however, for soundness we define $t = 0.5$ in this case.*

Here, the *average rating* $t$ indicates the degree to which past pieces of evidence support the truth of the proposition. It depends on the relative frequency of observations or pieces of evidence supporting the truth of the proposition. The extreme values can be interpreted as follows:

- average rating = 0: There is only evidence(s) contradicting the proposition.
- average rating = 1: There is only evidence(s) supporting the proposition.

The *certainty* $c$ indicates the degree to which the average rating is assumed to be representative for the future. It depends on the number of past observations (or collected pieces of evidence). The higher the certainty of an opinion is, the higher is the influence of the average rating on the expectation value in relation to the initial expectation. When the maximum level of certainty ($c = 1$) is reached, the average rating is assumed to be representative for the future outcomes. The extreme values can be interpreted as follows:

- certainty = 0: There is no evidence available.
- certainty = 1: The collected evidence(s) is considered to be representative.

The *initial expectation* $f$ expresses the assumption about the truth of a proposition in absence of evidence.

**Definition 2.2.3 (Expectation value of CT)**
*The* expectation value *of an opinion $E(t, c, f) \in [0, 1]$ is defined as $E(t, c, f) = t * c + (1 - c) * f$.*

It expresses the expectation about the truth of the proposition taking into account the initial expectation, the average rating and the certainty. In other words, the expectation value shifts from the initial expectation value ($f$) to the average rating ($t$) with increasing certainty ($c$). The expectation value, $E$, expresses trust of the trustor on trustee and referred to as *trust value* in CertainTrust model.

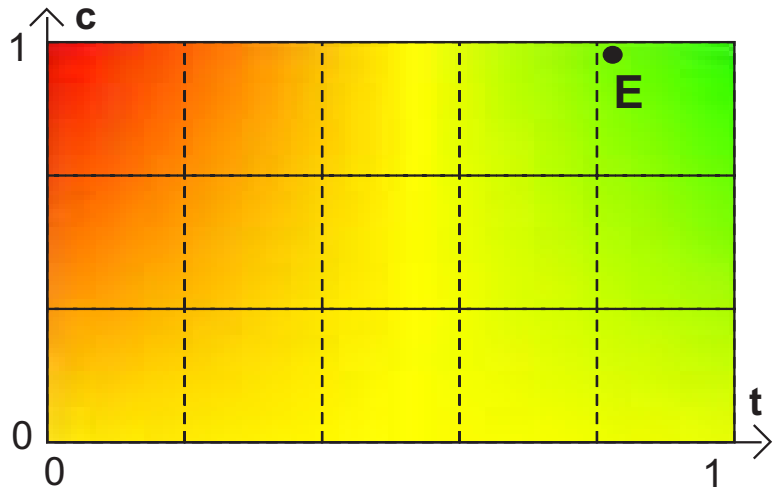



Figure 2.2: Human Trust Interface (HTI)

Ries provides a graphical representation of the CertainTrust opinion to allow easy interpretation and manipulation of trust values by human users. The so called Human Trust Interface (HTI) is based on the parameters introduced above. The two parameters, $t$ and $c$, are the basis for two-dimensional layout of the graphical representation. In Fig. 2.2, $x$-axis represents the *average rating* ($t$) and $y$-axis represents the certainty ($c$). The parameter, *trust value* ($E$), is represented by a red-yellow-green color gradient. The color gradient intuitively represents the semantics of the trust value. The red gradient indicates a low ($E = 0$), yellow a medium ($E = 0.5$) and a green indicates a high trust value ($E = 1$). The values in between are calculated using a linear combination of the colours, red, yellow, and green, in the RGB color model. Fig. 2.2 visualizes the *trust value* of a trustee, B. This is based on A's experience with B in a specific context. As an example, the collected pieces of evidence are represented as opinion, $o_A = (0.83, 0.96, 0.5)$. The trust value, $E = 0.82$, is calculated using Def. 2.2.3 and the value is visualized with a black dot.

**Operators:** *CT* contains a set of basic operators particularly designed for aggregating and weighting evidence units from multiple recommenders. The basic operators are denoted as *consensus* and *discounting* which are equivalent

to the *SL* consensus and discounting operators, as there exists a bijective mapping between *CT* and *SL* representation of opinions. In addition to the basic operators, *CT* contains *extended consensus* and *extended discounting* operators. These are particularly designed for mitigating Sybil attacks in the context of trust propagation [RA09]. The justification and evaluation of the operators are provided in [Rie09b].

The CertainTrust model is designed on the basis of evidence space. Thus, it provides a mapping [Rie09b] to the evidence space. In [Rie09b], it has been also shown that there exists a bijective mapping between the *CertainTrust*'s opinion representation and *Subjective Logic*'s representation of binomial opinion; thus, the representational models are equivalent. However, the models have certain advantages and disadvantages which demand a thorough technical discussion.

#### 2.2.2.3 Discussion on *Subjective Logic* and *CertainTrust* representational models

The advantage of both models are that they provide means for modelling trust in presence of uncertain pieces of evidence. However, by design, the parameters representing opinion in *SL* are dependent to each other, whereas parameters representing *CT* opinion are independent of each other. Let us discuss this issue with a set of intuitive examples:

**SL case:** Assume that an opinion about the truth of a proposition A is given as, $\omega_A = (b, d, u, a)$ where $A =$ This service provider offers good customer support. As parameter $a$ is not relevant for the discussion in this section, we can assume atomicity, $a = 0.5$.

- **Redundant parameter:** According to the given constraint, $b + d + u = 1$, the model has one parameter redundant. It means that the redundant parameter does not increase the expressiveness of the model as $u = 1 - b - d$ or $b = 1 - u - d$ or $d = 1 - b - u$. However, the author argued that the redundant parameter is kept to allow compact expressions of opinion operators.

- **Dependent parameters:** As stated above, the parameters of binomial opinion representation in SL are interrelated by $b + d + u = 1$. This has another consequence that the range of possible values for each parameter depends on the actual values of the other parameters. Let us discuss use cases regarding the consequence.

  - The *three* parameters cannot be chosen independently in the range of $[0, 1]$. For instance, if a user choose $b = 0.4, d = 0.5, u = 0.6$, then the constraint, $b + d + u = 1$, is not fulfilled.

- It is not even possible to choose *two* parameters independently in the range of $[0, 1]$. For example, if a user choose, $b = 0.5, d = 0.6$, it leads to $u = 1 - 0.5 - 0.6 = -0.1$, which is not in the range of $[0, 1]$.
- Finally, it is also not possible to change the value of a *single* parameter without affecting at least the value of other parameter. For example, a user wants to change a particular value ($u$) in the given opinion, $(0.1, 0.1, 0.8)$. In this example, it is not possible to change the value of $u = 0.8$ without changing the value of $b$ or $d$ or both together.

Now, in the context of evidence-based trust, the parameters can be categorized as follows:

- Parameter related to *(un-)certainty*, i.e., parameter $u$ indicates the *uncertainty* associated with an opinion.
- Parameters related to the *collected evidence units*, e.g., positive or negative evidence units.

This categorization is inspired by the given mapping [Jøs01] between SL opinion representation and the evidence space. The mapping is as follows:

$$\begin{aligned} b &= \frac{r}{r+s+2} \\ d &= \frac{s}{r+s+2} \\ u &= \frac{2}{r+s+2} \end{aligned} \tag{2.2}$$

where, $r$ and $s$ are referred to as positive and negative pieces of evidence respectively.

Following the discussion above, we conclude that the following statements are true for SL:

- The *parameters in the category collected evidences are dependent* to each other as $b + d = 1 - u$; thus, the parameters cannot be chosen independently in the range of $[0, 1]$, e.g., $b = 0.5$ and $d = 0.6$ is not possible.
- The *parameters across the categories are dependent.* For example, if uncertainty, $u = 0.5$, this directly influences the values of the parameters in the category *collected evidence units*, i.e., $b$ or $d$ should lie in the range of $[0, 0.5]$.

The influence of dependent parameters can also be reasoned by the following use case, particularly focusing on the *quantity* of collected evidences. Assume that the *trust value* (i.e., probability expectation value) associated with the proposition A is 0.5. This value can be based on two different set of opinions as follows.

- If the opinion is based on few representative evidence units, the values of the opinion are as follows: (b,d,u)=$(0.1, 0.1, 0.8)$.
- If the opinion is based on relatively more representative evidence units, the values of the opinion are as follows: (b,d,u)=$(0.4, 0.4, 0.2)$.

We observe that both opinions *differ only based on the amount of collected evidences*, but *all three parameters have to be changed*.

**CT case:** Assume that an opinion about the truth of a proposition A is given as, $o_A = (t, c, f)$ where $A =$ This service provider offers good customer support. Again, as parameter $f$ is not relevant for the discussion, we can assume $f = 0.5$.

From the given Def. 2.2.2, we see that the parameters in the CT model are *independent* and the $CT$ representational model does not contain any redundant parameter. It means that

- A user can choose any value for the parameter $t$ independently in the range of $[0, 1]$ without any influence by the value of the parameter, $c$.
- A user can choose any value for the parameter, $c$, independently in the range of $[0, 1]$ without any influence by the value of the parameter, $t$.
- A user may want to change a value of a particular parameter (e.g., to reflect high certainty) in the given $CT$ opinion, $(t, c) = (0.6, 0.1)$. In this case, the user only has to adjust the value of the parameter $c$. For example, the adjusted opinion may appear as $(t, c) = (0.6, 0.7)$.

Let us discuss the issue of *independent* parameters focusing on the categories introduced in the *SL* discussion above.

- Parameter related to *(un-)certainty*, i.e., parameter $c$ indicates the *certainty* associated with an opinion; uncertainty would be $1 - c$.
- Parameters related to *collected evidence units*, i.e., parameter $t$ indicates relative frequency of the collected pieces of evidence.

As stated above, there exists a mapping between the CT opinion space and the evidence space. The mapping is as follows:

$$t = \frac{r}{r+s}$$
$$c = \frac{N.(r+s)}{2.(N-(r+s)) + N.(r+s)} \tag{2.3}$$

where, $r$, $s$, and $N$ are referred to as *positive*, *negative* pieces of evidence and *maximum expected evidence units* respectively. The mapping with the definition of $N$ is given in Section 6.1.2.

Following the discussion above, we conclude that the following statements are true for *CT*.

- The parameters can be assigned, adjusted or changed across the categories *independently*.
- The parameters are *independent* within the categories, as each category has only one parameter.

Finally, the influence of independent parameters can be reasoned following the same use case as discussed in the *SL* discussion. Assume that the *trust value* (i.e., expectation value, $E$) associated with the proposition A is 0.5. This value can be based on two different set of opinions as follows.

- If the opinion is based on few representative pieces of evidence, the values of the opinion are as follows: $(t, c) = (0.5, 0.2)$.
- If the opinion is based on more representative pieces of evidence, the values of the opinion are as follows: (t,c)=$(0.5, 0.8)$.

We see that both opinions *differ only based on the amount of collected pieces of evidence*, and *only the parameter for certainty c has to be changed*.

According to the discussion above, we conclude that the *independence* feature of the parameters in *CT* opinion representation and the way the parameters are chosen are an advantage of *CT* representational model. The model is flexible and simpler in contrast to the *SL* model, i.e., parameters of the *CT* opinion representation can be adjusted and interpreted in a more intuitive manner than the parameters in the *SL* opinion representation. However, both representational models are equally *expressive* in representing evidence-based information, as a bijective mapping exists between those two models. Moreover, graphical representation (i.e., HTI) of *CT* model is more user-friendly [RS08], than the one (i.e., opinion triangle) of *SL* model. This is considered as an important requirement (cf. Section 3.1.1) in the context of this thesis. The *extended consensus* and *extended discounting* operators for allowing Sybil-resistant trust propagation (i.e., robust evidence aggregation) are also considered as an requirement (cf. Section 3.1.2) for this thesis.

## 2.3 Summary

This chapter provides a elementary introduction to the trust concepts from two complementary fields. Moreover, a dedicated technical section is provided to give a basic idea of the trust establishment process in virtual environments. The chapter can be summarised as follows:

- In distributed service environments, a consumer's trust on a service provider can be expressed as *subjective belief* where belief has implications on the service attributes, e.g., security attributes, performance attributes.
- Bayesian approach is a good choice, which model trust as a *subjective probability*. The reason is that the approach allows direct integration of collected evidence units in the past, which are subject to uncertainty.
- Extended Bayesian approach such as *CertainTrust* has significant advantages with respect to its representational model. However, the model lacks trust operators, which are important means for trust establishment mechanisms in distributed service environments.
- Finally, assurances regarding hard security mechanisms seems complementary to trust establishment in distributed service environments. Thus, existing approaches considering evidence on security attributes are worth to investigate in the field of trust establishment.

# 3

# State-of-the-Art: Trust Establishment Mechanisms

This chapter presents state-of-the-art trust establishment mechanisms which have been proposed by research communities as well as in industry. The mechanisms that are driven by computational trust models and reputation systems are the main focus of this section. For brevity, we refer to these models and systems as *trust systems*. In both communities, a number of trust systems are proposed for different application scenarios. Beside these trust systems, a number of technologies provide means for establishing trust in distributed service environments. Hereby, we focus on three topics that are closely related to the contributions in this thesis: 1) commercial and research trends of trust systems, 2) applied technologies and 3) research trends of trust management systems.

## 3.1 Requirements

In existing literature [HRM10, HHRM12, HRM11], several requirements have been pointed out to design trust systems for distributed service environments, e.g., cloud computing. The requirements are grouped into functional and non-functional requirements.

### 3.1.1 Functional requirements

The requirements that are essential for a trust system to be functional in distributed service environments are as follows:

FR1 **Multi-faceted Trust Computation:** The computation of trust should consider opinion (cf. Section 2.2.2) about multiple attributes (cf. Section 4.2.2 for detailed description), which refer to competencies and capabilities of a service provider. These competencies and capabilities can be regarding different attributes, e.g., security, compliance, data governance or customer support. Considering multiple attributes in the computation of trust introduce further challenges, which are discussed in the next section. Note that criteria, categories, and aspects are interchangeably used to denote *attributes* where applicable in this thesis.

- Multi-attribute: In order to assess trustworthiness of a service provider, required mechanisms should consider all relevant opinions about different attributes. These opinions usually resemble different qualities of a service which service providers offer. Furthermore, each of the attributes can be composed of several other attributes. Therefore, aggregation of opinions about different attributes, independent of how the opinions are assessed, is a major challenge.
- Multi-source: When considering multiple attributes, the quantitative or qualitative information (opinions) that being factored into the trust establishment process can be derived from different sources. Additionally, one has to consider that these sources might have different characteristics; for instance, information derived from a trusted platform module (TPM) or certificates provided by a property attestation authority need to be handled differently from information derived from user feedback or expert ratings. Therefore, combining information about service-specific attributes derived from different sources is a major challenge.
- Multi-context: A service provider may offer services of different types or in different contexts, which require consumers to consider a different set of attributes to evaluate the trustworthiness of that provider in different contexts. For example, different service delivery models resemble different contexts in cloud computing marketplaces; a service provider might be trustworthy in the context of Software as a Service (SaaS) but not in the context of Platform as a Service (PaaS). Hence, trust systems should be context-aware in order to enable the evaluation of trustworthiness of a service provider regarding service-specific attributes, which are suitable for specific contexts. This leads to another challenge that how to transfer trust across contexts. For example, transferring trust established in the *SaaS* context to the *PaaS* context is not a trivial task.

FR2 **Trust Customization:** It is important to consider the subjective interests and requirements of the customers when assessing the trustworthiness of a service provider. Based on the individual interests and requirements, each customer gets a local (subjective) or customised trust value of a service provider. Subjective trust values provide means for integrating the preference of each customer in detail. Customers may give priority to specific sources of trust information or to a specific attributes based on their interests and requirements or both. Thus, trust systems require mechanisms to deliver customized trust values to the users.

FR3 **Trust Evaluation:** In distributed service environments, a service or system usually consists of several subsystems and components managed by multiple providers. Therefore, trust establishment mechanisms require knowledge about the architecture of the service as well as trustworthiness of its service instances, components, and subsystems in order to evaluate the trustworthiness of the whole system or service. Recently, a categorisation of such mechanisms is introduced in [HHRM12]. The categorisation is as follows:

- Black box: In black box approach, trustworthiness of an entity or a service is evaluated based on the observed output, for example, only based on user feedback. Trust systems and models in this class treat the service as a black box, in other words, these systems do not consider any knowledge about the internal processes and components of the service.
- Inside-out: This approach evaluates trustworthiness of an entity or a service based on the knowledge about the architecture of the service and the trustworthiness of its components (or subsystems). This approach seems to be a perfect fit considering the composite and distributed nature of services in cloud computing environments.
- Outside-in: This particular approach requires knowledge about the internal architecture of a service and its components as input as well as information about observed behaviour of a service. The goal of this approach is to assess and evaluate trustworthiness of internal components of a service composition based on its external behaviour. This is far from trivial, but can be successful when some instances are re-used in multiple services and if certain errors in the behaviour of the service composition can be backtracked to the originating instance.

FR4 **Trust Representation:** The representation of trust needs to be transparent and comprehensible so that the consumers can make trust-based

decision in a convenient and confident manner. In order to make an appropriate decision about which service provider to select in a competitive marketplace, users require an intuitive representation of trust together with additional information regarding relevant attributes.

### 3.1.2 Non-functional requirements

The requirements that are related to the mechanisms for assessing and evaluating trustworthiness of service providers and systems are as follows.

NR1 **Trust Computation under Uncertainty:** In real world applications, the information (opinions) about trustworthiness of the systems and its components are subject to *uncertainty* (cf. Section 2.2.2). For example, trustworthiness values based on expert assessment might be based on insufficient information and the current solutions in the field of trusted computing, that assess the *trustworthy* behaviour of a system, is not able to effectively capture dynamic changes in trust. Thus, mechanisms for evaluating trustworthiness should be able to calculate and express the degree of uncertainty associated to the derived trustworthiness of the overall system.

NR2 **Trust Computation under Conflict:** The information about trustworthiness of the service providers as well as their underlying services and systems can be derived from multiple sources, e.g., experts, users, accreditators. These sources might provide conflicting information which could influence the evaluation of trustworthiness of a service provider. In order to ensure representative trustworthiness value, the computational trust mechanisms should be able to calculate and express the degree of conflict as well as consider it during the evaluation of trustworthiness.

NR3 **Attack Resistance:** As soon as the influence of trust mechanisms on decision-making of customers will grow, the interests in manipulating those values will grow accordingly, as already have been seen in distributed service environments [KC09]. A number of different attacks, e.g., playbook, proliferation attacks, reputation lag attacks, false praise or accusation (collusion), whitewashing (re-entry), sybil attacks, against trust systems have been discussed in [KC09, JG09]. Effective mechanisms for resisting these attacks are essential to develop robust trust systems in distributed service environments.

## 3.2 Commercial and Research trends

In this section, first we present the state-of-the-art *trust systems*, i.e., trust models, reputation systems, trust models in the field of Trusted Comput-

ing. Second, the trust systems are analysed based on the requirements that are outlined in Section 3.1. The objective is to examine the underlying mechanisms of the existing models and systems whether these address the requirements for establishing trust in distributed service environments. Finally, in the discussion section the lacking requirements are identified based on the analysis of existing trust systems.

### 3.2.1 Trust systems

There are a number of commercial trust models, as well as numerous proposals in different research communities, targeting various application areas such as eCommerce, product review sites, Peer to Peer (P2P) networks, Online Social Networks (OSNs), Wireless Sensor Networks (WSNs), ubiquitous, grid and cloud computing. At first, sixteen promising trust systems and models from different application areas are described. Then, these systems and models are analysed with respect to the requirements mentioned in Section 3.1.

#### 3.2.1.1 Commercial systems

Trust systems used in current eCommerce applications and product review sites fall into this category. Most of these systems are centralized and use summation-based models for aggregating feedback given by different users. A significant advantage of these models are that aggregated feedback can easily be transferred to reputation information which is used to establish trust between users. The most widely known commercial reputation systems are those offered by Ebay and Amazon. There is, however, a multitude of similar systems available in online marketplaces, such as Epinions, AllExperts and Bizrate, to name but a few. A brief description about the characteristics of two such systems are given in the following:

**eBay:** [1] eBay is a popular online auction site, allowing sellers to put up items for sale and buyers to bid for those items, with the highest bid winning the auction. After each transaction, sellers and buyers get the opportunity to rate and/or comment on each other through a Feedback Forum. eBay's primary rating system, the *general feedback*, is based upon ratings of three types – positive, negative or neutral. An user's general feedback *score* is determined by the number of positive feedback ratings the user received, while her general feedback *rating*, i.e. reputation, is defined as the quotient of the number of positive ratings by the total number of positive and negative ratings. In order to determine the recent behaviour of a participant, the total number of positive, negative and neutral ratings are also displayed for different periods (i.e., within the past month, the past 6 months and over the preceding 12 months).

[1] http://www.ebay.com/

Beyond general feedback, eBay includes *detailed seller ratings* that provide users with information on eBay sellers, according to four distinct categories:

- *item as described* is defined as the measurement of the accuracy of the item description.
- *communication* is defined as the measurement of degree of satisfaction along with the way and timeliness by which a seller address questions and concerns.
- *shipping time* is defined as the measurement of the time at which bought items were dispatched.
- *shipping and handling charges* is defined as the measurement of the appropriateness of fees incurred by mailing and packaging items, as well as charges levied for time spent on packaging and mailing.

*Detailed seller ratings* category displays average reputation that a seller has for each of these categories, measured on a 5 grade rating scale, for instance, 1 (poor) to 5 (excellent) stars. While the general rating system is personalized, i.e., a user can identify the rating from another user. However, detailed seller ratings are independent of the general rating system, for instance, they do not affect each other.

**Epinions:** [2] It is a review site on which users can comment on and rate their experience with various kinds of objects or services (e.g. places, products, movies, companies), rate reviews given by other users and choose to trust or block (i.e., distrust) specific reviewers. A review generally consists of a quantitative rating ranging from 1 to 5 stars. In addition to the star rating, additional means of rating are as follows: a short summary briefly outlining the benefits (the *Pros*), the shortcomings (the *Cons*) and overall impression (the *Bottom Line*) about a product. The additional means are complemented with qualitative textual description that vary in length between twenty to 199 words for *express reviews* and more than 200 words for *regular reviews*. Depending on the type of item or service reviewed, the quantitative rating can be supplemented by sub-ratings for category-specific products or service aspects, e.g. ease of use, battery life or durability when rating hand held electronic devices or ease of ordering, customer service and on-time delivery when rating online shops.

Furthermore, registered users of the Epinions community can rate reviews by other users. The rating scale for assessing the quality (or *helpfulness*) of reviews is based upon discrete verbal statements incorporating verbal hedges (i.e., Not Helpful, Somewhat Helpful, Helpful, and Very Helpful). The rating for the reviews basically express the helpfulness that those are accorded,

[2] http://www.epinions.com/

how prominently a given review can be placed, and overall the status of the reviewer. A member can obtain the status *Advisor, Top Reviewer or Category Lead* depending on the quality and company of his or her reviews, as well as taking into account the trust and block relationships within the Epinions community.

#### 3.2.1.2 Application-specific trust models

The trust models that are proposed for different application scenarios are described in this section.

***FIRE* model for multi-agent systems:** Huynh et al. [HJS06] tailored *FIRE* particularly to the requirements of agents operating in multi-agent environments. They consider that such a trust model should be distributed and take a wide variety of information sources into consideration. Thus, agents should be able to evaluate trust accorded to other agents subjectively and the trust model, furthermore, should be robust to lying. The resulting system includes four distinct types of modules contributing in the computation of a trust score, although FIRE is extensible to include further modules. The four modules presented in [HJS06] are as follows:

1. **interaction trust module** derives a score based upon prior *direct interaction* experience between the evaluating agent (trustor) and the agent under evaluation (trustee).

2. **role-based trust module** derives a score based upon the, for instance institutional, role of the trustee within the context as determined by the interaction context.

3. **interaction trust module** derives a score based upon *recommendations* regarding the trustee received by the trustor from neighbouring agents; takes into account the recommenders' prior performance in making correct recommendations.

4. **interaction trust module** derives a score based upon *certificates*, i.e., positive references from other agents involved in prior interactions, presented by the trustee to the trustor.

Each individual module supplies a module-specific value that is associated with a corresponding reliability value. In order to derive the final trust value for a trustee agent, the trustor agent aggregates each of the individually determined scores. The aggregation is in the form of a weighted sum, weighing the separate scores by a pre-determined module weight and their associated reliability value. This value is composited from the module-specific reliability scores. The FIRE model is flexible – by allowing the user to extend it

with further modules – and combines and adapts approaches of other trust models for agent systems, such as REGRET [SS02a, Sab03] or the model by Ramchurn et al [RHJ04].

***socialReGreT* model for multi-agent systems:** Sabater et al. proposed a reputation system, socialReGreT [SS02b], that is based on three dimensions of reputation: i) individual dimension, ii) social dimension, and iii) ontological dimension. They argued that the social relations in the social networks can be used to analyse trust and reputation in multi-agent system environments. Therefore, instead of using direct interactions and the information provided by other agents in the community about their past experiences, social relations should be taken into account when analysing reputation of an agent. Direct interactions and the information that comes from other members as well as the social relations are classified as *individual dimension* and *social dimension* respectively. Moreover, the *socialREGRET* system considers multi-facet concept of reputation, e.g., reputation of being a good travel agent summarizes the reputation of having a good air-carrier, a good hotel and good food during the travel. These different types of reputation and their combination into a single type is classified as *ontological dimension*.

***EigenTrust* model for P2P environments:** Kamvar et al. proposed the EigenTrust [KSGM03] model that manages reputation in file-sharing P2P networks. The major aim of the approach lies in eliminating users spreading inauthentic files from the network. Therefore, *EigenTrust* computes and assigns a unique global trust value to each peer, based on the peer's previous upload behaviour. The basic unit of evidence in the computation of *EigenTrust* is a binary rating assigned to the sharer by its peers, signifying their satisfaction with the sharer. Behavioural information is, conceptually, stored in a matrix of aggregated ratings for each pair of peers. Trust estimation in large networks and for remote peers achieved by consulting the vector corresponding to the peer under evaluation. This computation is aided by a statistical convergence of the queried vector to the first eigenvector of the queried matrix – hence the name *EigenTrust*. The secure distributed aggregation for computing these global trust values is based on Power iteration method. Further design goals of *EigenTrust* include decentralization, anonymity, low computational overhead, collusion resistance, and providing no incentive for whitewashing by leaving and re-entering the system under a new pseudonym.

***BNTM* model for P2P environments:** Wang et al. [WV03] proposed a Bayesian Network-based Trust Model (BNTM) that models and aggregates trust from multiple criteria. The model is demonstrated in a P2P-based file sharing scenario and Bayesian network is leveraged to model the trust

between an agent and a file provider. The authors of the proposed model argued about two kinds of trust that a user builds on a service provider. One is the trust that a user build on provider's competence in providing a service. Another is the trust that a user build in another agent's reliability in recommending a service provider. Therefore, direct interaction and other agents' recommendation are taken into account for calculating the aggregated trust or updating the corresponding Bayesian network. In order to address the reliability of the recommenders, the model considered the following two aspects: i) whether the information from a recommender is truthful or not and ii) whether a user agent and a recommender agent have similar behaviour regarding different criteria; if criteria are similar, then two agents can trust each other otherwise not.

***Bucheggers's* model for P2P and Mobile Ad-hoc networks:** Buchegger et al. proposed a robust reputation system in order to cope with false disseminated information for P2P and mobile ad-hoc networks. The reputation system is based on a distributed modified Bayesian trust model. The authors of the proposed model extended the standard Bayesian model by integrating a discounting factor that serves as the *fading* (i.e., ageing) mechanism for past experiences. The main objective of the proposed reputation system is to cope with false ratings. In this case, information provided by recommender nodes are considered as long as they are similar to the direct experience of a node's itself. This is based on a *deviation test* that computes the absolute difference between the expectation value calculated based on direct experience and the expectation value calculated based on recommendation. Moreover, a static weighting factor is introduced to reduce the influence of recommendations in the proposed system.

***Billhardt's* model for Service-oriented environments:** Billhardt et al. proposed a integrated system that combine trust and reputation mechanisms with service discovery or match-making mechanisms in order to select the best provider in a service-oriented environment. They consider a user's confidence value (refer to as *direct interaction*) and the reliability of the confidence value to calculate the trust score of a service provider. The reliability calculation of a confidence value is based on the mechanism proposed in *FIRE* [HJS06], which considers the number of interactions a confidence value is based on, and the variability of the individual values over past interactions. In case, the reliability value falls below a given threshold, the confidence values provided by recommenders are taken into account to calculate the trust value. The authors of the proposed model also introduced a mechanism that demonstrates how to transfer trust across similar contexts (i.e., services) offered by a service provider. The idea is based on the assumptions that services from the same provider will have a similar quality and more similar

its service types are, the more similar the quality of these services will be.

***Hang's* model for Service-oriented environments:** Hang et al. [HS11] proposed a trust-aware service selection model in the context of service compositions. The proposed model considers *qualities* (e.g., latency, throughput, failure) of a service as well as their constituent services to enable trustworthy service selection in web service environments. They use Bayesian networks to model service compositions and the dependency of providing a good service quality between the composite and the constituent services. Beta-mixture approach is considered to learn about the quality distribution of the services and provides the information of the constituent services, i.e., how much each constituent service contributes to the quality of the composite service. The proposed model also includes a mechanism that can deal with incomplete observable data. The main objective of the model is to select service instances to form suitable compositions based on the desired qualities.

***TidalTrust* model for OSNs:** Golbeck [Gol05] proposes a trust model called TidalTrust, enabling a participant (the trustor) to infer trust about another (the trustee) by specifically considering the intermediate connections between the two in a web-based online social network (OSN). The TidalTrust model is more complex of two models proposed in [Gol05] as it is capable of dealing with continuous rather than merely binary trust ratings. Moreover, the model presents a breadth first search algorithm for traversing a graph formed by the connections among neighbors in a social network. The goal of the presented algorithm is to infer a trust value by traversing those edges, which form both the shortest path between the two participants and represent the strongest (intermediary) connections between the two. The strength of the connections is based upon the individual trust scores assigned by intermediary participants to the edges connecting to the next participant on the path from trustor to trustee. The final inference mechanism for computing the trust score is given by a weighted sum over the edges of the selected paths. Aside from presenting the trust inference mechanism, Golbeck validates her model in two different applications, i.e., FilmTrust, Trustmail, with regard to robustness and applicability in real-world scenarios.

***RFSN* model for WSNs:** Ganeriwal et al. [GBS08] proposed a trust model, RFSN, that is particularly tailored to application in sensor networks. RFSN is a distributed, symmetric reputation-based model that uses both first-hand and second-hand information for updating reputation values. The process for updating of reputation values in RFSN is based upon a Beta probability distribution, allowing for higher flexibility by enabling the system to process continuous rather than merely discrete values. It incorporates both ageing and updating mechanisms, as well as incorporating weighing of

witness opinions. For a more comprehensive overview of the numerous trust and reputation models targeting adhoc and sensor networks, we refer the readers to [STL+09].

***GridEigenTrust* model for Grid Environments:** Laszewiski et al. adapt EigenTrust [KSGM03] framework into a trust model [GvL05] for classical grids. They integrate the model into a QoS management framework. By using the framework, grid-resources are probabilistically pre-selected based on their likelihood of possessing the requested capabilities and capacities. The approach of Laszewiski et al. is one of the few approaches that aim to improve *QoS* management in grid environments by integrating a reputation service.

***Abawajy's* model for Cloud Environments:** Abawajy [Aba09] proposed a reputation manager in order to determine the trustworthiness of a cloud vendor for the purpose of service sharing among the vendors. Reputation rating is based on direct experience or observation and indirectly by sharing experiences with other vendors. Service-specific attributes that are considered important to select service providers in distributed service environments, e.g, cloud computing, are not taken into account in this model.

#### 3.2.1.3 Non-application specific Trust models

**Beta reputation system** Jøsang et al. [JI02] proposed a centralized reputation system for general e-commerce environments. The reputation system is based on *Subjective Logic* that combine the elements of Bayesian probability theory with belief theory. The *Subjective Logic* is explained in Section 2.2.2.1. The authors of the proposed system argued that in contrast to other similar systems the beta reputation system has a firm basis in the theory of statistics. The operators [Jøs01] of *subjective logic* are used to aggregate ratings per interaction partner from different sources. The system allows different weights to the feedback based on the age, i.e., old feedback is given less weight than recent feedback. Furthermore, another type of weighting mechanism is introduced to discount reputation rating based on the trustworthiness of the entity who provides the rating.

**CertainTrust model:** Ries [Rie09a, Rie09b] suggested a trust model called *CertainTrust* for selecting trustworthy interaction partners in opportunistic networks. The author argued that the model can also be used for the same purpose in eCommerce and Web 2.0, e.g., recommendations in online platforms, scenarios. The model is an extension of Bayesian trust models [JI02, BLB04] integrating context-dependent parameters, such as *dispositional trust* and *aging* of evidence. One such parameter, the *maximum number of evidence units*, allows the user to define the number representative

of pieces of evidence about an entity's behaviour in an application context. *CertainTrust* explicitly allows modelling of a trust score and the certainty attributed to that score, based upon an expectation, for instance derived from the evaluator's disposition to trust and the behaviour of the entity under evaluation. Furthermore, transformation functions are presented in order to transfer average rating ($t$) and certainty ($c$) into the opinion space of Jøsang's Subjective Logic [Jøs01]. A detailed discussion on the *CertainTrust* model is provided in Section 2.2.2.2.

#### 3.2.1.4 Trust models in Trusted Computing

Apart from the field of trust and reputation models, there are a number of approaches from the field of trusted computing designed to ensure trustworthy cloud infrastructure. Krautheim et al. developed a private virtual infrastructure (PVI), which is a security architecture for cloud computing and uses a trust model to share the responsibility of security between the service provider and client [Kra09]. Schiffman et al. constructed a hardware-based attestation mechanism to provide assurance of data processing protection in the cloud for customers [SMV+10]. There are further approaches such as property-based TPM virtualization [SSW08], which can be used in the cloud scenario to assure users about the fulfilment of security properties in cloud platforms using attestation concepts. However, in general, attestation concepts based on trusted computing, e.g., [SS04], focus on the evaluation of single platforms not on compositions. Moreover, Nagarajan et al. [NV11] argue that given the nature of property-based attestation mechanisms, an attestation requester (e.g., service consumer) cannot be absolutely certain that an attesting platform will behave as it is expected to behave.

Thus, Nagarajan et al. [NV11] proposed a *hybrid* trust model, *TESM*, based on soft trust model to address uncertainties arise from attestation mechanisms. They combine *hard* trust from certificate-based property attestation mechanism with *soft* trust from past experiences and recommendations regarding the properties in the proposed *hybrid* model. The hard trust module includes a logical language, ALOPA, to formalize authorisation derivation in trusted platforms. The soft trust module leverages Subjective Logic for modelling and assessing the uncertainties arise due to the nature of property attestation mechanism. The authors applied the hybrid trust model for authorisation evaluation in distributed service environments, e.g., web service platforms [Nag10]. In particular, they consider the platforms as *composition* of multiple platform instances and applied the soft trust module to evaluate trust in presence of composite platform. The *TESM* for authorisation evaluation is demonstrated to be more effective in comparison to the existing approaches when authorisation needs to be evaluated in presence of multiple security properties of composite distributed platforms and in presence of uncertain property assessment method.

### 3.2.2 Analysis of Trust Systems

First, we analyse the trust systems based on the functional requirements, i.e., FR1 (Multi-attribute), FR2 (Trust Customisation), FR3 (Trust Evaluation), and FR4 (Trust Representation). Then we analyse the trust systems based on the non-functional requirements, i.e., NR1 (Trust Computation under Uncertainty), NR2 (Trust Computation under Conflict), NR3 (Attack Resistance).

#### 3.2.2.1 Analysis based on Functional Requirements

Trust computation under multiple attributes (i.e., *Multi-attribute*) is not a mechanism that can be found usually in the existing trust systems. Commercial trust systems such as *eBay* and *Epinions* consider *multiple attributes* for computing trust ratings. However, eBay's seller ratings, displayed in four distinct categories, do not affect the overall rating process, i.e., categorical ratings are not taken into account to compute the overall rating. Only three models – *TESM*, *BNTM*, and *socialReGreT* – proposed by the research community, consider multiple criteria in computing trust. *TESM* model leverages soft trust operators for computing platform trust by aggregating trust information about different properties of platforms and their underlying components. *BNTM* model uses Bayesian network for representing trust values in different attributes of a service provider. Bayes rule is used to compute the trust value for each of the attribute in the model. *socialReGreT* system also enables multi-attribute concept for trust computation and the mechanism that drives the concept is defined as "ontological dimension".

None but one of the trust systems consider different trust information sources when computing trust based on multiple attributes. *Beta reputation* system leverages Subjective Logic operators to aggregate feedback from *multiple sources*. Unfortunately, this system does not differentiate between the sources, e.g., trust information derived from experts is different from the trust information derived from user feedback. Thus, the mechanism of the proposed system does not fulfil the requirement regarding *multi-source* trust computation.

Commercial models such as Epinions aggregate trust ratings from *multiple contexts* to provide an overall reputation score for an entity. Interestingly, commercial models like eBay and most of the trust systems proposed by the research community do not support the feature. A couple of trust systems, GridEigenTrust and BNTM, from the research community provide mechanisms to combine trust values from multiple contexts in order to provide an overall trust score. However, none of these systems are able to transfer trust across contexts. Conversely, Billhardt's model does not support the multi-context feature, but it is capable of transferring trust across contexts. Therefore, significant improvement is required for trust systems in

distributed service environments to support both of those features.

*Trust customization* is one of the requirements that is fulfilled in a number of commercial applications and research community's proposals. Most of the commercial trust models, e.g., *eBay*, *Epinions*, support a single trust rating for all customers. Most of the trust systems, proposed by the research community, support *local (subjective)* trust score that reflect customers' preferences except *Eigentrust*, *GridEigentTrust*, and *Abawajy's model*.

Types of *Trust evaluation* mechanism is one of the important requirement when trust is evaluated in distributed service environments. Evaluation mechanisms in most of the trust systems can be classified as *Black box* approach because they do not consider the knowledge of internal architecture or behaviour of internal processes in the trust evaluation. The only exception is the *Hang's model* that considers observed behaviour of composite service to assess and evaluate trustworthiness of constituent components. This is classified as *Outside-in* approach. This approach is useful to identify constituent service who is responsible for unsatisfactory quality of the composite service. In distributed service environments, particularly in cloud computing environments, the consumer would be interested to know the trustworthiness of composite service as well as their constituent services without even interacting with that composite service upfront. This is classified as *Inside-out* approach. Evaluation mechanisms of the *GridEigenTrust* and the *TESM* do follow the *Inside-out* approach, but the mechanisms does not provide a formal approach which make the mechanisms specific to grid environments and trusted platforms respectively.

Commercial models like *eBay* and *Epinions* provide a graphical interface (e.g., star rating) together with detailed information (e.g., detailed seller ratings, detailed opinions) to the customers. On the one hand, the graphical interface in commercial models does not provide comprehensive trust information but with the help of detailed information the models mitigate that problem. On the other hand, most of the trust models from the research community do not provide a graphical interface for trust representation except the *CertainTrust* model and the *Beta reputation* system.

Table 3.1: Characterization of state-of-the-art trust models and reputation systems based on Functional Requirements

| Functional Requirements | Trust Computation | | | Trust Customization | Trust Evaluation (Bb vs Io vs Oi)[1] | Trust Representation (UI/C)[2] |
|---|---|---|---|---|---|---|
| Trust systems | Multi-criteria | Multi-source | Multi-context | | | |
| eBay | N | N | N | N | Bb | UI/C |
| Epinions | Y | N | N | N | Bb | UI/C |
| Beta Reputation | N | N | N | N | Bb | UI/- |
| CertainTrust | N | N | N | Y | Bb | UI/C |
| FIRE | N | N | N | Y | Bb | -/- |
| EigenTrust | N | N | N | N | Bb | -/- |
| socialREGRET | Y | N | N | Y | Bb | -/- |
| TidalTrust | N | N | N | Y | Bb | -/- |
| RFSN | N | N | N | Y | Bb | -/- |
| GridEigenTrust | N | N | N | N | Io | -/- |
| Abawajy's model | N | N | N | N | Bb | -/- |
| TESM | Y | N | N | Y | Io | -/- |
| BNTM | Y | N | N | Y | Bb | -/- |
| Buchegger's model | N | N | N | Y | Bb | -/- |
| Billhardt's model | N | N | N | Y | Bb | -/- |
| Hang's model | N | N | N | Y | Oi | -/- |

[1](Bb=Black box; Io=Inside-out; Oi=Outside-in)

[2](UI=User Interface; C=Comprehensiveness)

#### 3.2.2.2 Analysis based on Non-functional Requirements

The trust systems that are discussed in Section 3.2.1 usually don't consider *uncertainty* in trust computation. Only three existing systems and models– *Beta reputation*, *CertainTrust*, and *TESM*– consider uncertainty while computing trust. The underlying mechanisms of *Beta reputation* system and *TESM* model use the *Subjective Logic* operators for computing trust under uncertainty. *CertainTrust* model is also able to model trust under uncertainty and it provide a intuitive graphical interface (i.e., HTI) to visualize trust under uncertainty. However, this model is lacking mechanisms that can compute trust of a service or a service provider in composition of different trust attributes.

None of the state-of-the-art trust systems provide mechanisms for computing trust under conflicting trust information. This requirement is essential when computing trust based on the information derived from multiple sources. Each of these sources might use different information reasoning mechanisms, which may produce different trustworthiness value. Thus, an entity, who wants to compute trust based on the information provided by these sources, have to deal with the deviation (i.e., degree of conflict) of information and reflect the deviation in trustworthiness value.

Most of the trust models are subject to different kinds of attacks, while a few of them are resistant to particular attacks such as False Praise or Accusation (FPA), Sybil (S) and Whitewashing (W) attacks. Considering these we limit our scope to those three attacks in order to make the comparisons concise in Table 3.2. *CertainTrust* model includes mechanisms to deal with sybil and FPA attacks, while *Buchegger's model* and *socialREGRET* are resistant to FPA attacks only. *EigenTrust* model does not provide a mechanism to deal with *Sybil* attacks. However it suggests that imposing a cost to create new IDs, e.g., integrating captcha [Cap] in their approach will make costly for an adversary to create Sybil entities. None of these models address mitigation mechanisms against *whitewashing* attacks.

Table 3.2: Characterization of state-of-the-art trust models and reputation systems based on Non-functional Requirements

| Non-functional Requirements<br>Trust systems | Trust Computation under Uncertainty | Trust Computation under Conflict | Attack Resistance (FPA/S/W)[1] |
|---|---|---|---|
| eBay | N | N | -/-/- |
| Epinions | N | N | -/-/- |
| Beta Reputation | Y | N | -/-/- |
| CertainTrust | Y | N | FPA/S/- |
| FIRE | N | N | -/-/- |
| EigenTrust | N | N | -/S/- |
| socialREGRET | N | N | FPA/-/- |
| TidalTrust | N | N | -/-/- |
| RFSN | N | N | -/-/- |
| GridEigenTrust | N | N | -/-/- |
| Abawajy's model | N | N | -/-/- |
| TESM | Y | N | -/-/- |
| BNTM | N | N | -/-/- |
| Buchegger's model | N | N | FPA/-/- |
| Billhardt's model | N | N | -/-/- |
| Hang's model | N | N | -/-/- |

[1](FPA=False Praise Accusation; S=Sybil attack; W=Whitewashing attack)

### 3.2.3 Discussion

According to the analysis of different *trust systems*, it is evident that none of the systems provide mechanisms that fulfil all the requirements for establishing trust in distributed service environments. However, mechanisms such as Subjective Logic and CertainTrust opinion model considered by trust systems – *TESM*, *Beta reputation*, *CertainTrust* – fulfil some of the important requirements for designing trust systems in distributed service environments. The mathematical foundation of these mechanisms rely on the *Bayesian* approach (cf. Section 2), which allows

- to model trust based on collected pieces of *evidence* in the past and subjective prior knowledge.
- trust to be modelled as *subjective probability.*

Additionally, mechanisms underlying those models offer following features required for trust establishment in distributed service environments.

