UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Smart Contracts for Blockchain-based Reputation Systems

by Ahmed Saud Almasoud

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Associate Professor Farookh Hussain

University of Technology Sydney
Faculty of Engineering and Information Technology

May 2020

Certificate of Authorship / Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.
....
Signature of Candidate

12 May 2020
Date

Acknowledgement

I start by thanking Allah, lord of the world, for he has given me all the possibilities, blessings, and exceptional experience. He has given me ways to reach what I have reached so far. When a dream comes true, you should look a little backward to thank those who have had a major role in this success. Success cannot be achieved without the joint effort of different parties.

I thank the owner of the noble and essential role in this success, Professor Farookh Hussain, who tolerated my mistakes and was not only a supervisor of my thesis, but also the older brother, the supreme leader and the glorious professor; as he never hesitated to give me or extend a helping hand when I needed it. Honestly, without him being by my side, I could not pass this stage of my life. I owe him for all this success.

I would also like to thank the University of Technology Sydney (UTS) for giving me this opportunity to study in it. It was like my second home.

I would like to thank my parents, my wife, and my colleagues; without their support, this work could not have been completed.

LIST OF PUBLICATIONS

Journal publication

1- Almasoud. A. S., Hussain, F. K., and Hussain, O. K., ‘Smart Contracts for Blockchain-based Reputation Systems: A Systematic Literature Review’ 2019, *Journal of Network and Computer Applications (JNCA).* **(ERA, CORE – Q1 Journal).**

Conference publication

1- A. S. Almasoud, M. M. Eljazzar and F. Hussain, "Toward a Self-Learned Smart Contracts," *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, Xi'an, 2018, pp. 269-273. **(ERA, COREB*-Rank).**

بسم الله الرحمن الرحيم
١٤٢٠ م

Table of Contents

List of Figures

List of Tables

Abstract

Smart contracts are computer protocols that are meant to oversee, enforce, or verify performances or negotiations of contracts. These protocols ensure that no third parties are involved as a part of the transaction and ensure the security and credibility of the contracts. Reputation systems have been widely implemented in e-commerce applications and websites. Reputation systems provide a platform through which users can measure the trustworthiness or reliability of people offering online services or products. Previous researchers have already proposed the use of Blockchain technology for Reputation Systems.

A Blockchain is generally built on a peer-to-peer (P2P) network and adheres to a certain protocol for the communication amongst the blocks and for the validation of new blocks. However, through a systematic literature review, we have identified that the existing literature has not proposed the use of smart contracts for blockchain-based reputation systems. Using smart contracts in reputation systems can play a vital role by adding an additional layer of openness and security.

The contracts are secured using hash signature, which makes the smart contract "immutable" to alteration. The decentralized application (Dapp) used in the system offers large file storage, protection of the user personal data, low costs of transactions as well as easy bug fixing. Smart contracts can be used to implement the proof of reputation (POR) consensus algorithm which aims at providing quantification for the various systems that are built using the blockchain technology.

POR aims at using the reputation of the participants in the system to secure the network. In the event that a participant attempts to cheat on the smart contracts, they stand to face serious consequences both financially as well as brand wise. The consensus ensures that reputation is paramount in the blockchain system. The primary objective of this study is to propose and develop an intelligent framework termed (FarMed) that is centered on smart contracts-based reputation system.

The intelligence built into Farmed provides automated and reliable mechanisms for the following: (i) determining the current reputation value of a service provider; (ii)

intelligent mechanisms for trust-based inferencing in different contexts; (iii) intelligently preventing people from manipulating reviews in the reputation system; and (iv) providing a platform for transferring the reputation value of a service provider to other service providers. Finally, in order to validate the performance and accuracy of the proposed framework in this thesis, software prototyping will be chosen as the model of choice.

Chapter 1 : Introduction

1.1 Introduction

Technology has over the years eased the way communication is carried out between interacting parties. The advancements in the technology sector allow for continued success in how various entities interact with each other. Online service provisioning typically takes place between entities who have not transacted with each other before (Audun et al., 2007). As a result, issues related to trust arise ensuring that a user (service requestor) accepts the risk of transacting with the other entity (service provider) even before it receives the service. This means that the user does not have the ability to test and see whether the individual whom he/she is in a transaction is actually offering the claimed (or said) services. This is where reputation systems come to the rescue of such users (Casassa et al. 2001). Reputation systems provide a platform through which such users can measure the legitimacy of people offering online services or products. Typically, reputation systems allow a service requestor to rate an individual providing online services (service provider) and the (aggregated or accumulated) score of the service provider can be used by other individuals to make a decision of whether or not they want to transact with the said individual.

One of the areas where reputation systems have been widely implemented is in e-commerce applications. E-commerce refers to purchasing and selling of goods and services over the internet (Christidis, 2018).

These business transactions occur either as consumer-to-consumer, consumer-to-business, business-to-business, or business-to-consumer. Various technologies have been implemented to facilitate e-commerce systems and these include mobile commerce, digital funds transfers, internet marketing, supply chain management, electronic data interchange (EDI), and inventory management system. Different e-commerce platforms are currently available over the World Wide Web (WWW) such as eBay, Amazon, Alibaba, Shopify, Magento, Wix, OpenCart, SquareSpace among many others.

Establishing an online presence is simple and easy, and an online presence provides little evidence of the trustworthiness of an individual. It is for this reason that reputation systems are crucial when it comes to ensuring that reliable information is available about the providers so that consumers can make a fact-driven reliable decision (Sherman, 2018). In addition, reputation systems play a pivotal role in the financial services industry. As Gopalan et al. (2011) noted, there are two kinds of trust here. The first one is trust between participants in the financial transaction, that the parties will honor their own side of the agreement, even if it means unforeseen losses for one or more of them, second is trust by the populace in general that the financial industry is focused on its core role of efficiently bringing savers and investors together in ways that enhance the allocation of private savings to investment in physical and human capital. Trust helps financial institutions build reputation systems over time and this boost the viability of the institutions.

1.2 Statement of the Problem

The above description provides an introduction to the important role played by reputation systems. It also outlines some of the issues with reputation systems. Previously, different techniques have been implemented in the development of reputation systems. Some of these techniques include having centralized reputation systems that store the ratings in one centralized location, where the user(s) can obtain the data whenever they need it from the centralized location. This type of reputation system opens up such as a system to several challenges including in the case of Denial or Service (DoS) attacks that would mean that the system would be completely ineffective. Another type of reputation system is one in which the reputation values are stored across multiple nodes in a distributed manner. Users can fetch ratings (for the provider) from other users in a distributed manner and make decisions. This addresses the issue mentioned about the centralized reputation systems but also opens up the system to security issues, rating fraud or rating manipulation. The next section discusses what can be done to fix the security issues faced by the distributed reputation systems (Casassa et al., 2001).

Reputation systems should be able to support a large number of users, ensure the integrity of the ratings or trust scores, and also provide reliable mechanisms to support new users to bootstrap into the reputation-based economy. These are current issues in the generation of reputation systems, both in research and in practice. Blockchain technologies have the potential to address these issues. *This research represents the first attempt in addressing these issues using Blockchain. It also represents the only approach of its type that proposes innovative intelligent-based algorithmics on top of Blockchain for carrying out reliable trust and reputation computations.*

1.3 Blockchain Technology

Blockchain technology allows the creation of a decentralized environment, where data and transactions cannot be controlled by any third-party (Caesar, 2018). Any transaction that is completed is recorded in a public ledger in a secure, verifiable, permanent, and transparent way, with a timestamp and other details.

A blockchain can be defined as a growing list of records that offers security to the transactions that have been conducted through cryptography (Christidis and Devetsikiotis, 2016). Each block is made up of the timestamp of the previous block, a cryptographic hash, and the transaction data. Blockchains are designed in such a way that they ensure that there is no data modification. The blockchain technology promotes a system where an open ledger is stored within a distributed network where records of various transactions between different parties are stored in an efficient, permanent and verifiable way. The blockchain technology generally takes advantage of peer-to-peer (P2P) networks that adhere to a certain protocol for the communication amongst the blocks, and the validation of new blocks. Once a transaction is complete, the action is irreversible and exists within a certain block within the network. This means that if the data within a certain block is altered, then it leads to the alteration of the data within the other blocks (Xu et al., 2016).

1.3.1 Blockchain Structure

As stated, no data can be altered within a block without altering data within other blocks and causing collusion within the network. This ensures that users can verify and perform checks for the transactions in a cheaper way. Blockchain records are thus maintained autonomously by individuals within a P2P network, verified through the mass collaboration of users within the network made possible by collective interests. This approach ensures that a unit is only transferred once meaning that double-spending is eliminated. Blockchains thus assign title rights to users since they provide records that need to be offered then accepted before the transfer of rights occur (Watanabe et. al, 2015).

1.3.2 Blocks

Blocks within the Blockchain technology contain valid transactions (Watanabe et al., 2015). Each block is made up of the cryptographic hash of the previous block within the chain providing the link between two blocks. This process ensures that there is integrity within the records stored by each of the blocks back to the first block. In some cases, some blocks are produced simultaneously creating what is known as a temporary block (Watanabe et al., 2015).

1.3.3 Block Time

According to Xu et al. (2016), the block time can be defined as the time taken for the Peer to Peer network to generate a block within the chain.

1.4 Smart Contracts

Smart contracts can be defined as computer protocols meant to oversee, enforce, or verify performances or negotiations of contracts (Delmolino et al., 2016).

These protocols thus ensure that no third parties are present during this processing of such transactions and ensure the credibility of transactions. Such transactions are irreversible, and the trail of the records can be tracked due to the record-keeping abilities of the smart contracts. Smart contracts ensure that transactions conducted within the confines of the WWW are secure and that the transaction costs are lower than what is found in traditional contracts (Buterin, 2017).

According to Delmolino et al. (2016), smart contracts provide several advantages when implemented:

i. Autonomy: Smart contracts provide autonomy since an individual does not have to contract third parties for a specific transaction to be conducted or more importantly authorized. This eliminates the chances of manipulation by third parties since the entire implementation is performed by the network.

ii. Trust: Transactions are stored and documented in a ledger that is distributed and shared between multiple parties (also known as nodes) hence there is no possibility of the loss of a ledger.

iii. Backup: There is a duplication of the ledger within the network hence transactions are always available.

iv. Safety: Encryption is performed in smart contracts hence the data within the ledgers remain safe.

v. Accuracy: The automated system of performing transactions and the elimination of the human factor ensure high accuracy in smart contracts.

However, despite their great potentials, smart contracts also have some disadvantages. The prime among these is the lack of regulation. Internationally, both Blockchain and Smart Contracts have not been regulated so far. In addition, the impossibility of changing smart contracts can be a disadvantage as well. When parties involved in the contract genuinely want to change some terms of the agreement, it becomes difficult.

1.5 The relationship between Smart Contracts and Blockchain

Smart contracts are implemented through the blockchain. A smart contract is akin to a traditional contract in the physical world, but it is completely digital and is characterized by a computer program that is stored inside a blockchain.

The blockchain based smart contracts have introduced a new epoch of computational law whereby contracts are supported and agreed on by a blockchain which is universally prevalent and unbiased.

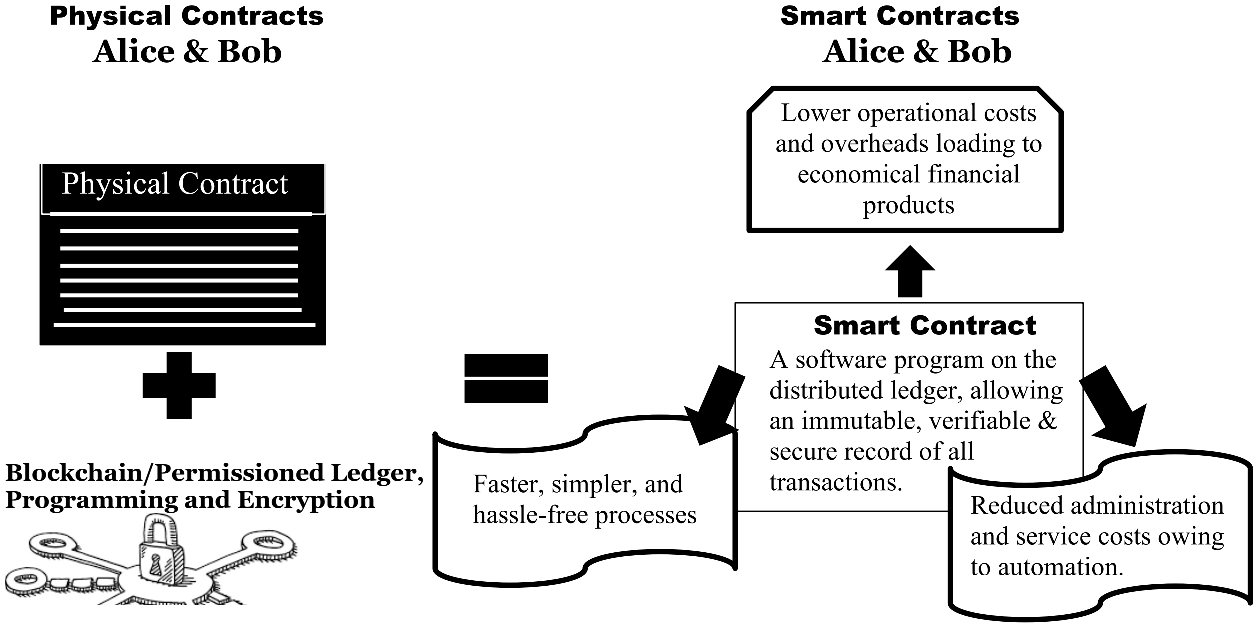

Physical Contracts
Alice & Bob
Physical Contract
Blockchain/Permissioned Ledger,
Programming and Encryption
Smart Contracts
Alice & Bob
Lower operational costs and overheads loading to economical financial products
Smart Contract
A software program on the distributed ledger, allowing an immutable, verifiable & secure record of all transactions.
Faster, simpler, and hassle-free processes
Reduced administration and service costs owing to automation.

Figure 1. 1. Relationship between smart contract and blockchain technology

1.6 Purpose of Using Smart Contracts in Blockchain Technology

Smart contracts represent the software codes that run over the blockchain to implement various types of transactions for satisfying particular conditions similar to traditional contracts (Toneli et al. 2018). According to Lauslahti et al. (2017), the smart contracts are algorithmic, self-enforcing, and self-executing computer programs. These smart contracts eliminate the necessity of a trusted third party in the transactions by allowing untrusted parties to manifest contract terms (Wohrer & Zdun 2018).

Cong & He (2018) explained how decentralized ledger technologies such as blockchains can simplify the creation of smart contracts. Furthermore, using smart contracts, the users are able to codify precisely their trust relations and agreements. Cong & He (2018) noted that these will be automatically executed by the platform like Ethereum after deployment. In fact, these smart contracts can facilitate economic activities by providing services effectively that are offered traditionally by intermediaries (e.g., notaries, banks, and courts) and trusted third parties.

In order to secure blockchains, Watanabe et al. (2016) proposed a new mechanism based on smart contracts that can be implemented in contracts management. A new consensus method was used in this mechanism that used credibility scores to create a hybrid blockchain. In this way, it was possible to thwart the attackers from monopolizing resources and ensure the security of the blockchains (Everts & Muller 2018). Cong & He (2018) also opined that with a large range of economic outcomes, smart contracts and blockchain can sustain market equilibria.

Kosba et al. (2016) presented a decentralized smart contract system called 'Hawk'. This new system can guarantee transactional privacy by ensuring that the system does not store information about financial transactions on the blockchain. In this way, it can help different parties to transact safely as the transactions are not exposed.

Buterin (2014) and Wood (2014) discussed a smart contract platform called 'Etherium'. The authors highlighted the issues regarding its design, implementation, and also the opportunities it provides. Some of the common security patterns on Ethereum were discussed by Wohrer & Zdun (2018). These included Checks-Effects-Interaction, Emergency Stop (Circuit Breaker), Speed Bump, Rate Limit, Mutex, and Balance Limit. Developers can address the security problems like harmful call-backs and uncontrollably high financial risks by applying these patterns.

Marino & Juels (2016) found the need to define a new set of standards for undoing and altering smart contracts, as the traditional tools often fail in this regard. The authors developed a new set of standards and tried to evaluate their performance on Ethereum, a popular smart contract platform. In the end, they succeeded in their approach and proved the value of such a framework.

Idelberger et al. (2016) tried to find out the technical and legal advantages of using logic-based languages instead of procedural languages for programming smart contracts in the blockchain system. The authors concluded that smart contracts based on logic-based languages are easier to work with for the developers. Furthermore, logic-based smart contracts can also reduce the risk of errors in the implementation. In addition, it is possible to ease the validation process by using this particular type of smart contract.

Gatteschi et al. (2018) tried to help insurers to decide whether to adopt blockchain technology or not by clarifying the concepts of blockchain and its advantages and disadvantages. The authors argued that blockchains and smart contracts can improve customer experience and reduce operating costs, minimize the overhead related to the verification of new customers and manual data entry, compute risk assessments and prevent frauds.

1.7 Peer-to-Peer Networks

Decker & Wattenhofer (2013) defined Peer-to-Peer (P2P) is an alternate network model that is delivered by a traditional client server design. P2P networks make use of a dispersed model in which every appliance, that is referred to as a peer, purposes as a client with its own layer of functionality server. A peer functions as the client and as the server at the same time, in that, the peer can start requests to other peers and at the same time reply to requests received from other peers on the same network. It varies from the traditional client-server model where a customer can only direct their needs to a server and then wait for the server to respond (Decker and Wattenhofer, 2013).

Through the client-server approach, the server's performance will deteriorate as the number of customers demanding services from the server increase. In P2P networks, as the number of peers in the network increases, the overall performance of the network improves. These peers can arrange themselves into ad-hoc clusters as they share information, cooperate and share bandwidth with each other in order to accomplish tasks (such as sharing of files). In P2P networks, each peer can upload as well as transfer at the same time; furthermore, new peers can join the network while old peers leave at any time.

Another feature of a P2P network is its ability to tolerate fault, also known as fault-tolerance. When a peer is detached from the main network, the P2P application will continue functioning by using other peers. For instance, in a BitTorrent system, any clients transferring a given file are also functioning as servers in the system. When a client discovers that one of the peers is not replying, he/she will search for other peers, recover parts from the old peer and continue with the downloading process. A P2P network is more fault tolerant compared to the client-server model, where all information sharing will halt if the server is down (Decker and Wattenhofer, 2013).

1.7.1 P2P Network Topology

All P2P topologies, even though they may appear different, have a single shared characteristic. All file transmissions occurring between peers are continuously done directly through a data connection that exists between the peer distributing the file and the peer demanding for it. The control process preceding the transfer of files can, however, be applied in numerous other ways. P2P file sharing networks can be categorized into four basic groups: the centralized, decentralized, hierarchical and ring systems. Although it is possible for these topologies to exist on their own, it is typically the practice for distributed systems to have a more multifaceted topology through the combination of several basic systems in order to create a hybrid system (Christidis and Devetsikiotis 2016).

1.8 Reputation Systems

Reputation systems provide users with the ability to share their experience with other users ensuring that such users have the ability to make sound decisions based on the feedback provided by individuals who have used the products or services (Greenwald, 2014). From a service provider’s point of view, this represents a marketing tool (Josang 2009), which ensures that the users do not risk damage to their reputation, as a result of poor-quality services.

According to Atzori (2016), the reputation of a specific user has a narrow link with the trustworthiness of that user. Reputation can be defined as what is said about a user (by others), what is believed about the said user, or their character or stance. As a result, reputation is derived from the observations of all the members of a social network.

Reputation can be observed to be a collaborative evaluation of trustworthiness that is based on the ratings or the referrals provided by members within a certain social network. As a result, a user can base their trust on the referrals that a trustee has and their personal experience. So as to avoid loops and dependencies, it is a requirement that referrals be obtained from first-hand experience only and not from other referrals (Buchegger and Le Boudec, 2013).

Sherman (2018) noted that blockchain presents a way to change how the online reputation system is managed. Building a decentralized reputation system as a smart contract will give a standard way of having access to the reputation data that has been accumulated, where more verifications across platforms can reinforce the reliability of any one reputation score. Integrating a proof-of-individuality framework in the verification system will guard against Sybil attacks. For example, it will prevent forging and creating multiple identities to manipulate scores.

Sherman (2018) also illustrated the use of this system in an e-commerce marketplace noting that it will involve the verification of user's identity, expelling bot or duplicate accounts, facilitating bi-directional reputation system mechanisms like down-voting or up-voting for users to leave feedback. It will also involve aggregation of individual user's reputations across different platforms on the network, quantification and tokenization of users' reputations on the platform based on different factors including the quality of service or product, after-sales support or response time.

According to Caesar (2018), Bitconch cryptocurrency proposed a proof of reputation algorithm that offers a new solution to the blockchain technology. The algorithm modes time, social network and contribution activities to create a decentralized reputation system. The users with high reputation values are defined as "mutual trust nodes". They can start the payment channels to fast-track offline transactions through micro-transactions.