- *Subjective logic* offers computational trust operators (cf. Section 2.2.2.1) that are able to deal with uncertainty. This model provides a trust representation by means of a construct namely *opinion* and its corresponding graphical interface, i.e., opinion triangle.
- *CertainTrust* model also offers computational operators (cf. Section 2.2.2.2) that are able to deal with uncertainty. This model provides a intuitive graphical trust representation, i.e., HTI, designed particularly for human users.

A detail discussion on *Subjective logic* and *CertainTrust* model is presented in Section 2.2.2.3. According to that discussion, we concluded that *CertainTrust* is a relatively a better choice in terms of its flexible and simple representational model.

In this thesis, *CertainTrust* model is extended according to the functional (cf. Section 3.1.1) and non-functional (cf. Section 3.1.2) requirements. The extension of CertainTrust model is required by means of computational trust operators that should be able to compute;

- trust based on information about multiple attributes and the information is subject to *uncertainty.*
- trust based on information derived from multiple sources and the information is subject to *uncertainty* as well as *conflicting.*

In order to fulfil the requirements– *trust customization*, *trust evaluation* – further mechanisms are required to

- customize trust value according to consumers' preference and interests.
- evaluate the trustworthiness of a composite service or system based on the trustworthiness of constituent components and subsystems.

## 3.3 Applied technologies

In this section, we present the technologies that support consumers to build trust on service providers in distributed service environments. The focus is given on the technologies that service providers in cloud computing marketplaces leverage to build trust.

**Service Level Agreements (SLAs):** In practice, one way to establish trust on service providers is the fulfilment of *SLAs*. *SLA* validation [HBS10] and monitoring [3Te09] schemes are used to quantify what exactly a cloud provider is offering and which assurances are actually met. In service oriented environments, customers are usually responsible for monitoring *SLA* violations (e.g., service downtime) and inform the providers for compensation. The compensation clauses in *SLAs* are written by the cloud providers in such a way so that the customers merely gets the opportunity to apply for compensation (e.g., service credits) due to *SLA* violation. This problem arises from not having standardised *SLAs* for service providers in the marketplaces, particularly in cloud marketplaces. Although, the problem is addressed by an industry driven initiative [Clo10] for establishing standardized *SLAs*, this initiative is far from completion and implementation in practice.

Researchers from the academia and industry – Irfan et al. [HAP+10], Wang et al. [WZWQ10], and Pawar et al. [PRNZ12] – has already demonstrated the practical use of *SLA* compliance for establishing trust on service providers in grid computing, web service, and cloud computing environments. Irfan et al. proposed a trust model based on certificates (i.e., PKI-based) and reputation-based trust system as a part of an SLA validation framework. Wang et al. proposed a trust model that takes multiple aspects (reputation, trustworthiness, and risk) into account for evaluating web services. Both approaches consider *SLA* validation as the main factor for establishing trust on the grid service and web service providers. The *SLA* compliance issue has also recently been considered in a trust model proposed by Pawar et al. [PRNZ12]. The trust model was developed in the context of a cloud-specific project [FHT+12] for evaluating trustworthiness of cloud infrastructure providers.

**Auditing:** Cloud providers use different audit standards (e.g., SAS 70 II, FISMA, ISO 27001) to assure users about their offered services and platforms. For example, Google lists SAS 70 II and FISMA certification to

assure users about the security and privacy measures taken for Google Apps. The audit SAS 70 II covers only the operational performance, e.g., policies and procedures inside datacenters, and relies on a highly specific set of goals and standards. They, however, are not sufficient in alleviating the users' security concerns [Sea09] and most cloud providers are not willing to share their audit reports, which also leads to a lack of transparency. Even though the audit-driven certificates have shortcomings, they serve as persistent trust anchors for transient online services, e.g., service offerings by cloud providers.

Recently, researchers have demonstrated the integration of certification processes into reputation-based trust models [HVHM12]. They argued that the integrated trust model will be able to mitigate the market entry problem for comparatively new service providers. Moreover, the model enables the certified service providers to serve as persistent trust anchors for more transient online services, such as service offerings by cloud providers. The authors demonstrated the integration of audit-driven certification processes into established *CertainTrust* model.

**Ratings & Measurements:** Recently, a cloud marketplace[3] has been launched to support consumers in identifying dependable cloud providers. Cloud providers are rated based on a questionnaire that needs to be filled in by current cloud consumers. In the future, CloudCommons aims to combine consumer feedback with technical measurements for assessing and comparing the trustworthiness of cloud providers. Furthermore, there is a new commercial cloud marketplace named SpotCloud[4] that provides a platform where cloud consumers can choose among potential providers based on *cost*, *quality*, and *location*. In this platform, cloud providers' ratings are displayed in an Amazon-like "star" interface with no documentation on how the ratings are computed.

**Self-assessment Questionnaires:** According to recent studies [Fuj10], lack of security control transparency is the leading inhibitor to the adoption of cloud services. In order to enable transparency, the Cloud Security Alliance (CSA) proposed a self-assessment framework [CSAc] that enables cloud providers to publish their security-specific capabilities of the services they offer. This framework includes a questionnaire, i.e., *CAIQ*, that provides means for cloud providers to document their capabilities in terms of different attributes, e.g., compliance, information security, governance.

The published capabilities serve as an indicator of trustworthiness for cloud providers. In order to enable the process, a metric is required to assess the published CAIQs and leverage the assessment result for evaluating trustworthiness of cloud providers. The CSA self-assessment framework as

[3] http://beta-www.cloudcommons.com/web/cc/about-smi
[4] http://www.spotcloud.com/

it stands does not provide such a metric. Recently, researchers have coined the need for such a metric in order to design a practical trust evaluation system [HHRM12] for cloud marketplaces.

## 3.4 Research trends: Trust Management (TM) Systems

According to [JKD05], *TM* systems should allow relying entities to reliably represent their capabilities and competencies of the underlying systems in terms of relevant attributes. Such systems should also allow reliant parties to make assessments and decisions regarding the dependability of potential transactions based on the available evidences. For the latter part, Bayesian trust systems provide means for assessing the trustworthiness of relying entities based on observations and evidence. Thus, TM system can be considered as a driving element in trust establishment process. Note that *trustee* is termed as relying party and *trustor* is termed as reliant party.

The TM systems developed in the last century, e.g., KeyNote [BFK98], REFEREE [CFL+97], IBM Role-based Access Control Model [HMM+00], assumed trust relationships to be monotonic and do not manage trust considering the notion of learning from the available information. These systems are useful for access control decisions where a service provider determines what a consumer is allowed to do, but, not in a scenario where trust is a negotiation process, e.g, cloud computing marketplaces.

To overcome these problems in the existing *TM*s, Grandison et al. [GS03] proposed a policy-based *TM* framework that includes notation for specifying trust concepts as well as software tools for analysing and monitoring trust specifications. However, the proposed framework does not address the concept of *uncertainty* as a part of trust specification language for specifying trust relationships between a trustor (e.g., consumer) and a trustee (e.g., service provider).

Modelling and representing uncertainty is important when a trustor assess trustworthiness of a trustee based on evidence units, which are incomplete, insufficient and derived from unreliable sources. Hence, *TM* systems directed for distributed service environments should consider a mechanism that is able to deal with uncertainty and reflect it explicitly in representation as well as in computation of trust. Moreover, distributed service environments contain composed services in addition to published service-specific attributes that can be composed of several other attributes. Thus, underlying trust mechanisms should be able to compute trust under composition of attributes and services. Furthermore, evidence regarding the service-specific attributes can be derived from multiple sources. Therefore, underlying trust mechanism of a *TM* system should consider the issues such as user preference in selecting sources and computation of trust under conflicting information. Nevertheless,

policy-based approach, e.g., Grandison et al. [GS03], is an important element of a *TM* system. In this thesis, policy orchestration is not considered as the main focus for designing a *TM* system.

## 3.5 Summary

This chapter provides an extensive review on the state-of-the-art trust mechanisms proposed in different application scenarios. These mechanisms are rigorously analysed with respect to a set of *functional* and *non-functional* requirements. The contents of this chapter is summarised as follows:

- There is no one-fits-all solution to establish trust in distributed service environments.
- However, there are a couple of mechanisms, i.e., Subjective Logic and CertainTrust, that fulfil important non-functional requirements. We choose the *CertainTrust* representational model, as it allows one to adjust the opinion parameters independently and one to take the advantage of intuitive graphical representation of trust.

In this thesis, the author aims to provide the following mechanisms in order to address the gaps identified in the state-of-the-art:

- A generalized formal framework to assess trustworthiness of composed distributed services and service providers based on their published service-specific attributes.
- By design, the framework should be able to customise trust values according to consumers' personal preferences and interests.
- An extended computational framework based on CertainTrust is required for enabling trustworthiness evaluation considering composed architecture of the systems and services as well as composed service-specific attributes.
- A *TM* system by integrating the generalized trust assessment framework and the computational framework for trust evaluation is required to enable trust establishment in distributed service environments.

# 4

# Formal Framework for Trust Establishment

In service oriented environments, computing resources such as computing power, data storage, software are modelled as services. These services are offered directly or composed into other services. Even if the services are offered directly, the services can still be composed of systems or subsystems that are distributed across the world and managed by several parties. These kind of services are widely adopted in distributed service environments, e.g., cloud computing. For example, a cloud-based video rendering service might be composed of several distributed services such as a storage service from cloud storage provider and compute service from another provider. In such a service environment, a number of providers may provide services with similar functionality. However, there might be huge differences regarding the provided quality level of those services as well as the capabilities of the service providers. These are referred to as *non-functional characteristics* of a service provider. Therefore, distinguishing service providers based on their non-functional characteristics is essential for potential consumers to identify a dependable service. Grandison [Gra07, Wu11] suggests a relationship between *trust* and *dependability* in the context of distributed system by stating that the trustworthiness of a service relies on the dependability level of a trustee.

Dependability is defined as the ability of any trustor (i.e., consumer) to rely on trustee's (i.e., service provider) *behaviour*. In the context of trust establishment, service provider's behaviour regarding non-functional requirements should be able to meet the expectation of a consumer. Thus, it is important to formally model the *expected behaviour* of a trustee from a trustor's perspective. Moreover, the formal model should also consider the behaviour of services that service providers offer. As these services are

increasingly aggregated to offer a composed service in emerging distributed service environments, it is also important to consider the concept of *composition* when modelling the expected behaviour of services. Hence, we propose a *formal framework* to model the expected behaviour regarding non-functional attributes of service providers as well as of underlying services that are composed of subsystems and components. It will become evident in Section 4.2.1 that fulfilling the *expected behaviour* of a consumer ultimately influence the *trustworthiness* assessment process of service providers in distributed service environments.

Firstly, we revisit the cloud computing example briefly discussed in Chapter 1 in order to illustrate the means and necessity of trust assessment in distributed service environments. Then, the required concepts behind the formal framework are discussed in detail. Finally, the trust assessment framework is formulated along with intuitive examples driven by a couple of cloud computing scenarios.

## 4.1 Revisited Cloud-based Healthcare Scenario

In scenario (cf. Fig. 4.1) one considers a healthcare provider who wishes to outsource their in-house application, for managing medical records, to a cloud-based service. The main goal of the healthcare provider, the cloud consumer in this case, is to minimize IT expenditure as well as allowing doctors, patients, and insurance companies seamless access to these medical records using the cloud-based service. The medical records contain private information and outsourcing them to a service in the cloud requires that the service provider who hosts the service is trustworthy in handling private information. The healthcare provider requires assurances on *compliance* with regulatory acts such as HIPAA (Health Insurance Portability and Accountability Act), data protection through *security* and *privacy* attributes, *geographical location* (e.g., data should not leave specific political border) as well as high *availability* of the service. The healthcare provider considers service providers as *trustworthy* if they possess the capabilities to fulfil these assurances. Since the cloud service market for offering medical record services is competitive, the healthcare provider faces the challenge of selecting a trustworthy service provider that is best-suited and most appropriate for meeting its requirements from several alternatives.

In order to select a trustworthy cloud provider, the consumer (i.e., the healthcare provider) should be able to compare the offered services or solutions independently. This task includes analysing the *SLA* whether it address consumer's requirements and check whether the provider conform to specific *audit* standards or not. In order to perform these processes for each of the cloud providers could turn out to be a difficult and cumbersome task. Moreover, *ratings* derived from consumers' feedback about the capabilities

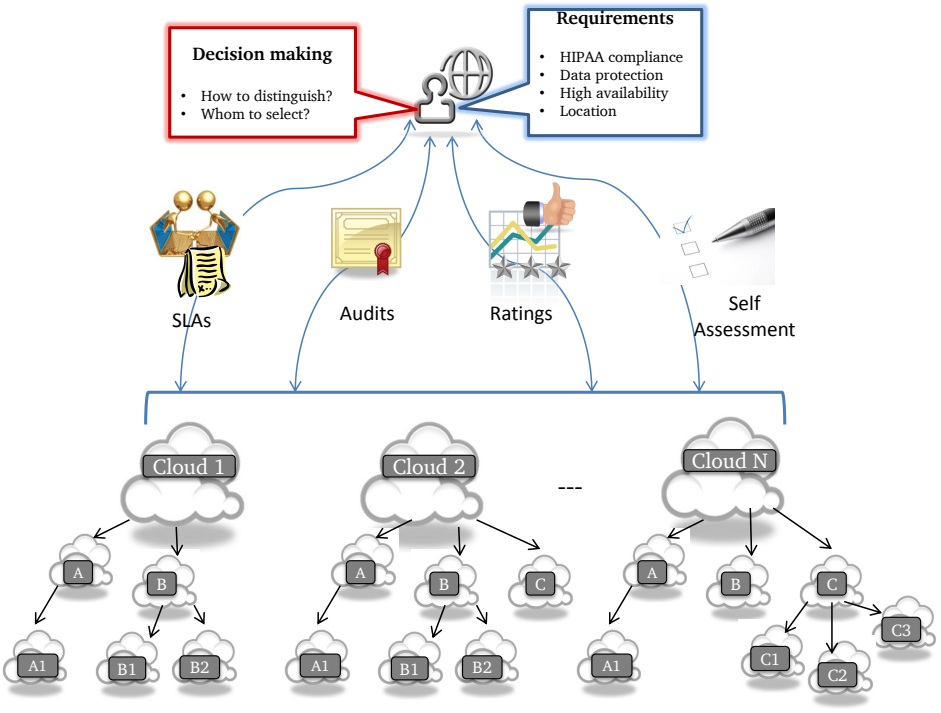



Figure 4.1: Means for Assessing Trustworthiness in Cloud Marketplaces

of cloud providers provide means to distinguish one service from the other. Recently, the *CSA* has introduced a *self-assessment* framework *CAIQ* [CSAc] that provides means to publish service-specific capabilities of cloud providers regarding various security and privacy attributes. The consumers can make use of these published capabilities from the STAR (Security, Trust & Assurance Registry) to get a better handle on the security attributes the cloud providers have. In the *CAIQ*, fulfilment of these attributes can be reasoned in a composite manner similar to service composition, for instance, a cloud provider is considered to fulfil the *Information Security (IS)* attribute given that the constituent attributes (IS-01–IS-34) are also fulfilled. However, the *CAIQ* framework as it stands does not provide a solution of how to analyse the *composite* structure in the context of trust assessment. Figure 4.1 summarizes the means for assessing the *trustworthiness* of service providers in the current scenario.

Moreover, cloud services (e.g., medical record management) are hosted in complex distributed systems (cf. Fig. 4.1), which are composed of subsystems and components. These systems are usually managed by multiple parties, i.e., service providers. Thus, it is essential to assess trustworthiness of cloud services considering (1) the trustworthiness of the underlying subsystems and atomic components (independent from how these trust values are assessed),

(2) information on how the system or a service combines its subsystems and components, and (3) the knowledge about which subsystems and components are redundant. The latter means that a system can be composed of identical components or subsystems, where each of those components and subsystems provides exactly the same functionality as others. Trustworthiness of a cloud service is assessed based on these information and the given requirements that users are interested of. For example, the healthcare provider might want to assess the trustworthiness of the cloud-based service regarding a security or an availability attribute.

## 4.2 Concepts behind the Formal Framework

The formal framework is designed to assess the composite structure of a service or system considering service-specific attributes given that these attributes are specified by the consumers. Moreover, the framework is adapted to the context of assessing the composite structure of attributes. In all these cases, it is assumed that service and system specification and the information about service-specific attributes are available. Thus, the framework only focus on assessing the composite structure of the service regarding the attributes and represent them in simplified meaningful terms. These representative terms serve as a formal basis for assessing the *trustworthiness* of service providers in distributed service environments, e.g., cloud computing marketplaces. Before introducing the formalisation of the framework, it is necessary to understand the rationale behind the construction of the framework.

### 4.2.1 Trustworthiness Assessment

Trust concepts are extensively discussed in the existing literatures as presented in Chapter 2. However, Chang et al. [CHD05] claimed that the related term *trustworthiness* has not been clearly distinguished in the existing literature. According to Chang et al., these literature are lacking a distinct definition comprising of *trust*, *trust values* and *trustworthiness*. To address the gap, Chang et al. defined *trustworthiness* as *a measure* of the level of trust that a trustor has in the trustee. They also argue that the trustworthiness is measured against the trustworthiness scale. According to the wide spread understanding of trust (cf. Section 2.1.1), it is subjective and in the context of this thesis it is modelled using *subjective probabilities*. Thus, the term "a measure" in the definition of trustworthiness refers to an *estimate* of the *degree of trust* which aligned with the concept of trustworthiness mentioned in [CHD05].

In order to assess the *trustworthiness* of a service provider according to consumers' requirements we model the consumers' notion of dependability in the context of distributed service environments by means of *propositional*

*statement*. For brevity, we refer the propositional statement as *proposition*. According to the stated relation between dependability and trustworthiness, we model dependability by means of *expected behaviour* of offered services or systems regarding different attributes. From a consumer perspective, the *expectation* of a consumer can be stated in the form of different attributes that a service and a service provider should have. On an abstract level, those attributes can come, for instance, from the fields of security, privacy, performance, customer support, and compliance. The following examples of *propositions* demonstrate how a consumer's expectation about the behaviour of a service, system and service provider regarding different attributes can be modelled.

- "Alice expects video rendering Service A to respond within $100ms$." (Performance)
- "Alice expects that storage Service provider B has the capability to provide data protection." (Security)
- "Bob expects Cloud A to provide 99.99% uptime in a yearly average for their storage service." (Availability)
- "Bob expects Cloud B to provide competent customer support for their platform service". (Customer support)
- "Charlie expects Cloud X's medical record management service comply with HIPAA" (Compliance)

Assessing *trustworthiness* in the presence of composite services and systems becomes non-trivial when each of those propositions need to be fulfilled by the constituent components and services. For example, Service A might be composed of several other services and subsystems. In this case, *trustworthiness* of Service A regarding performance attribute depends on the trustworthiness of constituent services and subsystems regarding the performance attribute. Moreover, each of the attributes may have been composed of sub-attributes, e.g., *Compliance* attribute may depend on a number of sub-attributes– HIPAA, ISO $270001 - 2005$, SAS 70 II. In this case, Charlie might expect that Cloud X's service complies with all three standards, i.e., sub-attributes.

From the discussion above, a question arise here is which attributes are important to consider in the trustworthiness assessment of service providers. Hence, the next section focuses on service-specific attributes that contribute into trustworthiness assessment of cloud providers.

### 4.2.2 Trust Attributes

According to the scenario in Section 4.1, there can be several service providers offering cloud-based healthcare services with similar functionalities. As stated

above, the consumers are interested to select cloud providers not only based on the functional characteristics (i.e., service types or methods) of the services but also based on *non-functional characteristics*. This refers to how well a service behaves and what sort of capabilities the providers possess regarding *non-functional* attributes. In cloud computing environments, according to Habib et al. [HRM10, HHRM12] those attributes go beyond the non-functional QoS parameters [WV07a], which are considered important for selecting trustworthy web service providers.

There are only two approaches [HRM10, UKJS10] known from the existing literature that attempt a systematic, well-founded, and comprehensive listing and identification of service-specific attributes. These attributes potentially contribute to the trustworthiness of a service, particularly services in cloud computing. One is described in [HRM10] and denoted as *QoS+*. It is based on a survey of existing literature in the field of cloud computing security, privacy and trust. The other one is described in [UKJS10] and denoted as *trust affectors*. These affectors are identified based on semi-structured interviews conducted over 33 persons representing cloud providers, cloud consumers, regulation authorities, and researchers from security, privacy, trust, and user experience (UX) field.

In the present thesis, the QoS+ serves as a list of attributes that influence trust establishment on cloud providers from a consumer's perspective. The trust affectors identified in [UKJS10] only demonstrates the need of such attributes that influence consumers to establish trust towards the service providers. Thus, we focus on identifying the *sources* of information and the *methods* for deriving information regarding the QoS+ attributes. Nevertheless, the motivating discussion of trust affectors in [UKJS10] demonstrates the need for those attributes in the trust establishment process. The information about the attributes are often available from multiple entities, e.g., Cloud Providers (CPs), Cloud Consumers (CCs), Cloud Accreditators (CAs), Cloud Brokers (CBs), Cloud Carriers (CCas). They provide the information regarding the service-specific attributes using different methods. In Table 4.1, the QoS+ parameters are listed along with their sources of information and the methods that can be used for extracting trust information.

1. **SLAs:** As discussed in Section 3.3, *SLA* is a common practice that service providers consider in order to build a contractual relationship with a potential consumer. In the context of *SLA*, a *CC* trust a *CP* to provide compensation in the case of violation of specific clauses in the agreement. The *SLA* violations are usually detected using *SLA* monitoring followed by validation mechanisms. These mechanisms assume that a *SLA* is machine-readable, however, it is not yet a standard practice in the cloud marketplaces. Moreover, the *SLA*s in the cloud marketplaces are lacking a standardised format which is important to distinguish *CP*s only based on the clauses and compensation amount

Table 4.1: QoS+ Parameters: Information sources and approaches

| QoS+ Parameters | Who provide the information? | How to derive the information? |
|---|---|---|
| SLA | CPs, CBs, CCs, CCas | SLA validation and monitoring mechanisms |
| Compliance | CAs, CSA | Audit Standards, CCM |
| Portability<br>Interoperability<br>Geographical Location | CPs | SLAs |
| Customer Support | CCs, CPs, CBs, CCas | SLAs, User Feedback |
| Performance | CBs, Independent Third-party, CCs, CPs | Measurement, User Feedback |
| Security | CSA, CPs, CAs | CSA CAIQ, Property Attestation mechanism, Audits |
| User Feedback | CCs | Measuring and Ratings (User Feedback) |
| Service Deployment Models<br>Service Delivery Models | CCs, CBs, CRs | Context Dependency and Similarity techniques |

formulated in the *SLA*s.

2. **Compliance:** *CP*s consider audit standards as an assurance for the existence of technical (e.g., security) and organizational policies related to the services they offer. *CA*s analyse and examine the systems, software applications, security policies, hardware components, and organisational policies on-site or remotely using automated techniques. Upon satisfaction of the necessary requirements and guidelines, *CA*s issue valid certificates. Additionally, the information regarding the compliance with different audit standards can be published via the *CSA STAR*. Therefore, information regarding the audit compliance can be obtained from the *CP*s as well as from the *CSA*.

3. **Portability, Interoperability, and Geographical Location:** In cloud computing environments, these three attributes are desirable by the consumers in order to ensure *portability* of their outsourced data once contract ends with the provider, *interoperability* of a composed service across multiple platforms hosted by multiple providers, and preferred *geo-location* of hosted service, e.g., data storage. The information regarding these attributes are usually obtainable from the *CP*s. The existence of terms and clauses related to portability and interoperability are usually mentioned in the SLAs. The geographical location of the required service can be selected from cloud service management platform (if supported) while deploying a service. In certain cases, information regarding the location of the datacenters are mentioned in the SLAs.

4. **Customer Support:** A dedicated and on-time customer support is unavoidable and desirable when outsourcing mission critical applications in the cloud. *CP*s usually provide assurances in the form of terms and clauses regarding "customer support" in the *SLA*s. CBs and CCas are also required to include similar terms in their *SLA*s for their respective consumers, e.g., *CP*s or *CB*s or *CC*s.

5. **Performance:** In cloud computing environments, the information about the performance attributes (e.g., availability, latency, bandwidth, elasticity) is obtained using service monitoring technologies [CA ]. Usually, *CP*s and *CB*s provide applications for monitoring those parameters, which take place only after service provisioning contract. *CC*s may hire third-party brokers (if required) to monitor those attributes before service provisioning takes place, e.g., in service trial period. However, real time data regarding availability and latency attributes of cloud services are publicly available through API status[1] website. The monitored or observed data regarding the performance attributes can be compared with the committed data specified in the *SLA*s by means of SLA validation mechanisms [LGO10]. Based on the validation results followed by relative comparison of those results among different providers may support CCs to select trustworthy *CP*s regarding performance attributes.

6. **Security:** *CC*s are interested to know about the existence of certain security and privacy attributes before outsourcing their computing resources to the cloud. *CP*s are able to publish these attributes and their detail information in the CSA STAR. In cloud computing marketplaces, this is the only way to learn about the capabilities of *CP*s regarding their service-specific security and privacy attributes before signing a contract with the respective providers. Moreover, *CP*s are increasingly hosting services in trusted platforms containing the Trusted Platform Module (TPM) introduced in the early 2000s by Trusted Computing Group (TCG) [TCG10]. In distributed service environments, e.g, cloud computing, consumers can learn about the security or non-security related behaviour of the software and hardware components running on those platforms using property attestation mechanism [NV11].

7. **User Feedback:** Feedback through recommendation, reviews, and experience from the consumers are valuable for service selection in e-marketplaces. User feedback may appear in quantitative (e.g., satisfaction score) and/or in qualitative (e.g., reviews) forms. Consumers' experience can be used to complement each of the above mentioned attributes to allow user control in evaluating the trustworthiness of service providers.

[1] `http://api-status.com/`

8. **Service Deployment and Delivery Models:** Trust models and mechanisms are usually context-specific. In cloud computing environments, the service delivery models such as SaaS, PaaS, IaaS can be considered as contextual parameters. Trusworthiness of a service provider might vary from one context to the another. Therefore, it makes sense to design context-aware trust systems in cloud computing environments. In order to develop such systems, context dependency and similarity techniques proposed by Jeh et al. [JW02] and Tavakolifard et al. [TKH08] should be taken into consideration.

## 4.3 Formal Framework

An overview of the formal framework is presented in Fig. 4.2. The framework follows two steps of formalization in order to assess composite services and systems regarding specific attributes. The first step is to analyse system and service descriptions considering given attributes and formally represent the whole system and service in *trustworthiness terms*. The second step is to convert the *trustworthiness* terms into a simple form called *Propositional Logic Terms* (PLTs). As an optional step, the *PLTs* can be converted to equivalent normal forms, i.e., Conjunctive Normal Form (CNF) and Disjunctive Normal Form (DNF).

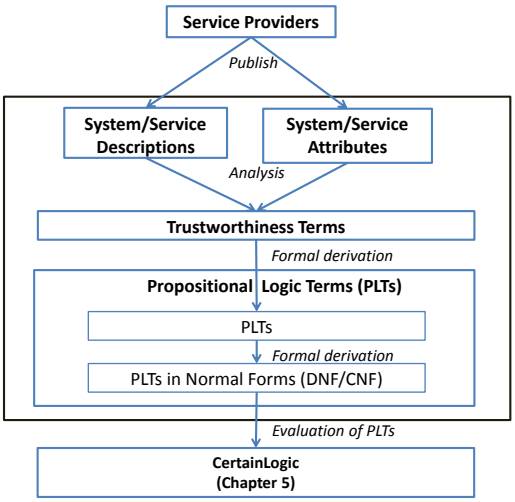



Figure 4.2: Formal Framework

In the next section, we discuss the basis of the formal framework and demonstrate how to represent a composite service/system using the formal apparatus of the framework.

### 4.3.1 Service/System Descriptions to *PLT*s

The basic idea of the framework rests on [Vv97, HKS00], which uses secret shares (cf. [BK05] for more details) and the concept that $k$ out of $n$ entities are needed for reconstructing the secret key and correspondingly decrypting a cipher text. We adapt this concept in the context of trust and propose a framework for modelling the composite services in distributed environments, where "$k$ out of $N$ constituent services/systems need to show trustworthy behaviour", in order not to compromise the *trustworthiness* of the composite service. In the case of Shamir's secret sharing mechanism [Sha79], the property $k$ out of $n$ means that we need to trust $k$ out of $N$ regarding availability and $n - k + 1$ out of $N$ regarding secrecy; where $N$ is the *set of share holders* and $n$ is the *number of share holders*. Thus, the choice of the actual value of $k$ is dependent on the security attribute (or 'requirement' of a consumer) under consideration. In contrast to the existing work in the aforementioned papers, which consider entities as homogeneous, our approach regard entities (i.e., services) as *heterogeneous* with respect to different requirements or attributes, e.g., security, compliance, availability by explicitly considering the itemized *set of* $N$. We use this model to formally represent the trustworthiness of a composite service regarding different attributes. Different attributes can lead to different representations. For example, in a system that implements a mixnet [Cha81], which routes messages sequentially through a *set* $N$ of $n$ anonymous nodes, each of the nodes must be trustworthy regarding availability attribute ($n$ out of $N$), while only one node needs to be trustworthy regarding anonymity (1 out of $N$).

#### 4.3.1.1 Transforming Composite Services/Systems into Trustworthiness Terms

Before presenting the formal definitions, the terminologies that are going to be used in this section need a brief discussion. First of all, the terms 'system' and 'service' are used to denote a composite service. Secondly, if a service has constituent services, which cannot be split further, these services are denoted as 'components' and the service is denoted as "atomic service". If a service has constituent services, which are further composed of constituent services, then the latter ones are denoted as "(sub)systems" and the service is denoted as "non-atomic service".

The definition of trustworthiness terms in terms of syntax and semantics follows the inductive definition of services and is provided by Definitions 1-4. For brevity, we introduce the abbreviation "*wrts.* $r$" (with regard to security requirement or attribute $r$). The informal notion "out of" used in the above discussion are formally used as 'out-of' in the following definitions. The motivation behind using 'out-of' term is to formally consider the

*composite structure* of services and reflect the service composition in the formal representation of trustworthiness regarding above mentioned security primitives.

Let $S$ be an atomic service with the set of components $A = \{A_1, \ldots, A_n\}$.

**Definition 1** A service $S$ can be described by the trustworthiness term ($k$ 'out-of' $|N|$), $k \in \{1, \ldots, |N|\}$, $N \subseteq A$, wrts. $r$

$$:\Leftrightarrow \begin{cases} \text{At least } k \text{ components 'out-of'} \\ N \text{ need to show trustworthy be-} \\ \text{haviour wrts. } r \text{ so that } S \text{ meets} \\ \text{requirement } r. \end{cases}$$ □

In order to get more flexible representations of requirements on atomic services, we define it in the following trustworthiness terms:

**Definition 2** A service $S$ can be described by the trustworthiness term (a) $((k_1 \textcircled{\wedge} \ldots \textcircled{\wedge} k_m)$ 'out-of' $(N_1, \ldots, N_m))$, (b) $((k_1 \textcircled{\vee} \ldots \textcircled{\vee} k_m)$ 'out-of' $(N_1, \ldots, N_m))$, $k_i \in \{1, \ldots, |N_i|\}$, $N_i \subseteq A \; \forall i$, wrts. $r$

$$:\Leftrightarrow \begin{cases} \text{For a) each } i \in \{1, \ldots, m\}, \text{ b)} \\ \text{any } i \in \{1, \ldots, m\}, \text{ at least } k_i \\ \text{components 'out-of' } N_i \text{ need} \\ \text{to show trustworthy behaviour} \\ \text{wrts. } r \text{ so that } S \text{ meets require-} \\ \text{ment } r. \end{cases}$$ □

With regard to non-atomic services, we define trustworthiness terms similarly:

Let $\{S_i\}_{i=1}^n$ be (sub)systems of a service $S$, and let system $S_i$ be described by the trustworthiness term ($k$ 'out-of' $l_i$) for all $i \in \{1, \ldots, n\}$.

**Definition 3** A service $S$ can be described by the trustworthiness term ($k$ 'out-of' $\{l_{i_1}, \ldots, l_{i_m}\}$), $k \in \{1, \ldots, m\}$, $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$, wrts. $r$

$$:\Leftrightarrow \begin{cases} \text{At least } k \text{ systems 'out-of'} \\ \{S_{i_1}, \ldots, S_{i_m}\} \text{ need to show} \\ \text{trustworthy behaviour wrts. } r \\ \text{so that } S \text{ meets requirement } r. \end{cases}$$ □

**Definition 4** A service $S$ can be described by the trustworthiness term a) $((k_1 \textcircled{\wedge} \ldots \textcircled{\wedge} k_m)$ 'out-of' $(Q_1, \ldots, Q_m))$, b) $((k_1 \textcircled{\vee} \ldots \textcircled{\vee} k_m)$ 'out-of' $(Q_1, \ldots, Q_m))$, $k_i \in \{1, \ldots, |Q_i|\}$, $Q_i \subseteq \{l_1, \ldots, l_n\} \; \forall i$, wrts. $r$

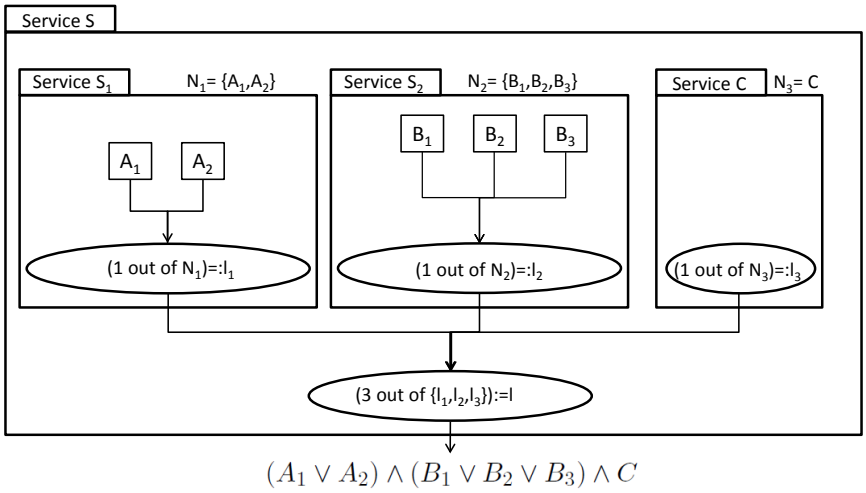



Figure 4.3: A MRM service: inductive determination of PLTs

$$:\Leftrightarrow \begin{cases} \text{For a) each } i \in \{1,\ldots,m\}, \\ \text{b) any } i \in \{1,\ldots,m\}, \text{ at} \\ \text{least } k_i \text{ systems 'out-of' the} \\ \text{set of systems for which } Q_i \\ \text{contains trustworthiness terms} \\ \text{need show trustworthy be-} \\ \text{haviour wrts. } r \text{ so that } S \text{ meets} \\ \text{requirement } r. \end{cases} \qquad \square$$

We now apply the inductive steps on the following two scenarios to determine the trustworthiness terms from given system/service descriptions.

**Example 1** Let us take an example from the field of cloud computing, and demonstrate how to analyse a composite cloud service considering its internal architecture and transform that service into trustworthiness terms. The objective is to assess trustworthiness of a simple Medical Record Management (MRM) service wrts. *availability* of the service.

In the example (cf. Fig. 4.3), the *MRM* service $S$ directly relies on two subsystems and an atomic component: $S_1$ provides authentication capabilities, $S_2$ offers storage capacity for medical records, and an atomic component $C$ is responsible for service-related billing. Subsystem $S_1$ consist of two authentication servers ($A_1$ and $A_2$), where at least *one* of the servers/systems has to be available for the service to be functional. Similarly, subsystem $S_2$ is composed of three redundant database servers and only *one* needs to be available. Figure 4.3 demonstrates the procedure of transforming a composite service into PLTs.

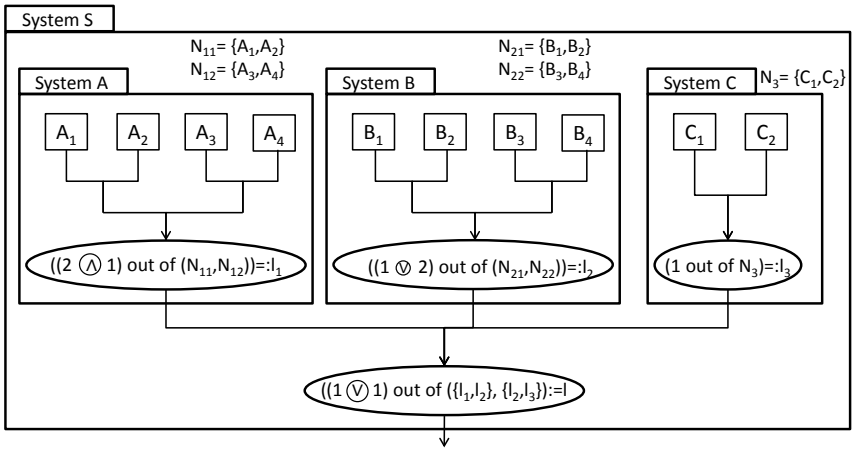
Applying definitions 1 and 3, the following trustworthiness terms are obtained with respect to the *availability* attribute:

- $A$: $\underbrace{(1\text{ 'out-of' }\{A_1, A_2\})}_{=:l_1}$ (def. 1)
- $B$: $\underbrace{(1\text{ 'out-of' }\{B_1, B_2, B_3\})}_{=:l_2}$ (def. 1)
- $C$: $\underbrace{(1\text{ 'out-of' }\{C\})}_{=:l_3}$ (def. 1)
- $S$: $(3\text{ 'out-of' }(\{l_1, l_2\, l_3\}))$ (def. 3)

**Example 2** Figure 4.4 demonstrates a composite web service scenario, in which a retailer (i.e., consumer) uses three web services in order to identify end customers' behavior at the end. Service $A$ offers data mining capabilities and stores sales data, including customer IDs. Service $B$ is offered by a financial service provider, who provides credit ratings of customers. Service $C$ provides storage capacities and stores master data on customers, including their customer IDs and identities. In this example, the retailer considers *secrecy* with regard to information, i.e., *which* customer has bought *what* under *which* financial conditions should not be revealed. *Secrecy* attribute is fulfilled if one of the providers $A$ and $B$ is trustworthy, or if one of $B$ and $C$ is trustworthy. With regard to subsystem $A$, we assume that this system accounts for *secrecy* by storing data on two components ($A_3$ and $A_4$) and implements a secret sharing mechanism [BK05]. Components $A_1$ and $A_2$ are responsible for distributed computation in terms of data mining; both components receive data from $A_3$ and $A_4$. With regard to financial service $B$, customer IDs generated by $B$ (they differ from customer IDs stored at $A$) are stored on $B_1$ and $B_2$ together with financial data by implementing a secret share mechanism. Components $B_3$ and $B_4$ store names of customers and customer IDs (generated by $B$) respectively. Analogous to $A$ and $B$, storage service $C$ implements a secret share mechanism when storing customer data.

In Figure 4.4, $l$ refers to the composite service/system and $l_i$ with $i \in \{1, 2, 3\}$ represents its constituent subsystems. Applying definitions 1, 2a, 2b, and 4b to the given scenario, we yield the following trustworthiness terms with respect to the *secrecy* attribute or requirement:

- $A$: $\underbrace{((2 \oslash 1)\text{ 'out-of' }(\{A_1, A_2\}, \{A_3, A_4\}))}_{=:l_1}$ (def. 2a)
- $B$: $\underbrace{((1 \otimes 2)\text{ 'out-of' }(\{B_1, B_2\}, \{B_3, B_4\}))}_{=:l_2}$ (def. 2b)
- $C$: $\underbrace{(1\text{ 'out-of' }\{C_1, C_2\})}_{=:l_3}$ (def. 1)
- $S$: $((1 \otimes 1)\text{ 'out-of' }(\{l_1, l_2\}, \{l_2, l_3\}))$ (def. 4b)




$$(A_1 \wedge A_2 \wedge (A_3 \vee A_4)) \vee (B_1 \vee B_2 \vee (B_3 \wedge B_4)) \vee (C_1 \vee C_2)$$

Figure 4.4: A web service: inductive determination of PLTs

#### 4.3.1.2 Mapping Trustworthiness Terms to Propostional Logic Terms (PLTs)

The examples show that representation of a composite service by means of *trustworthiness terms* can become complex even for simple services. In order to represent trustworthiness terms in a simplest possible format for allowing easy interpretation and evaluation, trustworthiness terms are transformed into *PLT*s. The *PLT*s are further transformed into semantically equivalent normal forms (i.e., *CNF* and *DNF*). These normal forms represent individual strengths of a composite service: i) CNF allows determining "weak configurations" such as single points of failure, ii) DNF allows determining "strong configurations" such as redundancy of the subsystems or components which leads to trustworthiness of the overall service, regardless of the trustworthiness of other components and subsystems. Thus, both normal forms are considered complimentary when representing trustworthiness of composite services and systems in formal terms.

**Theorem 4.3.1** *Let service $S$ consist of basic components $A = \{A_1, \ldots A_n\}$, and let $\{X_{A_1}, \ldots, X_{A_n}\}$ be literals with $\forall i{:}X_{A_i} = true$ , if $A_i$ is trustworthy wrts. $r$. Then, the trustworthiness term $l$ of $S$ can be mapped on a propositional logic formula $f(l)$ such that $S$ is trustworthy wrts. $r$ if and only if $f(l)$ is true.*

The proof regarding Theorem 4.3.1 is provided in Appendix A.1.

The examples presented in Fig. 4.3 and Figure 4.4 are used as running examples to illustrate how to determine PLTs from corresponding trustworthiness terms, namely $l_1, l_2, l_3$, and $l$.

**Example 3** The following example is the continuation of Example 1.

- $l_1 = (1$ 'out-of' $(\{A_1, A_2\}))$

$$\begin{aligned} \Rightarrow f(l_1) &\overset{(A.1)}{=} (A_1) \vee (A_2) \\ &= A_1 \vee A_2 =: f_A \end{aligned}$$

- $l_2 = 1$ 'out-of' $(\{B_1, B_2, B_3\})$

$$\begin{aligned} \Rightarrow f(l_2) &\overset{(A.1)}{=} (B_1) \vee (B_2) \vee (B_3) \\ &= B_1 \vee B_2 \vee B_3 =: f_B \end{aligned}$$

- $l_3 = (1$ 'out-of' $\{C\})$

$$\Rightarrow f(l_3) \overset{(A.1)}{=} C =: f_C$$

- $l = 3$ 'out-of' $(\{l_1, l_2, l_3\})$

$$\begin{aligned} \Rightarrow f(l) &\overset{(A.4)}{=} f((3 \text{ 'out-of' } \{l_1, l_2, l_3\})) \\ &= (f(l_1)) \wedge (f(l_2)) \wedge (f(l_3)) \\ &= (f_A) \vee (f_B) \vee (f_C) \\ &= (A_1 \vee A_2) \wedge (B_1 \vee B_2 \vee B_3) \wedge C \end{aligned} \tag{4.1}$$

□

Finally, we convert the resulting propositional logic term given in E4.1) into CNF.

$$\bigvee_{\substack{X \in \{A_1, A_2\} \\ Y \in \{B_1, B_2, B_3\}}} (X \wedge Y \wedge C) \tag{4.2}$$

The DNF formula given in Equation 4.1 shows that service $S$ is trustworthy with respect to *availability* attribute if all the subsystems are trustworthy wrts. *availability*.

**Example 4** The following example is the continuation of Example 2.

- $l_1 = ((2 \oslash 1)$ 'out-of' $(\{A_1, A_2\}, \{A_3, A_4\}))$

$$\begin{aligned} \Rightarrow f(l_1) &\overset{(A.2)}{=} (f((2 \text{ 'out-of' } \{A_1, A_2\})))\wedge \\ &\quad (f((1 \text{ 'out-of' } \{A_3, A_4\}))) \\ &\overset{(A.1)}{=} ((A_1 \wedge A_2)) \wedge ((A_3) \vee (A_4)) = A_1 \wedge \\ &\quad A_2 \wedge (A_3 \vee A_4) =: f_A \end{aligned}$$

- $l_2 = ((1 \circledcirc 2)\text{ 'out-of' } (\{B_1, B_2\}, \{B_3, B_4\}))$

$$\begin{aligned} \Rightarrow f(l_2) &\overset{(A.3)}{=} (f((1\text{ 'out-of' }\{B_1, B_2\})))\vee \\ &\quad (f((2\text{ 'out-of' }\{B_3, B_4\}))) \\ &\overset{(A.1)}{=} ((B_1 \vee B_2)) \vee ((B_3) \wedge B_4)) \\ &= B_1 \vee B_2 \vee (B_3 \wedge B_4) =: f_B \end{aligned}$$

- $l_3 = (1\text{ 'out-of' }\{C_1, C_2\})$

$$\Rightarrow f(l_3) \overset{(A.1)}{=} (C_1) \vee (C_2) = C_1 \vee C_2 =: f_C$$

- $l = ((1 \circledcirc 1)\text{ 'out-of' } (\{l_1, l_2\}, \{l_2, l_3\}))$

$$\begin{aligned} \Rightarrow f(l) &\overset{(A.6)}{=} (f((1\text{ 'out-of' }\{l_1, l_2\})))\vee \\ &\quad (f((2\text{ 'out-of' }\{l_2, l_3\}))) \\ &\overset{(A.4)}{=} (((f(l_1))) \vee ((f(l_2)))) \vee (((f(l_2)))\vee \\ &\quad ((f(l_3)))) \\ &= (f(l_1)) \vee (f(l_2)) \vee (f(l_3)) \\ &= (f_A) \vee (f_B) \vee (f_C) \\ &= (A_1 \wedge A_2 \wedge (A_3 \vee A_4))\vee \\ &\quad (B_1 \vee B_2 \vee (B_3 \wedge B_4)) \vee (C_1 \vee C_2) \end{aligned} \tag{4.3}$$

□

Finally, we convert the resulting propositional logic term given in (4.3) into CNF.

$$\bigwedge_{\substack{X \in \{A_1, A_2, A_3\} \\ Y \in \{A_1, A_2, A_4\} \\ Z \in \{B_3, B_4\}}} (X \vee Y \vee B_1 \vee B_2 \vee Z \vee C_1 \vee C_2) \tag{4.4}$$

(4.2) can be easily derived when we first transform (4.3) into DNF, given by

$$(A_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge A_2 \wedge A_4) \vee B_1 \vee B_2 \vee (B_3 \wedge B_4) \vee C_1 \vee C_2$$

The CNF formula given in (4.4) reveals that system $S$ is trustworthy with respect to *secrecy* attribute if at least *one* of the components $B_1, B_2, C_1, C_2$ is trustworthy wrts. $r$, which is a sufficient, but not necessary condition.

### 4.3.2 System/Service Attributes to *PLTs*

In a distributed service environment, service provider might not be interested in publishing service descriptions regarding different attributes. As an alternative, capabilities regarding service-specific attributes can be published using a self-assessment framework such as the CSA CAIQ (cf. Section 4.4.2 for more details). These attributes are composed of several other attributes. Thus, the proposed formal framework can also be applied to represent composite attributes in terms of *PLT*s.

For representing the composite attributes into PLTs, we simply replace the term service(s) with attribute(s) in the Definition 1, 2, 3 and 4. In the context of service-specific attributes, security requirement (i.e., $r$) term used in the previous definitions are replaced by consumer's requirement(s) ($R$) where consumers might prefer to personalise a set of requirements. Additionally, configuration of the *PLT*s depend on the consumers' preferred set of requirements from the published set of service-specific attributes whereas in the context of composite services it depends on the type of attributes, e.g., secrecy, availability. Note that "atomic attribute" and "non-atomic attribute" share the same notion of "atomic service" and "non-atomic service" respectively.

#### 4.3.2.1 Transforming System/Service Attributes into Trustworthiness Terms

The definition of trustworthiness terms in the context of published service-specific attributes, follow similar syntax and semantics of the definitions presented in Section 4.3.1.1. The formal inclusion of the 'out-of' term in the following definitions share the same motivation as given in the context of service composition. Here, the service attributes are considered as composite, i.e., a service attribute may consist of sub-attributes. Regarding service-specific composite attributes, the definitions are as follows.

Let $S$ be a service with atomic attributes, $P = \{P_i\}_{i=1}^{n}$. Assume that user requirements, $R = \{R_1, \ldots, R_n\}$ are subset of published attributes, $R \subseteq P$. In the case of single requirement, $r \subseteq P$. Every attribute $P_i$ assumed to have sub-attributes, $|N|$.

**Definition 5** An atomic property $P_i$ of a service $S$ can be described by the trustworthiness term ($k$ 'out-of' $N$), $k \in \{1, \ldots, |N|\}$, $N \subseteq P_i$, wrts. $r$

$$:\Leftrightarrow \begin{cases} \text{At least } k \text{ sub-attributes 'out-} \\ \text{of' } N \text{ need to be satisfied so} \\ \text{that } S \text{ meets requirement } r. \end{cases}$$ □

In order to describe atomic attributes of a service that satisfies more than one requirement, we define the following trustworthiness terms:

**Definition 6** Different atomic attributes $P$ of a service $S$ can be described by the trustworthiness term, $((k_1 \oslash \ldots \oslash k_m)$ 'out-of' $(N_1, \ldots, N_m))$; $\forall i$ $k_i \in \{1, \ldots, |N_i|\}$, $N_i \subseteq P$ , wrts. $R$

$$:\Leftrightarrow \begin{cases} \text{For each } i \in \{1, \ldots, m\} \text{ at} \\ \text{least } k_i \text{ sub-attributes 'out-of'} \\ N_i \text{ need to be satisfied so that} \\ S \text{ meets requirements } R. \end{cases}$$ □

In order to represent non-atomic attributes of a service by means of trustworthiness terms we define the following two definitions. Let $\{P_i\}_{i=1}^{n}$ be sub-attributes of a system $S$, and let property $P_i$ of the system be described by the following trustworthiness term $l_i, \forall i \in \{1, \ldots, n\}$. Assuming that user requirements, $R = \{R_1, \ldots, R_n\}$ are subset of published properties, $R \subseteq P_i$. In the case of single requirement, $r \subseteq P_i$.

**Definition 7** A non-atomic attribute $P_i$ of a service $S$ can be described by the trustworthiness term ($k$ 'out-of' $\{l_{i_1}, \ldots, l_{i_m}\})$, $k \in \{1, \ldots, m\}$, $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$, wrts. $r$

$$:\Leftrightarrow \begin{cases} \text{At least } k \text{ attributes 'out-of'} \\ \{P_{i_1}, \ldots, P_{i_m}\} \text{ need to be sat-} \\ \text{isfied so that } S \text{ meets require-} \\ \text{ment } r. \end{cases}$$ □

**Definition 8** Different non-atomic attributes $P$ of a system $S$ can be described by the trustworthiness term, $((k_1 \oslash \ldots \oslash k_m)$ 'out-of' $(Q_1, \ldots, Q_m))$;$\forall i$ $k_i \in \{1, \ldots, |Q_i|\}$, $Q_i \subseteq \{l_1, \ldots, l_n\}$ , wrts. $R$

$$:\Leftrightarrow \begin{cases} \text{For each } i \in \{1, \ldots, m\} \text{ at} \\ \text{least } k_i \text{ attributes 'out-of' the} \\ \text{set of attributes for which } Q_i \\ \text{contains trustworthiness terms} \\ \text{need to be satisfied so that } S \\ \text{meets requirements } R. \end{cases}$$ □

We demonstrate the analysis and determination of trustworthiness terms with example 5. Figure 4.5 demonstrates the inductive determination of trustworthiness terms from the given service-specific attributes.

**Example 5** Assume that a cloud provider has a set of attributes: security, compliance, data governance and the attributes are published according to the guidelines of *CSA CAIQ* in a public repository. A user wants to assess the trustworthiness of a cloud provider wrt. following requirements, security and compliance. We assume that a static mapping between the user requirements and published capabilities of cloud providers are already available. For instance, in order to satisfy the user's *security* requirement

cloud provider has to posses capabilities regarding three attributes: "Facility Security" (FS), Human resources Security (HS), and Information Security (IS). The above definitions are used to convert the service-specific attributes into trustworthiness terms. Applying definitions 5 and 7, the following trustworthiness terms are obtained according to given requirements of the user, i.e., *security* and *compliance*.

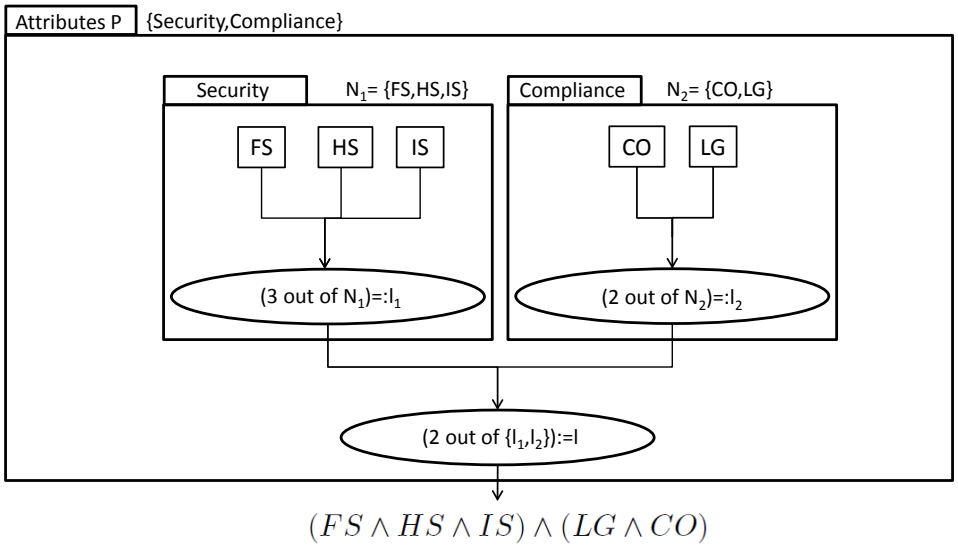



Figure 4.5: Composite attributes: inductive determination of PLTs

- *Security*: $\underbrace{(3\text{ 'out-of' }\{FS, HS, IS\})}_{=:l_1}$ (def. 5)
- *Compliance*: $\underbrace{(2\text{ 'out-of' }\{CO, LG\})}_{=:l_2}$ (def. 5)
- $\{Security, Compliance\}$: $(2\text{ 'out-of' }(\{l_1, l_2\}))$ (def. 7)

#### 4.3.2.2 Mapping Trustworthiness Terms to PLTs

Similar to Section 4.3.1.2, the trustworthiness terms derived from system/service attributes are mapped to *PLTs* to allow easy interpretation and evaluation of trustworthiness. The notion of *CNF* and *DNF* are not considered complimentary in the context of system/service attributes. A similar theorem is formulated to derive PLTs from trustworthiness terms in the context of system/service attributes. In the following theorem, the main focus is on the system or service attributes instead of system/service components or subsystems.

**Theorem 4.3.2** *Let attributes $P$ of system $S$ consist of sub-attributes $P = \{P_1, \dots P_n\}$ and let $\{X_{P_1}, \dots, X_{P_n}\}$ be literals with $\forall i$: $X_{P_i} = true$, if $R \subseteq P$ and $r \subseteq P_i$. Then, the trustworthiness term $l$ can be mapped to a propositional logic formula $f(l)$ such that $S$ is trustworthy wrts. $r$ or $R$ if and only if $f(l)$ is satisfied.*

The proof regarding Theorem 4.3.2 is provided in Appendix A.2.

We use the example shown in Figure 4.5 to illustrate how to determine the propositional logic formula of particular trustworthiness terms, namely $l_1, l_2, l_3$, and $l$.

**Example 6**

- $l_1 = (3$ 'out-of' $(\{FS, HS, IS\}))$

$$\begin{aligned} \Rightarrow f(l_1) &\overset{(A.7)}{=} (FS) \wedge (HS) \wedge (IS) \\ &= FS \wedge HS \wedge IS =: f_{security} \end{aligned}$$

- $l_2 = 1$ 'out-of' $(\{B_1, B_2, B_3\})$

$$\begin{aligned} \Rightarrow f(l_2) &\overset{(A.7)}{=} (LG) \wedge (CO) \\ &= LG \wedge CO =: f_{compliance} \end{aligned}$$

- $l = 2$ 'out-of' $(\{l_1, l_2\})$

$$\begin{aligned} \Rightarrow f(l) &\overset{(A.9)}{=} (f(l_1)) \wedge (f(l_2)) \\ &= (f_{security}) \wedge (f_{compliance}) \\ &= (FS \wedge HS \wedge IS) \wedge (LG \wedge CO) \end{aligned} \tag{4.5}$$

□

### 4.3.3 Evaluation of PLTs

The proposed framework provide means to model the trustworthiness of composite distributed services regarding different attributes. Particularly, the framework analyses the dependencies between the components and subsystems as well as their redundancy in respect of service-specific attributes under consideration. The PLTs allow to represent the specification of a composite service and system in a simplified formal term. However, it does not serve the purpose to quantitatively evaluate the trustworthiness of service providers. The quantitative evaluation is important to compare services and systems, offered by service providers, of similar non-functional attributes. In order to evaluate the trustworthiness of composite distributed services, there is a need for associating values with the PLTs. This means that each

component and subsystem of a service should be associated with a *value* that represents the satisfaction of a security attribute or service-specific attribute of a component or a subsystem. The values should be *aggregated* according to the derived specification (i.e., PLTs) of the composite services/systems.

According to existing approaches (cf. Chapter 2), such a *value* can appear in the form of *opinion*, which is based on available pieces of evidence. These are usually derived using tentative measurements such as past experience, past interactions, self-assessment, expert assessment or remote assessment of services in trusted platforms. Derived pieces of evidence can be incomplete, unreliable or indeterminable due to the type of measuring methods. Thus, opinions derived from this evidence is subject to *uncertainty*. Moreover, the opinions derived from different sources might be *conflicting* either due to the method they use, for assessing evidence, or simply because the sources are unreliable. The evaluation mechanisms should consider these issues in order to provide a representative trustworthiness value. The mechanism should also be able to aggregate opinions in *composition* as represented in *PLTs*.

The propositions in the *PLTs* are combined with logical operators (i.e., AND ($\wedge$) and OR ($\vee$)) and those operators should be able to combine opinions, which are subject to uncertain and conflicting pieces of evidence. In this vein, we propose novel definitions of the logical operators as well as non-standard (i.e, FUSION) operators in the next chapter. These definitions are particularly designed to deal with uncertain and conflicting opinions associated with propositions.

## 4.4 Domains of Application

This section illustrates a selected set of domains, where trust establishment mechanisms are applicable. In particular, the following domains demonstrate the need for assessing and evaluating the *trustworthiness* of service providers based on *composite* services they offer.

### 4.4.1 Cloud Marketplaces

Cloud marketplaces [LJ10] are emerging as a part of distributed service environments to facilitate seamless trading of IT commodities among cloud stakeholders. Cloud providers (sellers) and consumers (buyers) are the two main participating entities in a cloud marketplace. As the business market is growing rapidly in cloud marketplaces with new providers entering the market, the providers are expected to compete for customers by providing services with similar non-functional attributes, e.g., security, SLAs, performance, customer support. However, there can be notable differences among the service providers regarding the provided quality level as well as capabilities regarding the attributes. It is difficult to assess these notable differences when there are few solutions at hand, nor ones that are able to do so. Thus, cloud

consumers require a solution that can reliably assess the cloud providers in terms of their capabilities as well as support consumers to determine dependable and trustworthy cloud providers.

Looking at a cloud computing marketplace from a microscopic point of view, where consumers would like to provision a cloud based service from the most dependable and trustworthy cloud provider; the scope is limited to a single cloud provider in order to focus on the complexity of provider's internal service architecture. A company named MedicalVault (MV) offers medical record management service to millions of customers, e.g., hospitals, insurance companies, and patients. Obviously, medical records are sensitive information and should remain private and confidential.

MV hosts a small data center where they keep control over their proprietary APIs (Application Programming Interfaces) that contain data processing and mining algorithms and patented search technologies. For the rest of the services, it uses a public cloud environment provided by Cloud A, a Cloud provider located in Hamburg, Germany. Thus, we are dealing with a hybrid cloud.

As shown in Figure 4.6, Cloud A is internally using Cloud B, a public Cloud provider located in Bengaluru, India, for the following purposes (this information is not publicly available as Cloud A considers this to be a business advantage):

- medical image processing – using MV's ProImage software hosted on a remote server.
- advanced image processing tools (format conversion, filtering) – that ProImage does not support but Cloud A support by using software tools, TurboConv, TurboFilter from Cloud B and
- video archiving – using remote storage infrastructure

Cloud A site in Hamburg is responsible for hosting the ProImage software in their application server. After processing the images, those are stored in a temporary storage, marked (1) in figure 4.6.

Cloud A outsources advanced image processing tasks to Cloud B, a SaaS provider located in Bengaluru, India, in order to offload their infrastructure. On Cloud B's site, the advanced image rendering takes place using the TurboConv and TurboFilter services. Processed images are temporarily stored in Cloud B's (temporary) storage (2). Later, images are permanently archived in storage (3), (4), and (5) of Cloud A's storage infrastructure, located in Hongkong, China.

In this case (cf. Figure 4.6), MV comprises three types of service delivery models: platform-as-a-service (PaaS), software-as-a-service (SaaS), and infrastructure-as-a-service (IaaS). MV leverages virtual platforms (i.e., PaaS) at the Hamburg site of Cloud A to deploy the ProImage software. In the

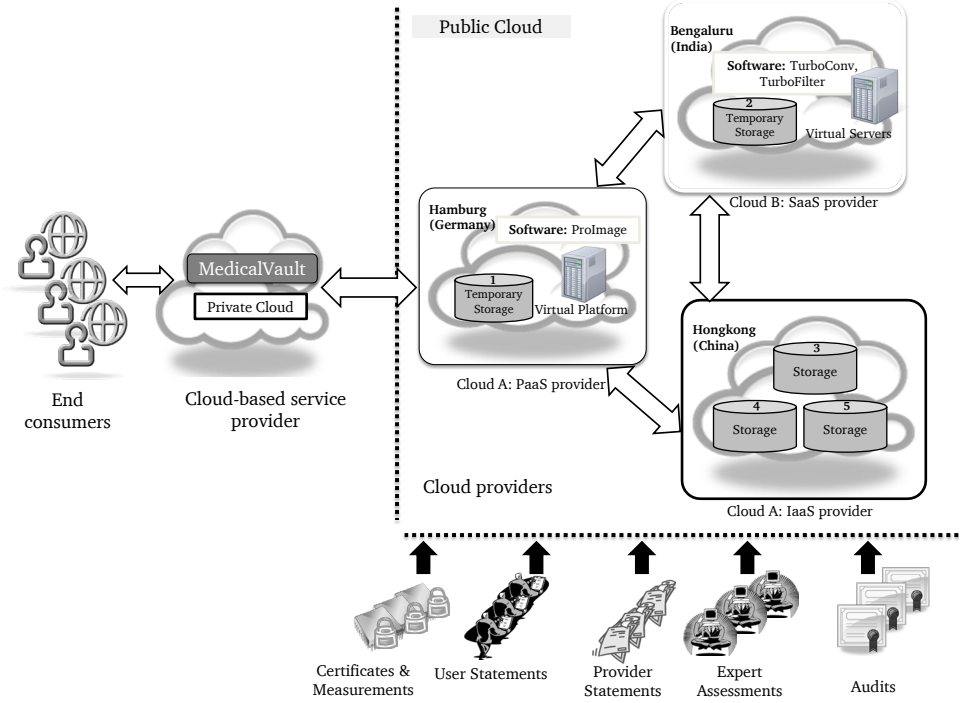



Figure 4.6: Cloud computing use case: microscopic view

SaaS case, MV uses the Cloud B's TurboConv and TurboFilter services. In the IaaS case, MV use the computing and storage infrastructure by accessing virtual servers and storages provisioned in the physical infrastructure of Cloud A.

Figure 4.6 illustrates how a composed service (e.g., medical record management) is processed and stored using several software and hardware components located in different geographical locations (e.g., Hamburg, Bengaluru, and Hongkong) around the globe. This distributed information sharing and processing of data builds upon non-transparent service compositions that are handled by several service providers (e.g., MV, Cloud A, and Cloud B). Consequently, the presented scenario leads to the question how much and in which contexts the end consumers can trust this kind of aggregated service, or whether they should refrain from using it.

The proposed formal framework can be applied to assess the composite architecture of the service, i.e., medical record management, and represent the service architecture in the form of PLTs. The formation of the PLTs depends on the service-specific attributes under consideration as discussed in Section 4.3. PLTs, representing a composite service, are associated with opinions, which represents the behaviour of the service and their underlying components regarding service-specific attributes. These opinions are often

available from different sources (cf. Figure 4.6) and are extracted using different methods. For evaluating (cf. Section 6.3 and Section 6.4) the trustworthiness of cloud providers, the opinions about their offered services regarding different attributes need to be aggregated and represented in a way so that the consumers can decide to provision a service from the most trustworthy provider out of similar ones.

### 4.4.2 Security Control Self-Assessment Framework

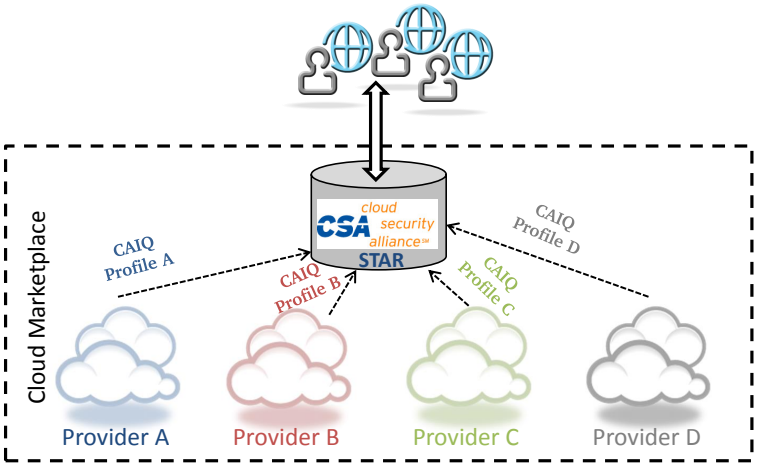



Figure 4.7: Current status of STAR

Cloud providers' capabilities regarding security attributes are important indicators of trustworthiness in competitive marketplaces. In order to encourage cloud providers, the CSA provides a self-assessment questionnaire, i.e., CAIQ, directed for cloud providers to publish capabilities regarding their service-specific attributes for potential consumers. In the last quarter of 2011, the CSA unveil the Security, Trust, and Assurance Registry (STAR) for publishing CAIQ profiles or reports regarding each of the services the cloud providers offer. The main reason behind this initiative is to support consumers to assess the security capabilities of providers before consumers consider contracting with them. The STAR and CAIQ frameworks as they stand (cf. Figure 4.7) does not offer consumers an automated solution to assess the capabilities regarding security attributes nor does they provide any means for consumers to specify their personal requirements when assessing the capabilities. Due to these shortcomings, consumers have to manually analyse the existence of security capabilities as well as manually check the compliance of those capabilties with their specific requirements. The manual process is undoubtedly a cumbersome task given that the CAIQ comprises of 11 domains which is further classified into 98 controls complemented with

197 questions and one has to manually assess the existence of capabilities regarding the domains (i.e., security attributes) by analysing the assertions given in response to the 197 questions.

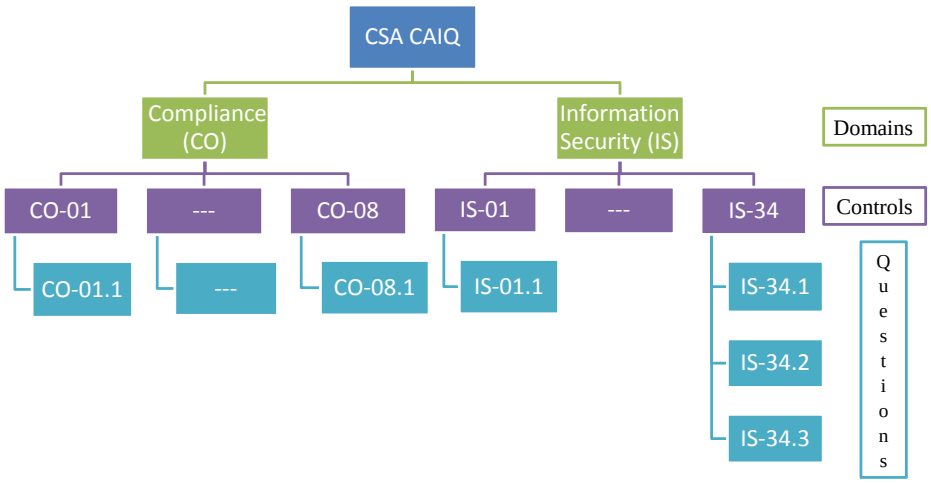



Figure 4.8: CSA CAIQ: Overview of basic structure

Each of the security attributes, as given in the CAIQ, maintain an hierarchical structure (cf. Figure 4.8). The top node is the domain (i.e, security attribute) followed by controls and control questions. For example, Compliance (*CO*) domain is consist of eight controls and each of the controls have one or more questions. Our proposed formal framework is applicable to model such compositions which are extensively discussed in Section 4.3 with intuitive examples. Modelling and representing controls and domains in terms of PLTs are the basis for automated assessment of completed CAIQs. For quantitative assessment of CAIQs, all controls and domains should be associated with numerical values (termed as "opinions"). These values can be extracted from the given assertions in reply to control questions by the service providers. The questions are designed in a way so that the providers are able to answer them in "yes/no" manner. In special cases, cloud providers also provide answers using other means, e.g., detailed comments, "Not Applicable" or skip a specific question. All these different types of assertions are the pieces of evidence, which refer to the existence of corresponding attributes or domains of a service offered by a cloud provider. In this case, evidence-based trust assessment mechanisms (cf. Section 2.2.2) is a good choice for extracting representative values, which in the end can contribute into trustworthiness evalution of cloud providers.

In evidence-based trust mechanisms, quality of interactions between a trustor and a trustee are usually measured based on the collected pieces of evidence. In cloud computing marketplaces, consumers have to make sure that they interact with the cloud providers who transparently publish their

security capabilities. Using the CSA CAIQ framework, cloud providers asserts the existence of security capabilities of the services they offer. The assertions are the pieces of evidence, which cloud providers provide based on their self-assessment. Unlike interaction based evidence units, the self-assessment based evidence units are not potentially infinite and cannot be classified as positive or negative evidence only. In order to deal with these special cases, evidence-based trust mechanism, i.e., *CertainTrust* representational model, needs to adapt on the basis of self-assessment based trustworthiness evaluation (cf. Section 6.1). Furthermore, potential consumers might want to evaluate the trustworthiness of cloud providers according to their selected domains or all domains as per the CAIQ specification. In this case, trustworthiness evaluation mechanism should be able to aggregate the provider-supplied assertions for calculating a trustworthiness value considering the domains that consumers are interested of.

### 4.4.3 Cyber-physical Service Marketplaces

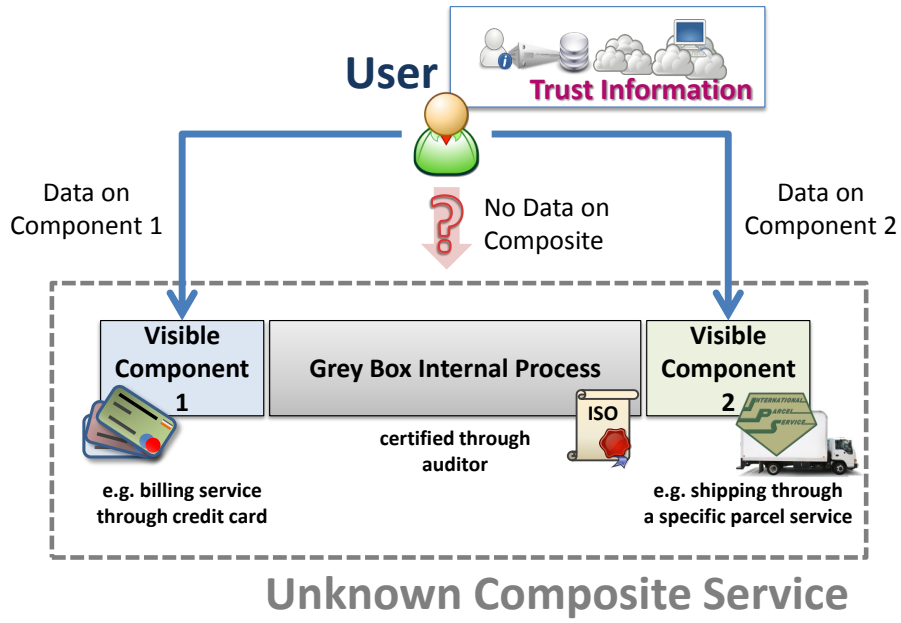



Figure 4.9: Cyber-physical Service Use Case

Cyber-physical service is defined as a service that involves virtual (i.e., internet-based) and physical (i.e., real-world) processes. For instance, ordering a product online and delivery of the product via postal shipping is a cyber-physical service. An ideal example of cyber-physical service marketplaces are eBay, Amazon, etc. The underlying trust systems in these

marketplaces only consider customer recommendation to calculate reputation of a seller (i.e., service provider) and these systems are usually termed as reputation-based trust systems [JIB07]. Moreover, it can be the case that a customer does not have any prior experience with a seller and the customer are not willing to depend on the recommenders or reliable recommenders are absent. In this setting, trust on the sellers can be established from other cues [HVHM12].

The cyber-physical services are generally not monolithic. These services are composed of multiple subsystems and components where some of the components are visible to the customer and associated with the entities that customers have interacted with before. Figure 4.9 illustrates a scenario where a customer would like to provision a composite service from an unknown provider. By necessity, multiple components of the service is visible to the customer, e.g., billing and shipping providers considered by the unknown provider. The billing and shipping components are 'visible' means that the customer are aware about the related providers by means of past experiences or assurances. For the core service (i.e., Grey box), the service provider does not reveal its internal processes to the customer directly. However, a certified auditor may certify the internal processes and issue a certificate (e.g., SAS 70 II), which might be an indicator of trustworthiness of the service provider.

Similar to the cloud marketplace scenario, customers may receive information regarding the unknown provider from multiple sources. Given the nature of the service composition in Figure 4.9, our proposed framework is able to analyse and represent the composite service architecture by means of PLTs. Obviously, the representation depends on how the internal processes are structured and which service-specific attribute are considered important by the customer.

## 4.5 Summary

The formal framework proposed in this chapter overcomes the limitations in the state-of-the-art for trust establishment in distributed service environments. The formal framework is designed to cover the following aspects which are not addressed by existing approaches:

- The proposed framework is able to assess the trustworthiness of a composite distributed service and a system considering the trustworthiness of constituent sub-systems and components. This approach is independent from how the associated opinions are assessed regarding different subsystems or components.

- The same framework is adapted to assess the trustworthiness of a service provider based on their published service-specific attributes.

This approach is also independent from how the opinions about the attributes are assessed.

# 5

# Novel Trust Establishment Mechanisms

The previous chapter has demonstrated a formal approach to analyse composite services regarding their service-specific composite attributes and represent them using Propositional Logic Terms (PLTs). In order to evaluate the PLTs, this chapter first provides the definitions of the logical operators that combine opinions associated with propositions. Then, definitions of the non-standard FUSION operators are provided to deal with conflicting opinions associated with propositions. Finally, a novel architecture for Trust Management (TM) system, which combines the formal framework (cf. Chapter 4) with computational trust operators, is proposed to mechanise trust establishment in cloud computing marketplaces.

In the previous chapter (cf. Section 4.3.3), we have stated that the value associated with a proposition can be termed as *opinion*. In a practical setting, an opinion resembles the formal notion of *information* about an entity's expected behaviour (cf. Section 4.2.1) regarding an attribute (cf. Section 4.2.1 and Section 4.2.2), which can be derived from multiple sources. Hence, on an abstract level, an opinion is associated with a proposition and is based on available pieces of evidence derived using different types of tentative measurements as discussed in Section 4.3.3. In this thesis, an opinion is formally represented using CertainTrust representational model, i.e., opinion $o = (t, c, f)$. The CertainTrust model is discussed in Section 2.2.2.2. In the next section, we introduce a novel extension of CertainTrust, *CertainLogic* that operates on opinions about an entity's expected behaviour regarding service-specific attributes.

## 5.1 CertainLogic: A Framework for Trustworthiness Evaluation

*CertainLogic* framework contains logical operators along with non-standard operators that operates on opinions represented by CertainTrust model. The logical operators are defined to combine opinions associated with propositions that are independent and the non-standard (FUSION) operators are defined to combine opinions associated with dependent propositions.

**Operators for Independent Propositions:** According to Section 4.2.1, in order to model the trustworthiness of a service or service provider, one can logically model the relevant attributes in the form of propositions and combine them using propositional logic. Precisely, the opinions on the fulfilment of those propositions are combined. As long as the propositions are considered to be independent, the logical operators ($AND$, $OR$, $NOT$) are sufficient. However, when the independence cannot be assumed (i.e., dependent propositions) those operators are no longer sufficient. For instance, this is the case when one has to combine two opinions based on the same observation made by different sources about the expected behaviour of an entity regarding two different attributes. An example regarding independent propositions and their relation to opinions are visualised in Figure 5.1.

**Operators for Dependent Propositions:** The dependency among propositions as well as opinions needs further discussion. For example, a consumer wants to know whether a service provider behaves as expected regarding an attribute or a set of attributes; the consumer can derive opinions about the service provider from different sources. If these sources, e.g., providers, consumers, accreditators, and experts, observe the behaviour regarding same attributes using similar methods and their estimates are equal, it is enough to take only one of the estimates into account. However, the sources may miss or misinterpret certain pieces of evidence collected from the same amount of observations, which can produce varying resulting opinions. Thus, while the individual opinions about the propositions vary from source to source, they are still dependent. The non-standard operator, e.g., FUSION, can be used to aggregate the opinions about dependent propositions. An example regarding dependent propositions and their relation to opinions are visualised in Figure 5.1.

**CertainTrust Opinion Representation Revisited:** According to CertainTrust definition (cf. Def. 2.2.2), an opinion $o$ is defined as a triple of values, $o = (t, c, f) \in \{[0,1] \times [0,1] \times [0,1]\}$, where $t$ denotes the *average rating*, $c$ denotes the *certainty* associated with the average rating, and $f$ denotes the *initial expectation* assigned to the truth of the proposition. Each

opinion $o = (t, c, f)$ is also associated with a expectation value, i.e., a point estimate, taking into account the initial expectation $f$, the average rating $t$, and the certainty $c$ as follows:

$$E(t, c, f) = t * c + (1 - c) * f \tag{5.1}$$

Thus, the expectation value shifts from the initial expectation value $f$ to the average rating $t$ with increasing certainty $c$.

Beyond providing means for explicitly modelling uncertainty, CertainTrust also provides a graphical representation (HTI), which supports an intuitive access for users (cf. Section 2.2.2.2).

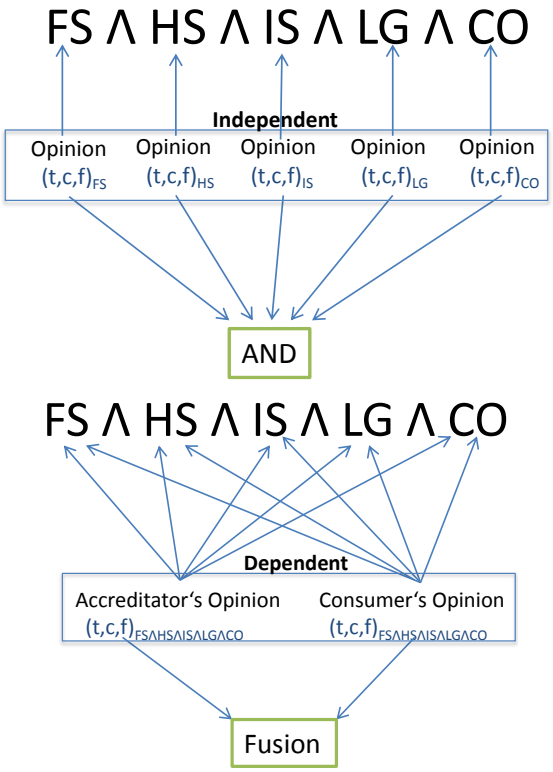



Figure 5.1: Visualisation of Independent and Dependent Propositions

**Assessment of Opinion:** The parameters for an opinion $o = (t, c, f)$ can be assessed in multiple ways [RHMV11b].

- Direct assessment: They can be assessed directly, e.g., based on the opinion of an expert, who estimates initial expectation value $f$ based on her overall knowledge of the context, the average rating $t$ is derived from the available pieces of evidence, and the certainty $c$ expresses the expert's confidence in the representativeness of the average rating.

- Derive from direct experience and recommendations: They can be derived from a trust system that considers one's past experience and recommendations from third parties. Particularly, CertainLogic can directly be applied to trust values derived using Bayesian trust models, e.g., [JI02, BLB04, TPJL06, RH08]. Furthermore, considering that those models provide operators for *discounting*, those models can be leveraged to increase the uncertainty when the information about the truth of a statement is received from a source that is not fully trusted.

- Subjective Logic: They can be derived from an opinion given in Subjective Logic (cf. Section 2.2.2.1).

- Beta probability distribution: CertainTrust model is an extended Bayesian model as discussed in Section 2.2.2. Thus, the parameters of the CertainTrust opinion can also be derived using Beta probability density function.

## 5.2 CertainLogic Operators for Combining Independent Propositions

In this section, we introduce novel *CertainLogic* operators to combine independent opinions associated with PLTs. In particular, we define the standard operators of propositional logic: $AND$, $OR$, and $NOT$. These operators are based on the previously established model, CertainTrust (cf. Section 2.2.2.2) that serve as a representational model for uncertain probabilities.

### 5.2.1 Definition of the Operators

The rationale behind the definitions of the CertainLogic's standard logical operators demand an analytical discussion. In standard binary logic, logical operators operates on propositions that only consider the values 'TRUE' or 'FALSE' (i.e., 1 or 0 respectively) as input arguments. In standard probabilistic logic, the logical operators operates on propositions that consider the values in the range of $[0, 1]$ (i.e., probabilities) as input arguments. However, logical operators in standard probabilistic approach is not able to consider *uncertainty* (cf. Section 2.2.2) about the probability values. Subjective Logic's logical operators are able to operate on opinions that consider uncertain probabilities as input arguments. Additionally, Subjective Logic's logical operators are generalized version of standard logic operators and probabilistic logic operators.

CertainLogic's logical operators operates on CertainTrust's opinions, which represents uncertain probabilities in a more flexible and simpler manner than the opinion representation in Subjective Logic (SL). Note that both CertainTrust's representation and Subjective Logic's representation

of opinions are *isomorphic* with the mapping provided in [Rie09b]. For detailed discussion on representational model of Subjective Logic's opinion and CertainTrust's opinion, we refer the readers to Section 2.2.2.3. The definitions of the CertainLogic's logical operators are formulated in a way that they are equivalent to the definitions of logical operators in Subjective Logic. This equivalence (cf. Section 5.2.6) serves as an argument for the *justification* and *mathematical validity* of the proposed definitions of our logical operators. Moreover, CertainLogic's logical operators are the special cases of binary logic and probabilistic logic operators which will become evident in Section 5.2.4 and Section 5.2.5.

#### 5.2.1.1 Operator $OR$ ($\vee$)

The operator $OR$ is applicable when opinions about two independent propositions need to form a new opinion reflecting the degree of truth for at least one out of two propositions.

**Definition 5.2.1 (Operator *OR*)** *Let $A$ and $B$ be two independent propositions and the opinions about the truth of these propositions be given as $o_A = (t_A, c_A, f_A)$ and $o_B = (t_B, c_B, f_B)$, respectively. Then, the resulting opinion is denoted as $o_{A\vee B} = (t_{A\vee B}, c_{A\vee B}, f_{A\vee B})$ where $t_{A\vee B}$, $c_{A\vee B}$, and $f_{A\vee B}$ are defined in Table 5.1 (OR). We use the symbol '$\vee$' to designate the operator OR and we define $o_{A\vee B} \equiv o_A \vee o_B$.*

The aggregation (using $OR$ operator) of opinions about independent propositions $A$ and $B$ are formulated in a way that the resulting initial expectation ($f$) are dependent on the initial expectation values, $f_A$ and $f_B$ assigned to $A$ and $B$ respectively. Following the equivalent definitions of Subjective Logic's normal disjunction operator and the basic characteristics of the same operator ($\vee$) in standard probabilistic logic, we define $f_{A\vee B} = f_A + f_B - f_A f_B$. The definitions for $c_{A\vee B}$ and $t_{A\vee B}$ are formulated in similar manner and the corresponding adjustments in the definitions are made to maintain the equivalence between the operators of Subjective Logic and CertainLogic.

#### 5.2.1.2 Operator $AND$ ($\wedge$)

The operator $AND$ is applicable when opinions for two independent propositions need to be aggregated to produce a new opinion reflecting the degree of truth of both propositions simultaneously.

**Definition 5.2.2 (Operator *AND*)** *Let $A$ and $B$ be two independent propositions and the opinions about the truth of these propositions be given as $o_A = (t_A, c_A, f_A)$ and $o_B = (t_B, c_B, f_B)$, respectively. Then, the resulting opinion is denoted as*

*$o_{A\wedge B} = (t_{A\wedge B}, c_{A\wedge B}, f_{A\wedge B})$ where $t_{A\wedge B}$, $c_{A\wedge B}$, and $f_{A\wedge B}$ are defined in Table 5.1 (AND). We use the symbol $'\wedge'$ to designate the operator AND and we define $o_{A\wedge B} \equiv o_A \wedge o_B$.*

The aggregation (using $AND$ operator) of opinions about independent propositions $A$ and $B$ are formulated in a way that the resulting initial expectation ($f$) are dependent on the initial expectation values, $f_A$ and $f_B$ assigned to $A$ and $B$ respectively. Following the equivalent definitions of Subjective Logic's normal conjunction operator and basic characteristics of the same operator ($\wedge$) in standard probabilistic logic, we define $f_{A\wedge B} = f_A f_B$. The definitions for $c_{A\wedge B}$ and $t_{A\wedge B}$ are formulated in similar manner and the corresponding adjustments in the definitions are made to maintain the equivalence between the operators of Subjective Logic and CertainLogic.

#### 5.2.1.3 Operator $NOT$ ($\neg$)

The operator $NOT$ is applicable when an opinion about an proposition needs to be negated.