1.9 Challenges Faced by the current Generation of Reputation System

While the model for a reputation system that has been developed by various scholars may prove to be effective to a large extent, it also has some challenges. As Vyshegorodtsev et al.

(2013) noted, it is possible for a mischievous user to build fake identities and use them to boost their own reputation. Also, according to Eyal and Sirer (2013), the image of Bitcoin as a currency is undergoing some troubles because some illicit behaviours and frauds have taken place during its short and recent life. Eyal and Sirer (2013) noted that among the most notable of these is the bankruptcy of the MtGox Exchange which used to be the biggest and the most visited site to sell or buy Bitcoins as against fiat currencies.

In addition, Poon and Dryja (2016) concluded that it is unlikely that a network with a huge number of low resourced users would have the ability to implement a workable reputation system. It would take a significant time after the deployment before the reputation system would become effective, gaining the essential data and feedback from users that would subsequently allow other users on the network to make good decisions regarding the trustworthiness of a peer. Therefore, it will take several months or even years before the potential of the reputation system will manifest and this is a key challenge.

The current reputation system finds it difficult to infer and model the trust value that a service provider should have in a specific context. This should normally be done by using the inference from existing trust values in related contexts. This study will solve this problem by using an ontology-based method to model the context-specific nature of reputation and trust.

Although Prisco (2015) proposed a system that solves some problems occurring in the current generation of reputation system, he explained that the risk of the unknown technical flaws that are in the cryptography used in securing the system can greatly undermine the entire security of the system. The study also concluded that reputation systems will never be able to defend all possible attacks with a hundred percent success rate.

To tackle some of the issues identified above, this study will involve the development of a framework that will integrate smart contracts, reputation systems and service-oriented computing. The study will also develop reliable and intelligent methods for transferring reputation from one platform to another so that no service provider loses the reputation they have gained. In addition, this research will evaluate the various methods that can be adopted to preserve the integrity of the reputation or trust values of a service provider.

1.10 Research Contribution

This proposed research aims to develop an intelligent Ethereum Smart contracts-based reputation system in a Blockchain network. This section discusses the contributions of this research.

1.10.1 Scientific Contribution

a. This is the first research that proposes and implements Ethereum smart contracts as the building block of a reputation system so as to preserve and ensure the integrity of the reviews.
b. This is the first and only work that makes available the developed Ethereum smart contracts solution for reputation system as-a-service to other consumers.
c. This is the first research that focuses on implementing algorithms to compute trust value in an Ethereum smart contract-based reputation system as a service in the Blockchain network.
d. This research is the first to propose an intelligent and reliable mechanism for transferring or trading reputation value from one service provider to another service provider.
e. This research is the first to propose an intelligent and accurate approach for the computation of reputation value of a service provider based on smart contracts.
f. This research is the first to propose an intelligent and accurate approach for inferring the reputation value of a service provider in a different context based on smart contracts.

1.10.2 Social Contribution

1. This study will help E-markets consumers to use applications and deal with service providers more confidently and securely. It also will be a step towards enabling and building online reliable trustworthy environments for e-commerce.
2. This study will help the good service providers exhibit their advantages more accurately and efficiently. Furthermore, it will help service providers to focus on other tasks which will in turn will increase productivity.

1.11 Thesis Plan

In this thesis, we proposed and developed various models, services and algorithms that enable the use of smart contracts for blockchain-based reputation systems. In order to achieve all the objectives of the study, this thesis has been organized in nine chapters as shown in Figure 1.2. We give a brief summary of each chapter in this section:

Chapter 2: Chapter two provides a systematic literature review of the existing literature on reputation systems in general and the use of blockchain technologies in a reputation system in particular. The primary aim of this chapter is to clarify that the issues that we intend to address through this thesis have not been previously addressed and resolved in a previous research.

Chapter 3: Chapter three formally defines all the issues that are to be addressed in this thesis. This is done by creating research questions from these issues. Thereafter, these questions are used in formulating research objectives which will be the targets and primary aims of the study.

Chapter 4: Chapter four presents the research methodology, that is, the methodological approach that is used in addressing the gaps and loopholes identified in the literature review. Specifically, the design science research approach was selected as the model to be used.

Chapter 5: Chapter five presents the solution developed to address Research Objectives 1 and 2. In precise terms, a *FarMed* service was built to integrate reputation systems, smart contracts, and service-oriented computing. Also, a model was developed to compute the reputation value of the service provider based on the trust values in the smart contracts. This model is outlined and discussed in detail in Chapter 5.

Chapter 6: Chapter six provides context-driven inferencing of trust value for service providers. The chapter includes a detailed description of service ontology while algorithms are modeled for context-based inference and for semantic distance computation. This is the solution to research objective 3.

Chapter 7: Chapter seven presents Reputation Auction Service (RAS). RAS provide a novel and intelligent method for bootstrapping of new service providers in the reputation-based economy. The service will address objective 4 of this research.

Chapter 8: Chapter eight provides the working of the prototype developed to answer all the research questions in this study. This demonstration was carried out with the use of screenshots and accompanied with appropriate and adequate explanations.

Chapter 9: Chapter nine concludes the thesis by providing a basic summary of what has been achieved and what can be done to expand the study in the future.

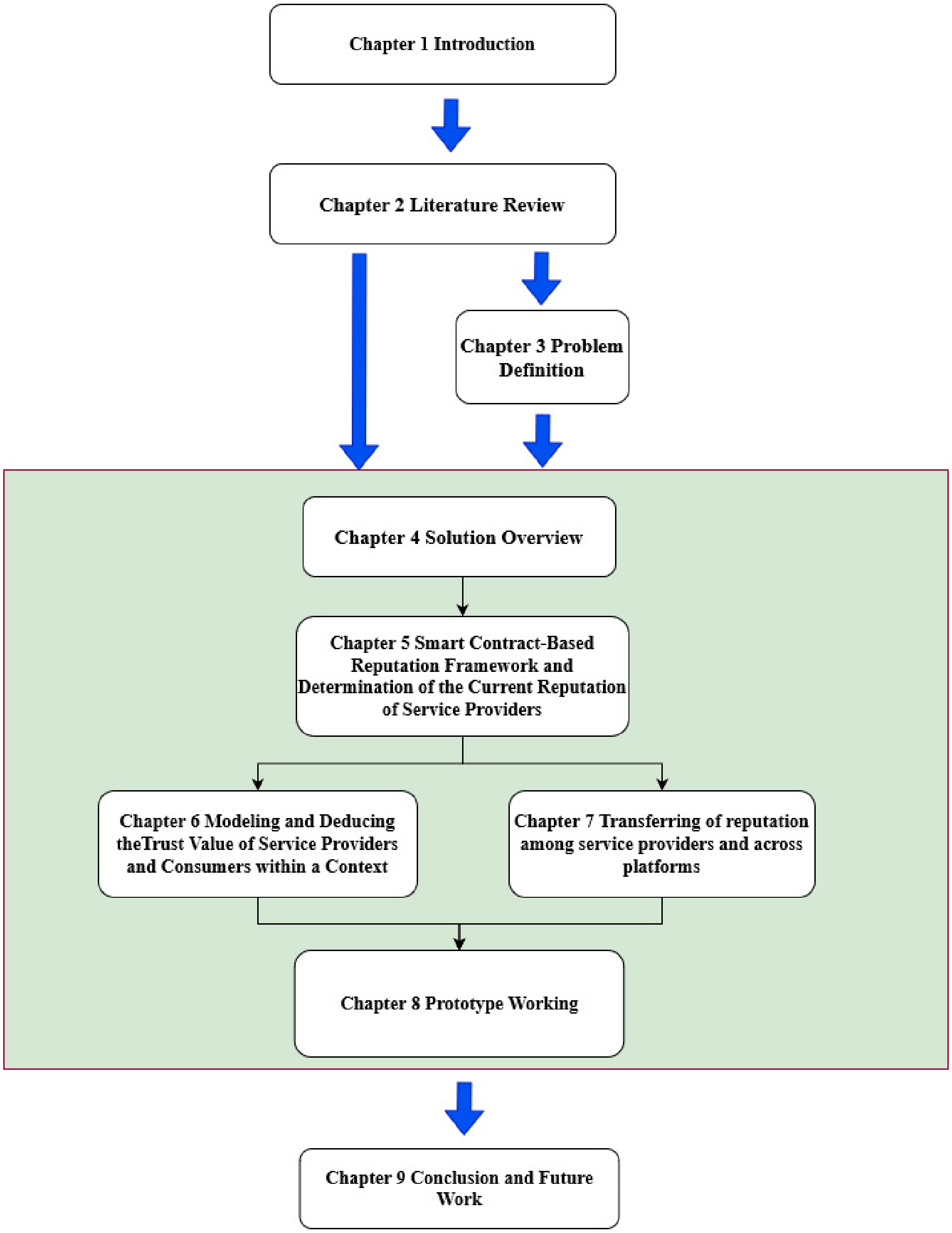

Chapter 1 Introduction
Chapter 2 Literature Review
Chapter 3 Problem Definition
Chapter 4 Solution Overview
Chapter 5 Smart Contract-Based Reputation Framework and Determination of the Current Reputation of Service Providers
Chapter 6 Modeling and Deducing theTrust Value of Service Providers and Consumers within a Context
Chapter 7 Transferring of reputation among service providers and across platforms
Chapter 8 Prototype Working
Chapter 9 Conclusion and Future Work

Figure 1. 2. The thesis structure

1.12 Conclusion

This chapter gave an introduction to reputation system, blockchain technology and smart contracts. The relationship between blockchain technology and smart contracts was also discussed. The problem statement of this study was highlighted, and all the challenges faced by the current generation of reputation systems were discussed. The chapter also presented the contribution that this study will make both scientifically and socially.

In the next chapter, a systematic literature review of all related papers is made. The objective of this systematic literature review is to ensure that the issue this study proposes to address has not been solved by a previous study.

Chapter 2 A Systematic Literature Review

2.1 Introduction

In this chapter, a systematic literature review on the use of smart contract for blockchain-based reputation systems will be carried out.

In a distributed decentralized environment, establishing an online presence is simple and easy but it provides little evidence about the trustworthiness of an individual. Thus, when service provisioning occurs between entities who have, hitherto, not made transactions with each other, the notion of trust and/or reputation is used to ensure that the service requestor accepts the risk of transacting with the service provider even before it receives the service (Josang et al., 2007). Reputation systems provide a platform through which such users can measure the legitimacy of people offering online services or products (Casassa et al. 2001). Typically, reputation systems allow a service requestor to rate an individual's (service provider's) ability to provide online services and the (aggregated or cumulated) score of the service provider can be used by other individuals to decide whether they want to transact with the said individual.

Over the years, reputation systems have been widely implemented in various industries such as e-commerce applications (Christidis, 2018), the financial services industry etc. In these areas, various advancements using reputation systems as their base have been made to facilitate the processes. For example, in e-commerce, various companies such as eBay, Amazon, Alibaba, Shopify, Magento, Wix, OpenCart, and SquareSpace use reputation systems to underpin technologies such as mobile commerce, digital funds transfers, electronic data interchange (EDI) etc. In financial systems, reputation systems play a pivotal role too in helping financial institutions build their reputation over time and boost the viability of their operations. However, in all these cases, the reputational information is stored in either a decentralized or centralized way. While storing the information in a centralized manner has advantages to the user(s) who wish to retrieve the data whenever they need it, it also has several challenges such as denial-of-service (DoS) attacks which render such a system completely ineffective. These drawbacks can be addressed by storing the information in a decentralized way where the reputation values are stored across multiple nodes. In such an environment, users can retrieve ratings (for the provider) from other users in a distributed manner and make decisions. But this platform has its own drawbacks in relation to security issues, such as rating fraud and rating manipulation. These drawbacks render the reputation systems useless as they are not able to support many users, ensure the integrity of the ratings or trust scores, and also provide reliable mechanisms to support new users to bootstrap into the reputation-based economy.

The advent of blockchain technologies is a means to address these drawbacks. As noted by Sherman (2018), blockchain is a way to change how online reputation systems are managed. By integrating a proof-of-individuality framework in the verification system, a blockchain model guards against Sybil attacks and prevents scenarios such as forgery, creation of multiple identities, manipulating scores etc. This technology has led researchers to make new advancements to further carry out processing efficiently. One such advancement is the development and implementation of smart contracts. Smart contracts are software codes that run over the blockchain technology to implement the different required transactions. As shown in Figure 2.1, they are akin to traditional contracts, but have the capabilities of being self-enforcing, far more efficient, and less onerous on the seller and buyer (Toneli et al. 2018).

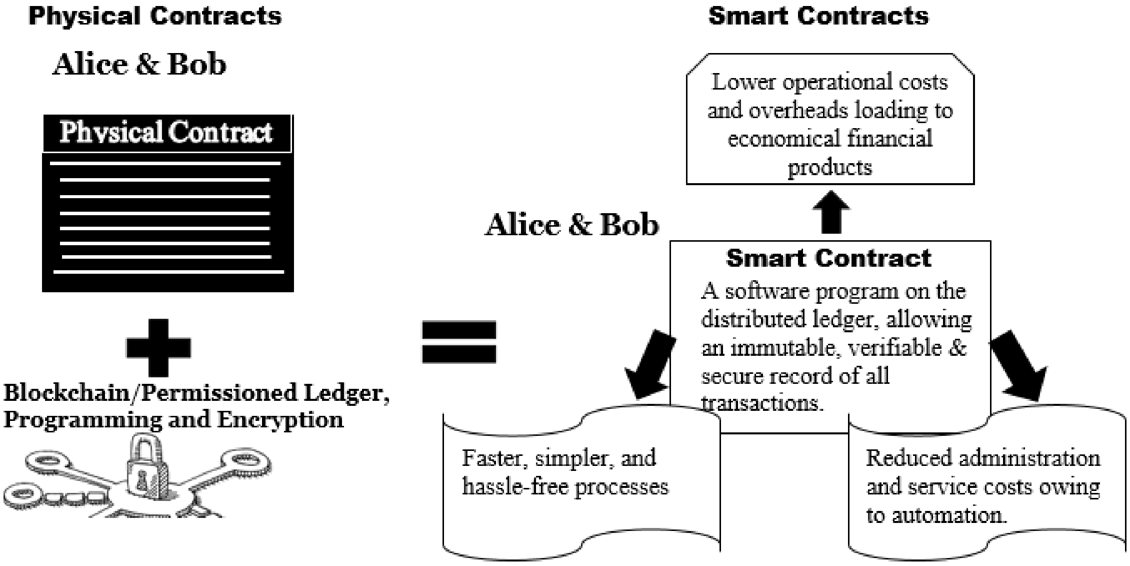

Physical Contracts
Alice & Bob
Physical Contract
Blockchain/Permissioned Ledger, Programming and Encryption
Smart Contracts
Lower operational costs and overheads loading to economical financial products
Alice & Bob
Smart Contract
A software program on the distributed ledger, allowing an immutable, verifiable & secure record of all transactions.
Faster, simpler, and hassle-free processes
Reduced administration and service costs owing to automation.

Figure 2.1. Relationship between Smart Contract and Blockchain Technology

However, as it is with any new technology, there are several open research issues that exist in the wide implementation of smart contracts. The objective of this paper is to understand such current issues by performing a systematic literature review (SLR). The SLR intends to interpret and evaluate the existing relevant literature in relation to the area of study using categorical analysis (Petticrew and Roberts 2006). The categorical analysis is undertaken in the five broad areas in which smart contracts and blockchain-based reputation systems have limitations. These are the inability of service users in (1) deriving the reputation value of service providers based on the values (or ratings) present in the smart contracts; (2) predicting the future trust value of a service provider based its trust values in the blocks; (3) considering the reputation of a service provider as a digital asset and moving across platforms; (4) detecting and dealing with reputation fraud such as bad mouthing, ballot stuffing, positive and negative discrimination, false feedback, and the value imbalance problem etc; and (5) mathematical models and algorithms that assist in addressing the abovementioned gaps. The need to address these drawbacks will be explained in the next section before performing an SLR to enrich smart contracts in these areas.

The structure of this chapter is as follows. Section 2.2 presents five key requirements that should serve as the pillars of a smart contract-based reputation system. In section 2.3, the adopted process of shortlisting the papers chosen for this SLR is discussed. This includes discussing the criteria for searching the literature as well as the inclusion and exclusion criteria. Section 2.4 presents a summary of all the papers that have been shortlisted, totaling 30 in all. In section 2.5, a framework named *FarMed* Service is proposed as a smart contract-based reputation system service and all its components are thoroughly discussed. Finally, section 2.6 concludes the SLR.

2.2 Key requirements from a Smart Contract

This section discusses factors to be considered in smart contracts that will form the basis of a comparison of different existing papers in this SLR. Smart contracts are computer protocols designed to oversee, enforce, or verify performances or negotiations of contracts

(Delmolino et al., 2016). By ensuring that no third parties are present during the processing of such transactions, which are irreversible and leave a trail of records for record-keeping, these protocols ensure their credibility (Buterin, 2017). These characteristics are beneficial and assist the user in the *during* phase of the smart contract. As shown in Figure 2.2, the smart contract phase between Alice (user) and Bob can be categorized into three different phases, namely *before*, *during* and *after*. *Before* is that phase in which Alice and Bob negotiate to decide on the specifics of the smart contract, prior to forming it. *During* is the phase of collaboration governed by the smart contract, while *after* is the phase from the expiration of the smart contract.

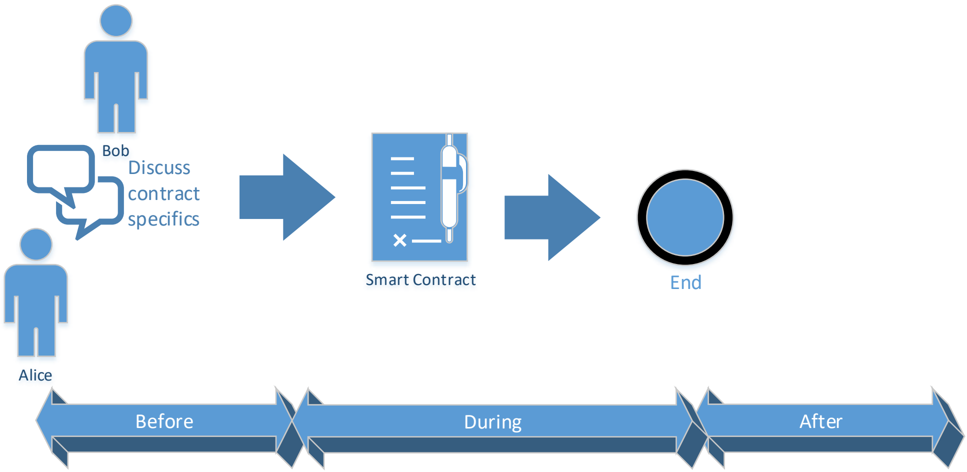


Figure 2.2. The three phases of a smart contract

For the wide application of smart contracts, apart from focusing on the during phase, users like Alice need certain requirements too, as follows:

2.2.1 Ability to derive overall reputation value of service providers based on the values (or ratings) present in the smart contracts (Req: 1)

Smart contracts allow transactions to be stored and documented in a ledger that is distributed and shared between multiple parties (also known as nodes) hence there is no possibility of the loss of a ledger. While they guarantee the truthfulness of operations in the *during* phase, they do not assist Alice in the *before* and *after* phases in tasks such as determining Bob’s credibility to complete the required tasks. However, this is necessary as reputation systems provide users with the ability to share their experience with other users ensuring that such users have the ability to make sound decisions based on the feedback provided by individuals who have used the products or services (Khaqqi et al., 2018). From a service provider’s point of view, having such information is necessary as this represents a marketing tool (Josang 2009). Without this, Alice does not have a comprehensive framework to determine either with whom to form a contract and why or whether to form a contract with Bob or not?

2.2.2 Ability to determine the trust value of a service provider in a context (Req: 2)

According to Atzori (2016), the reputation of a specific user has a narrow link with the trustworthiness of that user. Reputation can be defined as what is said about a user (by others); what is believed about the said user; or their character or stance. As a result, reputation is derived from the observations of all the members of a social network. A model is required to determine the reputation value of the service provider in a particular context. This will help in inferring Bob’s credibility and modelling his reputation in a specific context thereby assisting Alice in making a decision.

2.2.3 Reputation system as a digital asset and moving across platforms (Req: 3)

In most reputation systems, service providers like Bob start off with a neutral or zero ratings and have issues interacting with other users in the system due to their low or non-existent credibility. It becomes a difficult task for new entrants to provide their services to the market. However, as a reputation value is context specific, such providers may have expertise in other contexts which can be used as leverage in this context. The challenge is how can they use this so that their reputation value is not glued to a platform, instead, it will be platform agnostic with appropriate weightings. In other words, a service provider should have the opportunity to transfer reputation from one platform to another, thereby ensuring that the reputation they have earned is not lost and can be used for reputation exchange and bootstrapping of new users (Caesar, 2018).