**Definition 5.2.3 (Operator *NOT*)**
*Let $A$ be a proposition and the opinion about the truth of this proposition be given as $o_A = (t_A, c_A, f_A)$. Then, the resulting opinion is denoted as $\neg o_A = (t_{\neg A}, c_{\neg A}, f_{\neg A})$ where $t_{\neg A}$, $c_{\neg A}$, and $f_{\neg A}$ are given in Table 5.1 (NOT). We use the symbol $'\neg'$ to designate the operator NOT and we define, $o_{\neg A} \equiv \neg o_A$*

The negation of an opinion about proposition $A$ represents the opinion about $A$ being false. The definition of the $NOT$ ($\neg$) operator corresponds to $NOT$ operator in standard logic, i.e., binary logic.

Table 5.1: Definition of the operators

| *OR* |
|---|
| $c_{A\vee B} = \begin{cases} c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A)+(1-c_A)c_B f_A(1-t_B)}{f_A+f_B-f_A f_B} & \text{if } f_A f_B \neq 0, \\ \text{“undefined” else.} \end{cases}$ |
| $t_{A\vee B} = \begin{cases} \frac{1}{c_{A\vee B}}\left(c_A t_A + c_B t_B - c_A c_B t_A t_B\right) & \text{if } c_{A\vee B} \neq 0, \\ 0.5 \text{ else.} \end{cases}$ |
| $f_{A\vee B} = f_A + f_B - f_A f_B$ |
| *AND* |
| $c_{A\wedge B} = \begin{cases} c_A + c_B - c_A c_B - \frac{(1-c_A)c_B(1-f_A)t_B+c_A(1-c_B)(1-f_B)t_A}{1-f_A f_B} & \text{if } f_A f_B \neq 1, \\ \text{“undefined” else.} \end{cases}$ |
| $t_{A\wedge B} = \begin{cases} \frac{1}{c_{A\wedge B}}\left(c_A c_B t_A t_B + \frac{c_A(1-c_B)(1-f_A)f_B t_A+(1-c_A)c_B f_A(1-f_B)t_B}{1-f_A f_B}\right) & \text{if } c_{A\wedge B} \neq 0, \text{if } f_A f_B \neq 1, \\ 0.5 \text{ else.} \end{cases}$ |
| $f_{A\wedge B} = f_A f_B$ |
| *NOT* |
| $t_{\neg A} = 1 - t_A$, $c_{\neg A} = c_A$, and $f_{\neg A} = 1 - f_A$ |

### 5.2.2 Properties of the Operators

The operators for $AND$ and $OR$ can be shown commutative and associative. Both of the properties are desirable for the evaluation of propositional logic terms.

**Theorem 5.2.1 (Commutativity)**
*It holds* $o_{A\wedge B} = o_{B\wedge A}$ *and* $o_{A\vee B} = o_{B\vee A}$

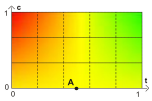
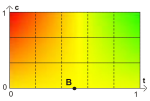
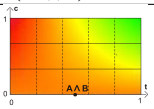
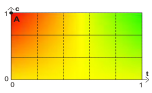
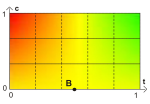
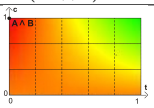
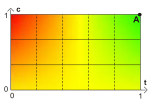
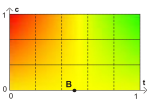
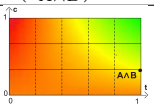
**Theorem 5.2.2 (Associativity)**
*It holds* $o_{A\wedge(B\wedge C)} = o_{(A\wedge B)\wedge C}$ *and* $o_{A\vee(B\vee C)} = o_{(A\vee B)\vee C}$.

The proofs are given in Appendices C.1, C.2, and C.3.

The operators are *not distributive*, i.e., it holds that $o_{A\wedge(B\vee C)} \neq o_{(A\wedge B)} \vee o_{(A\wedge C)}$, as $A\wedge B$ and $A\wedge C$ represents partially dependent propositions. Note that the evaluation of propositional operators in the standard probabilistic approach does not satisfy *distributivity* for the same reason.

The operators does not conform to *Idempotent* property. Conforming to this property means that the propositions $A$ and $B$ are identical which appears counter-intuitive for the assumption that is made while defining the operators. It must be always assumed that the the propositions as well as their input opinions are *independent*.

Table 5.2: Examples for the operator $AND$

| $o_A = (t_A, c_A, f_A)$ | $o_B = (t_B, c_B, f_B)$ | $o_{A \wedge B}$ |
|---|---|---|
| (0.5,0,0.5) | (0.5, 0, 0.5) | (0.5, 0, 0.25) |
| $E(o_A) = 0.5$ | $E(o_B) = 0.5$ | $E(o_{A \wedge B}) = 0.25$ |
|  |  |  |
| (0,1,0.5) | (0.5, 0, 0.5) | (0, 1, 0.25) |
| $E(o_A) = 0$ | $E(o_B) = 0.5$ | $E(o_{A \wedge B}) = 0$ |
|  |  |  |
| (1,1,0.5) | (0.5, 0, 0.5) | (1, 0.333, 0.25) |
| $E(o_A) = 1$ | $E(o_B) = 0.5$ | $E(o_{A \wedge B}) = 0.5$ |
|  |  |  |

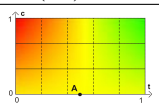
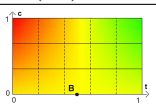
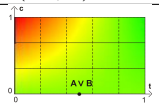
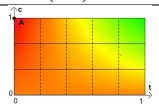
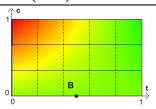
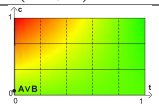
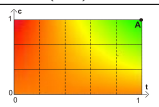
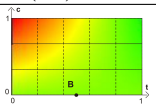
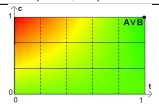
### 5.2.3 Examples

In the following, we present some examples to demonstrate the impact of the newly defined operators on opinions modeled in CertainLogic. Table 5.2 presents three examples for the $AND$ operator and Table 5.3 demonstrates another 3 examples for the $OR$ operator. For each example, we consider the opinions according to the CertainTrust representation and additionally we provide the expectation value ($E$, according to the Equation 5.1) and the graphical representation (HTI) of an opinion.

The color-gradient in the HTI indicates the expectation value ($E$) of each point in the interface. Therefore, the color of each point in the interface is calculated as a linear combination of the RGB-vectors of red ($E = 0$), yellow ($E = 0.5$), and green ($E = 1$). Note that the visualization of opinions along with the calculation of color-gradient in the background are generated using a Java application. The examples are basically screen shots from that application. In the first row, one can see how the $AND$ operator affects the initial expectation $f$. The initial expectation for propositions $A$ and $B$, it holds $f_A = f_B = 0.5$, it holds $f_{A \wedge B} = 0.25$, as $A$ and $B$ have to be true simultaneously. This is directly reflected by the color-gradient of HTI. As the certainty of $c_A = c_B = 0$, i.e., no evidence available, the certainty of the resulting opinion is $c_{A \wedge B} = 0$, and the expectation value ($E$) of each opinion is equivalent to the initial expectation, $f$.

In the second row, we provide an example where one is certain ($c_A = 1$) that proposition $A$ is false ($t_A = 0$) and for $B$ it holds $c_B = 0$, i.e., complete

Table 5.3: Examples for the operator $OR$

| $o_A = (t_A, c_A, f_A)$ | $o_B = (t_B, c_B, f_B)$ | $o_{A \vee B}$ |
|---|---|---|
| (0.5,0,0.5) | (0.5, 0, 0.5) | (0.5, 0, 0.75) |
| $E(o_A) = 0.5$ | $E(o_B) = 0.5$ | $E(o_{A \vee B}) = 0.75$ |
|  |  |  |
| (0,1,0.25) | (0.5, 0, 0.75) | (0, 0.0769, 0.8125) |
| $E(o_A) = 0$ | $E(o_B) = 0.75$ | $E(o_{A \vee B}) = 0.75$ |
|  |  |  |
| (1,1,0.25) | (0.5, 0, 0.75) | (1, 1, 0.8125) |
| $E(o_A) = 1$ | $E(o_B) = 0.75$ | $E(o_{A \vee B}) = 1$ |
|  |  |  |

uncertainty. The certainty of the resulting opinion is $c_{A \wedge B} = 1$ and the rating is $t = 0$. In this case, the knowledge about $A$ is sufficient to conclude that $A \wedge B$ is false.

However, the third row shows that if one is certain ($c_A = 1$) that proposition $A$ is true ($t_A = 1$) and for $B$ it holds $c_B = 0$, i.e., complete uncertainty, then, the certainty of the resulting opinion is only $c_{A \wedge B} = 0.33$. The resulting opinion demonstrates that even if the proposition $A$ is certainly true, it is not sufficient for the $AND$ operator to conclude that both propositions, $A$ and $B$, are true in composition.

The examples for the $OR$ operator follow a similar reasoning. In the first row, one can immediately see how the initial expectation value is influenced by the $OR$ operator; the resulting opinion's color-gradient is more 'greenish'. This is reasonable as the initial expectation value of the $A$ and $B$ are $f_A = f_B = 0.5$ and the resulting opinions expectation value is $f_{A \vee B} = 0.75$, as the chances that $A \vee B$ is true are higher than the chances for only $A$ or only $B$.

In the second row, the opinion, $(0, 1, 0.25)$, about proposition $A$ demonstrates that the proposition is false, whereas the opinion, $(0.5, 0, 0.75)$, about proposition $B$ shows that a source is completely uncertain about the truth of the proposition. In this case, the resulting opinion's certainty value, 0.0769, indicates that the propositions in composition are unlikely to be false because of the opinion about the proposition $B$ is based on zero pieces of evidence, i.e., no evidence available. Hence, high initial expectation value, i.e., $f_B = 0.75$,

of $o_B$ influence the expectation value as well as the initial expectation value, $E(o_{A\vee B}) = 0.75$; $f_{A\vee B} = 0.8125$, of propositions in composition. It interprets that the chances of $A \vee B$ is false are higher than the chances for only $A$ or equal to $B$ but with higher certainty value (0.0769).

In the third row, the opinion, $(1, 1, 0.25)$, about proposition $A$ is sufficient to conclude that the propositions in composition is true. The reason behind the conclusion is that the opinion about $A$ is based on maximum or infinite number of positive pieces of evidence and for the $OR$ operator this is a sufficient condition.

### 5.2.4 Equivalence: CertainLogic to Standard Logic

Standard logic is a special case of CertainLogic for the logical operators, $AND$, $OR$, and $NOT$. It means that when a corresponding operator is available in binary logic, and the input arguments are equivalent to TRUE (1) or FALSE (0) of binary logic, then the resulting opinion is equivalent to the result that binary logic definition will produce. The truth table (cf. Appendix B) is generated based on the corresponding definitions of CertainLogic operators that are applied to opinions with binary values, i.e., 1 and 0.

### 5.2.5 Equivalence: CertainLogic to Standard Probabilistic Logic

In the standard probabilistic logic, the operation for $AND$ is usually defined as $p(A \wedge B) = p(A)p(B)$, the operation for $OR$ is given as $p(A \vee B) = p(A) + p(B) - p(A)p(B)$, and the operation for $NOT$ is given as $p(\neg A) = 1 - p(A)$.

The expectation value $E(t, c, f)$ of an opinion can be interpreted as the probability for the truth of a proposition and it can be shown that the following statements are true:

**Theorem 5.2.3 (Equivalence)**
*The propositional logic operators for AND, OR, and NOT as defined in Table 5.1 are compliant with the standard probabilistic evaluation of propositional terms as it holds:*

1. $E(o_{A\wedge B}) = E(o_A)E(o_B)$ *(for AND)*
2. $E(o_{A\vee B}) = E(o_A) + E(o_B) - E(o_A)E(o_B)$ *(for OR)*
3. $E(o_{\neg A}) = 1 - E(o_A)$ *(for NOT)*

The proof is given in Appendix C.4.

Although, the standard probabilistic approach is compliant with CertainLogic, there are multiple advantages when combining opinions with CertainLogic:

- Our model can express the uncertainty, which is not possible in the standard probabilistic approach. This is important, because in real world scenarios probabilities are usually not known, but have to be estimated or derived from experiments, and thus they are subject to uncertainty.
- Our model does not only take the (un-)certainty as an input parameter, but also reflects the uncertainty in the calculated result. Thus, the certainty is a good indicator for the confidence associated to that result.

### 5.2.6 Equivalence: CertainLogic to Subjective Logic

*Subjective Logic (SL)* opinion model has been described in 2.2.2.1, and it combines elements from belief theory with Bayesian probabilities. In the following, we show that the logical operators of *CertainLogic* are equivalent to those of *SL*, which provides the argument for the mathematical validity and justification of our approach.

**Definition 5.2.4 (Belief representation (SL))**
*According to [Jøs01], an opinion is given by* $\omega = (b, d, u, a)$*, where* $b$ *models the* belief*,* $d$ *the* disbelief*,* $u$ *the* uncertainty*, and* $a$ *the atomicity.*

The mapping between CertainTrust (CT) and *SL* opinions is provided in [Rie09b]. The mapping of an opinion in *CT* to *SL* is denoted as a function $m_{SL}^{CT}$,which is defined below:

**Definition 5.2.5 (Mapping CT to SL)**
*The mapping from an opinion* $o = (t, c, f)$ *in* CT *to an opinion* $\omega = (b, d, u, a)$ *in* SL *is denoted as* $(b, d, u, a) = m_{SL}^{CT}(t, c, f)$ *and defined by* $b = t{*}c$*,* $d = (1 - t) * c$*,* $u = 1 - c$*, and* $a = f$*.*

The inverse mapping can be given as follows:

**Definition 5.2.6 (Mapping SL to CT)**
*The mapping from an opinion* $\omega = (b, d, u, a)$ *in* SL *to an opinion* $o = (t, c, f)$ *in* CT *is denoted as* $(t, c, f) = m_{CT}^{SL}(b, d, u, a)$ *and defined by* $c = 1-u$*,* $a = f$*, and* $t = \frac{b}{b+d}$ *for* $b + d \neq 0$*, else* $t = 0.5$*.*

**Theorem 5.2.4 (Equivalence of operators)** *Let* $A$ *and* $B$ *be independent propositions. In* SL*,* $\omega_A$ *and* $\omega_B$ *are two opinions about proposition* $A$ *and* $B$*, respectively. Using the mapping functions, i.e.,* $m_{SL}^{CT}$ *and* $m_{CT}^{SL}$*, our operators are equivalent to the* normalized versions *of* $AND$*,* $OR$*,* $NOT$ *operators of SL in [JM05]. This means that for* $op \in \{AND, OR\}$ *the first two statements and for* $NOT$ *the last two statements given in the following holds.*

1. $o_A = m_{CT}^{SL}(\omega_A)$ *and* $o_B = m_{CT}^{SL}(\omega_B) \Rightarrow \omega_{AopB} = m_{SL}^{CT}(o_{AopB})$

2. $\omega_A = m_{SL}^{CT}(o_A)$ *and* $\omega_B = m_{SL}^{CT}(o_B) \Rightarrow o_{AopB} = m_{CT}^{SL}(\omega_{AopB})$

3. $o_A = m_{CT}^{SL}(\omega_A) \Rightarrow \omega_{\neg A} = m_{SL}^{CT}(o_{\neg A})$

4. $\omega_A = m_{SL}^{CT}(o_A) \Rightarrow o_{\neg A} = m_{CT}^{SL}(\omega_{\neg A})$

The proof is given in Appendix C.5. As the mapping between opinions in *CT* and *SL* is bijective, this basically means that it is possible to switch between the representations as well as the logical operators. Although *SL* and *CT* provide capabilities for reasoning under uncertainty, there are several advantages that CertainLogic inherits from the CertainTrust representational model, which are explained in Section 2.2.2.3.

Finally, based on the mapping between CT and SL, and following the arguments provided in [JM05], the operators for $AND$ and $OR$ calculate the same expectation values as when doing the operation on Beta probability density functions. However, the variance in this case is not exact, but well approximated.

## 5.3 CertainLogic Operators for Dependent Propositions

In this section, we provide the definitions for the CertainLogic non-standard operators that operates on dependent opinions associated with *PLTs*. The non-standard operators can be used to fuse dependent opinions about a proposition from multiple sources. A detail discussion on dependent opinions and their relation to *PLTs* are provided in Section 5.1.

### 5.3.1 Definition of the Operators

We provide definitions for three types of fusion operators, i.e., operators that are suitable for aggregating dependent opinions on a proposition. At first, we introduce the *average fusion* operator. This operator is equivalent (cf. Appendix D.9) to the *averaging fusion* operator [Jøs09] and *consensus operator for dependent opinions* [JMP06] defined in Jøsang's Subjective Logic. The equivalence serves as an argument for the *justification* and the *mathematical validity* of our *average fusion* operator that we use as a starting point for introducing a novel fusion operator. The novel operator (i.e., conflict-aware fusion) is capable of dealing with conflict as well as preferences (as weights). Note that the *weighted fusion* operator is an intermediate step towards defining the novel *conflict-aware fusion* operator. Our *weighted fusion* operator should not be confused with the fusion operator that has been recently proposed by Zhou et al. [ZSLL11], as they consider two weights in their definition: one weight from the agent who provide the opinion and other weight from the agent who fuse the weighted opinions.

#### 5.3.1.1 Average Fusion Operator

Assume that multiple sources providing $n$ opinions about a proposition. The opinions are based on identical pieces of evidence and thus, derived opinions are dependent. Jøsang [Jøs09] argued that even if the opinions are based on identical pieces of evidence, the opinions might vary due to interpretation methods that considered by different sources. Thus, Jøsang proposed to apply *averaging rule* on the dependent opinions, which is also the rationale behind the definition of the CertainLogic average fusion ($A.FUSION$) operator.

Table 5.4: Definition of the Average Fusion Operator

$$t_{\widehat{\oplus}(A_1,A_2,\ldots,A_n)} = \begin{cases} \dfrac{\sum_{i=1}^{n} t_{A_i}}{n} & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ 0.5 & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0 \ , \\ \dfrac{\sum_{i=1}^{n}(c_{A_i} t_{A_i} \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n}(c_{A_i} \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} & \text{if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$c_{\widehat{\oplus}(A_1,A_2,\ldots,A_n)} = \begin{cases} 1 & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ \dfrac{\sum_{i=1}^{n}(c_{A_i} \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n}( \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} & \text{if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$f_{\widehat{\oplus}(A_1,A_2,\ldots,A_n)} = \frac{\sum_{i=1}^{n} f_{A_i}}{n}$$

**Definition 5.3.1 (*A.FUSION*)**

*Let $A$ be a proposition and let $o_{A_1} = (t_{A_1}, c_{A_1}, f_{A_1})$, $o_{A_2} = (t_{A_2}, c_{A_2}, f_{A_2}),\cdots,$ $o_{A_n} = (t_{A_n}, c_{A_n}, f_{A_n})$ be $n$ opinions associated to $A$. The **average fusion** is denoted as*

$$o_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)} = (t_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)}, c_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)}, f_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)})$$

*where $t_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)}$, $c_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)}$, $f_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)}$ are defined in Table 5.4. We use the symbol $(\widehat{\oplus})$ to designate the operator $A.FUSION$ and we define $o_{\widehat{\oplus}(A_1,A_2,\cdots,A_n)} \equiv \widehat{\oplus}((o_{A_1}), (o_{A_2}), \cdots, (o_{A_n}))$.*

Table 5.5: Definition of the Weighted Fusion Operator

$$t_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)} = \begin{cases} \dfrac{\sum_{i=1}^{n} w_i t_{A_i}}{\sum_{i=1}^{n} w_i} & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ 0.5 & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0 \ , \\ \dfrac{\sum_{i=1}^{n} (c_{A_i} t_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n} (c_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} & \text{if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$c_{\widehat{\oplus}_w(A_1,A_2,\ldots,A_n)} = \begin{cases} 1 & \text{if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ \dfrac{\sum_{i=1}^{n} (c_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n} (w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} & \text{if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$f_{\widehat{\oplus}_w(A_1,A_2,\ldots,A_n)} = \frac{\sum_{i=1}^{n} w_i f_{A_i}}{\sum_{i=1}^{n} w_i}$$

#### 5.3.1.2 Weighted Fusion Operator

Assume that consumers want to attach preferences with opinions while fusing the opinions derived from multiple sources. The preferential attachment might depend on consumers' personal preferences when selecting the sources, e.g., providers, experts, consumers, for deriving opinions. In order to extend the functionality of the state-of-the-art *A.FUSION* operator to deal with trustor's preferential attachments, we propose a novel operator, weighted fusion (*W.FUSION*), that operates on opinions associated with weights.

**Definition 5.3.2 (*W.FUSION*)**

*Let $A$ be a proposition and let $o_{A_1} = (t_{A_1}, c_{A_1}, f_{A_1})$, $o_{A_2} = (t_{A_2}, c_{A_2}, f_{A_2}), \cdots$, $o_{A_n} = (t_{A_n}, c_{A_n}, f_{A_n})$ be $n$ opinions associated to $A$. Furthermore, the weights $w_1$, $w_2$,$\cdots$, $w_n$ (with $w_1, w_2, \cdots, w_n \in \mathbb{R}_0^+$ and $w_1 + w_2 + \cdots + w_n \neq 0$) are assigned to the opinions $o_{A_1}$, $o_{A_2}$,$\cdots$, $o_{A_n}$, respectively. The **weighted fusion** is denoted as*

$$o_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)} = (t_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)}, c_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)}, f_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)})$$

*where $t_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)}$, $c_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)}$, $f_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)}$ are defined in Table 5.5. We use the symbol ($\widehat{\oplus}_w$) to designate the operator W.FUSION and*

*we define* $o_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)} \equiv \widehat{\oplus}_w((o_{A_1},w_1),(o_{A_2},w_2),\cdots,(o_{A_n},w_n))$.

#### 5.3.1.3 Conflict-aware Fusion Operator

Assume that consumers want to fuse conflicting opinions derived from multiple sources. In order to extend the functionality of the state-of-the-art $A.FUSION$ operator in terms of dealing with conflicting opinions, we propose a novel operator, conflict-aware fusion ($C.FUSION$), that operates on conflicting opinions and reflect the calculated degree of conflict ($DoC$) in the resulting fused opinion. Note that this operator is able to deal with preferential weights associated with opinions.

**Definition 5.3.3 (*C.FUSION*)**

*Let $A$ be a proposition and let $o_{A_1} = (t_{A_1}, c_{A_1}, f_{A_1})$, $o_{A_2} = (t_{A_2}, c_{A_2}, f_{A_2})$,$\cdots$, $o_{A_n} = (t_{A_n}, c_{A_n}, f_{A_n})$ be $n$ opinions associated to $A$. Furthermore, the weights $w_1$, $w_2$,$\cdots$, $w_n$ (with $w_1, w_2, \cdots, w_n \in \mathbb{R}_0^+$ and $w_1 + w_2 + \cdots + w_n \neq 0$) are assigned to the opinions $o_{A_1}$, $o_{A_2}$,$\cdots$, $o_{A_n}$, respectively. The **conflict-aware fusion** is denoted as*

$$o_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)} = ((t_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}, c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}, f_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}), DoC)$$

*where $t_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$, $c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$, $f_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$, the degree of conflict $DoC$ are defined in Table 5.6. We use the symbol ($\widehat{\oplus}_c$) to designate the operator $C.FUSION$ and we define $o_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)} \equiv \widehat{\oplus}_c((o_{A_1},w_1),(o_{A_2},w_2),\cdots,(o_{A_n},w_n))$.*

The rationale behind the definition of the *conflict-aware* fusion demands an extensive discussion. The basic concept of this operator is that the operator extends the *weighted fusion* by calculating the degree of conflict ($DoC$) between a pair of opinions. Then, the value of $(1 - DoC)$ is multiplied with the certainty ($c$) that would be calculated by the weighted fusion (the parameters for $t$ and $f$ are the same as in the weighted fusion).

Now, we discuss the calculation of the $DoC$ for two opinions. For the parameter, it holds $DoC \in [0, 1]$. This parameter depends on the average ratings ($t$), the certainty values ($c$), and the weights ($w$). The weights are assumed to be selected by the trustors (consumers) and the purpose of the weights is to model the preferences of the trustor when aggregating opinions from different sources. We assume that the compliance of their preferences are ensured under a policy negotiation phase. For example, users might be given three choices, High (2), Low (1) and No preference (0, i.e., opinion from a particular source is not considered), to express their preferences on selecting the sources that provide the opinions. Note that the weights are not introduced to model the reliability of sources. In this case, it would be appropriate to use the discounting operator [Rie09b, Jøs01] to explicitly consider reliability of sources and apply the fusion operator on the results to influence users' preferences. The values of $DoC$ can be interpreted as follows:

Table 5.6: Definition of the Conflict-aware Fusion Operator

$$t_{\widehat{\oplus_c}(A_1,A_2,\cdots,A_n)} = \begin{cases} \dfrac{\sum_{i=1}^{n} w_i t_{A_i}}{\sum_{i=1}^{n} w_i} \text{ if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ 0.5 \text{ if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0 \ , \\ \dfrac{\sum_{i=1}^{n}(c_{A_i} t_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n}(c_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} \text{ if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$c_{\widehat{\oplus_c}(A_1,A_2,\cdots,A_n)} = \begin{cases} 1 * (1 - DoC) \text{ if } c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1 \ , \\ \dfrac{\sum_{i=1}^{n}(c_{A_i} w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))}{\sum_{i=1}^{n}(w_i \prod_{j=1,\ j\neq i}^{n} (1 - c_{A_j}))} * (1 - DoC) \text{ if } \{c_{A_i}, c_{A_j}\} \neq 1 \ . \end{cases}$$

$$f_{\widehat{\oplus_c}(A_1,A_2,\cdots,A_n)} = \frac{\sum_{i=1}^{n} w_i f_{A_i}}{\sum_{i=1}^{n} w_i}$$

$$DoC = \frac{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} DoC_{A_i,A_j}}{\frac{n(n-1)}{2}}$$

$$DoC_{A_i,A_j} = \left| t_{A_i} - t_{A_j} \right| * c_{A_i} * c_{A_j} * \left( 1 - \left| \frac{w_i - w_j}{w_i + w_j} \right| \right)$$

- **No conflict ($DoC = 0$):** For $DoC = 0$, it holds that there is *no conflict* between the two opinions. This is true if both opinions agree on the average rating, i.e., $t_{A_1} = t_{A_2}$ or in case that at least one opinion has a certainty $c = 0$ (for completeness we have to state that it is also true if one of the weights is equal to 0, which means the opinion is not considered).

- **Total conflict ($DoC = 1$):** For $DoC = 1$, it holds that the two opinions are weighted equally ($w_1 = w_2$) and contradicts each other to a maximum. This means, that both opinions have a maximum certainty ($c_{A_1} = c_{A_2} = 1$) and maximum divergence in the average ratings, i.e., $t_{A_1} = 0$ and $t_{A_2} = 1$ (or $t_{A_1} = 1$ and $t_{A_2} = 0$).

- **Conflict ($DoC \in ]0,1[$):** For $DoC \in ]0,1[$, it holds that there are two opinions contradict each other to a certain degree. This means that the both opinions does not agree on the average ratings, i.e., $t_{A_1} \neq t_{A_2}$, having certainty values other than 0 and 1. The weights can be any real number other than 0.

Next, we argue for integrating the degree of conflict, $DoC$, into the resulting opinion by multiplying the certainty with $(1 - DoC)$. The argument is, in case that there are two (equally weighted) conflicting opinions, then this indicates that the information which these opinions are based on is not representative for the outcome of the assessment or experiment. Thus, for the sake of representativeness, in the case of total conflict (i.e., $DoC = 1$), we reduce the certainty ($c_{(o_{A_1},w_1)\widehat{\oplus}(o_{A_2},w_2)}$) of the resulting opinion by a multiplicative factor, $(1 - DoC)$. The certainty value is 0 in this case.

For $n$ opinions, degree of conflict (i.e., $DoC_{A_i,A_j}$) in Table 5.6 is calculated for each opinion pairs. The challenge is how to calculate the $DoC$ among $n$ opinions to adjust the certainty ($c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$) parameter of the resulting opinion. There are three possible ways that we have considered when calculating the $DoC$. These are as follows:

- One of the ways is to calculate the average of all possible $DoC_{A_i,A_j}$ values of all pairs. For instance, if there are $n$ opinions there can be at most $\frac{n(n-1)}{2}$ pairs and degree of conflict is calculated for each of those pairs individually. Finally, all the pair-wise $DoC$ values are averaged (i.e., averaging $\frac{n(n-1)}{2}$ pairs of $DoC_{A_i,A_j}$) to adjust the certainty (i.e., $c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$) parameter of the resulting opinion (cf. Table 5.6).

- Another way is to calculate the degree of conflict ($DoC$) for each pair of opinions and adjust the certainty ($c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$) $\frac{n(n-1)}{2}$ times if there are $n$ opinions. In this case, we get $\frac{n(n-1)}{2}$ certainty values which are then averaged to calculate the final certainty value.

- The other way is to calculate the degree of conflict ($DoC$) pair-wise and multiply all pair-wise values at once with the certainty ($c_{\widehat{\oplus}_c(A_1,A_2,\cdots,A_n)}$) of the resulting opinion. This approach has two drawbacks: i) it suffers from a multiplicative effect which means that the certainty is affected heavily with the increasing number of opinions, ii) it also heavily affect the certainty in case a single opinion radically conflict with others.

The first two approaches are equally capable of detecting conflicting opinions as the conflict analysis is done pair-wise. Either of these approaches performs better (in detecting conflicting information) than the third approach, particularly in a complex setting where a large collection of sources are available and only one of the sources radically conflicts with the other sources when providing opinions. In this case, either of the first two approaches

shifts half of the uncertainty on the outlier and others receive only $\frac{1}{2n(n-1)}$ of the extra uncertainty. Moreover, the first two approaches do not suffer from the multiplicative effect alike the third approach.

Finally, we see that the connection between $DoC$ and $c_{\widehat{\oplus_c}(A_1,A_2,\cdots,A_n)}$ (certainty) is linear. One can argue that this connection should be handled probabilistically rather than linearly. We choose the linear approach as it is simple, does not lead to unforeseen effects and allow good integration of weights, which is important for service oriented marketplaces. Moreover, due to linearity, *Weight-specific* properties (i.e., Section 5.3.2) hold for *conflict-aware* fusion operator as those properties hold for $W.FUSION$ operator.

In Table 5.4, 5.5 and 5.6, for all opinions if it holds $c_{A_i} = 0$ (complete uncertainty), the expectation values (cf. Definition 2.2.3) depends only on $f$. However, for soundness we define $t_{A_i} = 0.5$ in this case. The discussion of the fusion operators is supported by numerical and graphical (i.e., HTI) examples in Section 5.3.3.

### 5.3.2 Properties of the Operators

The desirable mathematical properties of the defined operators are classified into two types: i) Fusion-specific and ii) Weight-specific. The *Fusion-specific* properties are the ones which are shown desirable and necessary for the state-of-the-art fusion operators [JMP06, Jøs09]. The *Weight-specific* properties are useful to show the relationship among the average, weighted and conflict-aware fusion, that also extend to easier computation of the expectation value $E$ (cf. equation 1) of fused opinions. Moreover, these properties are aligned with the desirable properties for arithmetic mean-based averaging operations [BPC07]. As fusion operation belongs to the family of arithmetic mean-based averaging operations [BPC07], those particular properties are also desirable for our extended fusion operators.The properties that hold for our defined operators are outlined as follows:

1. Fusion-specific Properties: *Idempotency*, *Commutativity* & *Permutability* belong to this group.

2. Weight-specific Properties: *Weight Partitioning*, *Invariance to Weight Scaling* and three properties regarding *Weighted average of expectation value for common weight and/or certainty* belong to this particular group.

The formal theorems regarding the properties are discussed in the following. The proofs for the theorems are provided in Appendix D.1–D.8.

#### Fusion-specific Properties

**Idempotence:** When aggregating the same opinion twice, no additional information is gained for the resulting fused opinion. This should be reflected in the fusion operation by designing it to be idempotent. Formally, the following theorem 5.3.1 thus represents a desirable property of the fusion operator that holds for average, weighted and conflict-aware fusion.

**Theorem 5.3.1 (Idempotence)**
*It holds* $\widehat{\oplus}(o_{A_1}, o_{A_1}, \cdots, o_{A_1}) = o_{A_1}$ *and* $\widehat{\oplus}_w(o_{(A_1,w_1)}, o_{(A_1,w_2)}, \cdots, o_{(A_1,w_n)}) = o_{A_1}$ *and* $\widehat{\oplus}_c(o_{(A_1,w_1)}, o_{(A_1,w_2)}, \cdots, o_{(A_1,w_n)}) = o_{A_1}$.

**Commutativity and Permutability:** In averaging operations, the order of the operands should not affect the final outcome of the calculation. Therefore, the extended fusion operators are designed to be commutative as well as indifferent to a permutation of the operands. This makes them compliant with the following two theorems 5.3.2 and 5.3.3.

**Theorem 5.3.2 (Commutativity)**
*For two opinions, it holds*
$\widehat{\oplus}(o_{A_1}, o_{A_2}) = \widehat{\oplus}(o_{A_2} o_{A_1})$
$\widehat{\oplus}_w((o_{A_1}, w_1), (o_{A_2}, w_2)) = \widehat{\oplus}_w((o_{A_2}, w_2), (o_{A_1}, w_1))$
$\widehat{\oplus}_c((o_{A_1}, w_1), (o_{A_2}, w_2)) = \widehat{\oplus}_c((o_{A_2}, w_2), (o_{A_1}, w_1))$.

**Theorem 5.3.3 (Permutability of $n$ opinions)**
*Let* $\pi : [1, \ldots, n] \mapsto [1, \ldots, n]$ *denote a permutation such that* $o_{A_{\pi(i)}} = o_{A_i}$, *then it holds*
$\widehat{\oplus}_w(o_{A_1}, \cdots, o_{A_n}) = \widehat{\oplus}_w(o_{A_{\pi(1)}}, \cdots, o_{A_{\pi(n)}})$
$\widehat{\oplus}_c(o_{A_1}, \cdots, o_{A_n}) = \widehat{\oplus}_c(o_{A_{\pi(1)}}, \cdots, o_{A_{\pi(n)}})$

In this regard, one can argue that the associativity property is also desirable. But, it is not desirable as the defined operations for the fusion operators belong to the family of arithmetic mean based averaging operation. Note that for general arithmetic mean based averaging operations, associativity is not a desirable property [BPC07].

**Weight-specific Properties** The fusion operators fulfil a number of useful properties regarding the relationship between average and weighted fusion, that also extend to easier computation of the expectation value $E$ (cf. Definition 2.2.3) of fused opinions. In the following, we consider primarily *weighted fusion* with weight $w_i \in \mathbb{R}_0^+$ , $0 < i \leq n$, where $i, n \in \mathbb{N}$ and $\sum_{i=1}^{n} w_i \neq 0$.

**Weighted Fusion Partitioning to Average Fusion:** For rational weights, the weighted fusion operator ($W.FUSION$) and ($C.FUSION$) is isomorphic to the average fusion operator ($A.FUSION$).

**Theorem 5.3.4 (Weight Partitioning)**
*For* $w_1 = \frac{a_1}{b_1}, w_2 = \frac{a_2}{b_2}, \cdots, w_n = \frac{a_n}{b_n} \in \mathbb{Q}_0^+$ *it holds*

1. $\widehat{\oplus}_w((o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n)) =$
$\widehat{\oplus}_w(\underbrace{(o_{A_1}, 1), \cdots, (o_{A_1}, 1)}_{a_1 \cdot \prod_{j \neq 1} b_j \textbf{ times}}, \underbrace{(o_{A_2}, 1), \cdots, (o_{A_2}, 1)}_{a_2 \cdot \prod_{j \neq 2} b_j \textbf{ times}}, \cdots, \underbrace{(o_{A_n}, 1), \cdots, (o_{A_n}, 1)}_{a_n \cdot \prod_{j \neq n} b_j \textbf{ times}}) =$
$\widehat{\oplus}( \underbrace{o_{A_1}, \cdots, o_{A_1}}_{a_1 \cdot \prod_{j \neq 1} b_j \textbf{ times}}, \underbrace{o_{A_2}, \cdots, o_{A_2}}_{a_2 \cdot \prod_{j \neq 2} b_j \textbf{ times}}, \cdots, \underbrace{o_{A_n}, \cdots, o_{A_n}}_{a_n \cdot \prod_{j \neq n} b_j \textbf{ times}} )$

2. $\widehat{\oplus}_c((o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n)) =$
$\widehat{\oplus}_c(\underbrace{(o_{A_1}, 1), \cdots, (o_{A_1}, 1)}_{a_1 \cdot \prod_{j \neq 1} b_j \textbf{ times}}, \underbrace{(o_{A_2}, 1), \cdots, (o_{A_2}, 1)}_{a_2 \cdot \prod_{j \neq 2} b_j \textbf{ times}}, \cdots, \underbrace{(o_{A_n}, 1), \cdots, (o_{A_n}, 1)}_{a_n \cdot \prod_{j \neq n} b_j \textbf{ times}}) =$
$\widehat{\oplus}( \underbrace{o_{A_1}, \cdots, o_{A_1}}_{a_1 \cdot \prod_{j \neq 1} b_j \textbf{ times}}, \underbrace{o_{A_2}, \cdots, o_{A_2}}_{a_2 \cdot \prod_{j \neq 2} b_j \textbf{ times}}, \cdots, \underbrace{o_{A_n}, \cdots, o_{A_n}}_{a_n \cdot \prod_{j \neq n} b_j \textbf{ times}} )$

**Invariance to Weight Scaling:** As a weighted aggregation function, the weighted fusion operation ($W.FUSION$) and conflict-aware fusion operation ($C.FUSION$) is invariant to scaling of its weight terms by a constant.

**Theorem 5.3.5 (Invariance to Weight Scaling)**
$\forall k \neq 0$ *it holds*

1. $\widehat{\oplus}_w((o_{A_1}, w_1 * k), (o_{A_2}, w_2 * k), \cdots, (o_{A_n}, w_n * k)) =$
$\widehat{\oplus}_w((o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n))$

2. $\widehat{\oplus}_c((o_{A_1}, w_1 * k), (o_{A_2}, w_2 * k), \cdots, (o_{A_n}, w_n * k)) =$
$\widehat{\oplus}_c((o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n))$

**Weighted average of expectation value for common weight or common certainty:** The primary decision criterion in CertainTrust [Rie09b] is the expectation value, $E$ (cf. Definition 2.2.3) associated with an opinion $o = (t, c, f)$. Thus, in many cases, the computation of this expectation value is the final objective subsequent to applying the fusion operators. As $E \in \mathbb{R}_0^+$, averaging operations conducted on the expectation values using arithmetic operations, as opposed to the opinions using fusion, are computationally preferable. The following theorems 5.3.6, 5.3.7, 5.3.8 outline under which conditions this is possible.

**Theorem 5.3.6 (Weighted average of $E$: common $w$)**
*For* $w_1 = w_2 = \cdots = w_n = w$ *it holds*
$E(\widehat{\oplus}_w((o_{A_1}, w), (o_{A_2}, w), \cdots, (o_{A_n}, w))) = E(\widehat{\oplus}(o_{A_1}, o_{A_2}, \cdots, o_{A_n})) = \frac{\sum_{i=1}^{n} E(o_{A_i})}{n}$.

**Theorem 5.3.7 (Weighted average of $E$: common $c$)**
*For* $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = c$, *it holds*
$E(\widehat{\oplus}_w((o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n))) = \frac{\sum_{i=1}^{n} w_i E(o_{A_i})}{\sum_{i=1}^{n} w_i}$.

**Weighted average of expectation value for common certainty and common weights:** When defining the *W.FUSION* and *C.FUSION* we made sure that whenever one uses identical certainty values (i.e., $c_{A_1} = c_{A_2}$ or $c_{A_1} = c_{A_2} = \cdots = c_{A_n}$) and weights (i.e., $w_1 = w_2$ or $w_1 = w_2 = \cdots = w_n$), the result is the same as the result of the average fusion (cf. Thm. 5.3.6).

**Theorem 5.3.8 (Weighted average of $E$: common $c$ and $w$)**
*For* $w_1 = w_2 = \cdots = w_n = w$ *and* $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = c$, *it holds*

1. $\frac{\sum_{i=1}^{n} E(o_{A_i})}{n} = E(\widehat{\oplus}_w((o_{A_1}, w), (o_{A_2}, w), \cdots, (o_{A_n}, w)))$

2. $\frac{\sum_{i=1}^{n} E(o_{A_i})}{n} = E(\widehat{\oplus}_c((o_{A_1}, w), (o_{A_2}, w), \cdots, (o_{A_n}, w)))$

### 5.3.3 Examples of the Fusion Operators

In the following, we present some examples showing the impact of the newly defined operators that operates on opinions modelled with *CertainTrust* representational model.

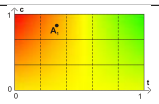
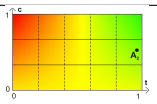
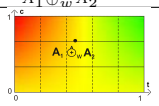
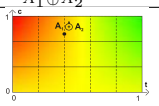
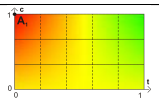
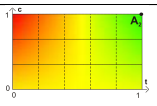
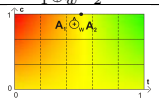
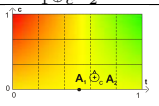
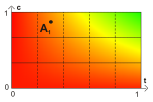
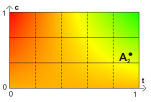
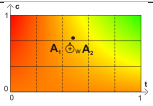
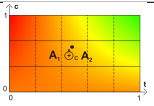
In the left part of Table 5.7, all the examples show the effect of the weighted fusion (*W.FUSION*) operator in different cases. The right part of the Table 5.7 shows the effect of average fusion (example 1) and conflict-aware fusion (example 2 & 3). We have developed a Java application in order to visualise the impact of *fusion* operators on opinions. The examples are basically screen shots from that application.

**Example 1:** The first example in Table 5.7 illustrates a comparison between the *W.FUSION* and *A.FUSION* operators.

While for the *A.FUSION* operator it holds that both opinions have the same impact on the results (which is equivalent to $w_1 = w_2$ in the weighted fusion), the *W.FUSION* operator supports the customization of the weights (in the example we use, $w_1 = 1$ and $w_2 = 2$ for the weighted fusion).