2.2.4 Detection of non-compliant behavior (Req: 4)

Sometimes, users may try to defeat the system by engaging in reputation fraud in smart contracts such as bad mouthing, ballot stuffing, positive and negative discrimination, false feedback, value imbalance problem etc. This can occur when a user provides ratings even when it has not transacted with a certain user thereby either giving an unfair advantage or hurting them in the process. This can also occur in cases where a service provider develops pseudonyms which they then use to rate themselves. The feedback can also be biased, which may hamper the effectiveness of a feedback-based reputation system (Tadelis, 2016). To avoid such instances, a smart contract should have validated methods of detecting and dealing with any reputation fraud it encounters.

2.2.5 Mathematical models and algorithms (Req: 5)

In order to compute the current trust value of service provider/s by addressing the above-mentioned requirements, specific mathematical models and algorithms are required. Such models need to be intelligent and should have the ability to automatically aggregate the ratings of a service provider and derive its reputation value from its overall rating. The algorithm or model should also be linked to the electronic marketplace (network) and should store specific parameters such as the service provider's address, service consumer's address, rating, and timestamp in a blockchain. This is important as since records on blockchain are immutable, it means the ratings and reputation values can never be altered.

Josang *et al.* (2007) noted that there are different measures of computing reputation and trust values. These models range from using different measures such as the use of simple summation or average of ratings; Bayesian model, which takes binary ratings as input and compute reputation values by a statistical update of beta probability density functions; belief model, which is related to the probability theory but the sum of probabilities here do not necessarily add up to 1 as the remaining probability is regarded as uncertainty; fuzzy models, which use linguistics to represent to what degree a service provider can be described e.g.

trustworthy or not trustworthy; and flow models, which use transitive iteration through looped or arbitrarily long chains to compute reputation values. The issue to be addressed here is what specific measure or a combination of measures to use according to the objective to be achieved for a smart contract-based reputation system.

These five factors form the basis of our investigation into the existing approaches in the literature to determine if they provide a solution to address the issues in the area of smart contracts. Therefore, questions about whether the existing methods assist in addressing these issues will be answered in the comparison. The next section details the approach used to shortlist the papers that are reviewed in the SLR.

2.3 Process adopted for shortlisting the papers for SLR

This section details on the process adopted to identify the relevant papers to be reviewed in the SLR (Sebastian and Dubravka, 2015). We adopted a four-step process:

Step 1: Searching the literature →This step involves defining the search terms, identifying the data sources and the process of data collection.

Step 2: Inclusion and exclusion criteria → Certain criteria are defined to guide the extraction of the most relevant studies.

Step 3: Quality Evaluation → Each of the articles or journal papers are reviewed based on two quality evaluation criteria.

Step 4: Data Analysis → After reviewing the selected studies, data is extracted and recorded.

2.3.1 Step 1: Criteria for searching the literature

This step involves deciding the following:

- Databases used: The electronic scientific databases and data sources selected to source the papers from for the SLR are as follows:

    1. Elsevier ScienceDirect (www.sciencedirect.com/)
    2. IEEE Xplore (www.ieexplore.ieee.org/Xplore/)
    3. Google Scholar (www.scholar.google.com.au/)

    These databases were selected primarily because they provide enough coverage of the literature that is relevant for this SLR.

- Search terms used: In order to create records for building the literature database, the search terms "smart contracts in blockchain", "blockchain and reputation systems" and "smart contracts for blockchain reputation systems" were entered into the publication databases. This resulted in the retrieval of 122 papers from 18 publication venues, as shown in Table 1. All the sources selected are those that include literature surveys or empirical studies.
- Required information from the selected papers: The specific information required for each record is the abstract and the full text document.
- Publication time of the records: For a paper to be considered for SLR analysis, it needed to be published after 2013. This is logical as there is very little consideration of the use of smart contracts and blockchain before this time period. Table 1 shows the selected journals and conference proceedings.

Table 2.1. Journals and Conference Proceedings

S/N	Journal or Conference Proceedings
1.	Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts
2.	International Journal of Human-Computer Studies
3.	International Journal on Semantic Web and Information Systems
4.	International Journal of Computational Intelligence and Applications
5.	Journal of Parallel and Distributed Computing
6.	IEEE Transactions on Information Forensics and Security
7.	Journal of Cases on Information Technology
8.	24th IEEE International Conference on Software Analysis, Evolution and Reengineering
9.	International Journal of Forecasting
10.	Central European Journal of Computer Science
11.	Proceedings of the Second Asia-Pacific conference on Conceptual modelling
12.	International Conference on Trust Management
13.	IEEE Symposium on Security and Privacy (SP)

14.	International Journal of Intelligent Information Technologies
15.	18th IEEE International Conference on Distributed Computing Systems
16.	IFIP International Information Security and Privacy Conference
17.	Journal of Interconnection Networks
18.	Journal of Computer and System Sciences

2.3.2 Step 2: Inclusion and Exclusion Criteria

Not all the records identified in the previous step are considered in the SLR. They are further shortlisted according to the following inclusion and exclusion criteria:

Inclusion criteria:

1. Must contain meta-analyses.
2. Present literature reviews and surveys with a defined search process, research question and data extraction. Regardless as to whether the literature review is only a part of the article or the main component, these articles are included.
3. All research works should be related to the study area.

Exclusion criteria:

The records are excluded if:

1. They are duplicate reports of a similar study.
2. Informal literature reviews that have no defined research questions, no defined data extraction process, or no defined search process.
3. They are not written in the English language

The stages of evaluating and selecting the relevant papers for SLR are summarized in Table 2.2.

Table 2.2. Summary of stages of evaluating and selecting relevant papers for SLR

Evaluation Stage	**Method**	**Assessment Criteria**
1st	Identify the related studies from the databases	Include the search terms
2nd	Eliminate studies based on date of publication	Exclude studies published before 2013
3rd	Eliminate studies based on title	If title includes the search terms (i.e. "smart contracts in blockchain", "blockchain and reputation systems" or "smart contracts for blockchain reputation systems"), include in the study; otherwise, exclude
4th	Eliminate studies based on abstract	If abstract shows study is relevant, include it; otherwise, exclude

As shown in Figure 2.3, 122 studies were found after searching the publication venues. Of these, some papers were eliminated based on their publication date which reduced the number of papers to 109. Then, the titles of the studies were evaluated to find only those that included the search terms; this further reduced the number to 88. Finally, the abstracts of the papers were read and evaluated to eliminate irrelevant papers. After this evaluation, the number of papers reduced to 38, as shown in Figure 2.3.

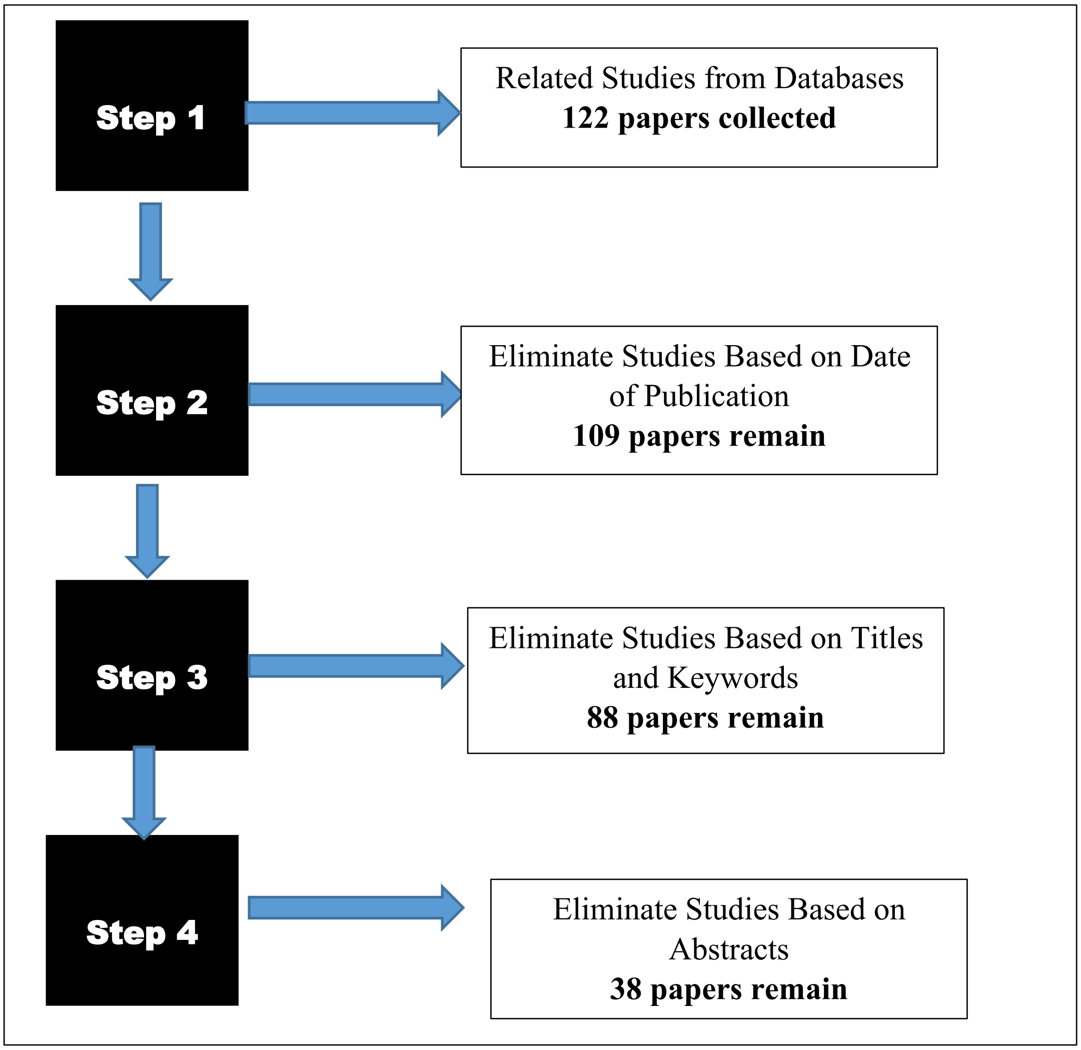

Step 1
Related Studies from Databases
122 papers collected
Step 2
Eliminate Studies Based on Date of Publication
109 papers remain
Step 3
Eliminate Studies Based on Titles and Keywords
88 papers remain
Step 4
Eliminate Studies Based on Abstracts
38 papers remain

Figure 2.3. The study selection process

2.3.3 Step 3: Quality Evaluation

The 38 selected papers from the knowledge record were retrieved and critically evaluated based on the three quality evaluation questions, as follows:

QE1: Does the paper cover relevant work and explore the research topics comprehensively?

QE2: Does the paper provide clear findings with justifiable results and conclusions?

QE3: Does the paper provide future directions?

Any paper which has at least two ‘yes’ answers to the three evaluation criteria questions is included in this SLR. Of the 38 papers, 30 of them satisfied the criteria as shown in Table 2.3.

2.3.4 Step 4: Shortlisted papers for SLR and categorizing them into broad areas

Each of the shortlisted papers was analyzed according to its scope, topic area, author’s information, country and the summary of its research questions and answers. Based on the analysis, the selected papers were categorized into one of the broad areas of *trust and reputation systems*, *blockchain-based reputation systems* or *smart contracts,* as shown in Table 2.3.

The papers in each area are summarized in the next section and their issues and limitations with respect to the key requirements needed from a smart contract, as defined in Section 2.2, are discussed.

2.4 Summary of papers shortlisted in the SLR

2.4.1 Trust and reputation systems

Online reputation systems represent an important kind of mechanism for establishing trust between interacting parties. According to Hendrikx et al. (2015), reputation systems facilitate trust between entities by increasing the effectiveness and efficiency of online services and communities. Such systems are increasingly gaining acceptance by players in different sectors such as health, transportation, finance etc. and are thus becoming an essential fabric of different websites and online services. A great potential exists for online trust and reputation systems and they can be implemented across different sectors. Reputation systems represent a trend when it comes to decision support systems for services offered through the Internet (Dennis and Owenson, 2016). The basic idea of reputation systems is that they allow various users to rate each other after a transaction is completed and show an aggregate rating of a user to provide a reputation score (for the service provider and the service requestor). This reputation score is used by other users to make decisions about whether they should engage with the user providing the service in the future. Dennis and Owenson (2016) argue that such systems provide an incentive for good behaviour within a market, thus ensuring that there is good market quality. For service providers to build their reputation value, they must first build trust. Guy et al. (2015) used two definitions of trust: *'decision trust'* and *'reliability trust'*. Reliability trust can be defined as the infallibility of an individual or something (Guy et al., 2015). Trust is, therefore, the relative probability that a *User A* expects another *User B* to perform a given action on which the welfare of *User A* depends. However, due to the complex nature of trust, it becomes difficult to make a decision on whether to enter into a situation of dependence with another individual or not. On the other hand, decision trust is defined as the extent to which a user is willing to depend on an individual or something at a given time with a feeling of subjective security, even in the case that negative consequences may occur. The abstract nature of decision trust is what provides a foundation of the broader notion of trust which includes dependence on the trusted party.

Study	Author	Date	Topic	Category
S1	Al-Bassam, M. [1]	2017	SCPKI: a smart contract-based PKI and identity system	Smart Contract
S2	Bogner, A., Chanson, M. [4]	2016	A decentralized sharing app running a smart contract on the Ethereum blockchain	Smart Contract
S3	Buechler, M. Earabathini, M. Hockenbrocht, C. Wan [5]	2015	*Decentralized Reputation System for Transaction Networks*	Trust and Reputation Systems
S4	Buterin, V. [6]	2014	A next-generation smart contract and decentralized application platform	Smart Contract
S5	Caesar, C. [7]	2018	How to build a reputation system on blockchain	Blockchain based reputation systems
S6	Cai, Y. Zhu, D. [8]	2016	Fraud detections for online businesses: a perspective from blockchain technology	Blockchain-based Reputation Systems
S7	Carboni, D. [9]	2015	Feedback-based reputation on top of the bitcoin blockchain	Blockchain-based Reputation Systems
S8	Chen, T., Li, X., Luo, X., & Zhang, X. [13]	2017	Under-optimized smart contracts devour your money	Smart Contract
S9	Christidis, K., & Devetsikiotis, M. [14]	2016	Blockchains and smart contracts for the Internet of Things	Smart Contract
S10	Cong, L. W., & He, Z. [15]	2018	Blockchain disruption and smart contracts	Blockchain-based Reputation Systems
S11	Decker, C. and Wattenhofer, R. [16]	2013	Information propagation in the bitcoin network	Blockchain-based reputation Systems
S12	Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. [19]	2016	Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab	Smart Contract
S13	Egbertsen, W., Hardeman, G., van den Hoven, M., van der Kolk, G., & van Rijsewijk, A. [20]	2016	Replacing paper contracts with Ethereum smart contracts	Smart Contract
S14	Frantz, C. K., & Nowostawski, M. [21]	2016	From institutions to code: towards automated generation of smart contracts	Smart Contract
S15	Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaría, V. [22]	2018	Blockchain and smart contracts for insurance: Is the technology mature enough?	Smart Contract

S16	Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. [23]	2016	Evaluation of logic-based smart contracts for blockchain systems	Smart Contract
S17	Bigi, G., Bracciali, A., Meacci, G., & Tuosto, E. [3]	2015	Validation of decentralized smart contracts through game theory and formal methods	Smart Contract
S18	Juels, A., Kosba, A., & Shi, E. [25]	2016	The ring of gyges: investigating the future of criminal smart contracts	Smart Contract
S19	Khaqqi, K. N., Sikorski,J. J., Hadinoto, K., & Kraft, M. [26]	2018	Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application	Blockchain-based Reputation Systems
S20	Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. [27]	2016	Hawk: The blockchain model of cryptography and privacy-preserving smart contracts	Smart Contract
S21	Lauslahti, K., Mattila, J., & Seppala, T. [28]	2017	Smart contracts–How will blockchain technology affect contractual practices?	Smart Contract
S22	Magazzeni, D., McBurney, P., & Nash, W. [29]	2017	Validation and verification of smart contracts: A research agenda	Smart Contract
S23	Marino, B., & Juels, A. [30]	2016	Setting standards for altering and undoing smart contracts	Smart Contract
S24	Schaub, A., Bazin, R., Hasan, O., & Brunie, L. [31]	2016	A trustless privacy-preserving reputation system	Blockchain-based Reputation Systems
S25	Tadelis, S. [33]	2016	Reputation and feedback systems in online platform markets	Trust and Reputation Systems
S26	Vandervort, D. [35]	2014	Challenges and opportunities associated with a bitcoin-based transaction rating system	Trust and Reputation Systems
S27	Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami [36]	2016	Blockchain contract: securing a blockchain applied to smart contracts	Blockchain-based Reputation Systems
S28	Wohrer, M., & Zdun [37]	2018	Smart contracts: security patterns in the Ethereum ecosystem and solidity	Smart Contract
S29	Wood, G. [38]	2014	Ethereum: A secure decentralized generalized transaction ledger	Blockchain-based Reputation Systems

S30	Zyskind, G., & Nathan, O. [40]	2015	Decentralizing privacy: using blockchain to protect personal data	Trust and Reputation Systems

Table 2.3. Listing of papers identified after the shortlisting in the SLR process

Trust and reputation systems on the Internet are motivated by several things. One of these motivations is to find usable substitutes for traditional signs of trust and reputation. Another motivation is to leverage the Internet infrastructure to develop efficient platforms or systems to collect ratings for service providers and derive reputation values to provide fundamental decision support systems to improve the quality of online platforms (Monir et al., 2013). There are several situations where some users are known for giving poor feedback to service providers. This is inimical to the reputation scores of service providers. Reputation systems should have an algorithm that can be used to model the trust values of a service provider. To achieve this, Resnick & Zeckhauser (2015) considered the following three properties that should be included in all reputation systems.

a) All entities in such reputation systems must be long-lived ensuring that in each transaction, there is an expected transaction by those entities in the future

b) All ratings for the present interactions should be distributed, and

c) The ratings for past interactions must be used as the decision-making basis for present-day transactions.

Table 2.4 presents a summary of the approaches that fall under the trust and reputation category and their commitment to the requirements of a smart contract. The requirements smart contracts in reputation systems is outlined in Section 2.

Table 2.4. Descriptions, issues and limitations of existing studies on trust and reputation systems

Study	Description of the Study	Issues/Limitations	Requirements of a Smart Contract				
			Req 1	**Req 2**	**Req 3**	**Req 4**	**Req 5**
S3	Developed a reputation algorithm called net flow convergence and a decentralized system that calculates reputation based on underlying network structure and allows users to look up and record the histories of transaction outcomes. The network inflow or outflow is the sum of complete set of edges, along with edge weight.	The system can only be used by researchers and programmers. For the theoretical utility of the system to be seen, it needs to reach widespread use. Not accurate enough to verify real-world reputation; a future system could use web crawlers to link nodes in the transactional network	√	√	X	X	√
S25	Describes how reputation helps facilitate trust and trade and explores some of the problems of bias in feedback and reputation systems.	Various solutions provided were not implemented in real-world scalability	X	X	√	√	X
S26	Considers three different models by which a reputation/rating system could be implemented in conjunction with Bitcoin transactions and considered the pros and cons of each. The paper found that each model faces challenges on both technological and social fronts. The rating system models examined include site-based systems, coin-based systems and wallet-based systems.	Some questions were not answered in the study. For example, can a site-based rating system interact with external wallets?	√	√	X	X	X

S30	Describes a decentralized personal data management system that allows users to take charge and control their own data.	The analysis only paid attention to storing pointers to encrypted data. Even though the approach is appropriate for random and storage, it is not effective or efficient for processing data.	X	X	X	X	X

2.4.3 Blockchain-based Reputation Systems

Blockchain, a technology on which Bitcoin has been implemented (Xu et al. 2017), can be used to address the issues surrounding reputation or review fraud. According to English et al. (2016), blockchain is a database that is transactional and globally shared, which is like the BitTorrent. The blockchain database can be accessed by all participants in the network. For those functions that need auditability, provenance and trusted computing, blockchain technology can be efficiently used (Zyskind & Nathan, 2015). The blockchain system can be applied to various domains. This technology utilizes a decentralized ledger (Khaqqi et al. 2018). As all transactions must be publicly broadcasted and be permanent, it can provide various types of services, such as product or service delivery verification in the supply chain industry, educational qualification verification in the education industry, money transfer security in the financial industry, and payment chargeback risk mitigation in e-commerce (Khan 2015). Another important application area for blockchain systems is financial fraud detection.

To facilitate business decision making, a variety of decision support systems (DSS) have been built in several domains or sectors. Given the information provided by users, a decision is made based on the decision-making model which may have built-in rules. Such systems significantly improve the effectiveness and efficiency of decision making, although they are vulnerable to manipulated (or fraudulent) input information, such as loan fraud (Zyskind & Nathan, 2015). For example, a decision on a loan application can be generated based on inputs of customers' personal information. When a user intends to apply for a loan through an online application system, he/she may falsify some of their personal financial information, such as a a fake repayment history, thus increasing the possibility of acceptance. Consequently, financial institutions may suffer tremendous losses due to loan fraud. Blockchain systems can keep historical transactions records that form input to DSS. The applicants cannot falsify information to obtain a favorable decision. Of all the application areas, we focus on the applications on rating fraud detection in the subsequent section.

In addition to the issues discussed above, reputation systems can also be treated as digital currency using blockchain. Through this, the reputation value of an entity will not be glued to a platform, instead, it will be platform agnostic. That is, a service provider has the opportunity to transfer reputation from one platform to another thereby ensuring that the reputation they have earned is not lost. Also, there will be an opportunity for reputation exchange and the bootstrapping of new users.

According to Coleman (2016), the seller is likely to promote his/her own product by encouraging people who provide fraudulent ratings to complete real transactions. These people may be offered free or significantly discounted products to solicit a positive review. This phenomenon has already been noted by Amazon.com. Amazon has removed the "verified purchase" badges from reviews associated with discounted transactions (Coleman, 2016). Furthermore, sellers can allow customers to first pay the full amount, submit ratings, and pay them back in other ways. Although transaction records are incorruptible in the blockchain-based reputation systems, fraudulent raters in such false "real transactions" are not detected. Cai and Zhu (2016) noted that although reputation systems are designed to serve as a mechanism that will reduce the risks related to online shopping, it is vulnerable to rating fraud. This fraud includes a situation where some raters will input unfairly low or high ratings into the system just to demote their competitors or promote their own products. They then explored the limitations of blockchain technology in subjective fraud and its effectiveness in objective fraud and concluded that blockchain-based reputation systems are efficient when they are deployed to prevent objective fraud, such as a loan application where the fraudulent information is fact-based. However, they noted that the effectiveness of blockchain-based reputation systems is limited in subjective information fraud where any ground-truth cannot be strongly confirmed.

Schaub *et al.* (2016) proposed how to utilize digital signatures to design reputation systems that can protect users' privacy. In a similar manner, Soska *et al.* (2016) proposed a system "Beaver", which protects users' privacy, while being resistant against Sybil attacks by charging fees. Dennis et al. (2016) designed reputation systems with underlying blockchain technology. These systems generate and broadcast a binary P2P rating on receiving the correct file. Table 2.5 presents a summary of the approaches that come under the blockchain category and their commitment to the requirements of a smart contract

Table 2.5. Descriptions and limitations of studies on blockchain-based reputation systems

Stud y	**Description**	**Issues/Limitations**	**Requirements of a Smart Contract**				
			Req 1	**Req 2**	**Req 3**	**Req 4**	**Req 5**
S5	Developed Bitconch chain, a new distributed web protocol using an innovative proof of reputation consensus algorithm. To build reputation on blockchain, the Bitconch chain mathematically models time, contribution activities and social network. Technologies used in achieving decentralization and scalability include Zero Knowledge-Proof, post-quantum encryption algorithm, directed acyclic graph etc.	Inability to bootstrap new users. The proposed model does not solve the problem faced by service providers who want to transfer reputation.	√	√	X	X	√
S6	This study explores rating fraud by differentiating subjective fraud from objective fraud. Then, it discusses the effectiveness of blockchain technology in objective fraud and its limitations in subjective fraud, especially rating fraud. The paper also carried out a systematic analysis of a blockchain-based reputation system in both objective fraud and subjective fraud.	Ballot stuffing, whitewashing attacks, and camouflage attacks, inability to bootstrap new users, cannot carry out context-based trust assessment and no predictive analytical approach to predict future trust value.	X	X	√	√	X
S7	This paper shows how a decentralized and distributed feedback management system can be built on top of the bitcoin blockchain. The primary objective of the paper is to avoid giving the control (centralization) of the feedback management system to internet firms alone.	Fake identities, not collusion resistant, Sybil attacks, inability to bootstrap new users, cannot carry out context-based trust assessment and no predictive analytical approach to predict future trust value.	√	X	X	X	X

S10	Analyzes how decentralization improves consensus effectiveness, and how the quintessential features of blockchain reshape industrial organizations and the landscape of competition.	The paper did not design a robust consensus protocol, nor did it provide the right incentives for maintaining consensus on specific blockchains.	X	√	√	X	X
S19	Incorporates blockchain technology to address the management of the emission trading scheme (ETS) and fraud issues and utilizes a reputation system in a new approach to improve ETS efficacy. A multi-criteria analysis was carried out to evaluate the proposed scheme in comparison to conventional ETS model. The result of the analysis showed that the proposed model is more efficient.	Attacks against reputation systems, like bad-mouthing, ballot stuffing etc.	√	X	√	X	√
S24	This paper presents a blockchain-based trustless reputation system and analyzes its correctness and the security guarantees it promises; and eliminates the need for users to trust fellow users or any third party. The paper used a blinded token exchange algorithm to verify that a customer was involved in a transaction before allowing such customer to rate the service provider.	Attacks against reputation systems, like bad-mouthing, ballot stuffing, Sybil attacks, and whitewashing. Others are inability to bootstrap new users, and no predictive analytical approach to predict future trust value.	√	√	X	X	√
S27	This paper proposes a new mechanism for securing a blockchain applied to contract management such as digital rights management. The study designed a new protocol that can be used to record a trail of consensus on the blockchain. A transaction is used as evidence of contractor consent in this protocol.	The mechanism was not implemented on actual cryptocurrency.	X	√	X	√	X

S29	The paper discusses the design of Ethereum, its implementation issues, the opportunities it provides and future problems.	Scalability remains an ongoing concern. With a generalized state transition function, it becomes difficult to partition and parallelize transactions to apply the divide-and-conquer strategy.	X	√	X	X	X

2.4.4 Smart Contracts for Reputation Systems

According to Lauslahti *et al.* (2017), smart contracts are algorithmic, self-enforcing, and self-executing computer programs. These smart contracts eliminate the necessity of a trusted third party in transactions by allowing untrusted parties to manifest contract terms (Wohrer & Zdun 2018).

Cong & He (2018) explained how decentralized ledger technologies such as blockchains can simplify the creation of smart contracts. Furthermore, using smart contracts, the users are able to codify precisely their trust relations and agreements. These will be automatically executed by platforms like Ethereum after deployment. In fact, these smart contracts can facilitate economic activities by effectively providing services that are offered traditionally by intermediaries (e.g., notaries, bank, and courts) and trusted third parties. In order to secure blockchains, Watanabe et al. (2016) proposed a new mechanism based on smart contracts that can be implemented in contract management. A new consensus method was used in this mechanism that used credibility scores to create a hybrid blockchain. In this way, it is possible to thwart attackers from monopolizing resources and thereby ensure the security of the blockchains (Everts & Muller 2018).

Cong & He (2018) also opined that with a large range of economic outcomes, smart contracts and blockchain can sustain market equilibria. Bigi et al. (2015) noted that protocols based on decentralized smart contracts can facilitate interaction among independent players without the interference of any coercing authority. The authors believed that these smart contracts could even be used in various applications in the future as a potentially enabling technology. This makes it essential to validate this technology.

Hence, the authors combined formal models and game theory for validating such bitcoin-based systems based on smart contracts. Christidis & Devetsikiotis (2016) examined the usefulness of using smart contracts and blockchains in Internet-of-Things (IoT) domain. The authors found that the powerful blockchain-IoT combination can cause substantial transformations across multiple industries. This can result in the creation of new distributed applications and business models. Kosba et al. (2016) presented a decentralized smart contract system called ‘Hawk’. This new system can guarantee transactional privacy by ensuring that the system does not store information about financial transactions on the blockchain. In this way, it can help different parties to transact safely as the transactions are not exposed.

In order to coordinate interactions between independent entities, like humans, agents, etc., Frantz & Nowostawski (2016) discussed the potential of blockchain technology. Blockchain technology often faces the challenge of ensuring the broader use of the correct and unambiguous specification of smart contracts (Toneli et al., 2018). The authors introduced a process to automate institutional constructs into rules that are contractual and machine-readable. Using this process, people with different levels of technical background will be able to easily generate smart contracts and use blockchain technology as an efficient coordination tool. Buterin (2014) and Wood (2014) discussed a smart contract platform called Ethereum. The authors highlighted the issues regarding its design, implementation, and also the opportunities it provides. Some of the common security patterns on Ethereum were discussed by Wohrer & Zdun (2018). These included Checks-Effects-Interaction, Emergency Stop (Circuit Breaker), Speed Bump, Rate Limit, Mutex, and Balance Limit. Developers can address security problems like harmful call-backs and uncontrollably high financial risks by applying these patterns. A Smart Contract-based Public Key Infrastructure (SCPKI) was proposed by Al-Bassam (2017). The SCPKI uses a transparent and decentralized design using a smart contract and a web-of-trust model on the Ethereum blockchain. The author argued that this PKI system is capable of detecting rogue certificates when they are published. In this way, it would be possible to ensure secure communication on the Internet.

Marino & Juels (2016) defined a new set of standards for undoing and altering smart contracts as traditional tools often fail in this regard. The authors developed a new set of standards and tried to prove their worth after applying to Ethereum, a popular smart contract platform. In the end, they succeeded in their approach and proved the value of such a framework. Idelberger et al. (2016) found the technical and legal advantages of using logic-based languages instead of procedural languages for programming smart contracts in the blockchain system. The authors concluded that smart contracts based on logic-based languages are easier for developers to work with. Furthermore, logic-based smart contracts can also reduce the risk of errors in the implementation. In addition, it is possible to ease the validation process by using this particular type of smart contract.

Lauslahti et al. (2017) analyzed smart contracts from the context of Finnish contract law and digital platforms. The authors found that smart contracts can be applied in a variety of ways, with different circumstances and goals. They concluded that smart contracts could generate legally binding obligations and rights to their parties, at least in some cases. Gatteschi et al. (2018) tried to help insurers decide whether to adopt blockchain technology or not by clarifying the concepts of blockchain and its advantages and disadvantages. The authors argued that blockchains and smart contracts can improve customer experience and reduce operating costs, minimize the overhead related to the verification of new customers and manual data entry, compute risk assessments and prevent frauds. Table 2.6 presents a summary of the approaches that come in the smart contract category and their commitment to the requirements needed for the facilitation of a smart contract.

Table 2.6. Descriptions, issues and limitations of existing studies on smart contracts

Study	**Description**	**Issues/Limitations**	**Requirements of a Smart Contract**				
			Req 1	**Req 2**	**Req 3**	**Req 4**	**Req 5**
S1	Smart contract-based public key infrastructure (SCPKI) is a substitute PKI system built on a transparent and decentralized design using a smart contract on the Ethereum blockchain and a web-of-trust model, to make it easily possible for rogue or fake certificates to be detected when they are published. The developed web of trust/confidence model is decentralized and highly fault tolerant.	Issues related to privacy. The system is only appropriate for publishing attributes that the user wishes to make public. It is not suitable for publishing more private identity attributes such as a personal address. Issues related to adaptability. The design of the system is such that all parties referenced by the system must already use the system.	X	X	√	√	X
S2	Demonstrated a decentralized app (DAPP) for the distribution of everyday objects based on a smart contract on the Ethereum blockchain. This contract allows users to register and rent devices without involving any trusted third party (TTP), the revelation of any personal information or prior signup to the service.	The paper identified significant fees for users and overbearing terms and conditions as part of the problems of the sharing economy but did not provide any solution for them.	X	X	X	X	X

S4	Created Ethereum, an alternative protocol for building decentralized applications. It is a blockchain with an in-built language that allows users to write smart contracts and create their own transaction formats.	Rudimentary Ethereum virtual machine implementation, primitive architecture and underdeveloped language. Ethereum is less optimized for one specific use case.	X	X	√	√	X
S8	The paper found out that smart contracts that are under-optimized cost more gas than normal thereby overcharging users. To address this, the study developed GASPER, a new tool to automatically locate costly gas patterns by analyzing the bytecodes of smart contracts.	The compilers need to be improved to produce gas-efficient bytecode.	X	√	X	X	X
S9	Reviewed how a blockchain-IoT combination can facilitate the sharing of resources and services thereby leading to the creation of a marketplace for devices and users. It also discussed how the combination allows automation of many time-consuming workflows in a cryptographically verifiable manner. The issues and challenges that need to be addressed before the deployment a blockchain-IoT system were also identified.	The paper did not provide specific solutions to most of the challenges identified.	X	X	X	√	X

S12	Discussed the common pitfalls in designing safe and secure smart contracts and how to fix them. This was based on insights gathered through years of pedagogical efforts on smart contracts.	Did not include programmers' learning adversarial thinking through attacking and amending their own code	X	X	√	√	X
S13	Identified what criteria Ethereum needs to fulfill to replace paper contracts. It discussed the privacy and security of the blockchain. The paper noted that it is not recommended to place paper contracts on Ethereum blockchain because of the huge privacy lapses and the variety of contract clauses.	The study did not provide any solution to privacy setbacks and the security issues identified.	X	X	√	X	X
S14	Proposed a modeling approach that supports the semi-automated translation of human-readable contract representations into computational equivalents in order to enable the codification of laws into verifiable and enforceable computational structures that reside within a public blockchain. The paper identified smart contract components that are obtainable in real-life institutions and proposed a mapping which was executed using a domain specific language.	Failed to explore the possibility of reversal, that is, given a blockchain contract, is it possible for humans or autonomous entities to verify the actual contractual semantics and obligations?	X	√	X	X	√

S15	Presented an overview of potential applications and use cases of blockchain and smart contracts in the insurance sector. Also, the paper provided a more general SWOT analysis of blockchain.	Study is restricted to the insurance sector alone.	X	√	√	X	X
S16	Provided insights on how to use logic-based smart contracts in combination with the blockchain. network. The paper noted that algorithms for logic approaches need to be efficient in the specific environment they are deployed. This was illustrated using various algorithms from defeasible logic-based frameworks.	While the paper identified procedural language as the most commonly used language to program smart contract, it does not justify why using logic-based languages will be a better alternative.	X	X	X	√	√
S17	Decentralized smart contracts represent the next step in the development of protocols that support the interaction of independent players without the presence of a coercing authority. The paper combined game theory and formal models to tackle the new challenges posed by the validation of decentralized smart contracts. The probabilistic framework adopted allowed to properly model of uncertainty and non-determinism in players' behavior. It also helped in exploiting statistical model checking to validate the smart contract.	The paper did not solve the problem of more complex and repeated games which require smart contracts exhibiting more sophisticated behavior.	X	X	√	√	√

S18	Explored the risk of smart contracts that can cause new criminal issues and demonstrated significantly that Criminal Smart Contracts (CSCs) for the leakage of secrets are efficiently realizable in existing scripting languages such as that in Ethereum.	Discussed only few Criminal Smart Contracts and did not point out potential countermeasures.	X	X	√	X	X
S20	Presented Hawk, a decentralized smart contract system that does not store financial transactions on the blockchain, thus retaining transactional privacy from the public's view.	Not integrated with any reputation system.	X	X	√	X	X
S21	The paper examined how the formation mechanisms of the general principles of contract law can be applied to the new technological framework of smart contracts.	Although this paper has described three examples of smart contracts, in reality, the number of possible applications may be practically infinite.	X	X	X	X	X
S22	This paper explored the issues and research challenges faced in the authentication and confirmation of smart contracts, especially the ones that run over blockchain.	Not integrated with any reputation system.	X	X	√	X	X

S23	This paper developed a new set of principles for changing and undoing smart contracts and thereafter applied these principles to a smart contract in existence.	No predictive analytical approach to predict future trust value.	X	X	X	X	X

2.4.5 Research gaps in the existing studies from the requirement perspective of a smart contract

As presented in Tables 2.4-2.6, researchers have focused on forming and facilitating smart contracts. However, from the perspective of *before* and *after* phases of a smart contract, the existing work has the following drawbacks:

- *Inability to derive the reputation value of service providers based on their performance in earlier smart contracts:* While existing work has focused on integrating reputation values with Bitcoin transactions (S26), they do not focus on using these values for other transactions. This is required in the *before* phase of a smart contract, where a user's need may be in different requirements from what the provider's value is in. To this end, (S3) focused on using the underlying network structure to calculate the reputation value, but not on how it can be interpreted in other contexts. (S30) proposed an approach for using decentralized data but that is for a user's own value and not for service providers. Without such an approach, the existing literature does not allow the service user to determine the reputation value of a provider in the context of forming a smart contract.

- *Inability to deduce the trust value of service provider in a specific context:* Another aspect which is required by a service user in the *before* phase of a smart contract is the ability to deduce the performance of a service provider in a specific context. While (S10) does initial work on investigating the use of decentralization, it does not represent how to use these values to transform it into context-specific requirements. Other researchers such as (S29, S27) proposed technologies for forming smart contracts, while others such as (S14, S8) focus on the technical aspects of forming one, but they focus on the *during* stage and not on the *before* aspects required for forming a smart contract.

- *Reputation system as a digital asset and moving across platforms:* The third requirement to facilitate smart contracts across different platforms is to enable service providers to move their reputation values across different platforms. This can be done when the reputation value is regarded as a digital asset and can be moved across platforms. While (S25) mention how reputation value helps to facilitate trade, it discussed how it can be transferred or moved. Other studies such as (S19, S24, S1, S4, S15) utilize blockchain in a specific domain but do not mention how it can be transferred across others. (S13) mentions privacy and security aspects as the key requirements to be addressed in a smart contract. These are more focused in the *during* stage of the smart contract but fail to mention how these can be achieved in the *before* stage, if such values are unable to be transferred across platforms.

- *Detection of non-compliant behavior and manipulation of reputation ratings:* The monitoring of non-complaint behavior has been studied widely in the literature. Studies such as (S6) investigated rating fraud by differentiating it with subjective fraud and objective fraud. Other approaches such as (S1) used the PKI system to detect fake certificates. (S17, S20) used a game theory-based approach to detect uncompliant behavior and a privacy-based approach respectively in a decentralized environment while (S18) used a secure environment to prevent leakage. However, the existing approaches focus on the *during* part of the smart contract and do not mention how to prevent inflating the existing reputation values on which the analysis in the *before* and *after* phases of the smart contract are built. Without these, the existing approaches cannot ensure that the shown values are indeed correct and free from bias.

- *Mathematical models and algorithms for reputation computation after reputation trading:* Some of the papers examined, like S3, S5, S17, S19 and S24, utilized mathematical models and algorithms to compute reputation values. (S3) developed a reputation algorithm called net flow convergence and a decentralized system that calculates reputation based on underlying network structure. (S5) mathematically modeled contribution activities, time and social network to build a decentralized reputation system. The model was established using a proof of reputation consensus algorithm. (S19) uses an algorithm called priority value (PV) to help buyers sort bids from sellers. (S24) made use of a blinded token exchange algorithm to verify that a customer was involved in a transaction before allowing such customer to rate the service provider. Finally (S17) adopted a probabilistic framework which allowed a proper modelling of uncertainty and non-determinism in players' behavior and it also helped in exploiting statistical model checking to validate the smart contract. However, the existing work only focused on the *before* and *during* phases of smart contracts. None of the papers in the examined or provided a reputation algorithm that can be utilized to compute the remaining reputation values of service providers *after* reputation trading or auctioning.

In the next section, we present our proposed framework to address the gaps.

2.5 Smart contracts-based reputation system service framework

This section discusses the methodological approach that is used to address the gaps identified in the literature. The proposed framework is termed *FarMed Service*, which is a smart contracts-based reputation system service framework that is driven by service-oriented computing (SOA). The framework is divided into two layers as shown in Figure 2.4.

Layer 1 – Blockchain data layer: In this layer, Ethereum-smart contracts verify that the reputation values of service provider/s and buyer/s have not been manipulated. Smart contracts in this layer store all the reputation values of all service provider/s and buyer/s.

Layer 2 – AI layer: This layer computes the analysis from information embedded in the smart contracts. In this layer, once data is acquired, it is passed to different intelligent modules to update the rating value of the service provider or buyer. The different modules that are in this layer are reputation computation, reputation predictive analytic, reputation trading and reputation auction service.

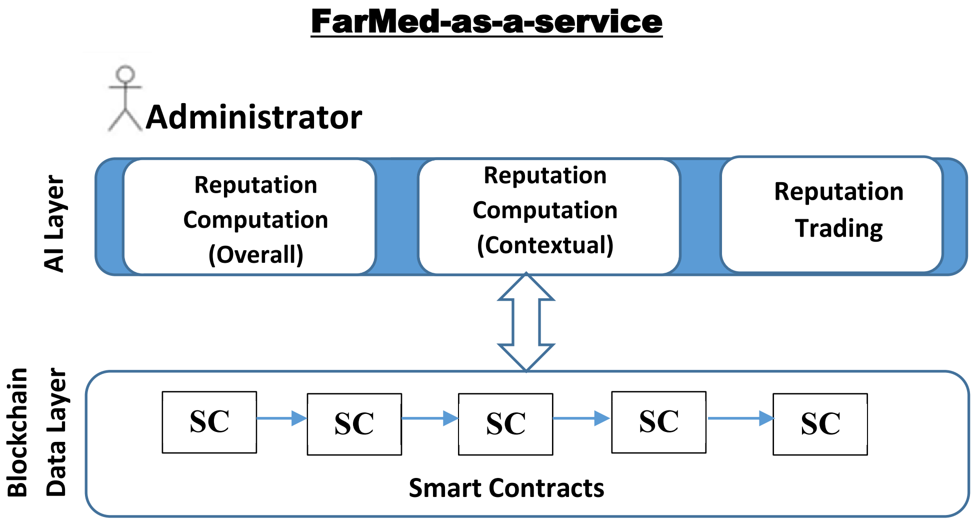


Figure 2.4: Details of the FarMed Service.