In the resulting opinions, one can observe the influence of the weights. In the *A.FUSION* (right), the resulting opinion $(0.4, 0.75, 0.5)$ is biased to $o_{A_1}$ because of the high certainty (0.833) associated with the opinion

Table 5.7: Examples for the Fusion Operators

| Input Opinions | | Resulting Opinions | |
|---|---|---|---|
| $o_{A_1} = (t_{A_1}, c_{A_1}, f_{A_1})$ | $o_{A_2} = (t_{A_2}, c_{A_2}, f_{A_2})$ | $o_{A_1 \widehat{\oplus} A_2}$ | |
| Example 1 | | | |
| | | *W.FUSION* $w_1 = 1; w_2 = 2$ | *A.FUSION* $w_1 = 1; w_2 = 1$ |
| $(0.3, 0.833, 0.5)$ | $(0.9, 0.5, 0.5)$ | $(0.4717, 0.6996, 0.5)$ | $(0.4, 0.75, 0.5)$ |
| $E(o_{A_1}) = 0.333$ | $E(o_{A_2}) = 0.7$ | $E(o_{A_1 \widehat{\oplus}_w A_2}) = 0.48$ | $E(o_{A_1 \widehat{\oplus} A_2}) = 0.425$ |
|  |  |  |  |
| Example 2 | | | |
| | | *W.FUSION* $w_1 = 1; w_2 = 1$ | *C.FUSION* $w_1 = 1; w_2 = 1$ |
| $(0, 1, 0.5)$ | $(1, 1, 0.5)$ | $(0.5, 1, 0.5)$ | $(0.5, 0, 0.5)$ $DoC = 1$ |
| $E(o_{A_1}) = 0$ | $E(o_{A_2}) = 1$ | $E(o_{A_1 \widehat{\oplus}_w A_2}) = 0.5$ | $E(o_{A_1 \widehat{\oplus}_c A_2}) = 0.5$ |
|  |  |  |  |
| Example 3 | | | |
| | | *W.FUSION* $w_1 = 1; w_2 = 2$ | *C.FUSION* $w_1 = 1; w_2 = 2$ |
| $(0.3, 0.833, 0.05)$ | $(0.9, 0.5, 0.35)$ | $(0.4717, 0.6996, 0.25)$ | $(0.4717, 0.5831, 0.25)$ $DoC = 0.166$ |
| $E(o_{A_1}) = 0.2582$ | $E(o_{A_2}) = 0.625$ | $E(o_{A_1 \widehat{\oplus}_w A_2}) = 0.4051$ | $E(o_{A_1 \widehat{\oplus}_c A_2}) = 0.3793$ |
|  |  |  |  |

$o_{A_1}$. However, using the *W.FUSION* (left) and giving a higher weight ($w_2 = 2$) to $o_{A_2}$ the resulting opinion ($(0.4717, 0.6996, 0.5)$) shows a shifted bias towards $o_{A_2}$. This example shows how the weighted fusion enables customised aggregation of opinions.

**Example 2:** The second example in Table 5.7 provides an interesting comparison between the *weighted fusion* (on the left) and the *conflict-aware fusion* on the right. In both of the cases, we combine two opinions with maximum certainty, but with conflicting average ratings, i.e., $o_{A_1} = (0, 1, 0.5)$ (strong negative opinion) and $o_{A_1} = (1, 1, 0.5)$ (strong positive opinion). When apply the *weighted fusion* the resulting opinion ($o_w$ for short) is $o_w = (0.5, 1, 0.5)$. For this opinion we have to note that the expectation value of the opinion is $E(o_w) = 0.5$, due to the average rating ($t_w = 0.5$), as the certainty value of this opinion is $c_w = 1$, which means that the average rating is representative for future outcomes (i.e., expectation value, $E$). This in turn means, that in a repeated series of experiments we can expect a similar

number of positive outcomes as negative outcomes (given a sufficiently high number of runs).

On the other hand, we have the resulting opinion ($o_c$ for short) is $o_c = (0.5, 0, 0.5)$ and the $DoC = 1$ (maximum) of the *conflict-aware fusion*. For this opinion, note that the expectation value of the opinion is $E(o_c) = 0.5$, too. However, this is due to the fact that the initial expectation value is $f_c = 0.5$. Furthermore, we see that the certainty value of this opinion is $c_c = 0$, which means that the average rating ($t_c = 0.5$) is not necessarily representative for future outcomes, i.e., it can easily change when new information becomes available.

Now, we can ask ourselves which of the resulting opinions reflects the situation better. The expectation value that the proposition under consideration is true, e.g., that the cloud provider has a competent customer service is 0.5 in both cases. In fact, if we think what would be the outcome of first request to the customer support, the information that we have collected propose that there is a probability of 0.5 for a positive experience and of 0.5 for a negative experience.

However, if we consider the case that we repeatedly run the experiment, e.g., repeated and subsequent interaction with the customer support, we should expect that the result of the second, third,$\cdots$, $n$ request is as satisfying (or unsatisfying) as the first one. Therefore, we conclude that this line of argumentation leads to the statement that the *conflict-aware fusion* produces a better result than the *weighted fusion*.

Finally, if one observes the result of the *weighted fusion*, i.e., $o_w = (0.5, 1, 0.5)$, this result is highly ambiguous and in fact, this could result from an infinite amount of opinions, e.g., $o_{A_1} = (0, 1, 0.5)$ and $o_{A_2} = (1, 1, 0.5)$. With the *conflict-aware fusion*, we address this problem by additionally providing the $DoC$.

**Example 3:** The third example in Table 5.7 provides another comparison between the weighted and the conflict-aware fusion. Here, the conflict between the input parameters (on the left) is not as extreme as in example 2 which is reflected by the $DoC = 0.166$ in the conflict-aware fusion (on the right). In this example, we also see that the reduction of the certainty (i.e., from $c_w = 0.6996$ to $c_c = 0.5831$) in the conflict-aware fusion usually leads to a lower expectation value (0.3793) than the expectation value (0.4051) in the weighted fusion. We argue that the lower expectation value in the conflict-aware fusion is justified in this example, as the average ratings of the input parameters are conflicting and thus, not representative. This effect comes from the reduction of the certainty (in the conflict-aware fusion) which in turn means that the expectation value is shifting closer to the initial expectation value (leading to a lower expectation value in this example).

Finally, the example also demonstrates how the choice of the initial expectation value ($f$) influence the HTI.

## 5.4 Novel Architecture for Trust Management (TM) using CertainLogic Framework

The mechanisms for assessing, representing and computing trustworthiness (considering uncertainty and conflict) are introduced in the previous sections. In this section, we propose a novel architecture (cf. Fig. 5.2) of a system for managing trust in cloud computing marketplaces. The idea of the system architecture rests on the definition of [JKD05], where the authors suggested that a *TM* should allow trustees to reliably represent their capabilities and allow trustors to make assessments and decisions regarding the dependability of the trustees. Moreover, the proposed architecture of the system integrates the formal framework (cf. Chapter 4) and the computational trust mechanisms (cf. Section 5.1) in order to comply with the requirements stated in Section 3.1 for trust systems.

The system is comprised of six modules, which are as follows.

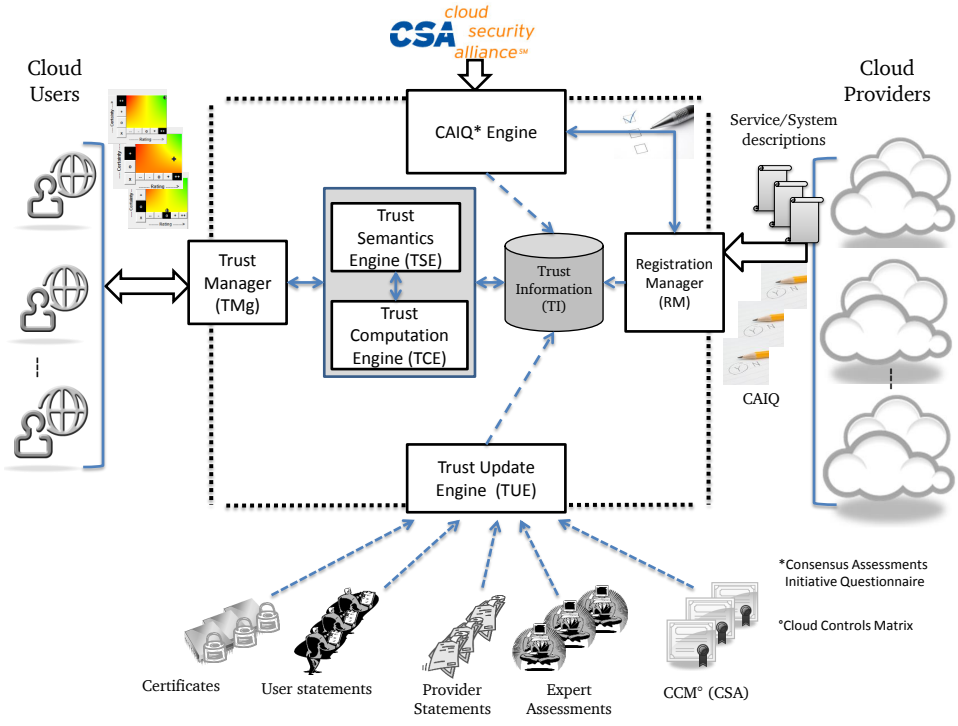



Figure 5.2: System Overview for Trust Management

### Registration Manager (RM)

Cloud providers register through the *RM* to be able to act as sellers in a cloud marketplace. They have to provide specifications regarding the services

they offer and fill in the *CAIQ* as a part of the cloud marketplace policy. The RM forwards the answers of the questionnaire and system/service description to the *CAIQ* engine and *TI* (Trust Information) respectively for further processing.

### Consensus Assessments Initiative Questionnaire (CAIQ) Engine

The *CAIQ* engine enables cloud providers to fill in the questionnaire by providing an intuitive graphical interface through the *RM*. The questionnaire allows cloud providers to represent their capabilities regarding service-specific attributes. The questions are designed to be answered in 'yes' or 'no'. All the assertions are stored in the *TI* for further processing. In order to utilize the completed *CAIQ*s in trust assessment process, we first need to assess the assertions and then, convert the assessment results into the CertainTrust opinion representation. The *CAIQ* assessment is discussed in Section 6.1.2. Experimental evaluation is performed using the completed *CAIQ*s, published in the *CSA STAR*, in Section 6.2.

### Trust Manager (TMg)

The *TMg* allows cloud consumers to specify their requirements before assessing the trustworthiness of cloud providers. It provides a front end to the users for specifying their requirements. Based on the requirements, the *TMg* provides the trust score of cloud providers by using the Trust Semantic Engine (*TSE*) and Trust Computation Engine (*TCE*). By default, users receive the trust value of a cloud provider based on their completed *CAIQ* and the assessment of their underlying services/systems. Otherwise, users can specify their own preferences, e.g., security and performance are preferred over customer support, according to their business policy and requirements in order to get a customised trust value of the cloud providers. Users may also choose the sources of trust information that need to be taken into account when computing the trustworthiness values of cloud providers. The *TMg* should also be able to provide trustworthiness value for every single attribute considered for the calculation of trustworthiness value by means of opinion $(t, c, f)$ representation and graphical interface (i.e., *HTI*). In the proposed architecture (cf. Fig. 5.2), the *TMg* is tightly coupled with the *TSE* and *TCE* to provide the above mentioned features to cloud consumers.

### Trust Semantics Engine (TSE)

The *TSE* is able to model which configuration of *PLT*s are required by the consumers in order to evaluate the trustworthiness of a cloud provider. A default configuration of *PLT*s is based on the *CAIQ* assertions stored in the repository (*TI*). For deriving the *PLT*s from system/service specifications,

the *TSE* integrates the formal framework proposed in 4.3.1. *PLT*s can also be derived from the *CAIQ*. The approach for deriving the *PLT*s from the *CAIQ* is discussed in Section 4.3.2.

Moreover, this engine supports users for expressing service-specific attributes as well as the sources of the information according to their choice. The *TSE* is able to customise the configuration of *PLT*s in order to reflect the users' preference. Customized *PLT*s are sent to the *TCE* for the final evaluation.

### Trust Computation Engine (TCE)

The *TCE* contains the definitions of CertainLogic operators in order to aggregate opinions associated with the PLTs. The *TCE* is tightly coupled with the *TSE* in order to enable the quantitative evaluation of *PLT*s. The outcome of the evaluation is a numerical score, i.e., trustworthiness value, along with a graphical representation of the score. The numerical values are archived in the *TI* repository after computation.

### Trust Update Engine (TUE)

The *TUE* is designed to collect opinions from various sources about the trustworthiness of cloud providers. The opinions collected here should be filtered appropriately so that users may use opinions that are valid according to their requirements. For example, spam filtering should be used to eliminate junk or useless information stored in the *TI* repository. Moreover, the sources should be authorised before they are able to provide opinions. The authorisation process should verify the identity of a source as well as trustworthiness of it's underlying platforms. The filtering of opinions and authorisation of sources are extremely important to ensure the reliability of trust assessment process inside the *TM* system.

## 5.5 Summary

This chapter provides a set of novel apparatus and mechanisms to evaluate the trustworthiness of service providers in distributed service environments, particularly in cloud computing marketplaces. The core aspects of this chapter can be summarized as follows:

1. CertainTrust is extended to a framework named *CertainLogic* that provides definitions of the logical operators ($AND$, $OR$, $NOT$) that combine opinions about independent propositions.

2. CertainLogic has further extended to provide definitions of non-standard operators, $A.FUSION$, $W.FUSION$, $C.FUSION$, which combine opinions about dependent propositions. $C.FUSION$ is a novel fusion

operator which is particularly designed to deal with weights (to model user preferences) and conflict among the opinions about dependent propositions.

3. Finally, a novel architecture for trust management is proposed to enable service providers to present their capabilities and consumers to evaluate the trustworthiness of service providers in marketplaces. CertainLogic operators and the formal framework are the nucleus of the proposed trust management system architecture. The proposed architecture is designed to manage trust information regarding multiple attributes and information that are derived from multiple sources to determine the trustworthiness of service providers.

Our proposed trust establishment mechanisms are evaluated in the next chapter using an instantiation of the trust management system architecture proposed in Section 5.4.

# 6 Evaluation

This chapter provides a quantitative evaluation of CertainLogic operators in real-world scenarios of cloud computing. In such scenarios, one would expect the evaluation of CertainLogic operators on opinions derived from evidence available from cloud providers in the real world. At present, only Cloud Security Alliance (CSA) provides a publicly accessible repository, i.e., STAR, where cloud providers disclose service-specific security attributes regarding their services. These attributes are basically the domains, e.g., Compliance (CO), Information Security (IS), in the top level of the CSA CAIQ framework. The assertions in reply to the lowest level control questions are the pieces of evidence, which infer the existence of a top level domain in the CAIQ framework. In this section, those pieces of evidence are leveraged to generate opinions and the CertainLogic operators along with the formal framework are applied on the opinions in order to evaluate the trustworthiness of cloud providers.

For the purpose of evaluation, we implement a realization of the TM system architecture proposed in Section 5.4. The realization of the system is discussed in Section 6.1. The realised system enables quantitative trustworthiness evaluation (cf. Section 6.2) of service providers based on the assertions (i.e., evidence) about their service/system attributes. The main reason for leveraging the STAR datasets in trust evaluation phase is to demonstrate the practicality of assessing and evaluating trustworthiness of cloud providers before consumers agree to a contract (e.g., SLA) with providers on provisioning offered services. In this chapter, several experiments are conducted using the datasets to justify the applicability of the formal framework and CertainLogic operators proposed in Chapter 4 and 5 respectively.

This chapter includes following three sets of experiments.

- In the first set of experiments, we demonstrate the applicability of CertainLogic $AND$ ($\wedge$) operator to combine opinions on independent service-specific capabilities of cloud providers. The real datasets from the *CSA STAR* [CSAd] are leveraged to conduct the experiments.
- In the second set of experiments, we demonstrate the applicability of CertainLogic $AND$ ($\wedge$) and $OR$ ($\vee$) operators to assess the trustworthiness of composite distributed systems and services in the context of cloud computing environments.
- In the third set of experiments, we consider a cloud computing marketplace scenario to demonstrate the applicability of CertainLogic FUSION operators to combine opinions from different sources. The CSA STAR datasets are leveraged to generate opinions for the experiments conducted using FUSION operators.

## 6.1 Realization of the TM System Architecture

This section discusses the building blocks of the realised system that leverage the security control self-assessment framework (CAIQ) as a basis for trust management. Firstly, we provide an overview of the *CSA CAIQ* which is important to understand the rationale behind the realised system. Secondly, we discuss how the formal framework and the proposed operators leverage the completed CAIQs to evaluate the trustworthiness of cloud providers. Finally, the key features of the realised system is discussed.

### 6.1.1 CSA CAIQ Revisited

The CAIQ includes 11 domains such as; Compliance (CO), Data Governance (DG), Facility Security (FS), Human Resources security (HR), Information Security (IS), LeGal (LG), OPerations management (OP), RIsk management (RI), Release Management (RM), ReSiliency (RS) and Security Architecture (SA). Each of these domains consists of several controls and control questions. The basic structure of the CAIQ is shown in Figure 6.1. For brevity, only 2 domains and their corresponding structures are shown. The CAIQ consists of 98 controls under 11 domains. On the one hand, these controls resemble requirements of the users who are interested to determine whether a cloud provider satisfies their requirements; while on the other hand, the same controls represents the cloud provider's capabilities. Each of these controls has one or more questions which are designed to be answered by the cloud providers in a 'yes/no' manner. Based on the given answers, consumers learn about the capabilities of cloud providers.

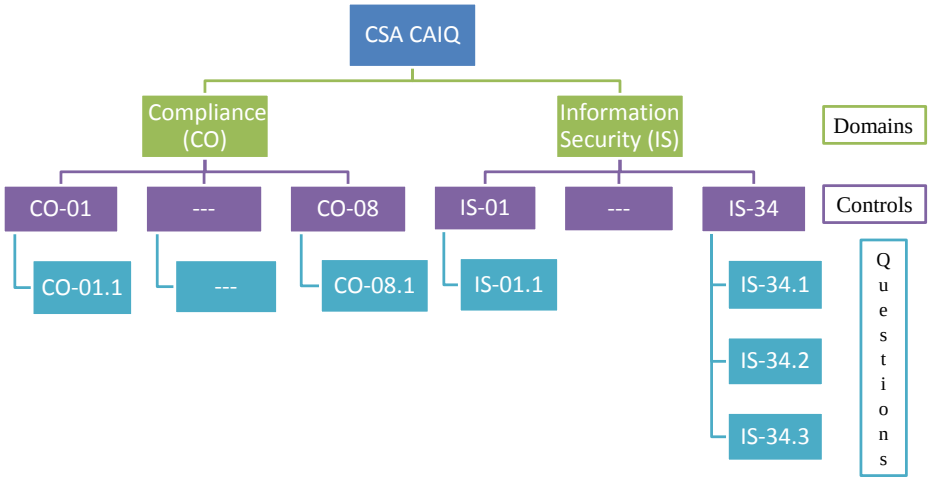



Figure 6.1: CSA CAIQ: Overview of basic structure

### 6.1.2 CAIQ Assessment

#### Assumptions

Our approach for the CAIQ assessment is based on the following assumptions. We assume that cloud providers provide one set of valid answers in response to the CAIQ for each of their services they offer and the answers are stored in the CSA STAR. The CSA is responsible for checking the authenticity and the basic accuracy of the answered questionnaires [CSAe].

#### Approach

In order to assess the CAIQ, we apply the formal framework (cf. Chapter 4) and necessary CertainLogic operators (cf. Chapter 5). The approach is detailed as follows:

1. Firstly, the formal framework is used to construct the PLTs from the given CAIQ domains. The TSE module of the proposed system (cf. Figure 5.2) is an instantiation of the formal framework that constructs the PLTs. In the context of the CAIQ, it means that the PLT configuration consists of 11 operands (i.e., domains) combined with 10 AND ($\wedge$) operators. Conceptually, PLT configuration is constructed as follows:

$$CO \wedge DG \wedge FS \wedge HR \wedge IS \wedge LG \wedge OP \wedge RI \wedge RM \wedge RS \wedge SA$$

2. Secondly, for evaluating the PLTs, the associated opinion $(t, c, f)$ for each of the propositions (CAIQ domains) is required. The opinions need to be extracted from each of the domains, where the underlying

questions are answered by the cloud providers. The answers are in the form of 'yes' or 'no', which upon analysis can be classified to *positive* and *negative* pieces of evidence. These evidence units correspond to the existence of each of the CAIQ domains, which in the end demonstrate the level of security capabilities a cloud provider has, regarding the services they offer in cloud marketplaces. In [Rie09b], there exists a mapping, which is proposed to map the collected pieces of evidence to the opinion space. The mapping (cf. Equation 6.1) has been proposed in the context of ubiquitous computing environments, where evidence units are collected based on a trustor's interaction experience with a trustee. In this context, positive ($r$) and negative ($s$) pieces of evidence are mapped to the CertainTrust opinion space. In the context of CAIQ, evidence units are based on the self-assessment of security capabilities that cloud providers possess regarding the services they offer. As the existence of the capabilities are reasoned based on the given assertions, the same mapping function is used to derive opinions from the given assertions, i.e., positive and negative pieces of evidence, under each of the domains. The mapping between the evidence space and the CertainTrust opinion space is as follows:

$$
\begin{aligned}
t &= \begin{cases} 0 & \text{if } r+s=0 \ , \\ \frac{r}{r+s} & \text{else} \ . \end{cases} \\
c &= \frac{N \cdot (r+s)}{2 \cdot (N-(r+s)) + N \cdot (r+s)} \\
f &= \text{consumer's expectation about each of the CAIQ domains} = 0.99
\end{aligned}
\tag{6.1}
$$

The detailed definitions of the parameters are as follows:

- Average rating, $t$, is calculated based on the number of *positive* assertions (i.e., $r$=yes) and the number of *negative* assertions ($s$=no) under each domain. If there are no questions answered with 'yes' or 'no', $t$ is 0. Otherwise, $t$ is the relative frequency of *positive* and negative assertions.
- Certainty, $c$, is calculated based on the total number of questions, $N$ and the number of positive and negative assertions under each domain. The $c$ is 1 when all questions under each domain are answered with *positive* or *negative* assertions and 0 if none are answered.

  The definition of $N$ is adjusted according to the context of CAIQ assessment. The total number of questions, $N$, not only consider *positive* and *negative* assertions but also the unanswered questions

under each domain. The unanswered questions can be of two types:

(a) Question that cloud providers left out for *unknown* reasons or an answer to a question is indeterminable to classify as 'yes' or 'no'.

(b) Question that does not fit the scope of the services (i.e., *Not Applicable*) that are offered by the cloud providers.

In order to deal with the above mentioned types of questions, we define $N$ as following:

- For type 2a, the *unknown* (we refer to as 'u') marked answer(s) to the corresponding question(s) under each domain are taken into account. That means, $N = r + s + u$.
- For type 2b, the *Not Applicable* (we refer to as 'NA') marked answers under each domain are not included in the calculation of $N$. That means, $N = (r+s+u)$ = Total number of questions $- NA$

- Initial expectation, $f$, is set as high (i.e., 0.99) for every single domain assuming that cloud providers publish information regarding their capabilities in the *STAR* repository truthfully and accuracy of those information are validated using CloudAudit [CSAa] framework. Recently, a practical approach to validate the CAIQ assertions is proposed in [HVM13a, HVM13b] using a hybrid trust framework, i.e., combining evidence-based trust mechanism (soft trust) with the certificate-based (hard trust) property attestation mechanism. The outcome of the validation is represented in the form of CertainTrust opinion representation, i.e., $(t, c)$. The initial expectation value $(f)$ then can be calculated based on the numerical values associated with $t$ and $c$ using the Equation 2.2.3 given in Section 2.2.2.2. Another mechanism for calculating $f$ from certification processes such as audits are demonstrated in [HVHM12]. In both of these cases, calculated $f$ is a continuous value between 0 and 1. Calculating $f$ is considered out of the scope of this thesis. Hence, both of the above mentioned methods for calculating $f$ are not taken into account in this thesis.

**Example**

In order to provide an in-depth insight to our approach, we illustrate it by means of an intuitive example. Let us apply our approach (cf. Section 6.1.2) to assess a completed CAIQ questionnaire. In order to keep it simple, the assessment approach considers two domains, *Compliance (CO)* and *Information Security (IS)*.

According to the latest version of CAIQ [CSAc], CO has 16 questions under eight different controls. Let us assume that cloud provider 'X' has answered the CO domain and the answers are classified as follows:

- Number of *positive* assertions, $r = 11$
- Number of *negative* assertions, $s = 2$
- Number of *unknown* assertions, $u = 2$
- Number of *Not Applicable* assertions, $NA = 1$
- Number of questions, $N = 15$, due to 1 NA question

Now, using the Equation 6.1, the resulting opinion $(t, c, f)$ is computed as follows:

$$\begin{aligned} t &= \frac{11}{(11+2)} = 0.8462 \\ c &= \frac{15 \cdot (11+2)}{2 \cdot (15-(11+2)) + 15 \cdot (11+2)} \\ &= 0.9798 \\ f &= 0.99 \end{aligned} \tag{6.2}$$

The corresponding opinion, i.e., $(t, c, f)$, on *IS* domain can be derived from the given assertions of cloud provider 'X' following the same approach as *CO* domain. According to the given assertions, the resulting opinion for the *IS* domain is, $(t, c, f) = (1, 1, 0.99)$. It means that *all* the questions are answered with positive assertions. Following the first step of our approach, we construct the PLTs $(CO \wedge IS)$ as shown earlier. Then, the opinions that are derived from the given assertions provide a basis for evaluating the PLTs.

3. Opinion on Cloud provider 'X' regarding their capabilities, i.e., Compliance (CO) and Information Security (IS), are combined based on the definition of the *AND* operator (cf. Table 5.1). The final opinion on PLTs (i.e., $(CO \wedge IS)$) means that the cloud provider 'X' has an average rating, $t = 0.8478$, with high certainty of 0.9898 under an initial expectation of 0.99 regarding *both* capabilities. The expectation value $(E = 0.8491)$, calculated based on the resulting opinion, interprets that the cloud provider is *trustworthy* regarding their capabilities to a degree of 0.8491. This value can also be denoted as *trustworthiness* value.

In the next section, a detailed discussion on the implemented CAIQ assessment tool is given.

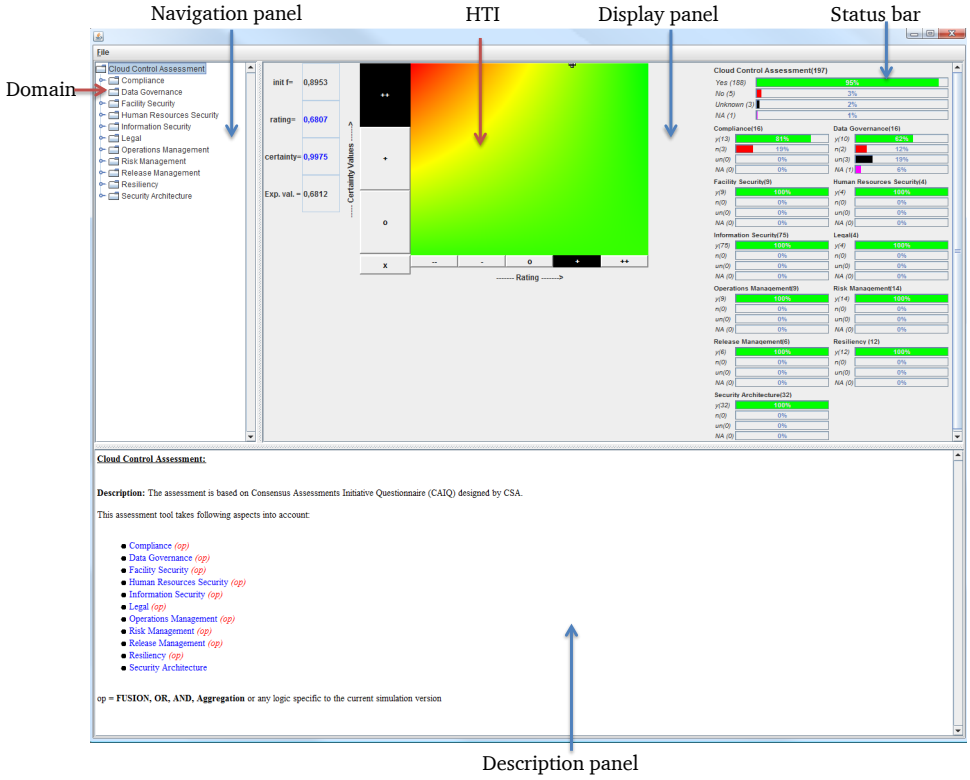



Figure 6.2: Visualization of CCA tool: Domain

### 6.1.3 Implementation

The implemented realization of the TM system is named "Cloud Control Assessment (CCA)" tool. The tool is developed to assess the assertions of the CAIQ and evaluate cloud providers based on the assessment. It has two special features in contrast to the current status of the *CSA CAIQ*:

1. The proposed tool provides a intuitive graphical interface for answering questions and supports interactive visualization of the assessment.

2. Our proposed tool also allows to load a completed questionnaire from the CSA STAR. For evaluating the questionnaire, CCA includes the features of the TSE to configure the PLTs and TCE (definition of the operators such as *AND* and *FUSION*) component as described in Section 5.4. The main objective is to quantify the CAIQ assessment into a trustworthiness score (i.e., expectation value) with complementary numerical and graphical opinion representations.

The graphical interface (cf. Figure 6.2 and Figure 6.3) of the tool has three panels and a menu bar:

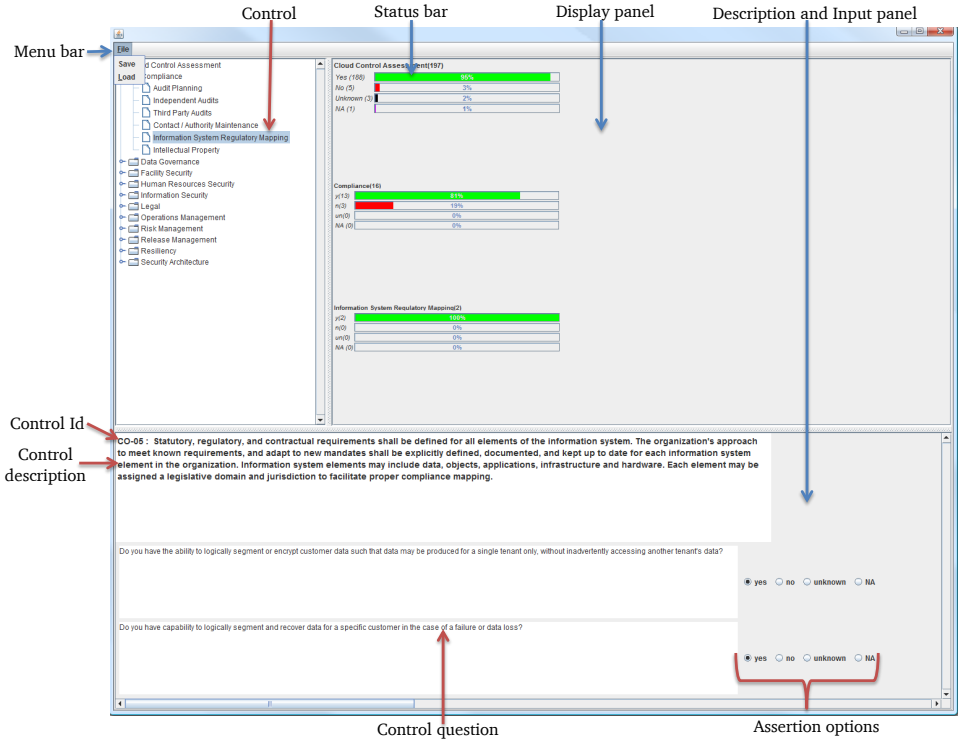



Figure 6.3: Visualization of CCA tool: Control Question

1. Navigation Panel: This panel has a tree-like structure to display and navigate the users to the CAIQ domains, their controls and control-specific questions.

2. Description and Input Panel: This panel provides description of each of the domains when selected in the navigation panel. Moreover, the control questions are displayed with their corresponding options (i.e., 'yes', 'no', 'unknown' and 'NA') when a particular control (e.g., Independent Audits) under a domain is selected in the navigation panel. These four options are given as radio buttons to allow faster input from the users (i.e., cloud providers) compared to the current approach (manual input in a Excel sheet) designed by CSA.

3. Display Panel: This panel displays results and progress or status by means of a graphical interface (i.e., CertainTrust HTI) and status bars, respectively. The HTI shows the opinion $(t, c, f)$ and the corresponding expectation value $(E)$ in a special panel on its left. The status bars are introduced for monitoring the progress of the assessment interactively. On the one hand, it provides cloud consumers a quick summary of the assessment based on the *assertions* given by the cloud providers. On

the other hand, cloud providers are able to monitor their progress in terms of numerical and graphical representations.

4. Menu bar: The bar includes 'Save' and 'Load' functions that are extremely important for lengthy questionnaire such as the CAIQ with 197 questions. To respond to 197 questions under 11 domains is quite a cumbersome task. Keeping that problem in mind, we developed this tool in a way so that the cloud providers can *save* their undone tasks while filling the questionnaire and *load* them at any convenient time. The load function is also essential for our evaluation phase. We use this function to load the questionnaires published in the *STAR* repository.

## 6.2 Experimental Evaluation: CertainLogic AND Operator

In this section, we discuss the experiments conducted using the *CAIQ* datasets published in the *STAR*. The objective of the experiments are to demonstrate the applicability of CertainLogic operators.

Following three cases is considered to demonstrate the applicability of CertainLogic AND ($\wedge$) operator:

1. **Best case.** In the *best case*, it is assumed that the Cloud provider 'X' provides all positive assertions when filling in the *CAIQ*. Hence, only positive synthetic assertions are taken into consideration to demonstrate the effect of CertainLogic AND ($\wedge$) operator on the resulting opinion, i.e., aggregated opinion associated with the CAIQ domains.

   The hypothesis we want to test here is: *CertainLogic AND operator calculates maximum trustworthiness value of a cloud provider if the provider asserts the existence (i.e., all positive assertions) of all the capabilties regarding the domains.*

2. **Practical case.** In this case, assertions derived from the *STAR* datasets are considered for the experiments. These experiments demonstrate the realistic effect of CertainLogic AND ($\wedge$) operator in the context of assessment and evaluation of cloud providers based on their assertions given in the STAR.

   The hypothesis we want to test here is: *CertainLogic AND operator calculates the trustworthiness value of a cloud provider based on the opinions derived from their given assertions published in the STAR and the operator behaves as expected (cf. Section 5.2.3).*

3. **Customised case.** This case considers personal preferences of a consumer on selecting attributes, e.g., CO, DG, SA domains in the context of CAIQ, when assessing the capabilities of cloud providers.

This particular case demonstrates the effect of AND ($\wedge$) operator by calculating the customised trustworthiness value according to the requirements specified by a consumer.

The hypothesis we want to test here is: *CertainLogic AND operator calculates the customised trustworthiness value based on the opinions associated with the domains that are specified by a consumer and the operator behaves as expected (cf. Section 5.2.3).*

Apart from the above mentioned three cases, there can be a worst case where a cloud provider might leave all the questions unanswered or answer all the questions with negative assertions. This particular case is assumed to be unrealistic as a cloud provider is not going to engage in such a practice which might lower its trustworthiness in a marketplace. One might think of a case where a cloud provider can be impersonated by a malicious entity and provides false information in order to hamper the cloud provider's trustworthiness. We assume that the *CSA* (as a trusted third party) checks the authenticity of the submissions as well as the identities of cloud providers and accuracy of the contents before publishing them in the *CSA STAR*.

### 6.2.1 Experiments: best case

Table 6.1: Cloud Control Assessment for Cloud 'X': Best case

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f); E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | |
| $DG$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | |
| $FS$ | 9 | 0 | 0 | 0 | 9 | (1,1,0.99) | |
| $HS$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | |
| $IS$ | 75 | 0 | 0 | 0 | 75 | (1,1,0.99) | |
| $LG$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | (1,1,0.8953);E=1 |
| $OM$ | 9 | 0 | 0 | 0 | 9 | (1,1,0.99) | |
| $RI$ | 14 | 0 | 0 | 0 | 14 | (1,1,0.99) | |
| $RM$ | 6 | 0 | 0 | 0 | 6 | (1,1,0.99) | |
| $RS$ | 12 | 0 | 0 | 0 | 12 | (1,1,0.99) | |
| $SA$ | 32 | 0 | 0 | 0 | 32 | (1,1,0.99) | |

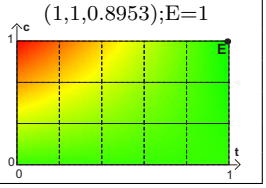



Table 6.1 shows the positive assertions in the evidence space and their resulting opinions $(t, c, f)$ using Equation 6.1. The last column of the table shows the final assessment based on the aggregation of all resulting opinions using the CertainLogic AND ($\wedge$) operator. The final assessment is given in opinion representation $(t, c, f)$ and expectation value $(E)$. In the final assessment, one can see how the AND ($\wedge$) operator affects the initial expectation,

$f$. It holds 0.99 (high expectation) for every single domain whereas for *all* domains it holds 0.8953. This is because of *all* assertions related to controls have to be true simultaneously.

### 6.2.2 Experiments: practical case

The *STAR* repository has several sets of completed questionnaires filled in by different cloud providers [CSAd]. We chose three sets of CAIQs completed by Cloud provider 'A', 'B' and 'S'. The identities of the cloud providers are *anonymized* due to usage restrictions of the *STAR*. At present, the *STAR* repository as it stands does not classify the completed CAIQs according to the service delivery models (e.g., SaaS, PaaS, IaaS) offered by the cloud providers. Thus, considering synthetic CAIQs which are assumed to be completed by Cloud provider 'Y' and Cloud provider 'Z'. Cloud provider 'Y' offer services with same functionalities as Cloud provider 'A' and Cloud provider 'Z' offer services with same functionalities as Cloud provider 'B'.

Table 6.2: Cloud Control Assessment for Cloud 'A' (anonymized): Practical case

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 0 | 2 | 0 | 9 | (1,0.9403,0.99) | |
| $HS$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | |
| $IS$ | 72 | 2 | 0 | 1 | 74 | (0.973,1,0.99) | |
| $LG$ | 2 | 0 | 0 | 2 | 2 | (1,1,0.99) | (0.5186,0.9945,0.8953);E=0.5207 |
| $OM$ | 4 | 3 | 0 | 2 | 7 | (0.5714,1,0.99) | |
| $RI$ | 12 | 0 | 1 | 1 | 13 | (1,0.9873,0.99) | |
| $RM$ | 5 | 0 | 0 | 1 | 5 | (1,1,0.99) | |
| $RS$ | 9 | 0 | 2 | 1 | 11 | (1,0.9612,0.99) | |
| $SA$ | 22 | 0 | 0 | 10 | 22 | (1,1,0.99) | |

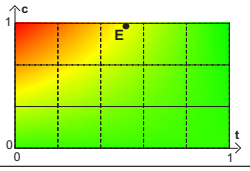
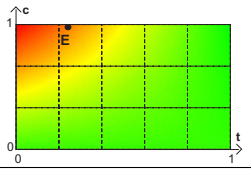



Table 6.2 and Table 6.3 present a summary of the assertions and the corresponding resulting opinions are calculated using the Equation 6.1, which maps the given assertions to opinion. According to the final assessment given in Table 6.2 and 6.3, cloud consumers can identify a potential cloud provider based on the computed expectation value. The expectation value of Cloud provider 'A' is much higher than that of Cloud provider 'Y'. It means that provider 'A' is more trustworthy than provider 'Y' based on the assessment of capabilities regarding security controls. Hence, the expectation value is a reasonable indicator for cloud consumers to identify a trustworthy

Table 6.3: Cloud Control Assessment for Cloud 'Y': Practical case using synthetic datasets

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 0 | 2 | 0 | 9 | (1,0.9403,0.99) | |
| $HS$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | |
| $IS$ | 72 | 2 | 0 | 1 | 74 | (0.973,1,0.9865) | |
| $LG$ | 2 | 2 | 0 | 0 | 4 | (0.5,1,0.99) | (0.2239,0.9976,0.8953);E=0.2255 |
| $OM$ | 4 | 3 | 0 | 2 | 7 | (0.5714,1,0.99) | |
| $RI$ | 12 | 1 | 1 | 0 | 14 | (0.9231,0.9891,0.99) | |
| $RM$ | 5 | 0 | 0 | 1 | 5 | (1,1,0.99) | |
| $RS$ | 9 | 0 | 2 | 1 | 11 | (1,0.9612,0.99) | |
| $SA$ | 22 | 0 | 0 | 10 | 22 | (1,1,0.99) | |




cloud provider, Cloud provider 'A' in this case. Note that in addition to the expectation value, the certainty ($c$) value is a good indicator of whether the aggregated average rating (t) is representative or whether further analysis is required. If the consumers need further analysis, they can browse each domain individually (using our *CCA* tool) for comprehensive assessment of the security controls released by the cloud provider(s).

Table 6.4 and Table 6.5 present another set of experiments where Cloud provider 'B' is assessed based on its published capabilities regarding the service that provider 'B' offers. Based on the CAIQ assessment, the computed expectation value ($E$) for Cloud provider 'B' is 0.1798 which is pretty low compared to the calculated expectation value, $E = 0.5912$, of Cloud provider 'Z'. It means that provider 'Z' is more trustworthy than provider 'B' based on the assessment of completed CAIQ. By analysing the assertions given in both of the tables, we conclude that provider 'Z' possess more capabilities compare to provider 'B' regarding several security controls. This is the rationale why the calculated expectation value ($E$) for provider 'Z' is higher than that of provider 'B'.

Table 6.4: Cloud Control Assessment for Cloud 'B' (anonymized): Practical case

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 13 | 1 | 0 | 2 | 14 | (0.9286,1,0.99) | |
| $DG$ | 14 | 2 | 0 | 0 | 16 | (0.875,1,0.99) | |
| $FS$ | 8 | 1 | 0 | 0 | 9 | (0.8889,1,0.99) | |
| $HS$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | |
| $IS$ | 64 | 8 | 0 | 3 | 72 | (0.8889,1,0.99) | |
| $LG$ | 2 | 0 | 0 | 2 | 2 | (1,1,0.99) | (0.1798,1,0.8953);E=0.1798 |
| $OM$ | 4 | 1 | 0 | 4 | 5 | (0.8,1,0.99) | |
| $RI$ | 12 | 1 | 0 | 1 | 13 | (0.9231,1,0.99) | |
| $RM$ | 3 | 2 | 0 | 1 | 5 | (0.6,1,0.99) | |
| $RS$ | 9 | 2 | 0 | 1 | 11 | (0.8182,1,0.99) | |
| $SA$ | 17 | 5 | 0 | 10 | 22 | (0.7727,1,0.99) | |

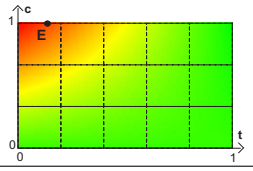
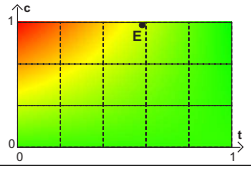



Table 6.5: Cloud Control Assessment for Cloud 'Z': Practical case using synthetic datasets

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 14 | 0 | 0 | 2 | 14 | (1,1,0.99) | |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 9 | 0 | 0 | 0 | 9 | (1,1,0.99) | |
| $HS$ | 4 | 0 | 0 | 0 | 4 | (1,1,0.99) | |
| $IS$ | 71 | 1 | 0 | 3 | 72 | (0.9861,1,0.9861) | |
| $LG$ | 2 | 0 | 0 | 2 | 2 | (1,1,0.99) | (0.5911,0.9997,0.8953);E=0.5912 |
| $OM$ | 5 | 0 | 0 | 4 | 5 | (1,1,0.99) | |
| $RI$ | 12 | 1 | 0 | 1 | 13 | (0.9231,1,0.99) | |
| $RM$ | 4 | 1 | 0 | 1 | 5 | (0.8,1,0.99) | |
| $RS$ | 10 | 1 | 0 | 1 | 11 | (0.9091,1,0.99) | |
| $SA$ | 20 | 1 | 1 | 10 | 22 | (0.9524,0.9957,0.99) | |



### 6.2.3 Experiments: customised case

Table 6.6: Cloud Control Assessment for Cloud 'A' (anonymised): Customised case

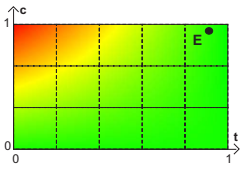
| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | (0.9109,0.9862,0.9606);E=0.9116  |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 0 | 2 | 0 | 9 | (1,0.9403,0.99) | |
| $IS$ | 72 | 2 | 0 | 1 | 74 | (0.973,1,0.99) | |



Table 6.7: Cloud Control Assessment for Cloud 'Y': Customised case

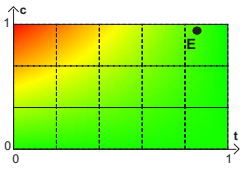
| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | (0.8532,0.987,0.9606);E=0.8546  |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 0 | 2 | 0 | 9 | (1,0.9403,0.99) | |
| $IS$ | 72 | 2 | 0 | 1 | 74 | (0.973,1,0.99) | |



Table 6.6 and Table 6.7 reflect consumer's preferences to evaluate a trustworthy provider in the cloud marketplace. In such a marketplace, a consumer might require cloud providers to possess capabilities regarding $CO$, $DG$, $FS$ and $IS$ domains as a part of their service provisioning policy. We demonstrate experiments on the completed $CAIQ$s by Cloud 'A' and Cloud 'Y'. By enabling *customization* feature of the $CCA$ tool on the completed CAIQs, we observe notable changes in opinion values as well as in expectation values calculated for both providers in comparison to the results given in Table 6.2 and Table 6.3. Moreover, the customisation feature allows the customers to get a personalized assessment of cloud providers' capabilities in contrast to the existing excel-based tool available on the $CSA$ website.