Figure 2.5 shows the overview of the *FarMed* framework from the stage of the user visiting the e-market to select an item, to purchasing the item or canceling the purchase intent. There are three phases namely the *marketplace phase*, *smart contract execution phase* and *trust value computation phase* in the FarMed framework. The *marketplace phase* represents the series of steps needed by the client to from a contract. Once a contract is formed, the workflow for the *smart contract execution phase* of the framework as shown in Figure 2.6 is executed. Taking the example of contract *C*, Figure 2.6 shows its initialization and the execution of its logic. X is the condition that a purchase which has not been rated exists. Z is the condition that a rating value has been provided by the service consumer. If any of conditions X and Z are not met, the smart contract execution will automatically fail. For the execution of the contract to be completed, both conditions X and Z must be met. When both these conditions have been met, the operation address Q is executed. This phase fulfills requirement 4 stated in section 2.4. This leads to phase 3, which is the *trust value computation phase*. As shown in Figure 2.5, this phase addresses requirements 1, 2, 3 and 5 mentioned in sections 2.2.1, 2.2.2, 2.2.3 and 2.2.5 respectively and includes the development of intelligent methods to compute overall reputation of service providers, compute reputation of a service provider in a context and allow service-provider reputation transfer. To achieve requirement 1, a five-star algorithm to compute and represent the service provider's reputation will be used in *FarMed*. For requirement 2, AKTiveRank algorithm will be used while for requirement 3, a modified instance of the five-star algorithm will be used. A brief description of how the *FarMed* framework aims to address the key requirements defined in Section 2.2 for the Smart Contract to address is explained in the next sub-sections.

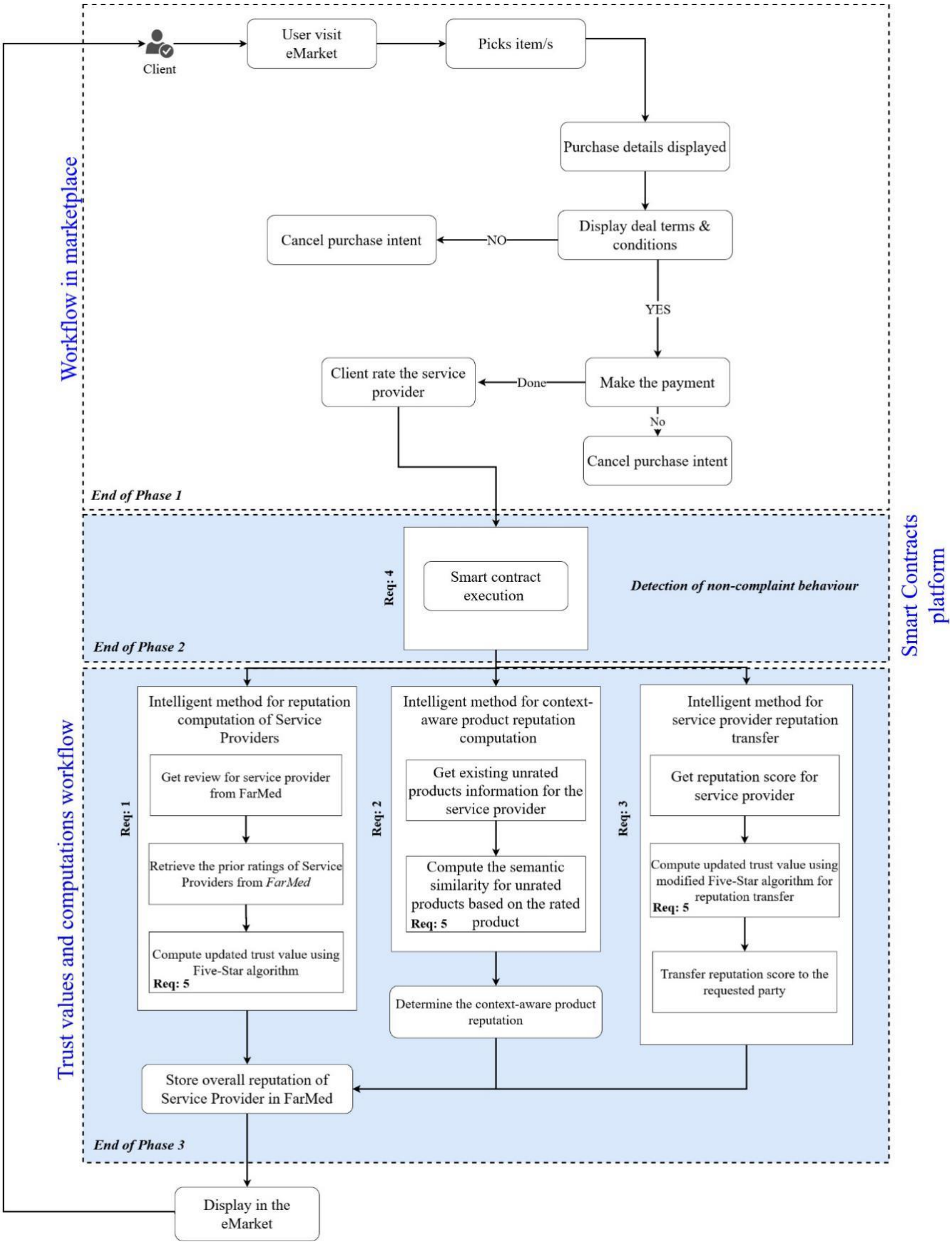

Workflow in marketplace
Client
User visit eMarket
Picks item/s
Purchase details displayed
Display deal terms & conditions
NO
Cancel purchase intent
YES
Make the payment
Done
Client rate the service provider
No
Cancel purchase intent
End of Phase 1
Smart Contracts platform
Req: 4
Smart contract execution
Detection of non-complaint behaviour
End of Phase 2
Trust values and computations workflow
Req: 1
Intelligent method for reputation computation of Service Providers
Get review for service provider from FarMed
Retrieve the prior ratings of Service Providers from FarMed
Compute updated trust value using Five-Star algorithm
Req: 5
Req: 2
Intelligent method for context-aware product reputation computation
Get existing unrated products information for the service provider
Compute the semantic similarity for unrated products based on the rated product
Req: 5
Determine the context-aware product reputation
Req: 3
Intelligent method for service provider reputation transfer
Get reputation score for service provider
Compute updated trust value using modified Five-Star algorithm for reputation transfer
Req: 5
Transfer reputation score to the requested party
Store overall reputation of Service Provider in FarMed
End of Phase 3
Display in the eMarket

Figure 2.5. Sequence of FarMed framework's working

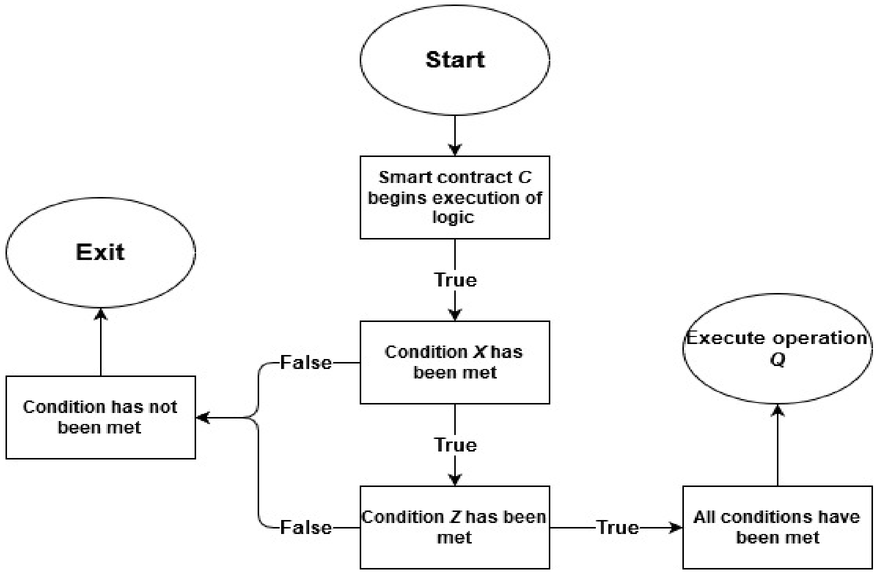

Start
Smart contract *C* begins execution of logic
True
Exit
Condition *X* has been met
False
Execute operation *Q*
Condition has not been met
True
Condition *Z* has been met
False
True
All conditions have been met

Figure 2.6. Sequence of working of the Smart contract execution phase of FarMed framework

2.5.1 Modelling and deducing the overall reputation value of service providers (Req: 1)

There is a need to model the overall reputation value of service providers based on the trust values stored in the smart contracts. To do this, a reputation algorithm will be used. The procedure involved in satisfying this requirement is shown in phase 3 in Figure 2.6 above. It involves getting the prior ratings of the service provider from *FarMed* and computing an updated reputation value which is then stored in *FarMed*.

2.5.2 Modelling and deducing the trust value of service providers within a context (Req: 2)

To model the reputation value of a service provider in a context, *FarMed* in the trust value computation phase develops a service ontology coupled with distance-based approaches. The use of ontologies in constructing a knowledge system is pervasive. Ontology provides common vocabularies for computers and humans to support semantics for knowledge sharing (Fensel *et al.,* 2000). For Req: 2 to be fulfilled, there is a need to compute the semantic similarity for each unrated product based on the rated product. Service ontology and AKTiveRank (Alani and Brewster, 2006) can be used for the computation. After getting the semantic distance, an algorithm can then be applied to model a context-based inference based on whether the semantic distance value meets a particular benchmark or not (depending on the value set by the researcher).

2.5.3 Transferring reputation among service providers and across platforms (Req: 3)

In order to bootstrap new service providers in the reputation-based economy, FarMed in the trust value computation phase will introduce the concept of *Reputation Auction*. The intention of this proposed method is that service providers with high or excess reputation values will auction part of their reputation ratings to other service providers and will receive payment for doing so. This is clearly demonstrated by Req: 3 in the phase 3 of Figure 2.5. The new sellers who have a zero-reputation score can buy from this auction to build their reputation in the market quickly while the offers in the auction will be made by sellers who usually have a high reputation score. The benefit for those who sell full or part of their reputation score is commercial. As long as they are building their reputation score by providing good services and products, their customers will rate them with a high rating. The providers can use this as a new source of income thus making the reputation score an important matter in the market. Figure 2.7 shows an example of reputation scores before and after the auction.

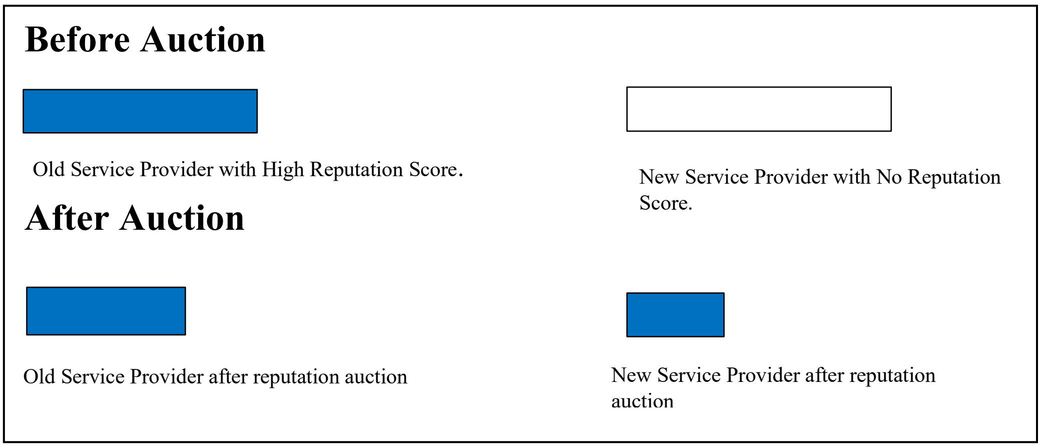

Before Auction
Old Service Provider with High Reputation Score.
New Service Provider with No Reputation Score.
After Auction
Old Service Provider after reputation auction
New Service Provider after reputation auction

Figure 2.7. Example of reputation scores before and after the auction

2.5.4 Detection of non-compliant behavior and manipulation of reputation ratings (Req: 4)

The smart contract needs to have validated methods of detecting and dealing with any reputation fraud. This is the most critical part of the smart contract. To achieve this, the smart contract needs to be evaluated from time to time by deploying the Ethereum Ropsten network. The Metamask plugin can be installed first so that it communicates with nodes on the remote server. After switching to the Ropsten network, a new account can be created, and the solidity compiler is used to deploy the contract. The contract is then tested to detect any non-compliant behavior using the provided interface. Phase 2 of the *FarMed* framework in Figure 2.5 represents the solution to this requirement and more information about it is provided in Figure 2.6.

2.5.5 Mathematical models and algorithms for reputation computation post-reputation trading (Req: 5)

To effectively compute the reputation values of service providers, generally and contextually as well as to enable reputation transfer, different mathematical models and algorithms mentioned earlier will be developed in this requirement. For the overall reputation computation, five-star algorithm will be used, for reputation computation in a specific context, semantic similarity and ontology service will be used, and for reputation transfer, a modified version of the five-star algorithm will be used.

2.6 Conclusion

This systematic literature review provides five key requirements that should be in a smart contract for reputation systems. Such requirements include the ability to derive reputation values and trust values of service providers based on the values present in the smart contracts and blocks respectively. Others include treating the reputation system as a digital asset and the detection of non-compliant behavior. After a thorough review of the existing literature, this chapter proposed *FarMed* Service as a smart contract-based reputation system framework. This framework will address all the key requirements of smart contracts.

This SLR represents the first attempt to address the issues observed in blockchain-based reputation systems using smart contracts. It also represents the only approach of its type that proposes innovative artificial intelligence (AI)-based algorithmics on top of blockchain to carry out reliable trust and reputation computations, deduce reputation values of service providers and carry out context-based trust assessments for service providers.

The next chapter will present all the gaps identified based on the literature review carried out in this chapter. Also, the next chapter will provide definitions to key terms used in this study and present all the research questions and objectives

Chapter 3 : Research Questions and Objectives

3.1 Introduction

This chapter identifies the research questions based on the comprehensive and systematic literature review provided in the previous chapters. These questions help in formulating the research objectives which are presented in this chapter as well.

3.2 Gaps in the Literature

From a thorough review of the existing literature documented in Chapter 2, we have identified the following five gaps:

1. There is no existing method for deriving the reputation value of service providers based on the values (or ratings) present in the smart contracts. In addition, there is no means of deducing the trust value of a service provider based on the trust values in the blocks.
2. There is no proposed framework by which the reputation is regarded as a digital asset and can be moved across platforms or from one service provider to another.
3. There are no methods to intelligently infer the reputation value of a service provider in a specific context based on existing trust values in various contexts.
4. There are no validated methods to detect and deal with reputation fraud in smart contracts such as bad mouthing, ballot stuffing, positive and negative discrimination, false feedback, value imbalance problem.
5. There is no mechanism by which new service providers can be bootstrapped into the reputation-based economy.

3.3 Key Definitions

In this section, we present the definitions of terms that will be used in this thesis.

Blockchain: The blockchain is a chain of blocks that are time-stamped and joined together using cryptographic hashes. It is organized in a peer-to-peer network consisting of nodes. Transactions belonging to many users can be found in a block while the block itself is publicly available to all the users of the network. In addition, each block contains the transaction data and the hash of the previous block, thus creating an immutable and secure, append-only chain. As each new block is added, the chain continues to increase in length (Yli-Huumo et al., 2016).

Smart Contracts: A smart contract is a computer protocol that has the primary objective of digitally facilitating, verifying, or enforcing the negotiation or performance of a contract. With smart contracts, credible transactions can be carried out without the third parties. The smart contracts are more transparent, offer high commercial efficiency and provide anonymity in transactions (Mark, 2017).

Reputation systems: This is a system that facilitates trust between entities thereby increasing the effectiveness and efficiency of online services and marketplaces. Reliance on reputation system in the online marketplace is prominent because most users don't have the direct and first-hand experience with other users. A reputation system works by enabling the collection, distribution, and aggregation of data about an entity, that can, in turn, be used to depict and predict that entity's future actions and services (Hendrikx et al., 2014).

Ethereum: According to Jani (2018), Ethereum is a peer-to-peer network of virtual machines that allows developers to run distributed applications (Dapps). Ethereum makes use of its own decentralized public blockchain to cryptographically store, protect and execute contracts. This distributed network of computers comfortably provides the reliability, security, and computing power necessary for carrying out designed arrangements.

E-commerce and Online marketplace: E-commerce means electronic commerce. E-commerce and online marketplace involve the use of digital information processing technology and electronic communications in business transactions to create, redefine and transform relationships for value creation between organizations, individuals or both (Nisha and Sangeeta, 2012). The main types of e-commerce include business-to-government (B2G); business-to-business (B2B); consumer-to-consumer (C2C); and business to- consumer (B2C).

Service-Oriented Computing: Service-Oriented Computing (SOC) involves the use of services as fundamental elements for developing applications. Services are platform-agnostic and self-describing computational elements that support low-cost and rapid composition of distributed applications. Service-Oriented Computing allows organizations to expose their core competencies programmatically over the internet using standard protocols and languages and to be implemented through a self-describing interface that is based on open standards (Papazoglou, 2003)**.**

Reputation Score (Trust Value): Trust value is a factor that helps to predict the behavior of an entity in the marketplace. It shows the level of vulnerability of relying on another party in the online marketplace.

3.4 Research Questions

From the systematic literature review documented in Chapter 2, and the shortcomings outlined in Section 2.5, it is clear that there are several gaps in the existing literature on smart contracts in reputation systems. Based on these gaps, the main research question is identified as follows:

How can a reputation system based on smart contracts be used for accurate reputation modelling in service-based e-commerce?

The main research question can then be subdivided into five sub-questions:

RQ 1: How to develop a framework for integrating *reputation systems*, *smart contracts,* and *service-oriented computing*?

RQ 2: How to develop intelligent methods to compute the current reputation values of the service provider from the values in smart contracts?

RQ 3: How to develop intelligent methods to both model and infer the trust value of a service provider in a specific context? Inference in a given context will be based on existing trust values in related contexts.

RQ 4: How to develop intelligent and reliable methods for transferring the reputation from one service provider to another?

RQ 5: How to validate and verify the proposed methods using proof-concept simulation framework/s?

3.5 Research Objectives

Based on the above question and sub-questions, the research objectives of this research are as follows:

Objective 1: To develop an intelligent framework that executes an Ethereum smart contact-based reputation system and provides this "as-a-service" to other consumers.

> *This objective will be achieved by developing the 'FarMed service' that is built to integrate reputation systems, smart contracts, and service-oriented computing. Ethereum is chosen because it is the most resilient and hack resistant blockchain now, thereby giving end users enough confidence on the trust value of the service provider.*

Objective 2: To develop an intelligent and reliable method to compute the current trust value of service provider/s based on trust values in the smart contracts.

> *This objective will be addressed by building the system on FarMed and building intelligence in FarMed through aggregating rating of a service provider stored across multiple blocks. An algorithm to derive the trust of the service provider based on the multiple ratings stored in the blocks of FarMed will be developed.*

Objective 3: To develop an intelligent method that can be used to both model and infer the trust value of a service provider in a specific context. Inference in a given context will be based on existing trust values in related contexts.

> *An ontology-based method will be used to model the context-specific nature of trust and reputation. Our developed solution involves the use of an existing ontology and that of existing measures to find the distance between two ontology concepts to carry out inferences about trust values in different context. The accuracy of the ontology-based method is computed by comparing the predicted context-aware value with the actual context-aware value.*

Objective 4: To develop an intelligent and reliable method for transferring the reputation from one service provider to another.

> *To achieve this objective, intelligent and reliable Blockchain-based protocols for transferring the reputation value from one provider to another will be developed*

in this research. The transfer protocol works with the buyer first making a request for purchase from the seller. Once the seller accepts the request, an agreement is made on the payback schedule and the same is stored in a smart contract. Then, the final reputation for each party is computed, stored in blockchain, and published in the marketplace.

Objective 5: To develop a software prototype, evaluate and test the effectiveness of the proposed methods in objective 1 to objective 4 for their accuracy.

The working of the proposed smart contract-based reputation service will be evaluated using Ethereum Ropsten network. This will be done by experimenting its effectiveness for addressing research question 1 to research question 4 along certain benchmarks.

3.6 Conclusion

This chapter presented the specific research questions which will be answered in this study. It also includes the research objectives that will be pursued and achieved through a systematic research approach. In addition, the chapter provides definitions to the key terms in this research.

In the next chapter, the research methodology along with an overview of solutions will be presented. This methodology will explain how the objectives will be achieved.

Chapter 4 : Research Methodology and Solution Overview

4.1 Introduction

This chapter discusses the methodological approach that will be used to address the gaps identified in the literature review. The overview of the proposed solution and how the research question will be solved are presented in this chapter. This chapter is organized as follows: (a) Section 4.2 presents the new key words used in this research. (b) Section 4.3 outlines the selected research methodology and justifies its solutions. (c) Section 4.2 - 4.8 presents an overview of research question 1 to research question 4 respectively. (d) Section 4.9 concludes the chapter.

4.2 Keys Definitions

FarMed: We use the term *FarMed* to the service built to integrate smart contracts with blockchain-based reputation system in order to achieve the prime objective of this research.

Reputation Auction Service (RAS): This is all about reputation trading. It is built to allow the bootstrapping of new members into the system. RAS allows service providers with high reputation value to sell part of it to new service providers who have not acquired any reputation on the platform. We define RAS as the approach by which reputation may be traded as a digital asset to bootstrap new member into the solution.

Service Provider: We define: A *service provider* as an agent who provides a given service that has financial value or economic value to the service requestor.

Service Consumer: We define: A *service requestor* as an agent who has requested a given service, from another agent, that has financial value or economic value.

Context: We define: A *context* as a significant aspect or property of trust that is usually not taken into account when defining and considering trust.

4.3 Selected Research Methodology

To achieve my research objectives, the Design Science Research approach (Ken *et al.,* 2009) is the model of choice. This is because it provides a methodological framework by coming up with an initial prototype, that is tested (by various stakeholders) to determine if it addresses the originally identified objectives. In the case that they have not been achieved, then it becomes necessary to go back to the research and development process and repeating and reiterating this process until the objectives have been achieved. Another reason for selecting the Design Science Research approach is that I will be producing a working artefact as a proof-of-concept and will be making it available as-a-service.

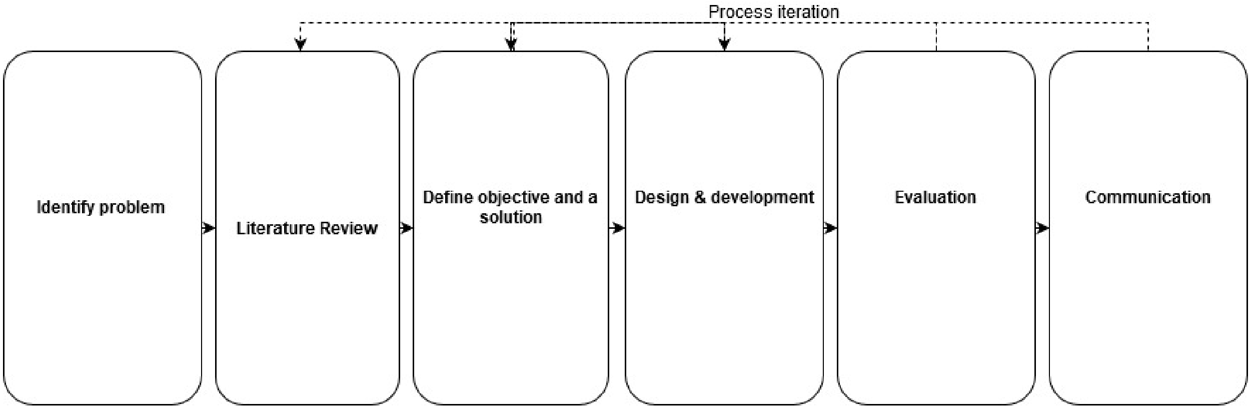


Figure 4.1. Design science research approach (Ken et. al., 2009)

The proposed solution is divided into five phases using design science research approach as shown in *Figure 4.1* above. The phases are explained as follows:

Phase 1: Problem Identification and Literature review phase: During this step, I reviewed the existing work to identify the gaps in the existing literature (both in practice and in research) for countering the tampering and changing of values in the reputation system. Reputation systems are subject to a range of attacks such as bad-mouthing attacks, etc. (Schaub *et al.,* 2016). Also, in this step, I identified and outlined the gaps in the existing state-of-the-art by carrying out critical evaluation of these works in relation to the research problem and moving the research forward. This process is documented in chapter two of this thesis.

Phase 2: Define the research objective and a solution phase: The primary objective of this research is to develop a reliable approach for making the reviews within the reputation system tamper-proof. To achieve this goal, I propose a blockchain based approach for the reputation systems. The solution that I propose is modular in the sense that it can be offered as-a-service to other reputation systems. This research is focused on developing approaches based on Artificial Intelligence to determine trust values of service providers based on the information present in the blockchain. This approach will enable bootstrapping of new service providers and transfer reputation from one provider to another. This phase is captured in chapters 3 and 4 of this thesis.

Phase 3: Design and development phase: During this phase, a software artefact will be built using both Artificial Intelligence methods and Blockchain as proof-of-concept. During this phase the solutions to research question 2, research question 3 and research question 4 will be developed and built into the artefact. This phase is dealt with and documented in chapters 5, 6, 7, and 8.

Phase 4: Evaluation and testing phase: During this phase, the developed software artefact will be evaluated using a number of metrics. Furthermore, the performance of the proposed blockchain based approach will be benchmarked against the current reputation systems using a number of metrics. This will be done as part of Research Question 5 in this thesis. This phase is captured in chapter 8.

Phase 5: Communication phase: This phase is for scholarly publication in international peer-reviewed journals and conferences. This phase is carried out continually.

Phase 6: Process iteration**:** The Design & Development phase as well as the Evaluation phase are iteratively performed during the research work based on the obtained results. The primary reason for this iteration, which is a flow from partial completion of the research back to objective definition phase, is to allow for a deductive cognitive process. That is, as the solution is being developed and evaluated, new premises about the artifact and its environment become obvious.

4.4 Solution Overview

This section discusses the overview of the "FarMed" service that is built to integrate reputation systems, smart contracts, and service-oriented computing.

4.4.1 Architecture of the FarMed

Smart contracts-based reputation system service framework driven by service-oriented computing (SOA) are divided into three layers as shown below in *Figure 4.2*.

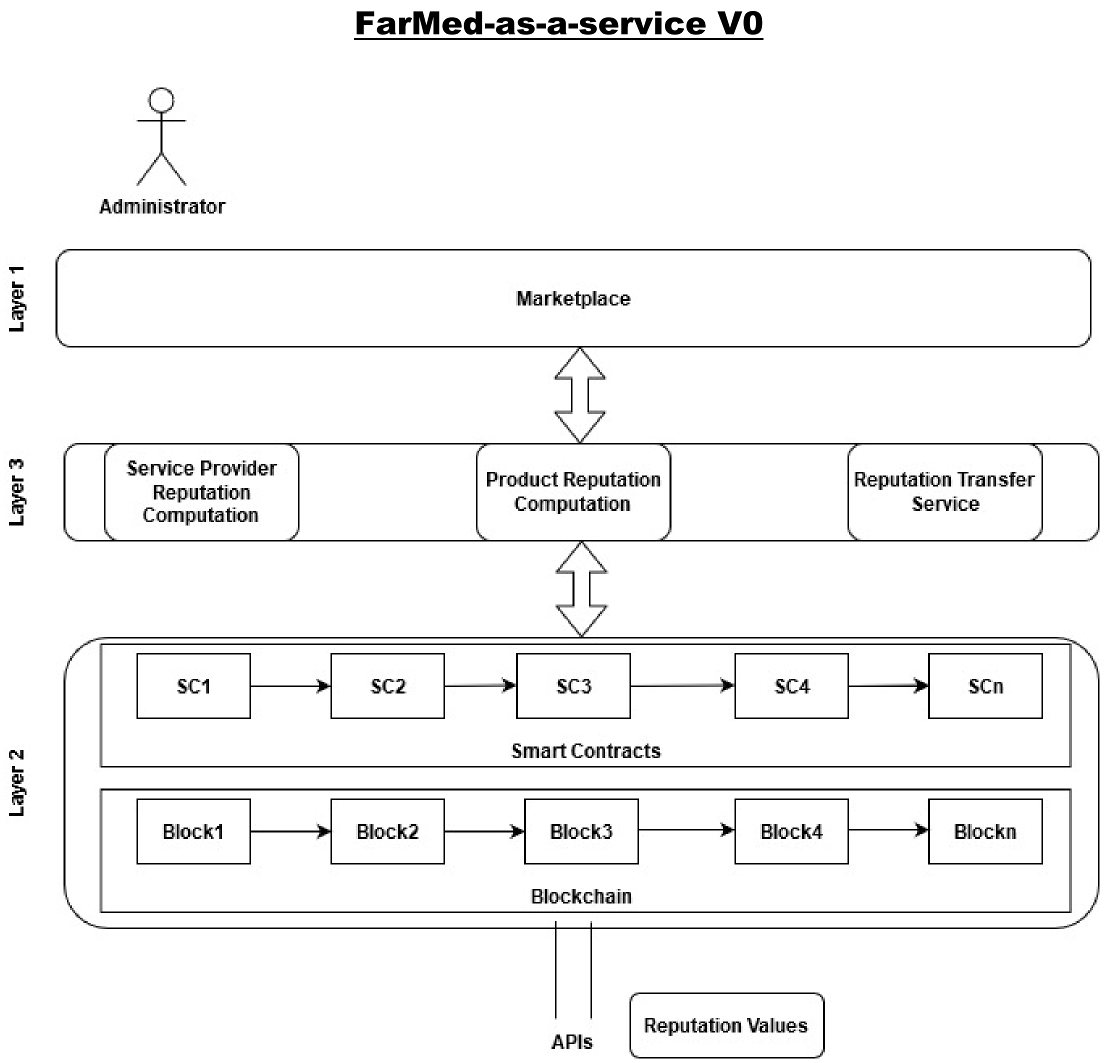


Figure 4.2. Architecture of FarMed Service

Layer 1: This is the electronic market level where the required data to run *FarMed* service comes from such as service provider's details, their ratings by the service consumers, and the purchase details.

Layer 2: This is where the Ethereum-smart contracts verify that the reputation values of service provider/s and service consumer/s are not manipulated. Smart contracts in this layer will store all the reputation values of all service provider/s.

Layer 3: This is for computations on top of information embedded in the smart contracts. In this layer, all data will be acquired from layer two only then passed on to different intelligent modules such as reputation algorithms in order to update the rating value of the service provider.

The overview of the entire working of *FarMed* is shown in Figure 4.3. In Section 4.5 to Section 4.8, we present an overview of the solution of each of the objectives in this thesis.

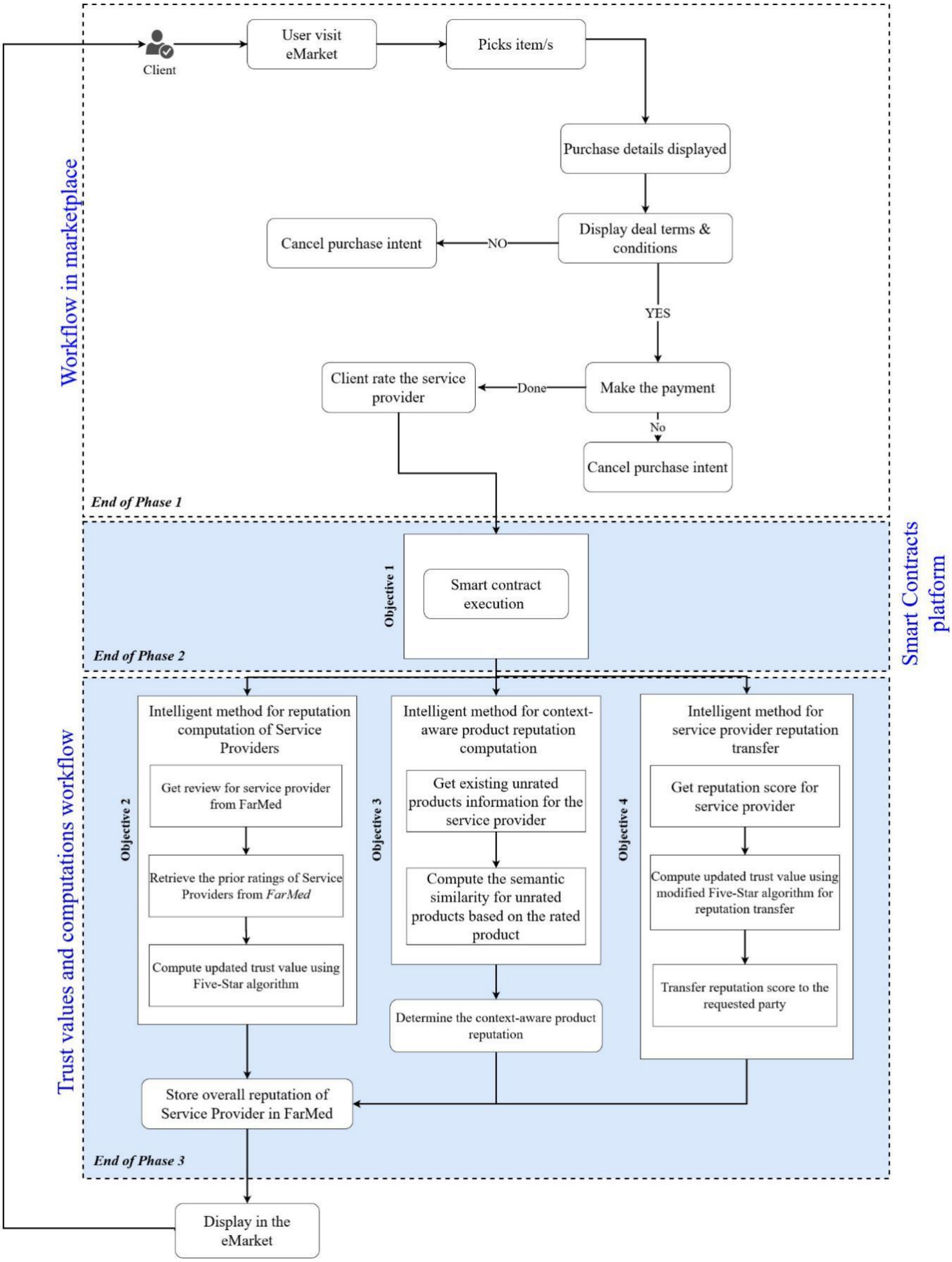

Client
User visit eMarket
Picks item/s
Purchase details displayed
Display deal terms & conditions
NO
Cancel purchase intent
YES
Make the payment
Done
Client rate the service provider
No
Cancel purchase intent
Workflow in marketplace
End of Phase 1
Objective 1
Smart contract execution
End of Phase 2
Smart Contracts platform
Trust values and computations workflow
Objective 2
Intelligent method for reputation computation of Service Providers
Get review for service provider from FarMed
Retrieve the prior ratings of Service Providers from FarMed
Compute updated trust value using Five-Star algorithm
Objective 3
Intelligent method for context-aware product reputation computation
Get existing unrated products information for the service provider
Compute the semantic similarity for unrated products based on the rated product
Determine the context-aware product reputation
Objective 4
Intelligent method for service provider reputation transfer
Get reputation score for service provider
Compute updated trust value using modified Five-Star algorithm for reputation transfer
Transfer reputation score to the requested party
Store overall reputation of Service Provider in FarMed
End of Phase 3
Display in the eMarket

Figure 4.3. Overview of the Smart contract-based reputation system service framework

4.5 Overview of solution for RQ2

When a service consumer rates a service provider, the rating value is stored on the blockchain. The service consumer carries this out by appending signature using the MetaMask Ethereum network. For this research question, the Ropstein Ethereum network is used. The network validates the service provider's rating through a signature and adds the rating to the network. The information stored in the network includes the raw rating value, the service provider's address and the service consumer's address.

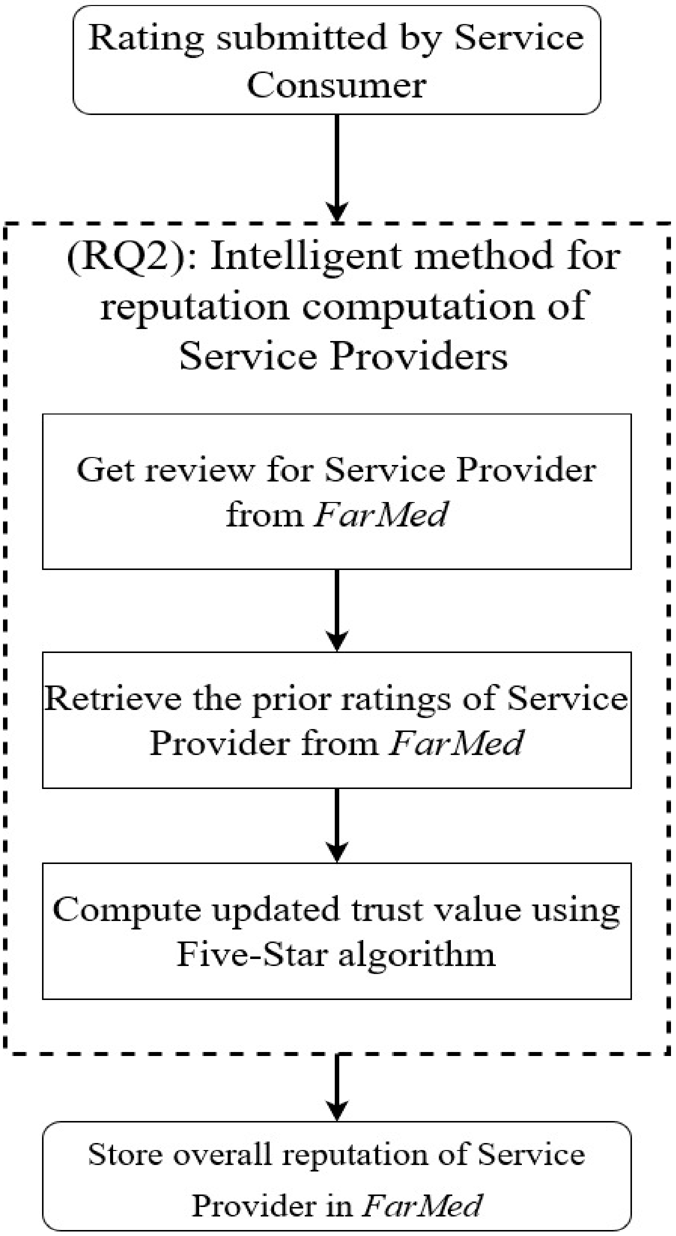

Rating submitted by Service Consumer
(RQ2): Intelligent method for reputation computation of Service Providers
Get review for Service Provider from *FarMed*
Retrieve the prior ratings of Service Provider from *FarMed*
Compute updated trust value using Five-Star algorithm
Store overall reputation of Service Provider in *FarMed*

Figure 4.4. Intelligent Method for Reputation Computation

This system is intelligent because it automatically aggregates the ratings for a service provider and uses an algorithm to derive the trust value from the overall rating of that service provider. The ratings are reliable because it's only service consumers that has patronized a service provider that are allowed to rate the service provider. All the ratings are done on blockchain. Since records on blockchain are immutable, it means the existing rating can never be altered. Furthermore, given the architecture of *FarMed* and because it is based on Blockchain, it is not possible to add spurious or falsified ratings. This is because every added reputation score has to go through a majority consensus. The workflow of intelligent method reputation computation is presented in *Figure 4.4*.

4.6 Overview of solution for RQ3

For answer this research question, an ontology-based method will be used to model the context-specific nature of trust and reputation. Ontology provides common vocabularies for computers and humans to support semantics for knowledge sharing (Fensel et al., 2000). For this study, the concept of transport service ontology is used, and this followed the ontological structure proposed by Dong et al. (2008b).

The workflow for the proposed solution is presented in Figure 4.5. As shown in the figure, we first get information about the existing unrated products from the service provider. After this, we compute the semantic similarity for each of the unrated product based on the rated product. To determine the semantic similarity, we used existing measures for capturing semantic distance between ontology concepts to capture and model the similarity between two products. Finally, using the base ontology and coupling it with the 'distance' between the two products, we used an algorithm to model a context-based inference. The algorithm was only applied when semantic distance is up to 80% and above.