In continuation, there will now be an analysis of the documented results in Table 6.6 and 6.7. The expectation values, calculated based on the customization of the domains, are an improvement than the values calculated in Table 6.2 and Table 6.3. The reason behind the deflection of the expectation values is that the assertions related to the required capabilities are more 'positive' than that of Table 6.2 and Table 6.3. It means that the cloud providers in Table 6.6 and 6.7 are more trustworthy in the customised case than in the practical case. Based on the results of the final assessment in the customised case, we conclude that cloud provider 'A' is more trustworthy than cloud provider 'Y'. The reason here is that 'A' possess *all* the capabilities under *CO* domain whereas provider 'Y' lacks one capability under that domain.

In this section, we have demonstrated the applicability of the formal framework and CertainLogic operator for combining opinions on independent propositions. The propositions are constructed according to the independent *domains* given in the *CSA CAIQ*. Opinions on the propositions are derived from the assertions given by the cloud providers in the *STAR*. Considering CertainLogic AND ($\wedge$) operator for combining opinions in this context demonstrates the operator's applicability in a real world setting.

## 6.3 Experimental Evaluation: CertainLogic AND and OR Operators

In order to demonstrate the applicability of CertainLogic AND and OR operators together, we revisit the cloud computing scenario discussed briefly in Chapter 4 Section 4.3. That scenario demonstrates how the evaluation of trustworthiness of a complex distributed system can be carried out, if there is an appropriate approach for constructing the PLTs from system specification. The approach is discussed in Chapter 4 with intuitive examples. Thus, in this section, we only focus on the quantitative evaluation of PLTs (i.e., formal representation of a composite distributed system) using CertainLogic operators. In this case, the *TCE* component, implemented for the realised system (cf. Section 6.1), is used to evaluate the PLTs.

### 6.3.1 Scenario: A Composite Service in Cloud Computing

Assuming that we evaluate the trustworthiness of a simple cloud-based Medical Record Management (MRM) service focusing on its *availability*. We firstly stick to a single service provider, i.e., Cloud provider 'X' offering a MRM service $S$.

In this case (cf. Fig. 6.4), the MRM service $S$ directly relies on two subsystems, $S_1$ providing authentication capabilities, $S_2$ offering storage capacity for sales data and data mining capabilities, and an atomic component

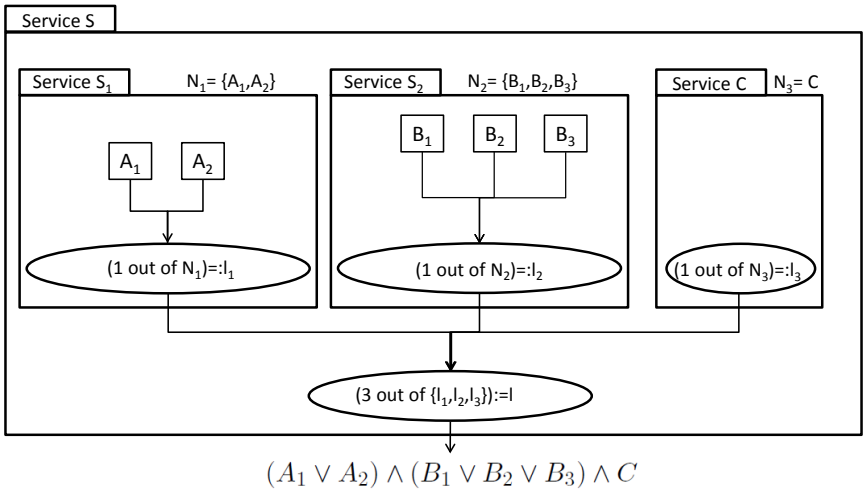



Figure 6.4: A MRM service: determination of PLTs

$C$ for the service-specific billing. Subsystem $S_1$ consist of two authentication servers ($A_1$ and $A_2$), where at least *one* of the servers has to be available. Similarly, subsystem $S_2$ is composed of three redundant database servers, where *one* server needs to be available.

We assume that the collected information (i.e., opinions) about the trustworthiness of the subsystems and atomic components are known. Based on the given system description and opinions, trustworthiness evaluation of the composite system can be carried out by evaluating the following PLTs:

$$(A_1 \vee A_2) \wedge (B_1 \vee B_2 \vee B_3) \wedge C$$

where $A_1$ is a proposition, which is true if component $A_1$ behaves as expected, e.g., the component replies to requests within a certain time limit; the interpretations of the other propositions are assigned in the same way. Although we restricted the scope of our example to *availability*, note that it is possible to model statements about the fulfilment of other relevant properties (e.g., attested / self-evaluated security properties of a component or subsystem) as propositions and to consider them in the evaluation of the overall trustworthiness of the system using PLTs. However, as the knowledge about the fulfilment of the propositions is subject to uncertainty, the evaluation method has to take this uncertainty into account when calculating the trustworthiness of the overall system.

### 6.3.2 Evaluation

Here, we show how the AND ($\wedge$) and OR ($\vee$) operators of CertainLogic can be applied to the scenario presented in Section 6.3.1. The propositional logic

term for evaluating the trustworthiness of the system in the scenario is given as:

$$(A_1 \vee A_2) \wedge (B_1 \vee B_2 \vee B_3) \wedge C$$

For the evaluation, we assume that we have good knowledge about the components of subsystem $S_1$ (consisting of $A_1$ and $A_2$) and subsystem $S_2$ (consisting of $B_1$, $B_2$, and $B_3$) and that the components are highly available. The opinions $o_{A_1}$ and $o_{A_2}$ as well as the resulting opinion $o_{A_1 \vee A_2} = o_{S_1}$ are given in Table 6.8(a). The opinions $o_{B_1}$, $o_{B_2}$, and $o_{B_3}$ as well as the resulting opinion $o_{B_1 \vee B_2 \vee B_3} = o_{S_2}$ are given in Table 6.8(b). In both cases, the subsystems are highly trustworthy ($E(o_{S_1}) = 0.9963$ and $E(o_{S_2}) = 0.9503$) and the certainty for both systems is also high ($c_{S_1} = 0.9956$ and $c_{S_2} = 0.9608$).

We demonstrate the advantage of CertainLogic operators by considering two different test cases regarding the trustworthiness of the atomic component $C$. Depending on whether the component is hosted by the owner of the overall system or by a third party, the *certainty* about the behaviour of this component might be higher or lower. The hypothesis we want to test with these two cases is: *the trustworthiness of a single subsystem influences the trustworthiness of the overall system/service.* The third test case is to demonstrate the application of operators for comparing two competitive services/systems based on the redundancy of the underlying components. For this case, we consider adding a redundant component under the subsystem $B$, i.e., $B_4$. The hypothesis we want to test here is: *adding a redundant component in service increases the level of trustworthiness of the overall system/service regarding availability attribute.*

Table 6.8: Cloud provider 'X': Resulting opinions for $S_1$ and $S_2$

(a) $S_1$:

| | |
|---|---|
| $o_{A_1}$ | $(0.90, 0.98, 0.5)$ |
| $o_{A_2}$ | $(0.99, 0.95, 0.5)$ |
| $o_{A_1 \vee A_2} = o_{S_1}$ | $(0.9974, 0.9956, 0.75)$ $E = 0.9963$ |

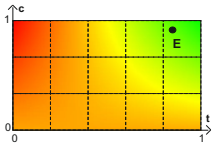
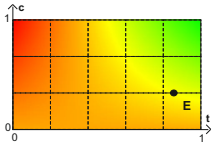
(b) $S_2$:

| | |
|---|---|
| $o_{B_1}$ | $(0.70, 0.80, 0.50)$ |
| $o_{B_2}$ | $(0.75, 0.80, 0.50)$ |
| $o_{B_3}$ | $(0.70, 0.90, 0.50)$ |
| $o_{B_1 \vee B_2 \vee B_3} = o_{S_2}$ | $(0.9584, 0.9608, 0.875)$ $E = 0.9503$ |

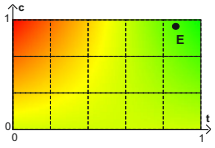
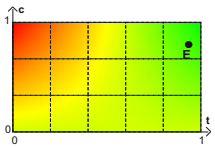
The experiments regarding the above mentioned three cases are as follows.

Table 6.9: Cloud provider 'X': Resulting opinions for $S$

(a) *Case 1:*

| | $o_C$ | $o_{S_1 \wedge S_2 \wedge C} = o_S$ |
|---|---|---|
| high certainty | $(0.90, 0.90, 0.50)$ | $(0.8604, 0.9211, 0.3281)$<br>$E(o_S) = 0.8184$<br>  |
| low certainty | $(0.90, 0.10, 0.50)$ | $(0.853, 0.3539, 0.3281)$<br>$E(o_S) = 0.5139$<br>  |

(b) *Case 2:*

| | $o_C$ | $o_{S_1 \wedge S_2 \wedge C} = o_S$ |
|---|---|---|
| high certainty | $(0.90, 0.90, 0.90)$ | $(0.8665, 0.9635, 0.5906)$<br>$E(o_S) = 0.8564$<br>  |
| low certainty | $(0.90, 0.10, 0.90)$ | $(0.933, 0.7764, 0.5906)$<br>$E(o_S) = 0.8564$<br>  |

- *Case 1:* We assume that the trustworthiness of $C$ is given as $o_C = (0.90, 0.90, 0.50)$ [*high certainty*] and as $o_C = (0.90, 0.10, 0.50)$ [*low certainty*]. For brevity, the trustworthiness of the overall system $S$ (consisting of $S_1$, $S_2$, and $C$) are given in Table 6.9(a). In the first row, we see that the *high certainty in* $o_C$ is also reflected in the resulting opinion ($c_S = 0.9211$), whereas the *low certainty in* $o_C$ is reflected in the resulting opinion ($c_S = 0.3539$) in the second row. In this example, we have different expectation values for $o_C$ (depending on the certainty), and thus also different expectation values for $o_S$.

- *Case 2:* Here, we assume that the trustworthiness of $C$ is given as $o_C = (0.90, 0.90, 0.90)$ [*high certainty*] or as $o_C = (0.90, 0.10, 0.90)$ [*low certainty*]. Both of these opinions lead to the same expectation values. The expectation value for the trustworthiness of the overall system is also the same. It is due to the compliance of the logical operators of CertainLogic with the same operators in the standard probabilistic approach. However, in our approach the different values for the certainty in the input parameters are still visible in the final result, for the certainty it holds $c_S = 0.9635$ [*high certainty*] and $c_S = 0.7764$ [*low certainty*] (cf. Table 6.9(b)).

Table 6.10: Resulting Opinions for $S_2$

(a) Case 3: $S_2$ *of Cloud provider 'X'*

| | |
|---|---|
| $o_{B_1}$ | $(0.70, 0.80, 0.50)$ |
| $o_{B_2}$ | $(0.75, 0.80, 0.50)$ |
| $o_{B_3}$ | $(0.70, 0.90, 0.50)$ |
| $o_{B_1 \vee B_2 \vee B_3} = o_{S_2}$ | $(0.9584, 0.9608, 0.875)$<br>$E = 0.9503$ |

(b) Case 3: $S_2$ *of Cloud provider 'Z'*

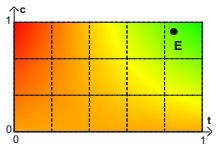
| | |
|---|---|
| $o_{B_1}$ | $(0.70, 0.80, 0.50)$ |
| $o_{B_2}$ | $(0.75, 0.80, 0.50)$ |
| $o_{B_3}$ | $(0.70, 0.90, 0.50)$ |
| $o_{B_4}$ | $(0.99, 0.99, 0.50)$ |
| $o_{B_1 \vee B_2 \vee B_3 \vee B_4} = o_{S_2}$ | $(0.9992, 0.9987, 0.9375)$<br>$E = 0.9991$ |

- *Case 3:* In a real world setting, one would assume that a MRM service might be offered by another provider, e.g., Cloud provider 'Z'. The service offered by Cloud provider 'Z' assumed to have one redundant component (under $S_2$) more than the subsystem $S_2$ of MRM service of Cloud provider 'X' offers. The redundant component is denoted as $B_4$ and it is highly trustworthy component, i.e., $(0.99, 0.99, 0.50)$,

which is evident just by looking at the opinion representation of that component. Table 6.10(a) and Table 6.10(b) lists the opinions of the individual components as well as the resulting opinion of the subsystems of provider 'X' and 'Z'. Now, if we compare only the resulting opinion, $(0.9584, 0.9608, 0.875)$, of $S_2$ of Cloud provider 'X' with the resulting opinion, $(0.9992, 0.9987, 0.9375)$, of Cloud provider 'Z', it is evident that by adding an additional redundant component under $S_2$ of Cloud provider 'Z' lifts up the trustworthiness value, i.e., $E = 0.9991$, of subsystem $S_2$ compare to the trustworthiness value, i.e., $E = 0.9503$, of subsystem $S_2$ of provider 'X'. This is because of the influence of CertainLogic OR operator on aggregating the opinions associated with the redundant components under the subsystem $S_2$.

Table 6.11: Comparison between MRM service ($S$) of Cloud provider 'X' and Cloud provider 'Z'

(a) Case 3: *Resulting opinion of MRM service of provider 'X'*

| $o_{S_1}$ | $o_{S_2}$ | $o_C$ | $o_S$ |
|---|---|---|---|
| $(0.9974, 0.9956, 0.75)$ | $(0.9584, 0.9608, 0.875)$ | $(0.90, 0.90, 0.50)$ | $(0.8604, 0.9211, 0.3281)$ $E(o_S) = 0.8184$   |

(b) *Case 3: Resulting opinion of MRM service of provider 'Z'*

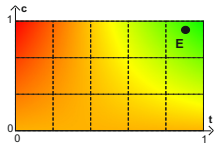
| $o_{S_1}$ | $o_{S_2}$ | $o_C$ | $o_S$ |
|---|---|---|---|
| $(0.9974, 0.9956, 0.75)$ | $(0.9992, 0.9987, 0.9375)$ | $(0.90, 0.90, 0.50)$ | $(0.899, 0.9216, 0.3516)$ $E(o_S) = 0.8561$   |

Table 6.11 lists the calculated trustworthiness value, i.e., $E$, of overall service $S$ offered by Cloud provider 'X' and 'Z' in Table 6.11(a) and 6.11(b) respectively. In Table 6.11(b), we see that the expectation value ($E(o_S) = 0.8561$) of the MRM service offered by Cloud provider 'Z' is higher than the expectation value ($E(o_S) = 0.8184$) calculated for the same service offered by Cloud provider 'X'. This is only because of the additional redundant component $B_4$ considered for the MRM service of provider 'Z', as all other components and subsystems as

well as their values are the same for both services offered by different providers.

## 6.4 Experimental Evaluation: CertainLogic FUSION Operators

In this section, we consider a cloud marketplace scenario (similar to the scenario in Section 4.4.1) and demonstrate the applicability of the proposed *fusion* operators in that scenario. The objectives are two-fold.

- Firstly, we demonstrate the effect of considering preferential weights when combining opinions from different sources.
- Secondly, we demonstrate how the novel fusion operator, i.e., conflict-aware, provides the most representative assessment of trustworthiness of cloud providers (e.g., Cloud 'A' and Cloud 'B') compared to the existing operators, i.e., average fusion and weighted fusion.

### 6.4.1 Cloud Marketplace Scenario

The cloud marketplace scenario discussed in Section 4.4.1 is revisited to highlight the integration of multiple trust information sources. The sources are assumed to provide opinions on the security-specific capabilities of cloud providers.

In the revisited scenario (cf. Figure 6.5), the cloud marketplace offers cloud services to the users as well as support them to identify trustworthy providers. The marketplace considers CSA's CAIQ framework as a basis to evaluate the trustworthiness of cloud providers.

The cloud marketplace aims to determine trustworthy cloud providers by using a reliable and transparent mechanism. The CCA tool provides means to assess security-specific capabilities published by the cloud providers and evaluate trustworthiness based on the assessment. In the previous section, we demonstrated how to combine different capabilities where the opinions about the capabilities are only considered to be given by a cloud provider. However, in the current scenario, the opinions about the capabilities are considered to be provided by various sources, e.g., cloud consumers, experts, and accreditators. For the sake of simplicity, a single provider is considered in the running scenario, i.e., Cloud 'A'.

Cloud 'A' fills in the CAIQ. It is published in the STAR as a policy of the marketplace. The *CCA* tool is used to assess the published *CAIQ* and the assessment is denoted as 'Q'. To ensure a reliable assessment of trustworthiness of Cloud A, consumers incorporate opinions based on various types of assessment from three other sources, i.e., experts ($E$), consumers' feedback ($F$) and accreditators ($A$).

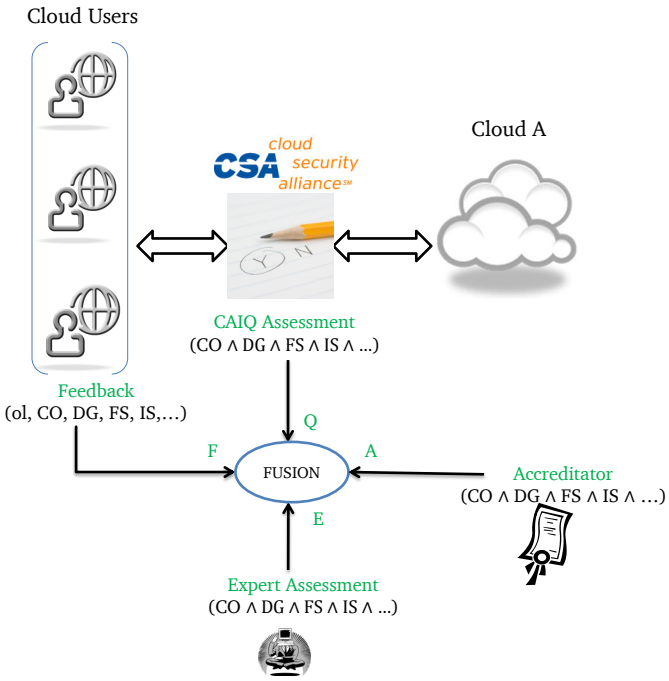



Figure 6.5: Cloud Marketplace – Fusion of Opinions from Multiple Sources

We see that opinions about Cloud A's trustworthiness are extracted from different sources. Let us assume that a user requires the following four capabilities: $CO$, $DG$, $FS$, $IS$ to be fulfilled by potential trustworthy cloud provider(s). Using our formal framework the capabilities are modelled in terms of propositional logic terms and these terms are associated with opinions. Alternatively, feedback based opinions can be given as an overall statement ($ol$) on the trustworthiness of the cloud provider.

In the given scenario (cf. Figure 6.5), the opinions (derived from expert assessment, CAIQ assessment, and accreditators) on the fulfilment of those propositions are combined using CertainLogic AND operator (i.e., $(CO \wedge DG \wedge FS \wedge IS)$). Users' opinions on the above mentioned attributes can be an overall rating ($ol$) or individual feedback on each of the attributes. A number of users' feedback on different attributes are assumed to be combined using consensus and discounting operators [Rie09b] and we denote the construction as $(ol, CO, DG, FS, IS)$ in Figure 6.5. The consensus operator is used to combine opinions from different recommenders and discounting operator is used to discount individual opinions according to the trustworthiness of those recommenders.

Finally, when combining the opinions from those different sources, a consumer may prefer one source over another. The consumer may give higher weights on $E$, $Q$ and $A$ than $F$ based on their preferences.

The fusion (aggregation) of opinions derived from different sources is especially challenging, as the differing sources' opinions may be conflicting. It may based on incomplete information or unreliable sources, and thus, it is subject to uncertainty. Therefore, the evaluation mechanism (i.e., fusion operation) should reflect the preferences, degree of conflict ($DoC$) and the uncertainty when combining multiple opinions (on propositions) to calculate the overall trustworthiness of Cloud 'A'.

### 6.4.2 Evaluation

In this section, we assume that the propositional logic terms (as shown in Figure 6.5) representing the trustworthiness of Cloud 'A' have already been evaluated using CertainLogic AND ($\wedge$) operator. Thus, we are in a situation where we have to aggregate four opinions ($Q$, $E$, $F$, and $A$) on the trustworthiness of Cloud 'A', i.e., we have to compute $\widehat{\oplus}_c(o_Q, o_E, o_F, o_A)$.

#### 6.4.2.1 Experimental Setup and Test Cases

The resulting four opinions are extracted in the following manner:

1. CAIQ assessment ($Q$): The resulting opinion on the trustworthiness of Cloud 'A' is extracted from their completed CAIQ published by CSA STAR. Our developed CCA tool is used for this purpose.

2. Accreditators ($A$): Accreditators use the CCA tool to assess the capabilities of Cloud 'A'. The resulting opinion ($A$) is then extracted based on the assessment. The opinion is represented using CertainTrust model.

3. Expert Assessment ($E$): The capabilities of Cloud 'A' are assessed by the experts leveraging the CCA tool. The resulting opinion is then derived using the CCA tool by the experts. The opinion is represented using CertainTrust model.

4. Feedback ($F$): The resulting opinion is extracted from other users' feedback on the trustworthiness of Cloud 'A' regarding their published capabilities. We assume that the feedback is represented using CertainTrust model.

In order to derive opinion $o_Q$, the published CAIQs in the STAR are considered. Hence, we use the resulting opinion (derived from CAIQs) documented in Table 6.6 for the experiments conducted in this section. The accreditators and experts may analyse the assertions of CAIQs in a different manner which results in two different outcomes (cf. Table 6.12 and Table 6.13). Thus, the resulting opinions ($o_A$ and $o_E$) derived from Cloud provider A's CAIQ profile are different than the opinion $Q$. The opinion

($o_F$) that resemble consumers feedback on the trustworthiness of Cloud 'A' are synthetically generated using the CCA tool.

Table 6.12: Cloud Control Assessment for Cloud 'A' (anonymised): *Accreditator* perspective

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $o_A$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | (0.8994,0.9938,0.9606);E=0.8998 |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 8 | 0 | 1 | 0 | 9 | (1,0.973,0.99) | |
| $IS$ | 72 | 3 | 0 | 1 | 75 | (0.96,1,0.99) | |

Table 6.13: Cloud Control Assessment for Cloud 'A' (anonymised): *Expert* perspective

| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $o_E$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 16 | 0 | 0 | 0 | 16 | (1,1,0.99) | (0.8003,0.9937,0.9606);E=0.8013 |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 1 | 1 | 0 | 9 | (0.875,0.973,0.99) | |
| $IS$ | 73 | 2 | 0 | 0 | 75 | (0.9733,1,0.99) | |

In the following experiments, we assume an initial expectation value ($f = f_Q = f_A = f_F = f_E = 0.1$), which reflects a rather pessimistic initial expectation of the consumers. Note that the consumer could either consider the calculated $f$ associated with the opinions or replace the calculated value with her own assumption.

We consider the following test cases for fusion experiments.

- Case 1: We conduct two experiments using average fusion and weighted fusion operators. The objective is to show the effect by considering

Table 6.14: Preferential Weights in Different Cases

| Test Cases | $w_Q$ | $w_A$ | $w_E$ | $w_F$ |
|---|---|---|---|---|
| Case 1 | 2 | 2 | 2 | 0.1 |
| Case 2 | 2 | 2 | 2 | 1 |
| Case 3 | 2 | 2 | 2 | 2 |
| Case 4 | 2 | 2 | 2 | 1 |

preferential weights when fusing opinions from different sources.

The hypothesis we want to test here is: *weighted fusion operator (W.FUSION) reflects variable preferential weights associated with the input opinions on the resulting fusioned opinion whereas average fusion operator (A.FUSION) does not reflect the variable weights on the resulting fusioned opinion.*

- Case 2: We conduct two experiments to demonstrate the comparison between weighted fusion and conflict-aware fusion operators. The main objective is to illustrate the capabilities in handling the conflicting opinions from different sources.

  The next hypothesis to be tested is: *conflict-aware fusion operator (C.FUSION) reflects the degree of conflict (DoC) among input opinions as well as the variable preferential weights associated with the input opinions on the resulting fusioned opinion whereas weighted fusion operator (W.FUSION) only reflects variable weights on the resulting fusioned opinion.*

- Case 3: In this case, two experiments are conducted to demonstrate the novelty of conflict-aware fusion operator over the state-of-the-art average fusion operator.

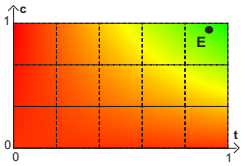
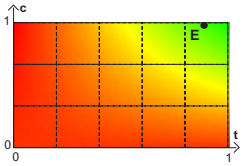
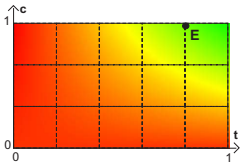
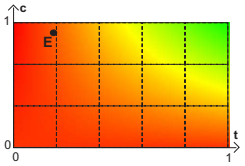
  The hypothesis we want to test here is: *conflict-aware fusion operator (C.FUSION) reflects degree of conflict (DoC) among input opinions as well as variable preferential weights associated with the input opinions on the resulting fusioned opinion whereas average fusion operator (A.FUSION) does not reflect variable weights as well as conflict among the input opinions on the resulting fusioned opinion.*

- Case 4: In the last case, the experiments are conducted to demonstrate a real world setting. In this setting, a user is interested to choose between a couple of cloud providers in terms of their trustworthiness.

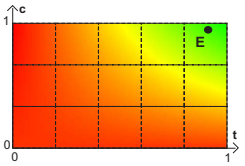
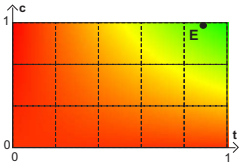
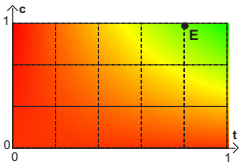
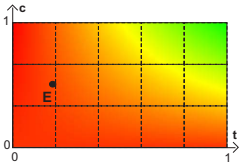
  The hypothesis we want to test here is: *conflict-aware fusion operator (C.FUSION) calculates more representative trustworthiness value compared to the trustworthiness value calculated using non conflict-aware fusion operators.*

Table 6.15: Opinions ($Q$, $A$, $E$, and $F$) on Trustworthiness of **Cloud 'A'**

(a) Opinions ($o_F$ with **high certainty**)

| | |
|---|---|
| $o_Q$ | $(0.9109, 0.9862, 0.1); E = 0.8997$  |
| $o_A$ | $(0.8994, 0.9938, 0.1); E = 0.8944$  |
| $o_E$ | $(0.8003, 0.9937, 0.1); E = 0.7959$  |
| $o_F$ | $(0.1943, 0.899, 0.1); E = 0.1843$  |

(b) Opinions ($o_F$ with **low certainty**)

| | |
|---|---|
| $o_Q$ | $(0.9109, 0.9862, 0.1); E = 0.8997$  |
| $o_A$ | $(0.8994, 0.9938, 0.1); E = 0.8944$  |
| $o_E$ | $(0.8003, 0.9937, 0.1); E = 0.7959$  |
| $o_F$ | $(0.1943, 0.50, 0.1); E = 0.1472$  |

#### 6.4.2.2 Experiments

In this section, a detail discussion on the conducted experiments is given.

**Case** 1. The objective here is to compare the *weighted fusion* operator with state-of-the-art *average fusion* operator in terms of handling preferential weights. We apply the *weighted fusion* operator to deal with variable weights given in Table 6.14 for Case 1. However, the average fusion operator is not designed to handle variable weights. In Table 6.16, user's preferences on the opinions, $o_Q$, $o_A$, and $o_F$, are reflected in the resulting opinion $(0.8606, 0.9922, 0.1)$ calculated using weighted fusion operator, but not in the resulting opinion $(0.8464, 0.99, 0.1)$ computed using average fusion operator. In the case of average fusion operation, all the opinions are given similar preference. The influence of variable weights are also evident in the expectation values, $E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8389$, $E(o_{\widehat{\oplus}_w(Q,E,A,F)}) = 0.8546$.

Table 6.16: Comparison between Average Fusion and Weighted Fusion Operators

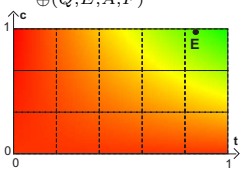
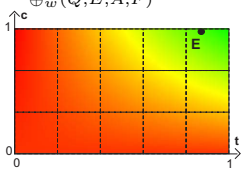
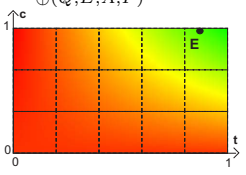
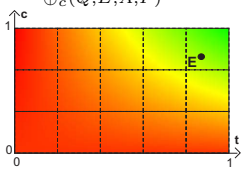
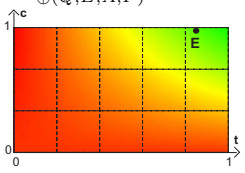
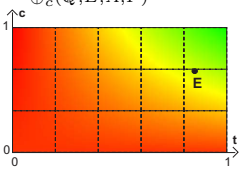
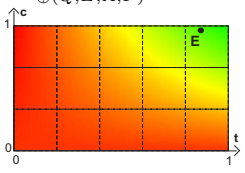
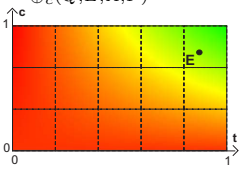
| Test Case | Cloud A: $o_{\widehat{\oplus}(Q,E,A,F)}$ | Cloud A: $o_{\widehat{\oplus}_w(Q,E,A,F)}$ |
|---|---|---|
| Case 1 | $(0.8464, 0.99, 0.1)$<br>$E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8389$<br>  | $(0.8606, 0.9922, 0.1)$<br>$E(o_{\widehat{\oplus}_w(Q,E,A,F)}) = 0.8546$<br>  |

Table 6.17: Comparison between Weighted Fusion and Conflict-aware Fusion Operators

| Test Case | Cloud A: $o_{\widehat{\oplus}_w(Q,E,A,F)}$ | Cloud A: $o_{\widehat{\oplus}_c(Q,E,A,F)}$ |
|---|---|---|
| Case 2 | $(0.8538, 0.9911, 0.1)$<br>$E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8470$<br>  | $(0.8538, 0.7562, 0.1)$<br>$E(o_{\widehat{\oplus}_c(Q,E,A,F)}) = 0.67$<br>  |

**Case 2.** The objective here is to compare the *conflict-aware fusion* operator with the *weighted fusion* operator in terms of handling conflicting opinions. From the previous experiment, we know that the weighted fusion operator is able to deal with variable weights. However, this operator is not able to handle *conflicts* among the opinions. It is evident when we apply *weighted* and *conflict-aware* fusion operator to aggregate the opinions given in Table 6.15. By applying conflict-aware fusion operator, the opinion surrounding trustworthiness of Cloud A is calculated as $(0.8538, 0.7562, 0.1)$, whereas by using the weighted fusion operator the opinion is calculated as $(0.8538, 0.9911, 0.1)$ in Table 6.17. The impact of the *conflict-aware fusion* is clearly visible on the certainty value 0.7562 compare to the certainty value 0.9911 calculated using *weighted fusion* operator. The expectation value $(E = 0.67)$ is also affected in the case of *conflict-aware* fusion compare to the value $(E = 0.8470)$ in *weighted fusion* when *conflict* among the opinions are taken into account.

Table 6.18: Comparison between Average Fusion and Conflict-aware Fusion Operators

| Test Case | Cloud A: $o_{\widehat{\oplus}(Q,E,A,F)}$ | Cloud A: $o_{\widehat{\oplus}_c(Q,E,A,F)}$ |
|---|---|---|
| Case 3 ($o_F$ with **high certainty**) | $(0.8464, 0.99, 0.1)$ $E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8470$   | $(0.8464, 0.6560, 0.1)$ $E(o_{\widehat{\oplus}_c(Q,E,A,F)}) = 0.5896$   |
| Case 3 ($o_F$ with **low certainty**) | $(0.8596, 0.9898, 0.1)$ $E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8518$   | $(0.8596, 0.7882, 0.1)$ $E(o_{\widehat{\oplus}_c(Q,E,A,F)}) = 0.6883$   |

**Case 3.** The objective in this case is to demonstrate the novelty of the *conflict-aware fusion* operator compare to the *average fusion* operator. Note that the average fusion operator is equivalent (cf. Appendix D.9) to the state-of-the-art fusion operator for dealing with dependent opinions in *subjective logic*. As an average fusion operator is not designed to deal with preferential weights, we consider the same weights for the opinions in this case.

We apply average fusion and conflict-aware fusion operators to combine the opinions given in Table 6.15. In Table 6.18, aggregated opinions on the trustworthiness of Cloud A are $(0.8464, 0.99, 0.1)$ (average fusion) and $(0.8464, 0.6560, 0.1)$ (conflict-aware fusion). The impact of the *conflict-aware* operator is clearly visible on the certainty value 0.6560 compared to the certainty value 0.99. The impact is mainly due to the conflicting opinion ($o_F = (0.1943, 0.899, 0.1)$) given by the users with high certainty. In the second row, the impact is low on the certainty value (0.7882) due to the conflicting opinion ($o_F = (0.1943, 0.50, 0.1)$) given by the users with low certainty (cf. Table 6.15(b)). The expectation values, $E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8470$; $E(o_{\widehat{\oplus}_c(Q,E,A,F)}) = 0.5896$, are also affected due to the same reasons that affect the certainty values. In the second row, the expectation values, $E(o_{\widehat{\oplus}(Q,E,A,F)}) = 0.8518$; $E(o_{\widehat{\oplus}_c(Q,E,A,F)}) = 0.6883$, are slightly better compare to the values in the first row. This is due to the conflicting opinion ($o_F = (0.1943, 0.50, 0.1)$) given by the users with low certainty (cf. Table 6.15(b)).

We conclude that the *conflict-aware fusion* operator provides the most

representative assessment of Cloud A's trustworthiness. Thus, this operator is best suited among the three operators that we have discussed. Note that the fusion operators in *subjective logic* do not consider preferential weights and conflicts when aggregating dependent opinions. Therefore, *conflict-aware fusion* operator is a better choice than the fusion operators in *subjective logic* when one requires the most representative trust assessment under uncertainty, conflict and personal preferences.

Table 6.19: Cloud Control Assessment for **Cloud 'S'** (anonymised): customised case ($o_Q$)

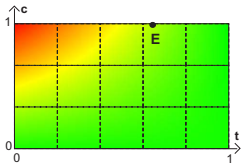
| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $(t, c, f)$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 13 | 1 | 0 | 2 | 14 | (0.9286,1,0.99) | (0.6366,0.9998,0.9606);E=0.6367   |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 2 | 0 | 0 | 9 | (0.7778,1,0.99) | |
| $IS$ | 63 | 4 | 3 | 5 | 70 | (0.9403,0.9987,0.99) | |

Table 6.20: Cloud Control Assessment for **Cloud 'S'** (anonymised): *Accreditator* perspective ($o_A$)

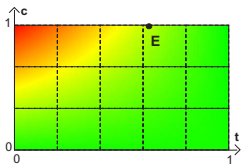
| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $o_A$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 11 | 1 | 2 | 2 | 14 | (0.9167,0.9767,0.99) | (0.6277,0.9957,0.9606);E=0.6291   |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 2 | 0 | 0 | 9 | (0.7778,1,0.99) | |
| $IS$ | 62 | 4 | 4 | 5 | 70 | (0.9394,0.9983,0.99) | |

Table 6.21: Cloud Control Assessment for **Cloud 'S'** (anonymised): *Expert* perspective ($o_E$)

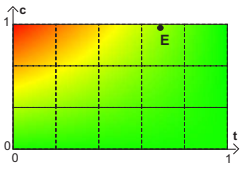
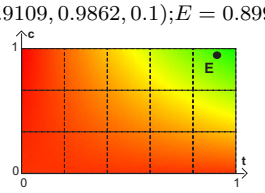
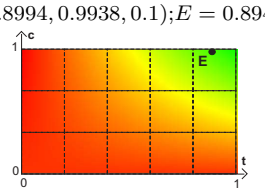
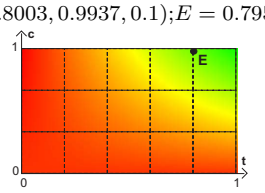
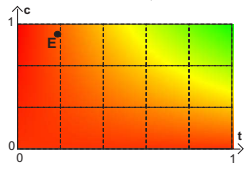
| Domains | $r$ | $s$ | $u$ | $NA$ | $N$ | Resulting Opinion $(t, c, f)$ | Final Assessment $o_E$;$E$ |
|---|---|---|---|---|---|---|---|
| $CO$ | 11 | 3 | 0 | 2 | 14 | (0.7857,1,0.99) | (0.6894,0.9893,0.9606);E=0.6923  |
| $DG$ | 15 | 1 | 0 | 0 | 16 | (0.9375,1,0.99) | |
| $FS$ | 7 | 0 | 2 | 0 | 9 | (1,0.9403,0.99) | |
| $IS$ | 63 | 4 | 3 | 5 | 75 | (0.9403,0.9987,0.99) | |

Table 6.22: Opinions ($Q$, $A$, $E$, and $F$) on Trustworthiness of Cloud providers

(a) Opinions on *Cloud provider 'A'*

| | |
|---|---|
| $o_Q$ | $(0.9109, 0.9862, 0.1); E = 0.8997$  |
| $o_A$ | $(0.8994, 0.9938, 0.1); E = 0.8944$  |
| $o_E$ | $(0.8003, 0.9937, 0.1); E = 0.7959$  |
| $o_F$ | $(0.1943, 0.899, 0.1); E = 0.1843$  |

(b) Opinions on *Cloud provider 'S'*

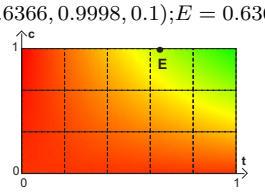
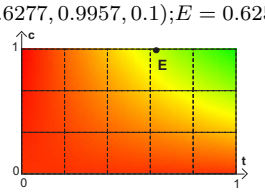
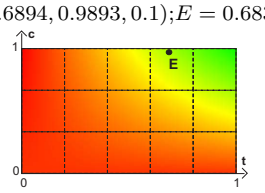
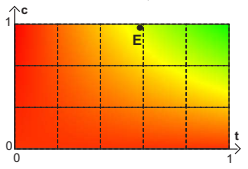
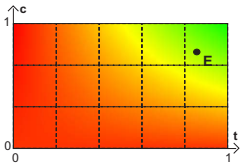
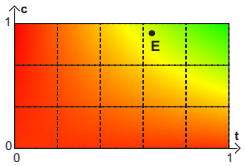
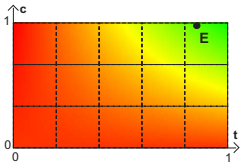
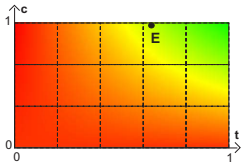
| | |
|---|---|
| $o_Q$ | $(0.6366, 0.9998, 0.1); E = 0.6365$  |
| $o_A$ | $(0.6277, 0.9957, 0.1); E = 0.6254$  |
| $o_E$ | $(0.6894, 0.9893, 0.1); E = 0.6831$  |
| $o_F$ | $(0.5943, 0.988, 0.1); E = 0.5884$  |

Table 6.23: Cloud 'A' Vs. Cloud 'S'

| *Conflict-aware* | Cloud provider 'A': $o_{\widehat{\oplus_c}(Q,A,E,F)}$ | Cloud provider 'S': $o_{\widehat{\oplus_c}(Q,A,E,F)}$ |
|---|---|---|
| Fusioned Opinion | $(0.8538, 0.7562, 0.1)$ | $(0.6368, 0.9604, 0.1)$ |
| Expectation value | $E = 0.67$   | $E = 0.6155$   |
| Degree of Conflict | $DoC = 0.2369$ | $DoC = 0.0389$ |
| *Non conflict-aware* | Cloud provider 'A': $o_{\widehat{\oplus_w}(Q,A,E,F)}$ | Cloud provider 'S': $o_{\widehat{\oplus_w}(Q,A,E,F)}$ |
| Fusioned Opinion | $(0.8538, 0.9911, 0.1)$ | $(0.6368, 0.0.9993, 0.1)$ |
| Expectation value | $E = 0.8470$   | $E = 0.6364$   |

**Case** 4. In a real world setting, one would assume that a user can choose between a couple of cloud providers, e.g., Cloud provider 'A' and 'S'. In this case, we propose to sort the cloud providers based by their expectation value ($E$) and using the $DoC$ as a second criteria if necessary. The opinions considered to calculate the fusioned opinion of Cloud provider 'A' and Cloud provider 'S' are given in Table 6.22. Table 6.19, 6.20 and 6.21 document the detailed CAIQ assessment of Cloud provider 'S' from the perspective of consumers, accreditators and experts respectively.

In the first set of experiments, *conflict-aware fusion* is applied to aggregate the opinions on trustworthiness of Cloud provider 'A' and 'S' given in Table 6.22(a) and 6.22(b) respectively. According to the calculated expectation values, Cloud provider 'A' is better ranked than Cloud provider 'S'. This comes from the fact that the given assertions by provider 'A' are more *positive* ($t = 0.8538$) than that of provider 'S' ($t = 0.6368$) regarding the required capabilities (CO, DG, FS, IS). Even though the certainty value (0.9604) indicates better representativeness of average rating of Cloud 'S', it is not enough for provider 'S' to receive a better expectation value ($E$) than provider 'A'. Additionally, the value, $DoC = 0.0389$ of provider 'S' indicates that the opinions considered for fusion operation is less conflicting than that of the opinions considered for fusing in the case of provider 'A' ($DoC = 0.2369$).

In the second set of experiments, the expectation value (0.8470) of Cloud

provider 'A' is way better than that of the Cloud 'S' ($E = 0.6364$). This is due to the *weighted fusion* operator which is not able to handle conflicts among the opinions. Thus, conflicting opinions does not influence $E$ when *weighted fusion* operation is considered for trustworthiness evaluation. Now, if we compare the results derived using *non* conflict-aware fusion (weighted fusion) operator with the results derived using conflict-aware fusion, we see that additional information, e.g., $DoC$ and a *representative* certainty ($c$) value, is available in the latter case. These additional information supports users to reason about the capabilities of cloud providers in a more reliable and transparent manner compared to the existing approaches discussed in this section.

We conclude that the *conflict-aware* fusion operator is more desirable than the non conflict-aware operators when combining conflicting opinions as well as opinions associated with variable weights and the opinions are derived from multiple sources.

## 6.5 Summary

This chapter demonstrates the applicability of novel trust establishment mechanisms in different application scenarios as well as in real world settings. The developed mechanisms show that the requirements, outlined in Chapter 3, are fulfilled when novel mechanisms are applied to establish trust in distributed service environments, particularly in cloud computing environments. The evaluation can be summarized as follows:

- The first two experiments demonstrate that the *CertainLogic* logical operators along with the formal framework are able to aggregate opinions about multiple attributes and customise the trustworthiness values according to the consumers' preference. In particular, the second experiment demonstrates that the proposed operators are also able to evaluate the trustworthiness of a composite service and system, based on the knowledge of their architecture and the trustworthiness of their components and subsystems. Moreover, the operators are designed to deal with *uncertainty* when aggregating opinions as well as reflect it in the resulting trustworthiness values in order to support users for reliable decision making.

- The third experiment demonstrates that the novel CertainLogic fusion operator (i.e., $C.FUSION$) is able to aggregate opinions that are derived from multiple sources. In particular, the fusion operator is designed to deal with conflicting opinions. Additionally, the novel operator considers consumer preferences when deriving opinions from multiple sources. The experiments also demonstrated the impact of

conflicting opinions as well as consumers' preferential attachment with the opinions in the resulting trustworthiness value.

# 7 Conclusions and Outlook

Composite distributed services in the emerging service environments, e.g., cloud computing, are increasingly becoming a reality. This paradigm shift in service environments enable providers to offer more dynamic, scalable and cost-effective services than ever before. Consequently, consumers are facing considerable obstacles in such a complex setting to distinguish among the service providers based on their service-specific attributes. This is due to the lack of an integrated solution for assessing and evaluating the trustworthiness of service providers based on their published attributes. Moreover, the solution should eventually support consumers to distinguish the service providers based on their calculated trustworthiness value.

## 7.1 Conclusions

This thesis contributes to a new architecture of *trust management system* for allowing service providers to represent their service-specific attributes and consequently enable customers to assess and evaluate the trustworthiness of service providers according to their requirements. Additionally, we contribute to the required novel mechanisms for developing the proposed trust system and demonstrate its applicability in the context of cloud computing marketplaces.

In this thesis, requirements for designing trust systems in distributed service environments have been outlined in Section 3.1, which serve as a guideline to analyse and discuss the state-of-the-art systems and mechanisms in various complementary application domains. According to the discussion in Chapter 3, none of the existing trust systems comply with *all* the requirements.

However, a couple of non-application specific trust mechanisms, Subjective Logic and CertainTrust, comply with non-functional requirement, $NR1$. CertainTrust model is a better choice in terms of its flexible and simple representational model, intuitive graphical trust representation ($FR4$) and attack-resistant ($NR3$) operators for robust trust aggregation. Thus, the following core contributions are made in order to comply with rest of the requirements, which are essential for developing a trust system for cloud computing marketplaces.

- *QoS+* provides a list of service-specific attributes that potentially contributes to the trustworthiness of service providers in cloud computing marketplaces. Additionally, we contribute to the identification of different sources that provide information regarding those attributes as well as different means for deriving those information. We conclude that the multi-faceted nature of *QoS+* attributes demonstrate the necessity to formulate the trustworthiness of service providers regarding those attributes in a generalised and simplified manner.

- *Formal framework* is designed and constructed to assess composite service architecture regarding their specific attributes. The framework allows composite service architecture and service-specific attributes to represent in simplified meaningful terms that we refer to as *PLT*s. The PLTs serve as a formal basis to evaluate the trustworthiness of service providers in distributed service environments. Particularly, the proposed framework consider consumer requirements to enable customisation of propositions in the *PLT*s. This in turn enables the proposed trust system to customise the evaluation of trustworthiness according to consumer requirements. The significant advantage of integrating the formal framework in the trust system architecture is that the framework is able to provide an abstract representation of underlying service architecture without analysing the service methods, i.e., functionality of a service it supports. We conclude that the integration of the formal framework in the proposed trust management system allows to fulfil the requirements $FR2$ and $FR3$, which are outlined in Section 3.1 for developing trust systems in distributed service environments.

- *CertainLogic*, a framework that contains computational operators that operates on trust information (i.e., opinions) about multiple attributes derived from multiple sources. The operators are able to operate on CertainTrust's representation of opinions, which are subject to uncertainty and when derived from multiple sources the opinions might be conflicting. The significant advantage of using the CertainLogic operators in the proposed trust system is that the definition of the operators enable aggregation of opinions independent of how opinions

are assessed. Note that the opinions are aggregated based on the propositional structure in the PLTs because an opinion is always associated with a proposition. We conclude that the integration of CertainLogic framework in the proposed trust management system allows to fulfil the requirements $FR1$ (except the *multi-context* feature), $FR3$, and $NR2$, which are outlined in Section 3.1 for developing trust systems in distributed service environments.

The evaluation of our contributions demonstrate their applicability in an instantiated scenario of distributed service environments. However, the generalised formalisation and the established mathematical foundations behind our contributions can lead their applicability to other complemented scenarios. Moreover, the trust establishment mechanisms that are presented and evaluated in this thesis lead to close the gaps identified in the state-of-the-art trust systems. Consequently, these mechanisms are valuable and significant in the field of evidence-based trust systems.

## 7.2 Outlook

The proposed architecture of the trust management system provides a first step towards assessment and evaluation of the trustworthiness of service providers in cloud marketplaces. In this thesis, we have provided the related concepts and proposed novel mechanisms for trust management architecture and proof-of-concept prototype. Beyond the contributions in this thesis, the following challenges are open for future research.

### Trust-aware Validation of Opinions

Our current realisation of the proposed trust management system assumes that opinions derived from a source, e.g., CSA's STAR, are based on truthful pieces of evidence. The assumption is realistic due to the CSA CloudAudit working group's proposal for integrating automated audit framework [CSAa] to validate the pieces of evidence. However, the current state of the CSA STAR as it stands does not provide such integration as well as the CloudAudit framework is in its early stage of development. Here, the trusted computing based techniques such as property based attestation would be interesting to investigate for enabling validation of attributes claimed by the service providers in the CSA STAR. In this regard, CSA's Cloud Trust Protocol (CTP) project [CSAb] would be also interesting to consider as the project proposal promises to develop mechanisms that will enable consumers to find out relevant pieces of information concerning security attributes. These pieces of information might serve as means to validate the opinions derived from the CSA STAR.

### Attack-resistant Trust Management

The proposed architecture of the trust management system relies on Sybil attack-resistant computational operators proposed in [RA09]. These operators are designed to mitigate the influence of Sybil recommenders when combining pieces of evidence from a number of users, and thus, the operators are applicable to recommender networks. The attack-resistant operators serve as means to protect the manipulation of opinions which in the end supports our proposed system architecture to deal with manipulated opinions derived from user feedback. When hosting such a system online, one should definitely consider designing attacks models and mitigation mechanisms in order to safe guard the proposed trust management system from known attacks. Moreover, the Trust Update Engine (TUE) module in the proposed system architecture allow multiple sources to provide opinions about service-specific attributes of service providers. Taking into account the hostile online environments, one should also consider novel authorisation mechanisms in order to ensure the reliability of the proposed trust management system. For example, compromised user platforms, i.e., hosting malicious software applications, can render the entire process of trust management useless by generating false opinions. Traditional user-centric authorisation mechanisms are only able to verify the identity of a source and necessary access control policies allowed for that source. In order to address the malicious behavior of the platforms in the authorisation process, existing trust enhanced mechanisms [Nag10] would be interesting to investigate.

### Multi-context aware Trust Mechanisms

Computational trust mechanisms underlying our proposed trust management system consider *context* as an embedded information given in a proposition. For example, "Alice expects that storage Service provider B has the capability to provide data protection"; where "storage service" is a SaaS context. Based on the derived opinions, one is only able to reason about the trustworthiness of a service in the given context. However, the proposed mechanisms in this thesis are not able to deal with cases where one has to reason about the trustworthiness of a service provider, who offers services in different contexts. For example, provider B also provides a database service and Alice expects that the provider B also has the capability to provide data protection. As the contexts are not the same, one cannot conclude that the provider B has the same level of trustworthiness regarding data protection attributes in both contexts. Reasoning overall trustworthiness of a service provider in such a multi-context scenario is non-trivial. Thus, future research should focus on developing computational trust mechanisms that are adaptive to multiple contexts as well as are able to transfer trust across those contexts.

### Generalisation of CertainLogic

The CertainLogic operators provided in this thesis are able to combine opinions about independent and dependent propositions. We have demonstrated that these operators allow one to combine an opinion about a security attribute with an attribute related to performance, e.g., availability. However, in a real-world setting, attributes such as latency or availability of a service might depend on the security attributes such as DDoS vulnerability or insecure encryption mechanism. In such a case, one has to consider the level of dependency between the attributes and how the dependency level can be transferred to the computation of opinions. Our proposed operators for combining opinions are not able to deal with partial dependency between the attributes. Thus, CertainLogic operators requires generalised definitions to deal with partial dependency between attributes and mechanisms for combining associated opinions.

# Bibliography


[3Te09] 3Tera Applogic. 3tera's Cloud Computing SLA goes live, March 31 2009.

[Aba09] Jemal Abawajy. Determining service trustworthiness in intercloud computing environments. *Int. Symposium on Parallel Architectures, Algorithms, and Networks*, 0:784–788, 2009.

[AFG+09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[AG07] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Web Semant.*, 5(2):58–71, 2007.

[BFIK99] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, chapter The Role of Trust Management in Distributed Systems Security, pages 185–210. Springer, 1999.

[BFK98] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. Keynote: Trust management for public-key infrastructures. In *Infrastructures (Position Paper). Lecture Notes in Computer Science 1550*, pages 59–63, 1998.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy (SP '96)*, pages 164 – 173. IEEE Computer Society, 1996.

[BK05] Robert Blakley and Gregory Kabatiansky. *Encyclopedia of Cryptography and Security.* Springer, 2005.

[BKL07] Samuel J. Best, Brian S. Krueger, and Jeffrey Ladewig. The effect of risk perceptions on online political participatory deci-

sions. *Journal of Information Technology and Politics*, 4:5–17, 2007.

[BLB04] Sonja Buchegger and Jean-Yves Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*, 2004.

[Bol04] William M. Bolstad. *Introduction to Bayesian Statistics*. John Wiley & Sons, Inc, 2004.

[BPC07] Gleb Beliakov, Ana Pradera, and Tomasa Calvo. Averaging functions. In *Aggregation Functions: A Guide for Practitioners*, volume 221 of *Studies in Fuzziness and Soft Computing*, pages 39–122. Springer Berlin / Heidelberg, 2007.

[CA ] CA technologies. CA Nimsoft Monitor. `http://www.nimsoft.com/solutions/nimsoft-monitor/cloud.html` Accessed Feb 14 2013.

[Cap] Captcha. Captcha Project. `http://www.captcha.net/` Accessed Feb 12 2013.

[CDC08] E. Chang, T. Dillon, and D. Calder. Human system interaction with confident computing. the mega trend. In *Human System Interactions, 2008 Conference on*, pages 1 –11, may 2008.

[CFL+97] Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Martin Strauss. Referee: trust management for web applications. *Comput. Netw. ISDN Syst.*, 29:953–964, September 1997.

[Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.

[CHD05] Elizabeth Chang, Farookh Hussain, and Tharam Dillon. *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence*. John Wiley & Sons, 2005.

[Clo10] Cloud Computing Use Case Discussion Group. Cloud computing use cases white paper- introducing slas. In *Technical Report*. Cloud Computing Use Case Discussion Group, 2010. `http://cloudusecases.org/`.

[CSAa] CSA. Cloud Audit. `https://cloudsecurityalliance.org/research/cloudaudit/` Accessed Jan 05 2013.

[CSAb] CSA. Cloud Trust Protocol (CTP). `https://cloudsecurityalliance.org/research/ctp/` Accessed Jan 31 2013.

[CSAc] CSA. Consensus Assessments Initiative (CAI) Questionnaire. `https://cloudsecurityalliance.org/research/cai/` Accessed Feb 20 2013.

[CSAd] CSA. Security, Assurance & Trust Registry (STAR). `https://cloudsecurityalliance.org/star/` Accessed Feb 20 2013.

[CSAe] CSA. Security, Assurance & Trust Registry (STAR) FAQ. `https://cloudsecurityalliance.org/star/faq/` Accessed Feb 20 2013.

[CSG+03] Vinny Cahill, Brian Shand, Elizabeth Gray, Nathan Dimmock, Andy Twigg, Jean Bacon, Colin English, Waleed Wagealla, Sotirios Terzis, Paddy Nixon, Ciaran Bryce, Giovanna di Marzo Serugendo, Jean-Marc Seigneur, Marco Carbone, Karl Krukow, Christian Jensen, Yong Chen, and Mogens Nielsen. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2/3:52–61, 2003.

[FHT+12] Ana Juan Ferrer, Francisco Hernández, Johan Tordsson, Erik Elmroth, Ahmed Ali-Eldin, Csilla Zsigri, Raül Sirvent, Jordi Guitart, Rosa M. Badia, Karim Djemame, Wolfgang Ziegler, Theo Dimitrakos, Srijith K. Nair, George Kousiouris, Kleopatra Konstanteli, Theodora Varvarigou, Benoit Hudzia, Alexander Kipp, Stefan Wesner, Marcelo Corrales, Nikolaus Forgó, Tabassum Sharif, and Craig Sheridan. Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, 28(1):66 – 77, 2012.

[Fuj10] Fujitsu Research Institute. Personal data in the cloud: A global survey of consumer attitudes, 2010.

[Gam90] Diego Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, New York, 1990.

[Gam00] Diego Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations, electronic edition*, chapter 13, pages 213–237. 2000.

[GBS08] Saurabh Ganeriwal, Laura K. Balzano, and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, 4(3):1–37, 2008.

[Gol05] J. Golbeck. *Computing and Applying Trust in Web-Based Social Networks*. PhD thesis, University of Maryland, USA, 2005.

[Gra03] Tyrone Grandison. *Trust Management for Internet Applications*. PhD thesis, Imperial College London, UK, 2003.

[Gra07] Tyron Grandison. *Conceptions of Trust: Definition, Constructs, and Models*, chapter 1, pages 1–28. Trust in E-Services: Technologies, Practices and Challenges. IGI Global, 2007.

[GS00] Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16, 2000.

[GS03] Tyrone Grandison and Morris Sloman. Trust management tools for internet applications. In Paddy Nixon and Sotirios Terzis, editors, *iTrust*, volume 2692 of *Lecture Notes in Computer Science*, pages 91–107. Springer, 2003.

[GvL05] I. Veljkovic G. von Laszewski, B. E. Alunkal. Towards reputable grids. *Scalable Computing: Practice and Experience*, 6(3):95–106, 2005.

[HAP+10] Irfan Ul Haq, Rehab Alnemr, Adrian Paschke, Erich Schikuta, Harold Boley, and Christoph Meinel. Distributed trust management for validating sla choreographies. In *Grids and Service-Oriented Architectures for Service Level Agreements*, pages 45–55. Springer US, 2010.

[HBS10] Irfan Ul Haq, Ivona Brandic, and Erich Schikuta. Sla validation in layered cloud infrastructures. In *GECON*, Lecture Notes in Computer Science, pages 153–164. Springer-Verlag, 2010.

[HHRM12] Sheikh Mahbub Habib, Sascha Hauke, Sebastian Ries, and Max Mühlhäuser. Trust as a facilitator in cloud computing: a survey. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(19):19, 2012.

[HJS04] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In Ramon López de Mántaras and Lorenza Saitta, editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI)*, pages 18–22. IOS Press, 2004.

[HJS06] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[HKS00] T. Hofmeister, M. Krause, and H.U. Simon. Optimal k out of n secret sharing schemes in visual cryptography. *Theoretical Computer Science*, 240:471–485, 2000.

[HMM+00] Amir Herzberg, Yosi Mass, Joris Michaeli, Dalit Naor, and Yiftach Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *In Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 2–14. IEEE Computer Society Press, 2000.

[HRHM12a] Sheikh Mahbub Habib, Sebastian Ries, Sascha Hauke, and Max Mühlhäuser. Fusion of opinions under uncertainty and conflict – application to trust assessment for cloud marketplaces. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 109 –118, June 2012.

[HRHM12b] Sheikh Mahbub Habib, Sebastian Ries, Sascha Hauke, and Max Mühlhäuser. Fusion of opinions under uncertainty and conflict – trust assessment for cloud marketplaces. Technical report TUD-CS-2012-0027, Technische Universität Darmstadt, 2012.

[HRM10] Sheikh Mahbub Habib, Sebastian Ries, and Max Mühlhäuser. Cloud computing landscape and research challenges regarding trust and reputation. *Symposia and Workshops on ATC/UIC*, 0:410–415, 2010.

[HRM11] Sheikh Mahbub Habib, Sebastian Ries, and Max Mühlhäuser. Towards a trust management system for cloud computing. *IEEE TrustCom/IEEE ICESS/FCST, International Joint Conference of*, 0:933–939, 2011.

[HRMV13] Sheikh Mahbub Habib, Sebastian Ries, Max Mühlhäuser, and Prabhu Varikkattu. Towards a trust management system for cloud computing marketplaces: using caiq as a trust information source. *Security and Communication Networks*, pages 1–16, 2013.

[HS05] Michael N. Huhns and Munindar P. Singh. *Service-Oriented Computing: Key Concepts and Principles*, volume 9. IEEE Educational Activities Department, Piscataway, NJ, USA, January 2005.

[HS11] Chung-Wei Hang and Munindar P. Singh. Trustworthy service selection and composition. *ACM Trans. Auton. Adapt. Syst.*, 6:5:1–5:17, February 2011.

[HVHM12] Sascha Hauke, Florian Volk, Sheikh Mahbub Habib, and Max Mühlhäuser. Integrating indicators of trustworthiness into reputation-based trust models - insurance, certification, and coalitions. In *IFIPTM*, pages 158–173, 2012.

[HVM13a] Sheikh Mahbub Habib, Vijay Varadharajan, and Max Mühlhäuser. A framework for evaluating trust of service providers in cloud marketplaces. In *28th ACM SAC*, volume 2, pages 1963–1965. ACM Press, March 2013.

[HVM13b] Sheikh Mahbub Habib, Vijay Varadharajan, and Max Mühlhäuser. A trust-aware framework for evaluating security controls of service providers in cloud marketplaces. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 IEEE 11th International Conference on*, pages 459–468, July 2013.

[HWS09] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2009.

[ITU97] ITU-T. ITU-T recommendation X.509, the directory: Authentication framework. http://www.itu.int/rec/T-REC-X.509-199708-S/e, June 1997.

[JG09] Audun Jøsang and Jennifer Golbeck. Challenges for robust of trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (STM 2009)*, 2009.

[JHF03] Audun Jøsang, Shane Hird, and Eric Faccer. Simulating the effect of reputation systems on e-markets. In *Proceedings of the First International Conference on Trust Management (iTrust'03)*, pages 179–194, 2003.

[JI02] Audun Jøsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[JIB07] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[JKD05] Audun Jøsang, Claudia Keser, and Theodosis Dimitrakos. Can we manage trust? In *iTrust*, pages 93–107. Springer, 2005.

[JM05] Audun Jøsang and David McAnally. Multiplication and comultiplication of beliefs. *International Journal of Approximate Reasoning*, 38(1):19–51, 2005.

[JMP06] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring different types of trust propagation. In *In Proceedings of the 4th International Conference on Trust Management (iTrust)*, 2006.

[Jon99] S. Jones. Trust-ec: Requirements for trust and confidence in e-commerce. Technical Report EUR 18749 EN, European Communities EUR Report, April 1999.

[Jøs99] Audun Jøsang. Trust-based decision making for electronic transactions. In L. Yngström and T. Svensson, editors, *Proceedings of the Fourth Nordic Workshop on Secure IT Systems (NORDSEC'99)*, 1999.

[Jøs01] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

[Jøs07] Audun Jøsang. Probabilistic logic under uncertainty. In *Proceedings of the thirteenth Australasian symposium on Theory of computing - Volume 65*, CATS '07, pages 101–110, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[Jøs09] Audun Jøsang. Fission of opinions in subjective logic. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 1911 –1918, july 2009.

[JW02] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[KC98] A. Kini and J. Choobineh. Trust in electronic commerce: definition and theoretical considerations. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, volume 4, pages 51 –61 vol.4, jan 1998.

[KC09] Reid Kerr and Robin Cohen. Smart cheaters do prosper: defeating trust and reputation systems. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 993–1000, Richland, SC, 2009. IFAAMAS.

[KM10] Khaled M. Khan and Qutaibah Malluhi. Establishing trust in cloud computing. *IT Professional*, 12:20–27, 2010.

[Kra09] F. John Krautheim. Private virtual infrastructure for cloud computing. In *Proceedings of the HotCloud'09*, pages 5–5, Berkeley, CA, USA, 2009. USENIX Association.

[KSGM03] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th international conference on World Wide Web*, pages 640–651. ACM Press, 2003.

[KV11] Aarthi Krishna and Vijay Varadharajan. A hybrid trust model for authorisation using trusted platforms. In *Proceedings of the 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, TRUSTCOM '11, pages 288–295, Washington, DC, USA, 2011. IEEE Computer Society.

[LGO10] Bin Li, Lee Gillam, and John OŠLoughlin. Towards application-specific service level agreements: Experiments in clouds and grids. In Nick Antonopoulos and Lee Gillam, editors, *Cloud Computing*, volume 0 of *Computer Communications and Networks*, pages 361–372. Springer London, 2010.

[LJ10] Haifei Li and Jun-Jang Jeng. Ccmarketplace: a marketplace model for a hybrid cloud. In *Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, CASCON '10, pages 174–183, Riverton, NJ, USA, 2010. IBM Corp.

[MC96] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. Technical report, Management Information Systems Research Center, University of Minnesota, USA, 1996.

[MMA+01] Lik Mui, Mojdeh Mohtashemi, Cheewee Ang, Peter Szolovits, and Ari Halberstadt. Ratings in distributed systems: A Bayesian approach. In *Workshop on Information Technologies and Systems*, 2001.

[Nag10] Aarthi Nagarajan. *Techniques for Trust Enhanced Distributed Aurthorisation using Trusted Platforms*. PhD thesis, Macquarie University, August 2010.

[Nat12a] National Information Assurance Partnership. Common criteria for information technology security evaluation evaluation methodology 3.1 revision 4. Technical Report CCMB-2012-09-004, NIAP, USA, September 2012.

[Nat12b] National Information Assurance Partnership. Common criteria for information technology security evaluation part 1: Introduction and general model version 3.1 revision 4. Technical Report CCMB-2012-09-001, NIAP, USA, September 2012.

[Nat12c] National Information Assurance Partnership. Common criteria for information technology security evaluation part 2: Security functional components version 3.1 revision 4. Technical Report CCMB-2012-09-002, NIAP, USA, September 2012.

[Nat12d] National Information Assurance Partnership. Common criteria for information technology security evaluation part 3: Security assurance components version 3.1 revision 4. Technical Report CCMB-2012-09-003, NIAP, USA, September 2012.

[Nis99] Helen Nissenbaum. Can trust be secured online? a theoretical perspective. Number 2 in Etica & Politica / Ethics & Politics I. EUT Edizioni Università di Trieste, 1999.

[NV11] Aarthi Nagarajan and Vijay Varadharajan. Dynamic trust enhanced security model for trusted platform based services. *Future Gener. Comput. Syst.*, 27:564–573, May 2011.

[oD85] Department of Defense. Trusted computer system evaluation criteria (orange book). Technical Report DoD 5200.28-STD, December 1985.

[PRNZ12] P.S. Pawar, M. Rajarajan, S.Krishnan Nair, and A. Zisman. Trust model for optimized cloud services. In *Trust Management VI*, volume 374 of *IFIP Advances in Information and Communication Technology*, pages 97–112. Springer Berlin Heidelberg, 2012.

[PSHW04] Jonathan Poritz, Matthias Schunter, Els Van Herreweghen, and Michael Waidner. Property attestation–scalable and privacy-friendly security assessment of peer computers. Technical Report Report RZ 3548 (# 99559), IBM Research, October 2004.

[RA09] S. Ries and E. Aitenbichler. Limiting sybil attacks on bayesian trust models in open soa environments. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC '09. Symposia and Workshops on*, pages 178 –183, july 2009.

[RH08] Sebastian Ries and Andreas Heinemann. Analyzing the robustness of certaintrust. In Yücel Karabulut, John Mitchell, Peter Herrmann, and Christian Jensen, editors, *Trust Management II*, volume 263 of *IFIP Advances in Information and Communication Technology*, pages 51–67. Springer Boston, 2008.

[RHJ04] Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.

[RHMV11a] Sebastian Ries, Sheikh Mahbub Habib, Max Mühlhäuser, and Vijay Varadharajan. Certainlogic: A logic for modeling trust and uncertainty. In *Trust and Trustworthy Computing*, volume 6740 of *Lecture Notes in Computer Science*, pages 254–261. Springer Berlin / Heidelberg, 2011.

[RHMV11b] Sebastian Ries, Sheikh Mahbub Habib, Max Mühlhäuser, and Vijay Varadharajan. Certainlogic: A logic for modeling trust and uncertainty. Technical Report TUD-CS-2011-0104, Technische Universität Darmstadt, 2011.

[Rie09a] Sebastian Ries. Extending bayesian trust models regarding context-dependence and user friendly representation. In *Proceedings of the ACM SAC*, pages 1294–1301, New York, NY, USA, 2009. ACM.

[Rie09b] Sebastian Ries. *Trust in Ubiquitous Computing*. PhD thesis, Technische Universität Darmstadt, 2009.

[RS08] Sebastian Ries and Daniel Schreiber. Evaluating user representations for the trustworthiness of interaction partners. In *International Workshop on Recommendation and Collaboration (ReColl '08) in conjunction with the International Conference on Intelligent User Interfaces (IUI'08)*. ACM Press, 2008.

[Sab03] Jordi Sabater. *Trust and reputation for agent societies*. PhD thesis, Universitat Autnoma de Barcelona, Spain, 2003.

[Sea09] SearchCIO. Amazon gets SAS 70 Type II audit stamp, but analysts not satisfied, Nov 17 2009.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton Univ. Press. Princeton, NJ, 1976.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[SMV+10] Joshua Schiffman, Thomas Moyer, Hayawardh Vijayakumar, Trent Jaeger, and Patrick McDaniel. Seeding clouds with trust anchors. In *Proceedings of the ACM CCSW '10*, pages 43–46, New York, NY, USA, 2010. ACM.

[SS02a] Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. pages 475–482, 2002.

[SS02b] Jordi Sabater and Carles Sierra. Social ReGreT, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44–56, 2002.

[SS04] Ahmad-Reza Sadeghi and Christian Stüble. Property-based attestation for computing platforms: caring about properties, not mechanisms. In *Proceedings of the NSPW '04*, pages 67–77. ACM, 2004.

[SSW08] Ahmad-Reza Sadeghi, Christian Stüble, and Marcel Winandy. Property-based tpm virtualization. In *Information Security*, volume 5222 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin / Heidelberg, 2008.

[STL+09] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and trust-based system for ad-hoc and sensor networks. In A. Boukerche, editor, *Algorithms and Protocols for Wireless Ad-Hoc and Sensor Networks*, pages 375–402. Wiley & Sons, 2008-2009.

[SVRH11] Guido Schryen, Melanie Volkamer, Sebastian Ries, and Sheikh Mahbub Habib. A formal approach towards measuring trust in distributed systems. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 1739–1745, New York, NY, USA, 2011. ACM.

[TCG10] TCG. Trusted computing group (tcg), 2010.

[TCG11] TCG. Tpm main specification level 2 version 1.2, revision 116. Trusted Computing Group, Mar 2011.

[TKH08] M. Tavakolifard, S.J. Knapskog, and P. Herrmann. Trust transferability among similar contexts. In *Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks*, pages 91–97. ACM, 2008.

[TPJL06] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *AAMAS*, 12(2):183–198, 2006.

[UKJS10] I. Uusitalo, K. Karppinen, A. Juhola, and R. Savola. Trust and cloud services - an interview study. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 712 –720, 30 2010-dec. 3 2010.

[Vv97] E.R. Verheul and H. van Tilborg. Constructions and properties of k out of n visual secret sharing schemes. *Designs, Codes and Cryptography*, 11(2):179–196, 1997.

[WJI05] Andrew Whitby, Audun Jøsang, and Jadwiga Indulska. Filtering out unfair ratings in bayesian reputation systems. *The ICFAIN Journal of Management Research*, 4(2):48 – 64, 2005.

[Wu11] X Wu. Stable groupbased trust management scheme for mobile p2p networks. *International Journal of Digital Content Technology and its Applications*, 5:116–125, 2011.

[WV03] Yao Wang and Julita Vassileva. Bayesian network-based trust model. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, pages 372–, Washington, DC, USA, 2003. IEEE Computer Society.

[WV07a] Yao Wang and Julita Vassileva. A review on trust and reputation for web service selection. In *Proceedings of the 27th ICDCSW*, page 25, Washington, DC, USA, 2007. IEEE Computer Society.

[WV07b] Yao Wang and Julita Vassileva. Toward trust and reputation based web service selection: A survey. *International Transactions on Systems Science and Applications (ITSSA) Journal, special Issue on New tendencies on Web Services and Multi-agent Systems (WS-MAS)*, Vol 3(No. 2):118–132, 2007.

[WZWQ10] Shou-Xin Wang, Li Zhang, Shuai Wang, and Xiang Qiu. A cloud-based trust model for evaluating quality of web services. *Journal of Computer Science and Technology*, 25:1130–1142, 2010.

[Zad75] L. A. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30:407–428, 1975.

[ZSLL11] Hongwei Zhou, Wenchang Shi, Zhaohui Liang, and Bin Liang. Using new fusion operations to improve trust expressiveness of subjective logic. *Wuhan University Journal of Natural Sciences*, 16:376–382, 2011.

# A

# Proofs (Formal Framework)

## A.1 Proofs for Theorem 4.3.1

PROOF We prove the theorem along the inductive definition of trustworthiness terms, and we provide for each definition of trustworthiness terms the corresponding propositional logic formula. The principal idea of the proof is that we reformulate the expression "$k$ 'out-of' a set $L$" by explicitly considering all combinations of elements of $L$, where $L$ can be either a set of basic components or of trustworthiness terms of subsystems. The provision of such a mapping $f$ (of trustworthiness terms on propositional logic terms) proves the theorem.

- If $l = (k \text{ 'out-of' } N)$, $k \in \{1, \ldots, |N|\}$, $N \subseteq A$ (def. 1), then

$$f(l) := \bigvee_{\substack{\{A_{i_1}, \ldots, A_{i_k}\} \subseteq A \\ |\{A_{i_1}, \ldots, A_{i_k}\}| = k}} \left( \bigwedge_{j=i_1}^{i_k} A_j \right) \tag{A.1}$$

- If $l = ((k_1 \oslash \ldots \oslash k_m) \text{ 'out-of' } (N_1, \ldots, N_m))$, $N_i \subseteq A \ \forall i$ (def. 2a), then

$$f(l) := \bigwedge_{i=1}^{m} (f((k_i \text{ 'out-of' } N_i))) \tag{A.2}$$

- If $l = ((k_1 \otimes \ldots \otimes k_m) \text{ 'out-of' } (N_1, \ldots, N_m))$, $N_i \subseteq A \ \forall i$ (def. 2b), then

$$f(l) := \bigvee_{i=1}^{m} (f((k_i \text{ 'out-of' } N_i))) \tag{A.3}$$

- If $l = (k \text{ 'out-of' } \{l_{i_1}, \ldots, l_{i_m}\})$, $l_{i_j}$ trustworthiness terms, $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ (def. 3), then

$$f(l) := \bigvee_{\substack{\{j_1,\ldots,j_k\} \subseteq \{i_1,\ldots,i_m\} \\ |\{j_1,\ldots,j_k\}| = k}} \left( \bigwedge_{j=j_1}^{j_k} (f(l_j)) \right) \tag{A.4}$$

- If $l = ((k_1 \oslash \ldots \oslash k_m) \text{ 'outof' } (Q_1, \ldots, Q_m))$, $Q_i$ set of trustworthiness terms (def. 4a), then

$$f(l) := \bigwedge_{i=1}^{m} (f((k_i \text{ 'out-of' } Q_i))) \tag{A.5}$$

- If $l = ((k_1 \otimes \ldots \otimes k_m) \text{ 'out-of' } (Q_1, \ldots, Q_m))$, $Q_i$ set of trustworthiness terms (def. 4b), then

$$f(l) := \bigvee_{i=1}^{m} (f((k_i \text{ 'out-of' } Q_i))) \tag{A.6}$$

■

## A.2 Proofs for Theorem 4.3.2

Proof We prove the theorem along the inductive definition of trustworthiness terms, and we provide for each definition of trustworthiness terms the corresponding propositional logic formula. The principal idea of the proof is that we reformulate the expression "$k$ out of a set $L$" by explicitly considering all combinations of elements of $L$, where $L$ can be either a set of properties or of trustworthiness terms of sub-properties. The provision of such a mapping $f$ (of trustworthiness terms on propositional logic terms) proves the theorem.

- If $l = (k \text{ 'out-of' } N)$, $k \in \{1, \ldots, |N|\}$, $N \subseteq A$ (def. 5), then

$$f(l) := \bigvee_{\substack{\{A_{i_1},\ldots,A_{i_k}\} \subseteq A \\ |\{A_{i_1},\ldots,A_{i_k}\}| = k}} \left( \bigwedge_{j=i_1}^{i_k} A_j \right) \tag{A.7}$$

- If $l = ((k_1 \oslash \ldots \oslash k_m) \text{ 'out-of' } (N_1, \ldots, N_m))$, $N_i \subseteq A \; \forall i$ (def. 6), then

$$f(l) := \bigwedge_{i=1}^{m} (f((k_i \text{ 'out-of' } N_i))) \tag{A.8}$$

- If $l = (k \text{ 'out-of' } \{l_{i_1}, \ldots, l_{i_m}\})$, $l_{i_j}$ trustworthiness terms, $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ (def. 7), then

$$f(l) := \bigvee_{\substack{\{j_1, \ldots, j_k\} \subseteq \{i_1, \ldots, i_m\} \\ |\{j_1, \ldots, j_k\}| = k}} \left( \bigwedge_{j=j_1}^{j_k} (f(l_j)) \right) \tag{A.9}$$

- If $l = ((k_1 \oslash \ldots \oslash k_m) \text{ 'out-of' } (Q_1, \ldots, Q_m))$, $Q_i$ set of trustworthiness terms (def. 8), then

$$f(l) := \bigwedge_{i=1}^{m} (f((k_i \text{ 'out-of' } Q_i))) \tag{A.10}$$

■

# B

# Standard Logic Equivalent Truth Table using CertainLogic

## B.1 Truth Table using CertainLogic $AND$

The following truth table is generated using the definitions in Table 5.1 of CertainLogic $AND$ operator. The CertainLogic truth table for $AND$ operator is equivalent to the corresponding truth table in standard logic when input values are 1 and 0.

Table B.1: CertainLogic Truth Table using $AND$ Operator

| Average Rating | | | Certainty | | | Initial Expectation | | |
|---|---|---|---|---|---|---|---|---|
| $t_A$ | $t_B$ | $t_{A \wedge B}$ | $c_A$ | $c_B$ | $c_{A \wedge B}$ | $f_A$ | $f_B$ | $f_{A \wedge B}$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

## B.2 Truth Table using CertainLogic $OR$

The following truth table is generated using the definitions in Table 5.1 of CertainLogic $OR$ operator. The CertainLogic truth table for $OR$ operator is equivalent to the corresponding truth table in standard logic when input values are 1 and 0.

Table B.2: CertainLogic Truth Table using $OR$ Operator

| Average Rating | | | Certainty | | | Initial Expectation | | |
|---|---|---|---|---|---|---|---|---|
| $t_A$ | $t_B$ | $t_{A \vee B}$ | $c_A$ | $c_B$ | $c_{A \vee B}$ | $f_A$ | $f_B$ | $f_{A \vee B}$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

# C

# Proofs (CertainLogic)

According to Table 5.1, it holds $0 \leq t_A, t_B, c_A, c_B \leq 1$; $f_A = f_B \neq 1$ for $AND$ operator and $f_A = f_B \neq 0$ for $OR$ operator.

## C.1 Proof: Theorem 5.2.1 ($OR$;$AND$)

The proof for $o_{A \vee B} = o_{B \vee A}$ can be carried out component-wise by verifying $t_{A \vee B} = t_{B \vee A}$, $c_{A \vee B} = c_{B \vee A}$ , and $f_{A \vee B} = f_{B \vee A}$. Using Table 5.1(OR), we see that altering the positions for the opinion parameters do not affect the outcome of the result.

The proof for $o_{A \wedge B} = o_{B \wedge A}$ can be carried out analogously using Table 5.1(AND).

## C.2 Proof: Theorem 5.2.2 ($OR$)

Before proving the theorem, we introduce four Lemmas that we need for the proof.

**Lemma C.2.1** *As a Lemma we prove:* $c_{A \vee B} > 0$ *if* $c_A = 0$ *and* $c_B \neq 0$.

PROOF

$$0 < c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$$

Using $c_A = 0$ it holds:

$$0 < c_B - \frac{c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$$

$$0 < c_B f_A + c_B f_B - c_B f_A f_B - c_B f_A(1-t_B) \tag{C.1}$$

which is true as it holds $c_B f_A \geq c_B f_A(1-t_B)$ and $c_B f_A > c_B f_A f_B$. ∎

**Lemma C.2.2** *As a Lemma we proof:* $c_{A \vee B} > 0$ *if* $c_A \neq 0$ *and* $c_B = 0$.

PROOF

$$0 < c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$$

Using $c_B = 0$ it holds:

$$0 < c_A - \frac{c_A f_B(1-t_A)}{f_A + f_B - f_A f_B}$$

$$0 < c_A f_A + c_A f_B - c_A f_A f_B - c_A f_B(1-t_A) \tag{C.2}$$

which is true as it holds $c_A f_B \geq c_A f_B(1-t_A)$ and $c_A f_A > c_A f_A f_B$. ∎

**Lemma C.2.3** *As a Lemma we prove:* $c_{A \vee B} > 0$ *if* $c_A \neq 0$ *and* $c_B \neq 0$.

PROOF

$$0 < c_A + c_B - c_A c_B - \frac{c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)}{f_A + f_B - f_A f_B}$$

Expand and reorganize, using $f_A + f_B + f_A f_B > 0$ it holds:

$$c_A f_A + c_B f_A + c_A f_B + c_B f_B + c_A c_B f_A f_B > c_A f_A f_B + c_B f_A f_B + c_A c_B f_A + \\ + c_A c_B f_B + c_A(1-c_B)f_B(1-t_A) + (1-c_A)c_B f_A(1-t_B)$$

Simplify:

$$c_A f_A + c_B f_A + c_A f_B + c_B f_B + c_A c_B f_A f_B > c_A f_A f_B + c_B f_A f_B + \\ + c_A f_B(c_B + (1-c_B)(1-t_A)) + c_B f_A(c_A + (1-c_A)(1-t_B))$$

Using $c_A f_A > c_A f_A f_B$ and $c_B f_B > c_B f_A f_B$ it holds:

$$c_B f_A + c_A f_B + c_A c_B f_A f_B > c_A f_B(c_B + (1-c_B)(1-t_A)) + \\ c_B f_A(c_A + (1-c_A)(1-t_B))$$

Using $c_A f_B(c_B + (1-c_B)(1-t_A)) \leq c_A f_B(c_B + (1-c_B))$
and $c_B f_A(c_A + (1-c_A)(1-t_B)) \leq c_B f_A(c_A + (1-c_A))$ it holds:

$$c_A f_B + c_B f_A + c_A c_B f_A f_B > c_A f_B + c_B f_A$$

Simplify:

$$c_A c_B f_A f_B > 0 \tag{C.3}$$

which is true as it holds $f_A, f_B, c_A, c_B \neq 0$. ∎

**Lemma C.2.4** *As a Lemma we prove also:* $c_{A\vee B} = 0$ *if* $c_A = 0$ *and* $c_B = 0$.

PROOF Replacing $c_A = 0$ and $c_B = 0$ in $c_{A\vee B}$, we get $c_{A\vee B} = 0$, which is exactly what we want to prove. ∎

PROOF **Theorem 5.2.2 (*OR*)**:

The proof will be carried out componentwise by verifying $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$, $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$, and $f_{(A\vee B)\vee C} = f_{A\vee(B\vee C)}$.

**Proof for** $f_{(A\vee B)\vee C} = f_{A\vee(B\vee C)}$**:**

$$\begin{aligned} f_{(A\vee B)\vee C} &= f_{A\vee B} + f_C - f_{A\vee B} f_C \\ &= (f_A + f_B - f_A f_B) + f_C - (f_A + f_B - f_A f_B) f_C \\ &= f_A + f_B - f_A f_B + f_C - f_A f_C - f_B f_C + f_A f_B f_C \\ &= f_A + (f_B + f_C - f_B f_C) - f_A(f_B + f_C - f_B f_C) \\ &= f_A + f_{B\vee C} - f_A f_{B\vee C} = f_{A\vee(B\vee C)} \end{aligned} \tag{C.4}$$

**Proof for $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$:**

$$\begin{aligned} c_{(A\vee B)\vee C} &= c_{A\vee B} + c_C - c_{A\vee B}c_C - \\ &- \frac{c_{A\vee B}(1-c_B)f_B(1-t_{A\vee B}) + (1-c_{A\vee B})c_B f_{A\vee B}(1-t_B)}{f_{A\vee B} + f_B - f_{A\vee B}f_B} \\ &= \ldots \text{ [Expand } t_{A\vee B}, c_{A\vee B}, \text{ and } f_{A\vee B}\text{]...} \end{aligned} \tag{C.5}$$

$= (c_C((-1+f_A)(-1+f_B)f_C+(-f_A(-1+f_B)+f_B)t_C)+c_B(-(-1+f_A)f_B(1+(-1+c_C)f_C - c_C t_C) + t_B(-(-1+c_C)f_C + f_A(1+(-1+c_C)f_C - c_C t_C))) + c_A(f_A(1+(-1+c_B)f_B - c_B t_B)(1+(-1+c_C)f_C - c_C t_C) + t_A((-1+c_C)f_C(-1+c_B t_B) - (-1+c_B)f_B(1+(-1+c_C)f_C - c_C t_C))))/(f_A(-1+f_B)(-1+f_C) - f_B(-1+f_C) + f_C)$

$$\begin{aligned} &= \ldots \text{ [Concentrate } t_{B\vee C}, c_{B\vee C}, \text{ and } f_{B\vee C}\text{]...} \\ &= c_A + c_{B\vee C} - c_A c_{B\vee C} - \\ &- \frac{c_A(1-c_{B\vee C})f_B(1-t_A) + (1-c_A)c_{B\vee C}f_A(1-t_{B\vee C})}{f_A + f_{B\vee C} - f_A f_{B\vee C}} \\ &= c_{A\vee(B\vee C)} \end{aligned} \tag{C.6}$$

**Proof for $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$:**

For proving $t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$, we have to consider that there are two cases for calculating $t_{X\vee Y}$:

1. $c_{X\vee Y} \neq 0$
2. $c_{X\vee Y} = 0$.

For the proof, we use the observation that it holds $c_{X\vee Y} = 0$ if and only if $c_X = c_Y = 0$ (see Lemmas C.2.3 and C.2.4), which leads to 5 cases. The proofs are given for each of these cases seperately.