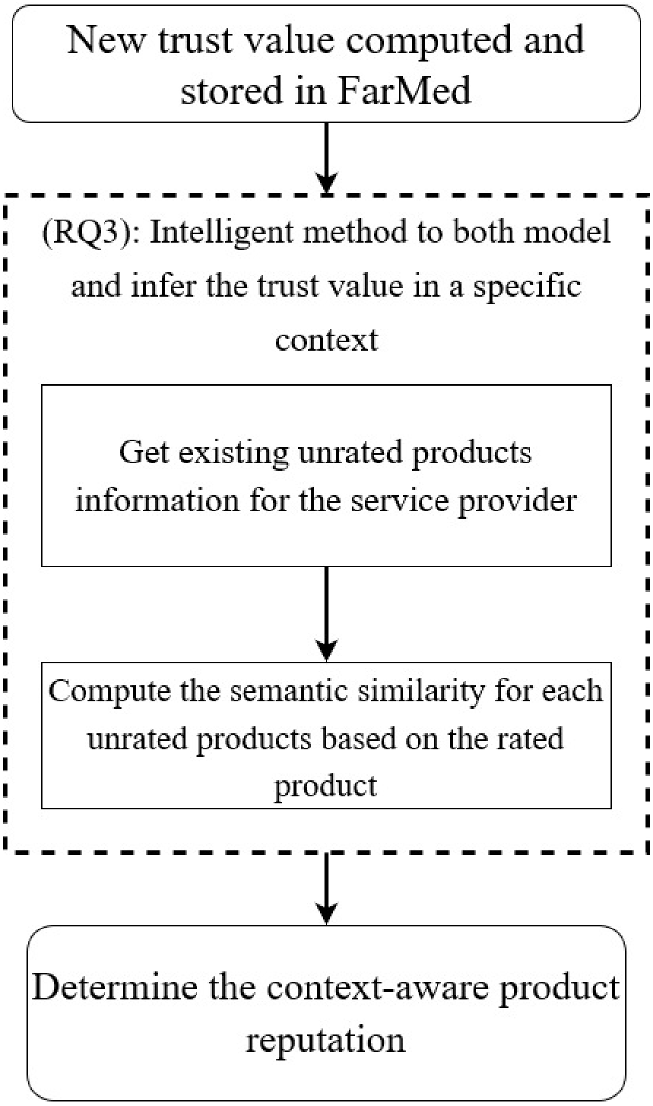

New trust value computed and stored in FarMed
(RQ3): Intelligent method to both model and infer the trust value in a specific context
Get existing unrated products information for the service provider
Compute the semantic similarity for each unrated products based on the rated product
Determine the context-aware product reputation

Figure 4.5. Intelligent Method for Context-driven inferencing of trust value

4.7 Overview of solution for RQ4

In a bid to answer Research Question 4, a Reputation Auction Service (RAS) is developed. This service provides a novel method for the bootstrapping of new service providers in the reputation-based economy. Service providers with high or excess reputation value will auction part of their reputation ratings to other service providers. The new service providers who have zero reputation score can buy from this auction to build their reputation in the market fast while the offers in the auction will be made by service providers who have a high reputation score. The benefit for the service providers is completely commercial. The workflow of intelligent method for service provider reputation transfer on auction is presented in *Figure 4.6*.

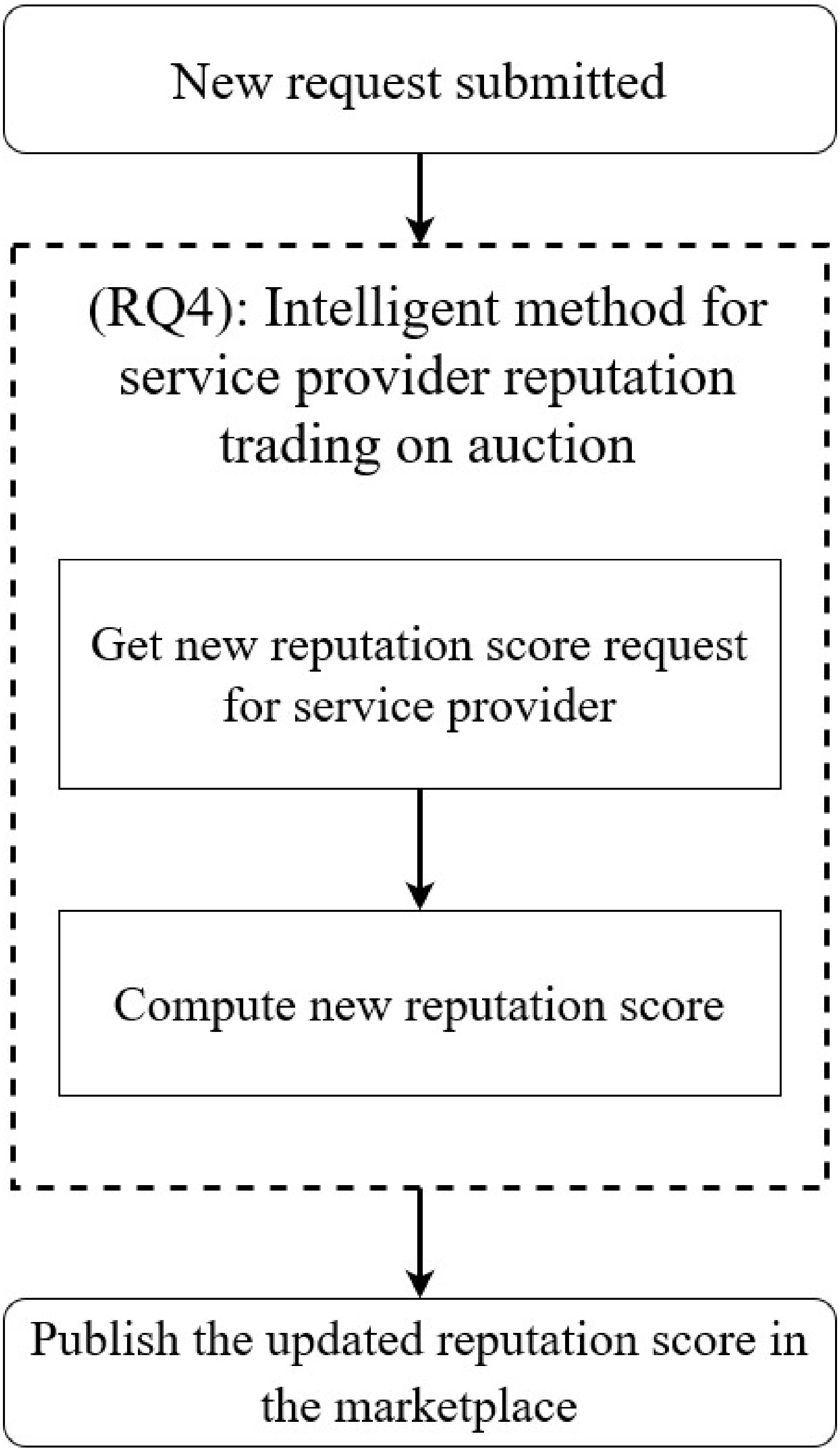

New request submitted
(RQ4): Intelligent method for service provider reputation trading on auction
Get new reputation score request for service provider
Compute new reputation score
Publish the updated reputation score in the marketplace

Figure 4.6. Intelligent Method for service provider reputation transfer on auction

4.8 Overview of solution for RQ5

To answer research question 5, the proposed smart contract-based reputation service was evaluated using the Ethereum Ropsten network. This is done primarily by experimenting the effectiveness for addressing Research Question 1 to Research Question 4 along certain benchmarks. The service built in this research works in a simple process. First, when service consumer makes a purchase of a product from the service provider's store, the transaction is completed through service consumer's Ethereum public address and recorded on the blockchain as a ledger. Once the purchase has been made, the service consumer is now eligible to rate the service provider. The service consumer signs every transaction by using the private key. This is essential because the network is anonymous; therefore, the transaction needs to be validated.

4.8.1 Testing and Verification of the Solution

Runtime Verification in collaboration with Formal Systems Laboratory (FSL) has built a mathematical model of the Ethereum Virtual Machine which makes it possible to verify the accuracy of smart contracts. In fact, the question of whether a program is correct cannot be asked if there is no mathematical model of the computing environment in which the program runs. The smart contracts can be executed based on a rigorous mathematical formalization of the Ethereum Virtual Machine.

4.9 Conclusion

In this chapter, I discussed the methodological approach that used in addressing the gaps identified in the literature review. Specifically, the design science research approach was selected as the model to be used.

Also, the process involved in building *FarMed* Service was discussed. *FarMed* service will integrate reputation systems, smart contracts, and service-oriented computing. Furthermore, an overview of the solution for each of the objectives involved in this research was presented in this chapter.

The next chapter will discuss the smart contract-based solution for computing the reputation value of a service provider based on the previous values stored in the Blockchain

Chapter 5 : Smart Contract-Based Reputation Framework and Determination of the Current Reputation of Service Providers

5.1 Introduction

In this chapter, firstly the processes involved in the building of the smart contract-based framework are discussed. There are basically three phases in creating the framework; each phase is explained in this chapter. Furthermore, a model will be proposed to compute the reputation value of the service provider based on the trust value in the smart contracts; this provides a complete solution to research objective 2. In this chapter, we also present the results of the validation and implementation of proposed solution to the research question two. To validate this research question, we created a blockchain-based reputation system and used the following technologies to test the system:

1. Blockchain: As defined in Chapter 4, a Blockchain is a decentralized and distributed digital ledger that is used to record transactions across different computers so that records are immutable and cannot be altered retroactively, without the alteration of other blocks. In this research, Blockchain was used for storing and retrieving ratings for service providers.

2. PHP: This is also known as Hypertext Preprocessor. It is an open source general-purpose scripting language (Watanabe *et al.,* 2016). We used PHP to run the algorithms. To run the PHP in a local machine, we made use of the XAMPP software, a cross-platform web server solution stack package consisting mainly of MariaDB database, Apache HTTP Server, and interpreters for scripts written in the Perl and PHP programming languages.

3. MetaMask: This is a browser plugin which allows users to make Ethereum transactions through regular platforms (Wohrer and Zdun, 2018). The MetaMask plugin helps in facilitating the adoption of Ethereum by bridging the gap between the user interface for Ethereum and the regular web like Firefox or Chrome. In our work, we used the MetaMask plugin in Chrome to link PHP with Blockchain.

The outline of this chapter is as follows. In Section 5.2, we outline the phases of the smart contracts-based reputation framework in detail. Subsequently, in Section 5.3 we explain how using the proposed framework in Section 5.2, the reputation values can be computed. Furthermore, in Section 5.4, we discuss the aims of the engineering the prototype system used for validating the proposed smart contracts-based reputation framework. In Section 5.5, we outline and explain the dataset used for validation purposes. In Section 5.6 we explain in detail in a stepwise manner the detailed working of the system prototype. Finally, in Section 5.7 we discuss the results obtained from the evaluation.

5.2 Phases in the Smart-Contracts based Reputation Framework

The three phases in the framework include the *marketplace phase*, *smart contract execution phase*, and *trust values and computations phase*. The overview of the complete framework is represented by Figure 5.1.

5.2.1 Working of the Marketplace Phase

An online marketplace is a place where service consumers make purchases of goods and/or services from service providers remotely (Tadelis, 2016). The marketplace phase represents the first phase of the framework. As shown in Figure 5.1, the following processes occur in the marketplace phase.

i. <u>(Step 1) User visits the eMarket</u>: Clients, who are also known as service consumers, visit the online marketplace.

ii. <u>(Step 2) Pick item/s</u>: On the platform, the client will see the item/s available and select the desired item/s.

iii. <u>(Step 3) Display the purchased items</u>: Once the preferred item(s) have been selected, the details of such item(s) will be displayed for the client to see.

iv. <u>(Step 4) Display deal terms and conditions</u>: Afterwards, the terms and conditions associated with the purchase are displayed to the client.

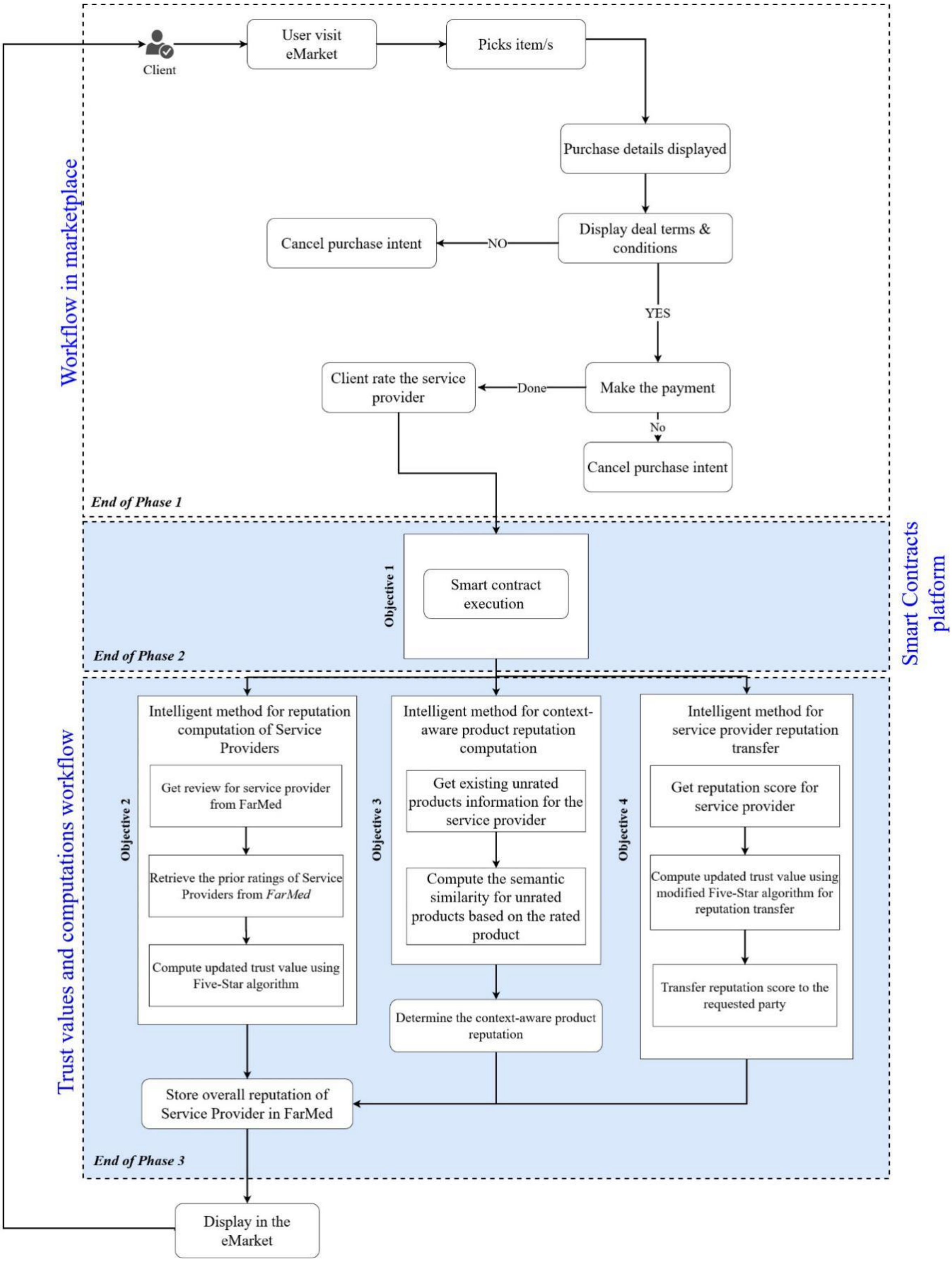

Workflow in marketplace
Client
User visit eMarket
Picks item/s
Purchase details displayed
Display deal terms & conditions
NO
Cancel purchase intent
YES
Make the payment
Done
Client rate the service provider
No
Cancel purchase intent
End of Phase 1
Smart Contracts platform
Objective 1
Smart contract execution
End of Phase 2
Trust values and computations workflow
Objective 2
Intelligent method for reputation computation of Service Providers
Get review for service provider from FarMed
Retrieve the prior ratings of Service Providers from FarMed
Compute updated trust value using Five-Star algorithm
Objective 3
Intelligent method for context-aware product reputation computation
Get existing unrated products information for the service provider
Compute the semantic similarity for unrated products based on the rated product
Determine the context-aware product reputation
Objective 4
Intelligent method for service provider reputation transfer
Get reputation score for service provider
Compute updated trust value using modified Five-Star algorithm for reputation transfer
Transfer reputation score to the requested party
Store overall reputation of Service Provider in FarMed
End of Phase 3
Display in the eMarket

Figure 5.1. Smart Contract-based Reputation Framework

v. (Step 5) Cancel Purchase Intent: If the client is not satisfied with the terms and conditions, there is an option of cancelling the purchase.

vi. (Step 6) Make Payment: However, if the client is satisfied with the terms and conditions, he/she can proceed to the option of making payment for the purchase. At this point, the client still has the option of cancelling purchase intent if he or she so desires.

vii. (Step 7) Rate the service provider: In the case that the client decides to complete the transaction by making the payment, then, there is an option of rating the service provider. This reputation rating marks the end of the marketplace phase in the framework and it equally marks the beginning of phase 2 which involves smart contract execution.

The outline and relationship between the above steps with the Marketplace Phase are pictorially represented in Figure 5.1.

5.2.2 Smart Contract Execution Phase

In phase two, the smart contract is executed. This phase is novel compared with the current generation of reputation systems. Our proposed framework is the first of its type to use smart contracts for computing the reputation value of the service providers.

Figure 5.2 shows our smart contract's execution logic. The execution can be broken down as in the following steps:

1. Step 1: C is the initialization of the smart contract; that is, it begins the execution of the logic.

2. Step 2: X is the condition that a purchase which has not been rated exists.

3. Step 3: Z is the condition that a rating value has been provided by the service consumer.

4. Step 4: If any of conditions X and Z is not met, the smart contract execution will automatically fail. For the execution to be completed, both conditions X and Z must be true.

5. Step 5: If X and Z are true, that means all the conditions have been met. Therefore,

the operation address Q is executed.

6. Step 6: Execute Five-Star algorithm (CMS, 2019) and store the new rating value in the blockchain. Although we have used Five-Star algorithm for computing the reputation value based on aggregated individual ratings, any existing algorithm for reputation computation that is based on aggregating individual ratings can be used in our proposed framework.

Figure 5.2 below shows the logical working of our proposed smart contracts using the above steps.

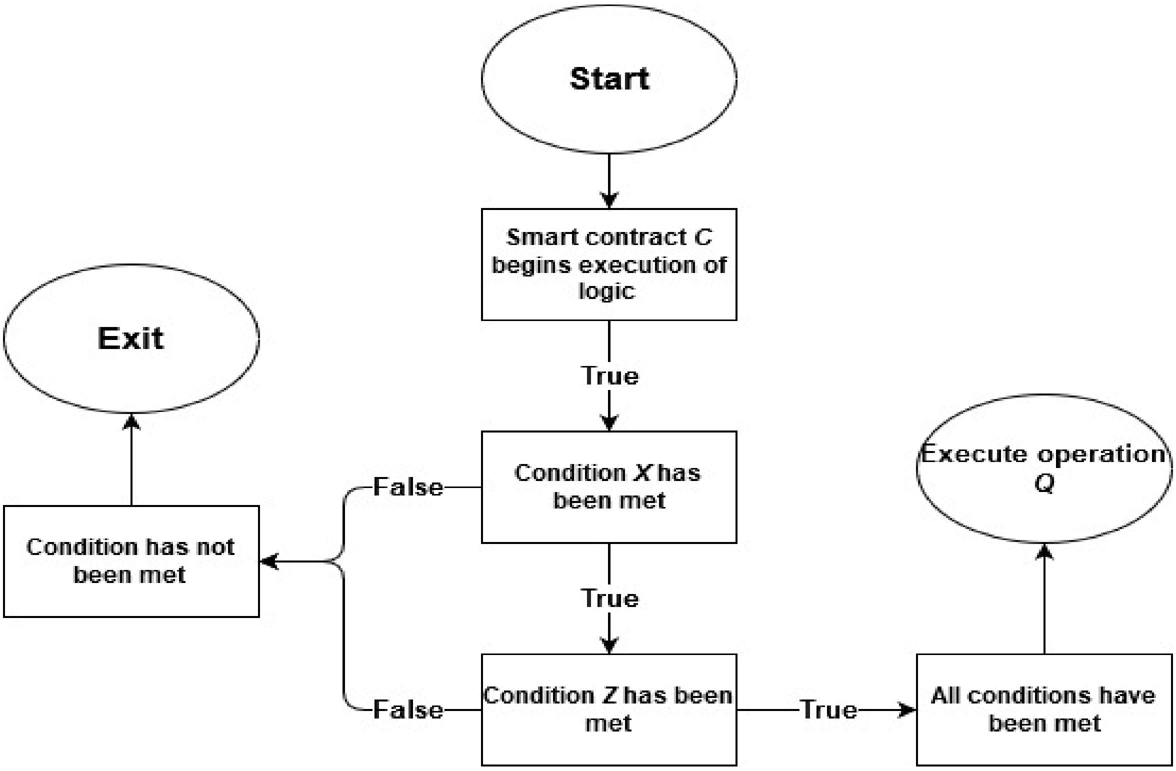

Start
Smart contract *C* begins execution of logic
True
Exit
Condition *X* has been met
False
Execute operation *Q*
Condition has not been met
True
Condition *Z* has been met
False
True
All conditions have been met

Figure 5.2. Smart Contract Execution Logic for Reputation Systems

5.2.3 Blockchain-Based Trust Values and Computations Phase

This phase relates to three objectives of this study. The pictorial representation and working of this phase are shown in Figure 5.3 and it is explained as follows:

i. Intelligent method for reputation computation using smart contracts: To achieve this, the service provider is provided with a review (or rating by other service consumers). As a result of this, its trust value is updated. Afterwards, the overall reputation rating of the service provider is stored in the Blockchain layer of *FarMed*. This is related to objective 2 of this research study.

ii. Intelligent method for inferring the reputation value in a different context: Here, the current review value of the product is first retrieved, and the inference of the trust value in a different context is computed using five-star algorithm. Then, the overall rating of the service provider for the specific product is computed and stored in the Blockchain layer of *FarMed*. This is related to objective 3 of this research study.

iii. Intelligent method for bootstrapping new service providers with low reputation values: To achieve this, we propose the notion of *Reputation Auction Service* (RAS). The RAS's primary objective it to enable new service providers to get started on the *FarMed* platform. Mediated by the RAS, in our proposed approach, new service providers request to purchase a reputation score from old service providers who have accumulated enough reputation on the platform. We call this process Reputation Lending. RAS provides a robust mechanism to ensure that the Reputation Lending is fair and equitable. This is related to objective 4 of this research study.

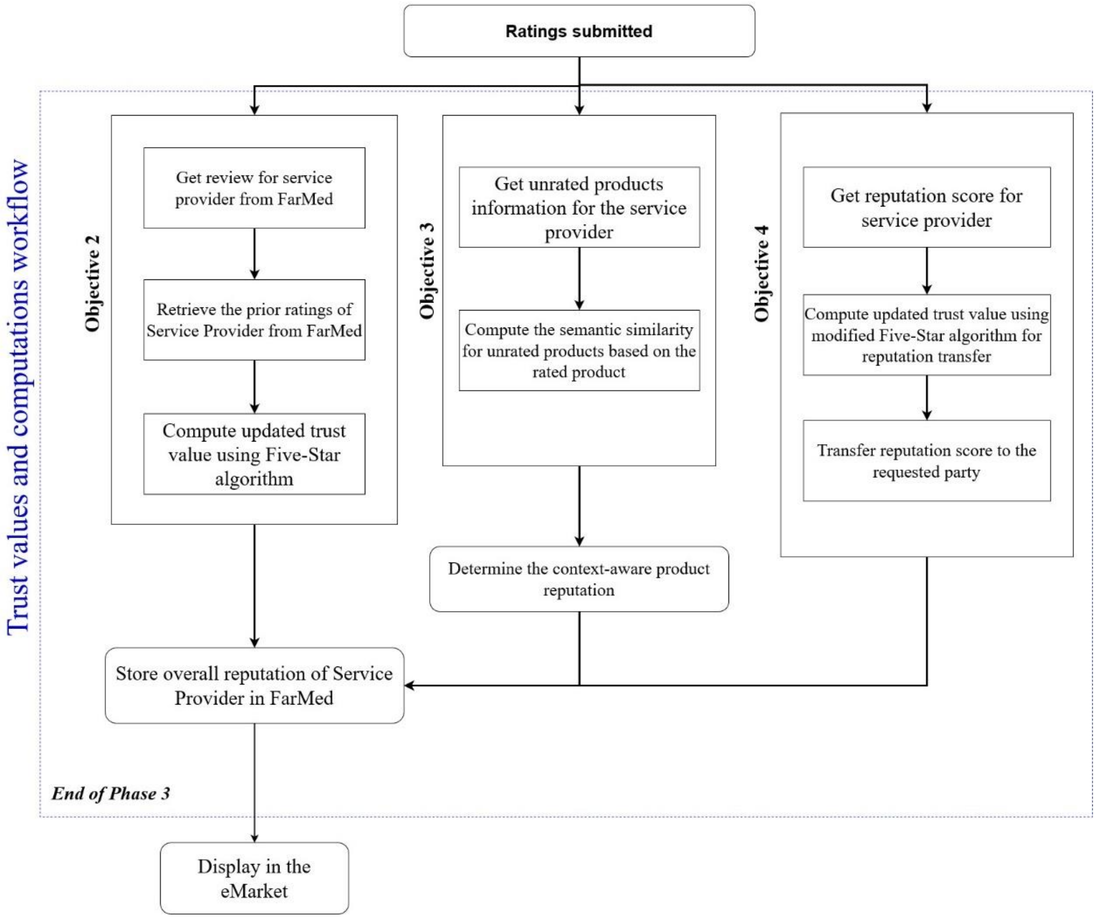

Ratings submitted
Trust values and computations workflow
Objective 2
Get review for service provider from FarMed
Retrieve the prior ratings of Service Provider from FarMed
Compute updated trust value using Five-Star algorithm
Objective 3
Get unrated products information for the service provider
Compute the semantic similarity for unrated products based on the rated product
Determine the context-aware product reputation
Objective 4
Get reputation score for service provider
Compute updated trust value using modified Five-Star algorithm for reputation transfer
Transfer reputation score to the requested party
Store overall reputation of Service Provider in FarMed
End of Phase 3
Display in the eMarket

Figure 5.3. Trust Values and Computations Phase

Once the actions required in cases is completed, the overall reputation of the service provider is displayed in the online marketplace. This marks the end of the third and last phase of the framework. Details on how the intelligent methods are created for each of the cases are provided in Section 5.3 (of this chapter), Chapter 6, and Chapter 7. The detailed algorithmic working of objective 2 is explained in Section 5.3. Furthermore, the detailed algorithmic workings of objective 3 and objective 4 are presented in Chapter 6 and 7 respectively.

5.3 Modelling and Determination of the Current Reputation Values of Service Providers

In this section, we will discuss how we model and determine the current reputation values of service providers thereby achieving objective 2 of this study.

5.3.1 Workflow for Solution

For research objective 2, in this research we used the Ropstein Ethereum network (Bogner *et al.,* 2016). The Ropstein Ethereum network validates the service provider's rating through a signature and adds the rating to the network. The information stored in the network includes the raw rating value, the service provider's ID and the service consumer's ID.

This system is intelligent because it automatically aggregates the ratings for a service provider and uses an algorithm to derive the overall reputation value based on all the previous ratings of that service provider. The ratings are reliable because it is only service consumers that have patronized a service provider that are allowed to rate the service provider. All the ratings are stored in *FarMed*'s Blockchain layer. Since the records of the blockchain are immutable, it means the rating can never be altered, except by Majority Consensus. The workflow of Intelligent Method for Reputation Computation is presented in Figure 5.4.

The five-star algorithm is used to compute the overall reputation value for the service provider as follows (CMS, 2019):

$$Rating = {}^{(T_1 n_1 + \ T_2 n_2 + T_3 n_3 + \cdots + T_f n_f)}\big/_{(n_1 + n_2 + n_3 + \cdots + n_f)}$$

Equation: 5.1

$$Rating = {}^{\sum Tn}\big/_{\sum n}$$

Equation: 5.2

where T is the rating value which can be 1, 2, 3, 4 or 5 only, due to smart contracts structure limitations.

n is the number of rating(s) having the same rating value

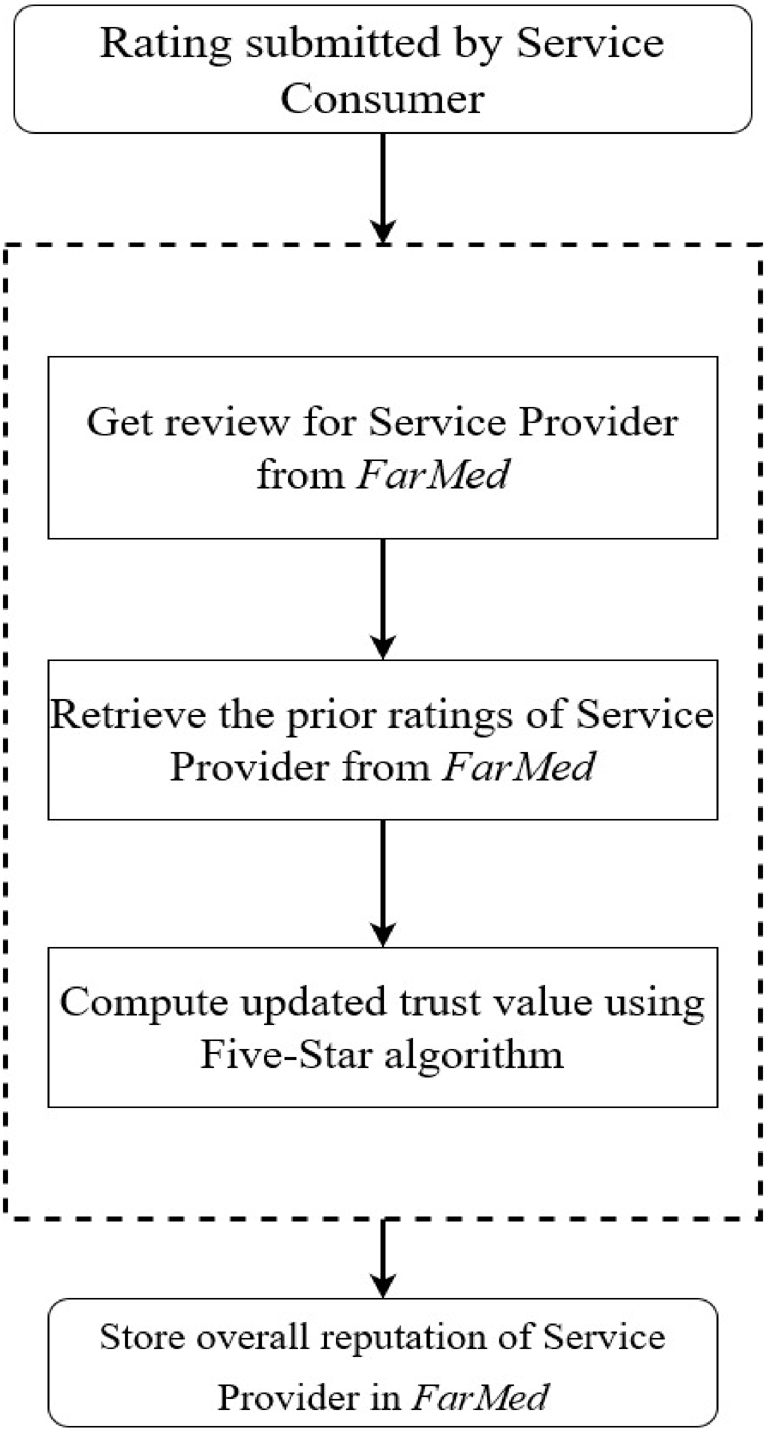

Rating submitted by Service Consumer
Get review for Service Provider from *FarMed*
Retrieve the prior ratings of Service Provider from *FarMed*
Compute updated trust value using Five-Star algorithm
Store overall reputation of Service Provider in *FarMed*

Figure 5.4. Intelligent Method for Reputation Computation

All the ratings referred to in Figure 5.4 are either retrieved from or stored in *FarMed*'s Blockchain layer. Equations 5.1 and 5.2 are about reputation computation (based on previous ratings). Each service provider and product in the marketplace will be associated with a reputation value. The parameters of the service provider and product are stored in *FarMed*'s Blockchain layer. The following attributes are stored for the service provider: service provider address, service consumer address, rating, and timestamp. The following attributes are stored for the product: product address, service provider address, service consumer address, rating, and timestamp.

5.3.2 Computing the updated Trust Value of a Service Provider

Once a new review for the service provider has been submitted, we propose the following workflow process to compute the trust value. The workflow is pictorially shown in Figure 5.5.

The steps involved in storing the new trust value and computing the updated reputation value of a given service provider is as below:

i. Request all previous ratings from *FarMed*: First, all the previous ratings that are related to the service provider and are stored in *FarMed* are requested.

ii. Add the newly submitted rating: The new review of the service provider is added to the existing reviews for that service provider. The addition of a new review is carried out by adding a block to *FarMed*.

iii. Compute new overall rating: The five-star algorithm is used to compute the new overall rating of the service provider. The Five-Star algorithm uses the prior ratings in *FarMed* as well as the new rating to compute the overall reputation value.

iv. Provide the new overall rating of the service provider to the service consumer: The updated overall trust value of the service provider is then stored in *FarMed*'s Blockchain layer.

v. Get confirmation of the submission from blockchain and publish the new overall rating of the service provider.

The logical working of the above steps is shown pictorially in Figure 5.5 below.

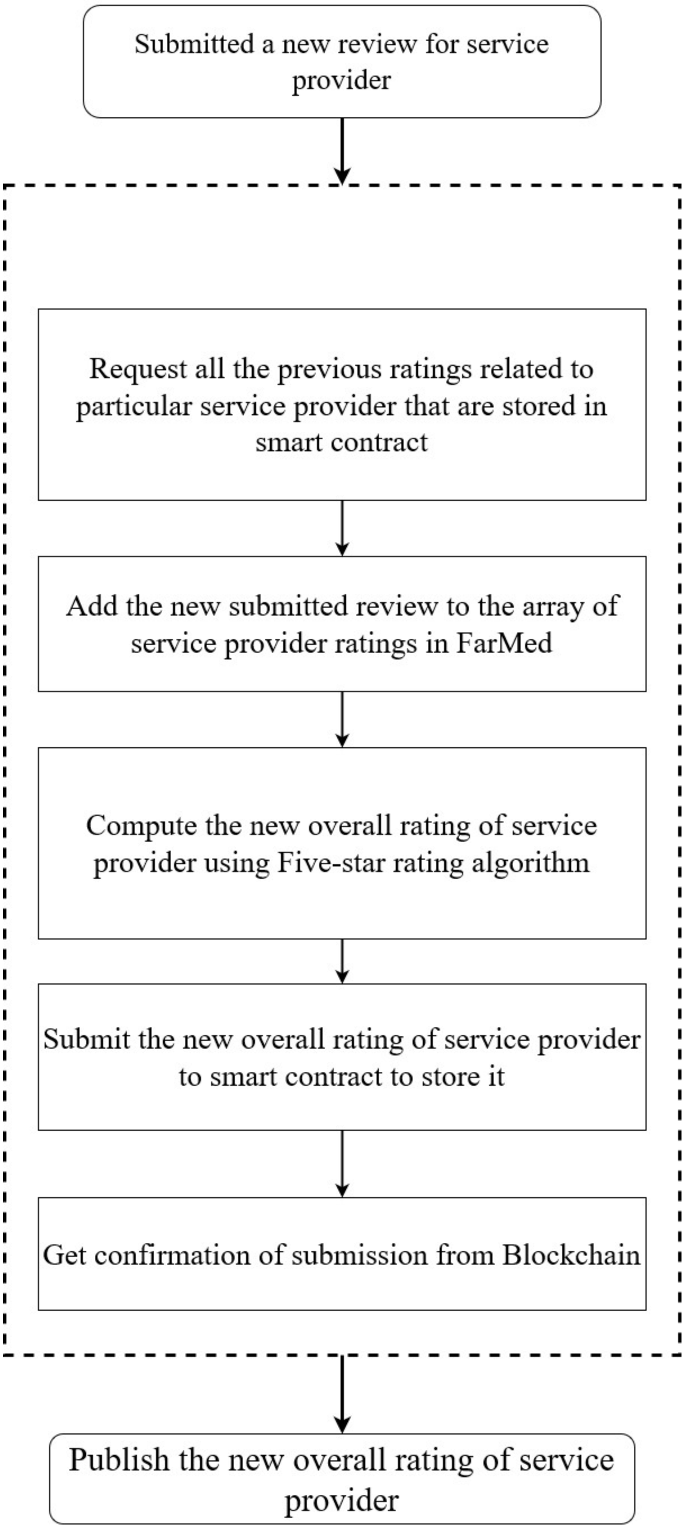

Submitted a new review for service provider
Request all the previous ratings related to particular service provider that are stored in smart contract
Add the new submitted review to the array of service provider ratings in FarMed
Compute the new overall rating of service provider using Five-star rating algorithm
Submit the new overall rating of service provider to smart contract to store it
Get confirmation of submission from Blockchain
Publish the new overall rating of service provider

Figure 5.5. Overview of steps in Computing Trust Values of Service Provider

5.4 Prototype Implantation

5.4.1 Aims of Engineering the Prototype System

The primary aim of engineering the prototype system was to simulate the working of the *FarMed* Service, an intelligent framework that executes Ethereum smart contract-based reputation system. Particularly, in this section, our objective was to use the developed *FarMed* prototype to test the efficacy of the methods proposed in Section 5.3 to compute the current reputation of service providers. For validating other objectives of this thesis, we have extended the prototype system (see Chapter 6 and Chapter 7)

As discussed in Chapter 2, in the current literature and also in practice, one of the major challenges of reputation systems is that mischievous users normally build fake identities which they use to boost their reputation in the online marketplace while many legitimate service providers are also subjected to attack (Cai and Zhu, 2016). With our intelligent framework which involves the integration of smart contracts, reputation systems and service-oriented computing, the integrity of reputation or trust values of a service provider can now be adequately preserved.

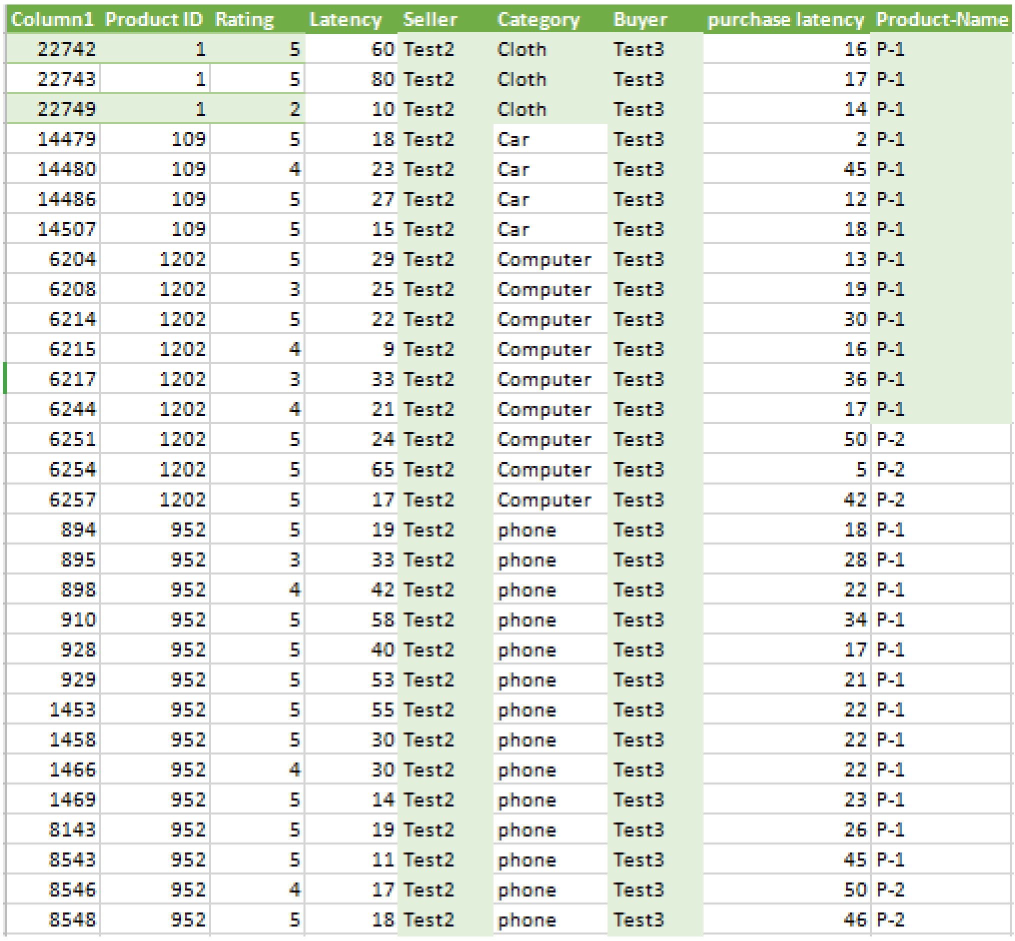
We used the traditional Ethereum technology as our computing platform and its programming language is Solidity (Wood, 2014). We also tried to use Dfinity (Butcher and Lunden, 2020) and Ethereum 2.0 (Rachel, 2020). However, both technologies are still at the development stage. In particular, Dfinity has just released an initial alpha version to enable developers to become familiar with the technology. Therefore, the technology is still in it’s infancy with clear indications that there are still a lot of changes to make before the comprehensive version. Also, since Dfinity does not support Solidity, it is impossible to use it to run Ethereum smart contracts. Ethereum 2.0 is faster than the traditional Ethereum and it can run the Ethereum smart contracts. However, just like Dfinity, Ethereum 2.0 is also in the development stage. Hence, it was not possible for use to use any other platform other than Ethereum.

5.5 Datasets Used for Validation

The dataset used for validation is publicly available. We used Women's Clothing E-commerce dataset revolving around ratings (and reviews) provided by customers. The data is real and commercial and was therefore anonymized. The names of the interacting parties were anonymized. Therefore, references to the company involved were replaced with 'retailer'. This dataset was taken from Kaggle (Nicapotato, 2018), a platform that offers a huge repository of community published data. The data set can be accessed from [https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews].