**Case 1.** $c_A \neq 0$, $c_B \neq 0$, and $c_C \neq 0$ or exactly one term out of $c_A$, $c_B$, and $c_C$ is equivalent to 0: In this case, it holds $c_{(A\vee B)\vee C} \neq 0$ and $c_{A\vee(B\vee C)} \neq 0$ (using Lemmas C.2.1 and C.2.2):

$$\begin{aligned} t_{(A\vee B)\vee C} &= \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C) \\ &= \ldots\text{[Expand } t_{A\vee B}, c_{A\vee B}, \text{ and } f_{A\vee B}\text{]...} \end{aligned} \tag{C.7}$$

$= ((f_A(-1+f_B)(-1+f_C) - f_B(-1+f_C) + f_C)(c_C t_C + c_B t_B(1-c_C t_C) + c_A t_A(-1+c_B t_B)(-1+c_C t_C)))/(c_C((-1+f_A)(-1+f_B)f_C + (-f_A(-1+f_B)+$

$f_B)t_C)+c_B(-(-1+f_A)f_B(1+(-1+c_C)f_C-c_Ct_C)+t_B(-(-1+c_C)f_C+f_A(1+(-1+c_C)f_C-c_Ct_C)))+c_A(f_A(1+(-1+c_B)f_B-c_Bt_B)(1+(-1+c_C)f_C-c_Ct_C)+t_A((-1+c_C)f_C(-1+c_Bt_B)-(-1+c_B)f_B(1+(-1+c_C)f_C-c_Ct_C))))$

$$\begin{aligned} &= \text{... [Concentrate } t_{B\vee C}\text{, } c_{B\vee C}\text{, and } f_{B\vee C}\text{]...} \\ &= \frac{1}{c_{A\vee(B\vee C)}}(c_At_A + c_{B\vee C}t_{B\vee C} - c_Ac_{B\vee C}t_At_{B\vee C}) \\ &= t_{A\vee(B\vee C)} \end{aligned} \tag{C.8}$$

Furthermore, there are four cases to consider:

1. $c_A = 0$, $c_B = 0$, and $c_C \neq 0$
2. $c_A = 0$, $c_B \neq 0$, and $c_C = 0$
3. $c_A \neq 0$, $c_B = 0$, and $c_C = 0$
4. $c_A = c_B = c_C = 0$

**Case 2.)** $c_A = 0$, $c_B = 0$, and $c_C \neq 0$

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_Ct_C - c_{A\vee B}c_Ct_{A\vee B}t_C)$$

Using $c_A = c_B = 0 \Rightarrow\ c_{A\vee B} = 0$ (Lemma C.2.4) it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_Ct_C$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_Ct_C$$

Using $c_A = c_B = 0$ and $c_{B\vee C}t_{B\vee C} = c_Bt_B + c_Ct_C - c_Bt_Bc_Ct_C$ it holds:

$$\begin{aligned} t_{(A\vee B)\vee C} &= \frac{1}{c_{A\vee(B\vee C)}}(c_At_A + c_{B\vee C}t_{B\vee C} - c_Ac_{B\vee C}t_At_{B\vee C}) \\ t_{(A\vee B)\vee C} &= t_{A\vee(B\vee C)} \end{aligned} \tag{C.9}$$

**Case 3.)** $c_A = 0$, $c_B \neq 0$, and $c_C = 0$

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C)$$

Using $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_{A\vee B}t_{A\vee B}$$

Using $c_{A\vee B}t_{A\vee B} = c_A t_A + c_B t_B + c_A t_A c_B t_B$ and $c_A = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_B t_B$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds: (C.10)

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_B t_B$$

Using $c_{B\vee C}t_{B\vee C} = c_B t_B + c_C t_C + c_B t_B c_C t_C$ and $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_{B\vee C}t_{B\vee C}$$

Using $c_A = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}(c_A t_A + c_{B\vee C}t_{B\vee C} - c_A t_A c_{B\vee C}t_{B\vee C})$$

$$t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$$

**Case 4.)** $c_A \neq 0$, $c_B = 0$, and $c_C = 0$:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}(c_{A\vee B}t_{A\vee B} + c_C t_C - c_{A\vee B}c_C t_{A\vee B}t_C)$$

Using $c_C = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{(A\vee B)\vee C}}c_{A\vee B}t_{A\vee B}$$

Using $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)}$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_{A\vee B}t_{A\vee B} \tag{C.11}$$

■

Using $c_{A\vee B}t_{A\vee B} = c_A t_A + c_B t_B - c_A t_A c_B t_B$ and $c_B = 0$ it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}c_A t_A$$

Using $c_{B\vee C} = 0$ (Lemma C.2.4) it holds:

$$t_{(A\vee B)\vee C} = \frac{1}{c_{A\vee(B\vee C)}}(c_A t_A + c_{B\vee C}t_{B\vee C} - c_A c_{B\vee C}t_A t_{B\vee C})$$

$$t_{(A\vee B)\vee C} = t_{A\vee(B\vee C)}$$

**Case 5.)** $c_A = c_B = c_C = 0$

In this case it holds $c_{(A\vee B)\vee C} = c_{A\vee(B\vee C)} = 0$ (applying the Lemma C.2.4 two times when calculating $c_{(A\vee B)\vee C}$ or $c_{A\vee(B\vee C)}$, respectively, and thus $c_{(A\vee B)\vee C} = t_{A\vee(B\vee C)} = 0.5$

## C.3 Proof: Theorem 5.2.2 ($AND$)

For proving $t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)}$, we have to consider that there are two cases for calculating $t_{X\wedge Y}$:

- (1) the case $c_{X\wedge Y} \neq 0$
- and (2) the case $c_{X\wedge Y} = 0$.

For the proof, we use the observation that it holds $c_{X\wedge Y} = 0$ if and only if $c_X = c_Y = 0$ (see Lemma C.3.1, C.3.2, C.3.3 and C.3.4), which leads to 5 cases. The proofs are given for each of these cases separately.

Before proving the theorem, we prove the following four Lemmas necessary for the proof.

**Lemma C.3.1** *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A = 0$ *and* $c_B \neq 0$.

Proof

$$\begin{aligned} 0 &< c_B - \frac{c_B(1-f_A)t_B}{1-f_Af_B} \\ 0 &< c_B(1-f_Af_B) - c_B(1-f_A)t_B \\ c_B(1-f_A)t_B &< c_B(1-f_Af_B) \end{aligned} \tag{C.12}$$

which is true as it holds $1 - f_A < 1 - f_Af_B$ and $c_Bt_B \leq c_B$. ∎

**Lemma C.3.2** *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A \neq 0$ *and* $c_B = 0$.

Proof

$$\begin{aligned} 0 &< c_A - \frac{c_A(1-f_B)t_A}{1-f_Af_B} \\ 0 &< c_A(1-f_Af_B) - c_A(1-f_B)t_A \\ c_A(1-f_B)t_A &< c_A(1-f_Af_B) \end{aligned} \tag{C.13}$$

which is true as it holds $1 - f_B < 1 - f_Af_B$ and $c_At_A \leq c_A$. ∎

**Lemma C.3.3** *As a Lemma we proof:* $c_{A\wedge B} > 0$ *if* $c_A \neq 0$ *and* $c_B \neq 0$.

Proof

$$\begin{aligned} 0 &< c_A + c_B - c_Ac_B - \frac{(1-c_A)c_B(1-f_A)t_B + c_A(1-c_B)(1-f_B)t_A}{1-f_Af_B} \\ 0 &< c_A(1-f_Af_B) + c_B(1-f_Af_B) - c_Ac_B(1-f_Af_B) - (1-c_A)c_B(1-f_A)t_B - \\ &\quad - c_A(1-c_B)(1-f_B)t_A \end{aligned} \tag{C.14}$$

To proof this we show that it holds A) and B):

A) $c_A(1-f_Af_B) > c_A(1-c_B)(1-f_B)t_A$

which is true as it holds $1 - f_A f_B > 1 - f_B$ and $c_A \geq c_A(1 - c_B)t_A$.

B) $c_B(1 - f_A f_B) > c_A c_B(1 - f_A f_B) + (1 - c_A)c_B(1 - f_A)t_B$ which is true as it holds (using $1 - f_A f_B > 1 - f_A$):

$c_A c_B(1 - f_A f_B) + (1 - c_A)c_B(1 - f_A)t_B$
$< c_A c_B(1 - f_A f_B) + (1 - c_A)c_B(1 - f_A f_B)t_B$
$< c_B(1 - f_A f_B)(c_A + (1 - c_A)t_B)$
$< c_B(1 - f_A f_B)(c_A + (1 - c_A))$
$< c_B(1 - f_A f_B)$ ∎

**Lemma C.3.4** *As a Lemma we proof:* $c_{A \wedge B} = 0$ *if* $c_A = 0$ *and* $c_B = 0$.

PROOF Replacing $c_A = 0$ and $c_B = 0$ in $c_{A \wedge B}$, we get $c_{A \wedge B} = 0$, which is exactly what we want to prove. ∎

PROOF **Theorem 5.2.2** ($AND$)

**Proof for** $f_{(A \wedge B) \wedge C} = f_{A \wedge (B \wedge C)}$**:**

$$f_{(A \wedge B) \wedge C} = f_A f_B f_C = f_{A \wedge (B \wedge C)} \tag{C.15}$$

**Proof for** $c_{(A \wedge B) \wedge C} = c_{A \wedge (B \wedge C)}$**:**

$$\begin{aligned}
&c_{(A \wedge B) \wedge C} = c_{A \wedge B} + c_C - c_{A \wedge B} c_C - \\
&- \frac{(1 - c_{A \wedge B})c_C(1 - f_{A \wedge B})t_C + c_{A \wedge B}(1 - c_C)(1 - f_C)t_{A \wedge B}}{1 - f_{A \wedge B} f_C} \\
&= \ldots \text{ Expand } t_{A \wedge B}, c_{A \wedge B}, \text{ and } f_{A \wedge B} \ldots \\
&= \ldots \text{ Concentrate } t_{B \wedge C}, c_{B \wedge C}, \text{ and } f_{B \wedge C} \ldots \\
&= c_A + c_{B \wedge C} - c_A c_{B \wedge C} - \\
&- \frac{(1 - c_A)c_{B \wedge C}(1 - f_A)t_{B \wedge C} + c_A(1 - c_{B \wedge C})(1 - f_{B \wedge C})t_A}{1 - f_A f_{B \wedge C}} \\
&= c_{A \wedge (B \wedge C)}
\end{aligned} \tag{C.16}$$

**Proof for** $t_{(A \wedge B) \wedge C} = t_{A \wedge (B \wedge C)}$**:**

**Case 1.)** $c_A \neq 0$, $c_B \neq 0$, and $c_C \neq 0$.

In this case it holds:

$$
\begin{aligned}
t_{(A\wedge B)\wedge C} &= \frac{1}{c_{(A\wedge B)\wedge C}}(c_{A\wedge B}c_C t_{A\wedge B}t_C + \\
&\frac{c_{A\wedge B}(1-c_C)(1-f_{A\wedge B})f_C t_{A\wedge B}}{1-f_{A\wedge B}f_C} + \\
&+ \frac{(1-c_{A\wedge B})c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C}) \\
&= \ldots \text{ Expand } t_{A\wedge B},\ c_{A\wedge B}, \text{ and } f_{A\wedge B} \ldots \\
&= \ldots \text{ Concentrate } t_{B\wedge C},\ c_{B\wedge C}, \text{ and } f_{B\wedge C} \ldots \\
&= \frac{1}{c_{A\wedge(B\wedge C)}}(c_A c_{B\wedge C} t_A t_{B\wedge C} + \\
&+ \frac{c_A(1-c_{B\wedge C})(1-f_A)f_{B\wedge C}t_A}{1-f_A f_{B\wedge C}} + \\
&+ \frac{(1-c_A)c_{B\wedge C}f_A(1-f_{B\wedge C})t_{B\wedge C}}{1-f_A f_{B\wedge C}})
\end{aligned}
\tag{C.17}
$$

Furthermore, there are four other cases to consider:

1. $c_A = 0$, $c_B = 0$, and $c_C \neq 0$

2. $c_A = 0$, $c_B \neq 0$, and $c_C = 0$

3. $c_A \neq 0$, $c_B = 0$, and $c_C = 0$

4. $c_A = c_B = c_C = 0$

**Case 2.)** $c_A = c_B = 0$

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}(c_{A\wedge B}c_C t_{A\wedge B}t_C + \frac{c_{A\wedge B}(1-c_C)(1-f_{A\wedge B})f_C t_{A\wedge B} + (1-c_{A\wedge B})c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C})$$

Using $c_A = c_B = 0$ it holds $c_{A\wedge B} = 0$ (Lemma C.3.4), and thus:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}\frac{c_C f_{A\wedge B}(1-f_C)t_C}{1-f_{A\wedge B}f_C}$$

Using $f_{A\wedge B} = f_A f_B$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{(A\wedge B)\wedge C}}\frac{c_C f_A f_B(1-f_C)t_C}{1-f_A f_B f_C}$$

Using $\frac{1}{c_{(A\wedge B)\wedge C}} = \frac{1}{c_{A\wedge(B\wedge C)}}$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_C f_A f_B(1-f_C)t_C}{1-f_A f_B f_C} \tag{C.18}$$

Expanding with $(1-f_B f_C)$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_C f_A f_B(1-f_C)t_C(1-f_B f_C)}{(1-f_A f_B f_C)(1-f_B f_C)}$$

Using $c_B = 0$ and $c_{B\wedge C}t_{B\wedge C} = \frac{c_C f_B(1-f_C)t_C}{1-f_B f_C}$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}\frac{c_{B\wedge C}f_A(1-f_{B\wedge C})t_{B\wedge C}}{1-f_A f_{B\wedge C}}$$

Using $c_A = 0$ it holds:

$$t_{(A\wedge B)\wedge C} = \frac{1}{c_{A\wedge(B\wedge C)}}(c_A c_C t_A t_C + \frac{c_A(1-c_{B\wedge C})(1-f_A)f_{B\wedge C}t_A + (1-c_A)c_{B\wedge C}f_A(1-f_{B\wedge C})t_C}{1-f_A f_{B\wedge C}})$$

$$t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)}$$

**Case 3.)** $c_A = 0$, $c_B \neq 0$, and $c_C = 0$
The algebriac proof is analogous to Case 2.
**Case 4.)** $c_A \neq 0$, $c_B = 0$, and $c_C = 0$
The algebriac proof is analogous to Case 2.
**Case 5.)** $c_A = c_B = c_C = 0$
In this case, it holds $c_{(A\wedge B)\wedge C} = c_{A\wedge(B\wedge C)} = 0$ (applying Lemma C.3.4 two times, when calculating $c_{(A\wedge B)\wedge C}$ and $c_{A\wedge(B\wedge C)}$, respectively, and thus $t_{(A\wedge B)\wedge C} = t_{A\wedge(B\wedge C)} = 0.5$. ∎

## C.4 Proof: Theorem 5.2.3

Proof We can prove each of the equations in the theorem separately. The detail algebraic simplifications are omitted for first two proofs.

1. $E(o_{A\wedge B}) = E(t_{A\wedge B}, c_{A\wedge B}, f_{A\wedge B})$[Using Definition 5.2.2]
   $= t_{A\wedge B} * c_{A\wedge B} + (1 - c_{A\wedge B}) * f_{A\wedge B}$[Using Definition 2.2.3]
   $= \ldots$ [Substitution of $t_{A\wedge B}$, $c_{A\wedge B}$, and $f_{A\wedge B}$ using Table 5.1 ($AND$) and algebraic simplifications]
   $= (t_A * c_A + (1 - c_A) * f_A)(t_B * c_B + (1 - c_B) * f_B)$
   $= E(o_A)E(o_B)$[Using Definition 2.2.2 and Definition 2.2.3]

2. $E(o_{A\vee B}) = E(t_{A\vee B}, c_{A\vee B}, f_{A\vee B})$[Using Definition 5.2.1]
   $= t_{A\vee B} * c_{A\vee B} + (1 - c_{A\vee B}) * f_{A\vee B}$[Using Definition 2.2.3]
   $= \ldots$ [Substitution of $t_{A\vee B}$, $c_{A\vee B}$, and $f_{A\vee B}$ using Table 5.1 ($OR$) and algebraic simplifications]
   $= (t_A * c_A + (1 - c_A) * f_A) + (t_B * c_B + (1 - c_B) * f_B) - (t_A * c_A + (1 - c_A) * f_A)(t_B * c_B + (1 - c_B) * f_B)$
   $= E(o_A) + E(o_B) - E(o_A)E(o_B)$
   [Using Definition 2.2.2 and Definition 2.2.3]

3. $E(o_{\neg A}) = E(t_{\neg A}, c_{\neg A}, f_{\neg A})$[Using Definition 5.2.3]
   $= (t_{\neg A} * c_{\neg A}) + (1 - c_{\neg A}) * f_{\neg A}$[Using Definition 2.2.3]
   $= (1-t_A)*c_A+(1-c_A)*(1-f_A)$[Substitution using Table 5.1 ($NOT$)]
   $= c_A - t_A * c_A + 1 - c_A - f_A + f_A * c_A$
   $= 1 - (t_A * c_A + (1 - c_A) * f_A)$
   $= 1 - E(o_A)$[Using Definition 2.2.2 and Definition 2.2.3] ∎

## C.5 Sketch of Proof: Theorem 5.2.4

PROOF In the following we provide the for the $NOT$ operator and show sketches for $AND$ and $OR$.

1. $\omega_{\neg A} = (b_{\neg A}, d_{\neg A}, u_{\neg A}, a_{\neg A})$
   $= (d_A, b_A, u_A, 1 - a_A)$[Using Theorem 6 in [Jøs01]]
   $= ((1 - t_A)c_A, t_A c_A, 1 - c_A, 1 - f_A)$[Using $o_A = m_{CT}^{SL}(\omega_A)$]
   $= m_{SL}^{CT}(1 - t_A, c_A, 1 - f_A)$[Using Definition 5.2.5]
   $= m_{SL}^{CT}(t_{\neg A}, c_{\neg A}, f_{\neg A})$[Using Table 5.1($NOT$)]
   $= m_{SL}^{CT}(o_{\neg A})$

2. $o_{\neg A} = (t_{\neg A}, c_{\neg A}, f_{\neg A})$
   $= (1 - t_A, c_A, 1 - f_A)$[Using Table 5.1 ($NOT$)]
   $= m_{CT}^{SL}((1 - t_A)c_A, t_A c_A, 1 - c_A, 1 - f_A)$[Using Definition 5.2.6]
   $= m_{CT}^{SL}(d_A, b_A, u_A, 1 - a_A)$[Using $\omega_A = m_{SL}^{CT} o_A$]
   $= m_{CT}^{SL}(\omega_{\neg A})$[Using Theorem 6 in [Jøs01]]

The proof for the operators $AND$ and $OR$ can be carried out analogous to the proof for $NOT$. However, here we just provide a sketch of the proofs.

1. $\omega_{A \wedge B} = (b_{A \wedge B}, d_{A \wedge B}, u_{A \wedge B}, a_{A \wedge B})$[Using 5.2.4]
   $= \ldots$ [Substitution of $b_{A \wedge B}$, $d_{A \wedge B}$, $u_{A \wedge B}$,$a_{A \wedge B}$ with
   $b_A$, $d_A$, $u_A$, $a_A$, $b_B$, $d_B$, $u_B$, and $a_B$ ...
   ... as defined by the normal multiplication in [JM05] ...
   ... applying $o_A = m_{CT}^{SL}(\omega_A)$ and $o_B = m_{CT}^{SL}(\omega_B)$ ...
   ... algebraic simplifications and applying Definition 5.2.5]
   $= m_{SL}^{CT}(t_{A \wedge B}, c_{A \wedge B}, f_{A \wedge B})$
   $= m_{SL}^{CT}(o_{A \wedge B})$

2. $o_{A \wedge B} = (t_{A \wedge B}, c_{A \wedge B}, f_{A \wedge B})$
   $= \ldots$ [Introduce $t_A$, $c_A$,... and replace them by $b_A$, $d_A$,...]
   $= m_{CT}^{SL}(b_{A \wedge B}, d_{A \wedge B}, u_{A \wedge B}, a_{A \wedge B})$
   $= m_{CT}^{SL}(\omega_{A \wedge B})$

3. $\omega_{A \vee B} = (b_{A \vee B}, d_{A \vee B}, u_{A \vee B}, a_{A \vee B})$[Using 5.2.4]
   $= \ldots$ [Introduce $b_A$, $d_A$,... and replace them by $t_A$, $c_A$,...]
   $= m_{SL}^{CT}(t_{A \vee B}, c_{A \vee B}, f_{A \vee B})$
   $= m_{SL}^{CT}(o_{A \vee B})$

4. $o_{A \vee B} = (t_{A \vee B}, c_{A \vee B}, f_{A \vee B})$[Using 2.2.2]
   $= \ldots$ [Introduce $t_A$, $c_A$,... and replace them by $b_A$, $d_A$,...]
   $= m_{CT}^{SL}(b_{A \vee B}, d_{A \vee B}, u_{A \vee B}, a_{A \vee B})$
   $= m_{CT}^{SL}(\omega_{A \vee B})$ ∎

# D

# Proofs (CertainLogic FUSION)

## D.1 Proof: Theorem 5.3.1

PROOF We prove the theorem component-wise for average fusion ($A.FUSION$) operator by verifying the following under different cases:

$$t_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = t_{A_1}; c_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = c_{A_1}; f_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = f_{A_1}. \tag{D.1}$$

At first, we prove $t_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = t_{A_1}$ under following three cases:

- Case 1: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1$.
- Case 2: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0$.
- Case 3: if $\{c_{A_i}, c_{A_j}\} \neq 1$.

**Proof for Case 1:**

$$\begin{aligned} t_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} &= \frac{t_{A_1} + t_{A_1} + \cdots + t_{A_1}}{n} \text{ [Using Table 5.4 and replace all } A_i\text{'s by } A_1] \\ &= \frac{n * t_{A_1}}{n} \\ &= t_{A_1} \end{aligned} \tag{D.2}$$

**Proof for Case 2:**

$$t_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = 0.5 \text{ [Using Table 5.4 and replace } A_i\text{'s by } A_1] = t_{A_1} \tag{D.3}$$

**Proof for Case 3:**

$$
\begin{aligned}
&t_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} \\
&= \frac{(c_{A_1}t_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))+\cdots+(c_{A_1}t_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))}{(c_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))+\cdots+(c_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))} \\
&\text{[Using Table 5.4 and replace } A_i\text{'s by } A_1\text{]} \\
&= \frac{n*(t_{A_1}c_{A_1}(1-c_{A_1})}{n*(c_{A_1}(1-c_{A_1}))} \\
&= t_{A_1}
\end{aligned}
\tag{D.4}
$$

Next, we prove $c_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = c_{A_1}$ under following two cases:

- Case 1: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1$.
- Case 2: if $\{c_{A_i}, c_{A_j}\} \neq 1$.

**Proof for Case 1:**

$$
\begin{aligned}
&c_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} \\
&= 1 \text{ [Using Table 5.4 and replace } A_i\text{'s by } A_1\text{]} \\
&= c_{A_1}
\end{aligned}
\tag{D.5}
$$

**Proof for Case 2:**

$$
\begin{aligned}
&c_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} \\
&= \frac{(c_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))+\cdots+(c_{A_1}(1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))}{((1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))+\cdots+((1-c_{A_1})(1-c_{A_1})\cdots(1-c_{A_1}))} \\
&\text{[Using Table 5.4 and replace } A_i\text{'s by } A_1\text{]} \\
&= \frac{n*(c_{A_1}(1-c_{A_1}))}{n*(1-c_{A_1})} \\
&= c_{A_1}
\end{aligned}
\tag{D.6}
$$

Finally, we prove $f_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = f_{A_1}$

$$
\begin{aligned}
&f_{\widehat{\oplus}(A_1,A_1,\cdots,A_1)} = \frac{f_{A_1}+f_{A_1}+\cdots+f_{A_1}}{n} \text{ [Using Table 5.4 and replace } A_i\text{'s by } A_1\text{]} \\
&= \frac{n*f_{A_1}}{n} = f_{A_1}
\end{aligned}
\tag{D.7}
$$

■

The proof for the *W.FUSION* and *C.FUSION* operators of Theorem 5.3.1 can be carried out analogously.

## D.2 Sketch of the Proof: Theorem 5.3.2

PROOF The proof for $\widehat{\oplus}(o_{A_1}, o_{A_2}) = \widehat{\oplus}(o_{A_2}, o_{A_1})$ can be carried out component-wise by verifying $\widehat{\oplus}(t_{A_1}, t_{A_2}) = \widehat{\oplus}(t_{A_2}, t_{A_1})$, $\widehat{\oplus}(c_{A_1}, c_{A_2}) = \widehat{\oplus}(c_{A_2}, c_{A_1})$ , and $\widehat{\oplus}(f_{A_1}, f_{A_2}) = \widehat{\oplus}(f_{A_2}, f_{A_1})$. Using Table 5.4, these can be verified. ∎

The proof for $\widehat{\oplus}_w(o_{A_1}, o_{A_2}) = \widehat{\oplus}_w(o_{A_2}, o_{A_1})$ and $\widehat{\oplus}_c(o_{A_1}, o_{A_2}) = \widehat{\oplus}(o_{A_2}, o_{A_1})$ can be carried out analogously, using Table 5.5 and Table 5.6 respectively.

## D.3 Sketch of the Proof: Theorem 5.3.3

Looking at the definitions of the fusion operators (see Table 5.4, 5.5, 5.6), one sees that the theorem holds due to the commutativity of the summation.

## D.4 Proof: Theorem 5.3.4

PROOF We prove the theorem component-wise by verifying the following under different cases:

$$\widehat{\oplus}_w(t_{A_1}, t_{A_2}, \cdots, t_{A_n}) = \widehat{\oplus}(\underbrace{t_{A_1}, \cdots, t_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{t_{A_2}, \cdots, t_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{t_{A_n}, \cdots, t_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}}),$$

$$\widehat{\oplus}_w(c_{A_1}, c_{A_2}, \cdots, c_{A_n}) = \widehat{\oplus}(\underbrace{c_{A_1}, \cdots, c_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{c_{A_2}, \cdots, c_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{c_{A_n}, \cdots, c_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}}),$$

$$\widehat{\oplus}_w(f_{A_1}, f_{A_2}, \cdots, f_{A_n}) = \widehat{\oplus}(\underbrace{f_{A_1}, \cdots, f_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{f_{A_2}, \cdots, f_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{f_{A_n}, \cdots, f_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}}).$$

We prove $\widehat{\oplus}_w(t_{A_1}, t_{A_2}, \cdots, t_{A_n}) =$
$\widehat{\oplus}(\underbrace{t_{A_1}, \cdots, t_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{t_{A_2}, \cdots, t_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{t_{A_n}, \cdots, t_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}})$ under following three cases:

- Case 1: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0$.
- Case 2: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1$.
- Case 3: if $\{c_{A_i}, c_{A_j}\} \neq 1$.

**Proof for Case 1:**

$$\begin{aligned} &\widehat{\oplus}_w(t_{A_1}, t_{A_2}, \cdots, t_{A_n}) \\ &= t_{\widehat{\oplus}_w A_1, A_2, \cdots, A_n} \\ &= 0.5 = t_{\widehat{\oplus} A_1, A_2, \cdots, A_n} \end{aligned} \tag{D.8}$$

**Proof for Case 2:**

$$
\begin{aligned}
&\widehat{\oplus}_w(t_{A_1}, t_{A_2}, \cdots, t_{A_n}) \\
&= t_{\widehat{\oplus}_w A_1, A_2, \cdots, A_n} \\
&= \frac{\sum_{i=1}^{n} w_i t_{A_i}}{\sum_{i=1}^{n} w_i} \text{ [Using Table 5.5]} \\
&= \frac{\sum_{i=1}^{n} \frac{a_i}{b_i} t_{A_i}}{\sum_{i=1}^{n} \frac{a_i}{b_i}} \\
&= \frac{\sum_{i=1}^{n} \frac{a_i \cdot \prod_{j\neq i}^{n} b_j}{\prod_{k=1}^{n} b_k} t_{A_i}}{\sum_{i=1}^{n} \frac{a_i \cdot \prod_{j\neq i}^{n} b_j}{\prod_{k=1}^{n} b_k}} \\
&= \frac{\sum_{i=1}^{n} t_{A_i} a_i \cdot \prod_{j\neq i}^{n} b_j}{\sum_{i=1}^{n} a_i \cdot \prod_{j\neq i}^{n} b_j} \\
&= \frac{\sum_{i=1}^{n} t_{A_i} m_i}{\sum_{i=1}^{n} m_i} \text{ [Replace } a_i \cdot \prod_{j\neq i}^{n} b_j \text{ with } m_i\text{;product of integers is an integer]} \\
&= \frac{\sum_{i=1}^{n} (\sum_{j=1}^{m_i} t_{A_i})}{\sum_{i=1}^{n} m_i} \\
&= \widehat{\oplus}( \underbrace{t_{A_1}, \cdots, t_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}} , \underbrace{t_{A_2}, \cdots, t_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}} , \cdots , \underbrace{t_{A_n}, \cdots, t_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}} ) \text{ [where } m_i = a_i \cdot \prod_{j\neq i}^{n} b_j]
\end{aligned}
\tag{D.9}
$$

**Proof for Case 3:**

$$\widehat{\oplus}_w(t_{A_1}, t_{A_2}, \cdots, t_{A_n})$$

$$= t_{\widehat{\oplus}_w A_1, A_2, \cdots, A_n}$$

$$= \frac{\sum_{i=1}^{n}(c_{A_i} t_{A_i} w_i \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}(c_{A_i} w_i \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))} \text{ [Using Table 5.5]}$$

$$= \frac{\sum_{i=1}^{n}\frac{a_i}{b_i}(c_{A_i} t_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}\frac{a_i}{b_i}(c_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))} = \frac{\sum_{i=1}^{n}\frac{a_i \cdot \prod_{j\neq i}^{n} b_j}{\prod_{k=1}^{n} b_k}(c_{A_i} t_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}\frac{a_i \cdot \prod_{j\neq i}^{n} b_j}{\prod_{k=1}^{n} b_k}(c_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}$$

$$= \frac{\sum_{i=1}^{n}(a_i \cdot \prod_{j\neq i}^{n} b_j \cdot c_{A_i} t_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}(a_i \cdot \prod_{j\neq i}^{n} b_j \cdot c_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}$$

$$= \frac{\sum_{i=1}^{n}(m_i \cdot c_{A_i} t_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}(m_i \cdot c_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}$$

[Replace $a_i \cdot \prod_{j\neq i}^{n} b_j$ with $m_i$;product of integers is an integer]

$$= \frac{\sum_{i=1}^{n}(\sum_{i=1}^{m_i} c_{A_i} t_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}{\sum_{i=1}^{n}(m_i \cdot c_{A_i} \prod_{i=1,\ j\neq i}^{n}(1 - c_{A_j}))}$$

$$= \frac{\sum_{i=1}^{n}(\sum_{j=1}^{m_i} t_{A_i})}{\sum_{i=1}^{n} m_i} \text{ [Expanding the equation and reducing the common terms]}$$

$$= \widehat{\oplus}(\underbrace{t_{A_1}, \cdots, t_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{t_{A_2}, \cdots, t_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{t_{A_n}, \cdots, t_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}})$$

[where $m_i = a_i \cdot \prod_{j\neq i}^{n} b_j$]

(D.10)

The proof for the following two equations can be carried out analogously using Table 5.4 and Table 5.5:

$$\widehat{\oplus}_w(c_{A_1}, c_{A_2}, \cdots, c_{A_n}) = \widehat{\oplus}(\underbrace{c_{A_1}, \cdots, c_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{c_{A_2}, \cdots, c_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{c_{A_n}, \cdots, c_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}}),$$

$$\widehat{\oplus}_w(f_{A_1}, f_{A_2}, \cdots, f_{A_n}) = \widehat{\oplus}(\underbrace{f_{A_1}, \cdots, f_{A_1}}_{a_1 \cdot \prod_{j\neq 1} b_j \textbf{ times}}, \underbrace{f_{A_2}, \cdots, f_{A_2}}_{a_2 \cdot \prod_{j\neq 2} b_j \textbf{ times}}, \cdots, \underbrace{f_{A_n}, \cdots, f_{A_n}}_{a_n \cdot \prod_{j\neq n} b_j \textbf{ times}})$$

Finally, the component-wise algebraic verifications implies the verification of the theorem itself. ∎

The proof of the same theorem for *C.FUSION* operator can be carried out analogously using Table 5.4 and Table 5.6

## D.5 Proof: Theorem 5.3.5

Proof We prove the theorem component-wise by verifying the following equations under different cases:
$\widehat{\oplus}_w((t_{A_1}, w_1 * k), (t_{A_2}, w_2 * k), \cdots, (t_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((t_{A_1}, w_1), (t_{A_2}, w_2), \cdots, (t_{A_n}, w_n)$
$\widehat{\oplus}_w((c_{A_1}, w_1 * k), (c_{A_2}, c_2 * k), \cdots, (c_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((c_{A_1}, w_1), (c_{A_2}, w_2), \cdots, (c_{A_n}, w_n)$
$\widehat{\oplus}_w((f_{A_1}, w_1 * k), (f_{A_2}, w_2 * k), \cdots, (f_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((f_{A_1}, w_1), (f_{A_2}, w_2), \cdots, (f_{A_n}, w_n)$

We prove $\widehat{\oplus}_w((t_{A_1}, w_1 * k), (t_{A_2}, w_2 * k), \cdots, (t_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((t_{A_1}, w_1), (t_{A_2}, w_2), \cdots, (t_{A_n}, w_n))$ under following three cases:

- Case 1: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 1$.
- Case 2: If $c_{A_1} = c_{A_2} = \cdots = c_{A_n} = 0$.
- Case 3: if $\{c_{A_i}, c_{A_j}\} \neq 1$.

**Proof for Case 1:**

$$
\begin{aligned}
&\widehat{\oplus}_w((t_{A_1}, w_1 * k), (t_{A_2}, w_2 * k), \cdots, (t_{A_n}, w_n * k) \\
&= \frac{\sum_{i=1}^{n} k * w_i t_{A_i}}{\sum_{i=1}^{n} w_i} \text{ [Using Table 5.5 and replace } w_i \text{ with scaling factor } k] \\
&= \frac{\sum_{i=1}^{n} k \cdot w_i t_{A_i}}{\sum_{i=1}^{n} k \cdot w_i} \text{ [Multiply with a scaling factor } k] \\
&= \frac{k \cdot w_1 t_{A_1} + k \cdot w_2 t_{A_2} + \cdots + k \cdot w_n t_{A_n}}{k \cdot w_1 + k \cdot w_2 + \cdots + k \cdot w_n} \\
&= \frac{\sum_{i=1}^{n} w_i t_{A_i}}{\sum_{i=1}^{n} w_i} \text{ [Reduce common constant } k] \\
&= \widehat{\oplus}_w((t_{A_1}, w_1), (t_{A_2}, w_2), \cdots, (t_{A_n}, w_n))
\end{aligned}
\tag{D.11}
$$

The proof for Case 2 and 3 can be carried out analogously using Table 5.5. Moreover, the proof for the following two equations can be carried out analogously using Table 5.5:
$\widehat{\oplus}_w((c_{A_1}, w_1 * k), (c_{A_2}, c_2 * k), \cdots, (c_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((c_{A_1}, w_1), (c_{A_2}, w_2), \cdots, (c_{A_n}, w_n)$
$\widehat{\oplus}_w((f_{A_1}, w_1 * k), (f_{A_2}, w_2 * k), \cdots, (f_{A_n}, w_n * k) =$
$\widehat{\oplus}_w((f_{A_1}, w_1), (f_{A_2}, w_2), \cdots, (f_{A_n}, w_n)$

Finally, the component-wise algebraic verifications implies the verification of the theorem itself. ∎

The proof of the same theorem for *C.FUSION* operator can be carried out analogously using Table 5.4 and Table 5.6.

## D.6 Proof: Theorem 5.3.6

Proof We prove the theorem and omit detail algebraic simplifications for brevity.

First, we prove

$$E(\widehat{\oplus}_w(o_{A_1}, w), (o_{A_2}, w), \cdots, (o_{A_n}, w)) = E(\widehat{\oplus}(o_{A_1}, o_{A_2}, ..., o_{A_n}))$$

$$
\begin{aligned}
&E(\widehat{\oplus}_w(o_{A_1},w),(o_{A_2},w),\cdots,(o_{A_n},w))\\
&= E(\widehat{\oplus}_w(t_{(A_1,w),\cdots,(A_n,w)}),\widehat{\oplus}_w(c_{(A_1,w),\cdots,(A_n,w)}),\widehat{\oplus}_w(f_{(A_1,w),\cdots,(A_n,w)}))\\
&= t_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))} * c_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))} + (1 - c_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))})*\\
&f_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))}\\
&\ldots [\text{Substitution of } t_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))}, c_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))}, f_{\widehat{\oplus}_w((A_1,w),\cdots,(A_n,w))}]\\
&[\text{using Table 5.5}]\\
&\ldots [\text{replace } w_i \text{ by } w \text{ where } i = 1\cdots n \text{ and algebraic simplifications}]\\
&= t_{\widehat{\oplus}(A_1,\cdots,A_n)} * c_{\widehat{\oplus}(A_1,\cdots,A_n)} + (1 - c_{\widehat{\oplus}(A_1,\cdots,A_n)}) * f_{\widehat{\oplus}(A_1,\cdots,A_n)}\\
&= E(\widehat{\oplus}(o_{A_1},o_{A_2},...,o_{A_n})) \text{ [Using the Equation 5.1]}
\end{aligned}
\tag{D.12}
$$

Next, we prove

$$E(\widehat{\oplus}(o_{A_1},o_{A_2},...,o_{A_n})) = \frac{\sum_{i=1}^{n} E(o_{A_i})}{n}$$

$$
\begin{aligned}
&E(\widehat{\oplus}(o_{A_1},o_{A_2},...,o_{A_n}))\\
&= t_{\widehat{\oplus}(A_1,\cdots,A_n)} * c_{\widehat{\oplus}(A_1,\cdots,A_n)} + (1 - c_{\widehat{\oplus}(A_1,\cdots,A_n)}) * f_{\widehat{\oplus}(A_1,\cdots,A_n)}\\
&\ldots [\text{Substitution of } t_{\widehat{\oplus}(A_1,\cdots,A_n)}, c_{\widehat{\oplus}_w(A_1,\cdots,A_n)}, f_{\widehat{\oplus}_w(A_1,\cdots,A_n)} \text{ under different cases}]\\
&[\text{using Table 5.4}]\\
&\ldots [\text{after several steps of algebraic simplifications and using the Equation 5.1 for each } A_i]\\
&= \frac{E(o_{A_1}) + E(o_{A_2}) + \cdots + E(o_{A_n})}{n} = \frac{\sum_{i=1}^{n} E(o_{A_i})}{n}
\end{aligned}
\tag{D.13}
$$

■

## D.7 Proof: Theorem 5.3.7

PROOF We prove the theorem and omit detail algebraic simplifications for brevity.

$E(\widehat{\oplus}_w(o_{A_1}, w_1), (o_{A_2}, w_2), \cdots, (o_{A_n}, w_n))$
$= E(\widehat{\oplus}_w(t_{(A_1,w_1),\cdots,(A_n,w_n)}), \widehat{\oplus}_w(c_{(A_1,w_1),\cdots,(A_n,w_n)}), \widehat{\oplus}_w(f_{(A_1,w_1),\cdots,(A_n,w_n)}))$
$= t_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))} * c_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))} + (1 - c_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))})*$
$f_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))}$
$\ldots$ [Substitution of $t_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))}, c_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))}, f_{\widehat{\oplus}_w((A_1,w_1),\cdots,(A_n,w_n))}$]
[using Table 5.5]
$\ldots$ [replace $c_i$ by $c$ where $i = 1 \cdots n$ and algebraic simplifications]
$= \dfrac{w_1 * (t_{A_1} * c + (1 - c) * f_{A_1}) + \cdots + w_i * (t_{A_n} * c + (1 - c) * f_{A_n})}{w_1 + w_2 + \cdots + w_i}$
$= \dfrac{\sum_{i=1}^{n} w_i E(o_{A_i})}{\sum_{i=1}^{n} w_i}$ [Using Equation 5.1 for each $A_i$ and $c = c_{A_1} = c_{A_2} = \cdots = c_{A_n}$]

(D.14)

■

## D.8 Proof: Theorem 5.3.8

PROOF We prove the theorem and omit detail algebraic simplifications for brevity.

$\dfrac{\sum_{i=1}^{n} E(o_{A_i})}{n} = \dfrac{E(t_{A_1}, c, f_{A_1}) + E(t_{A_2}, c, f_{A_2}) + \cdots + E(t_{A_n}, c, f_{A_n})}{n}$
[If $c = c_{A_1} = c_{A_2} = \cdots = c_{A_n}$]
$= \dfrac{(t_{A_1} * c + (1 - c) * f_{A_1}) + (t_{A_2} * c + (1 - c) * f_{A_2}) + \cdots + (t_{A_n} * c + (1 - c) * f_{A_n})}{n}$
[Using Equation 5.1]
$= \dfrac{(t_{A_1} + t_{A_2} + \cdots + t_{A_n}) * c + (f_{A_1} + f_{A_2} + \cdots + f_{A_n}) * (1 - c)}{n}$
$= t_{\widehat{\oplus}_w(A_1,A_2,\cdots,A_n)} * c + f_{(A_1,A_2,\cdots,A_n)} * (1 - c)$
[Substitute $c_{A_i}$ by $c$ and $w_i$ by $w$ where $i = 1 \cdots n$ in Table 5.5]
$= E(\widehat{\oplus}_w((o_{A_1}, w), (o_{A_2}, w), \cdots, (o_{A_n}, w)))$
[Using Equation 5.1 and Definition 5.3.2]

(D.15)

■

The proof of the same theorem for *C.FUSION* operator can be carried out analogously using Table 5.6.

## D.9 Sketch of the Proof: Equivalence with Averaging Fusion in Subjective Logic

The proof regarding the equivalence between CertainLogic's $A.FUSION$ and Subjective Logic's averaging fusion operator follow the same algebraic process as demonstrated in Appendix C.5. Hereby, we only provide the sketch of the proof.

A bijective mapping (cf. Definition 5.2.5 and 5.2.6) between an opinion in CertainTrust given by its parameters, $o = (t, c, f)$ and Subjective Logic, where the opinion is given as $o = (b, d, u, a)$ has been provided in [Rie09b]. To prove the equivalence between the *average fusion* proposed in this thesis and the *averaging fusion and consensus operator for dependent opinions* proposed for subjective logic in [Jøs09, JMP06] one can start with the definition of the average fusion/consensus operator for dependent opinions in subjective logic and replace the parameters $b$, $d$, $u$, and $a$ by the corresponding parameters of CertainLogic following the bijective mapping. Finally, one applies the bijective mapping another time in order to convert the resulting equations, which are in the form of $b$, $d$, $u$ and $a$ to calculate the parameters of CertainLogic $t$, $c$, and $f$. The result will be equivalent to the *average fusion* defined in this thesis.

**Erklärung**[1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades "Dr. rer. nat." mit dem Titel *Trust Establishment Mechanisms for Distributed Service Environments* selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, 09. Aug 2013 Sheikh M. Habib

[1] gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

**Wissenschaftlicher Werdegang des Verfassers**[2]

| | |
|---|---|
| 06/1999 – 09/2003 | Studium (B.Sc.Engg.)<br>Computer Science and Engineering<br>Khulna University of Engineering and Technology<br>Bangladesch<br>Bachelorarbeitsthema: *Genetic Programming: An Approach to Automatic Synthesis of Digital Circuits* |
| 10/2004 – 01/2007 | Beschäftigung als Lecturer an der<br>University of Science and Technology Chittagong<br>Bangladesch<br>Wissenschaftliche Vertiefung u.a. in den Bereichen Algorithmen, Analyse und Entwurf von Informationssystemen, Formale Grundlagen der Informatik (Theory of Computing) |
| 09/2007 – 07/2009 | Studium (M.Sc.)<br>Networks and Distributed Systems<br>Chalmers University of Technology<br>Schweden<br>Masterarbeitsthema: *Security Evaluation of Windows Mobile Operating System* |
| 10/2009 – 09/2013 | Stipendiat im CASED GraduiertenSchule<br>Fachgebiet Telekooperation, Fachbereich Informatik<br>Technische Universität Darmstadt<br>Deutschland |

[2] gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt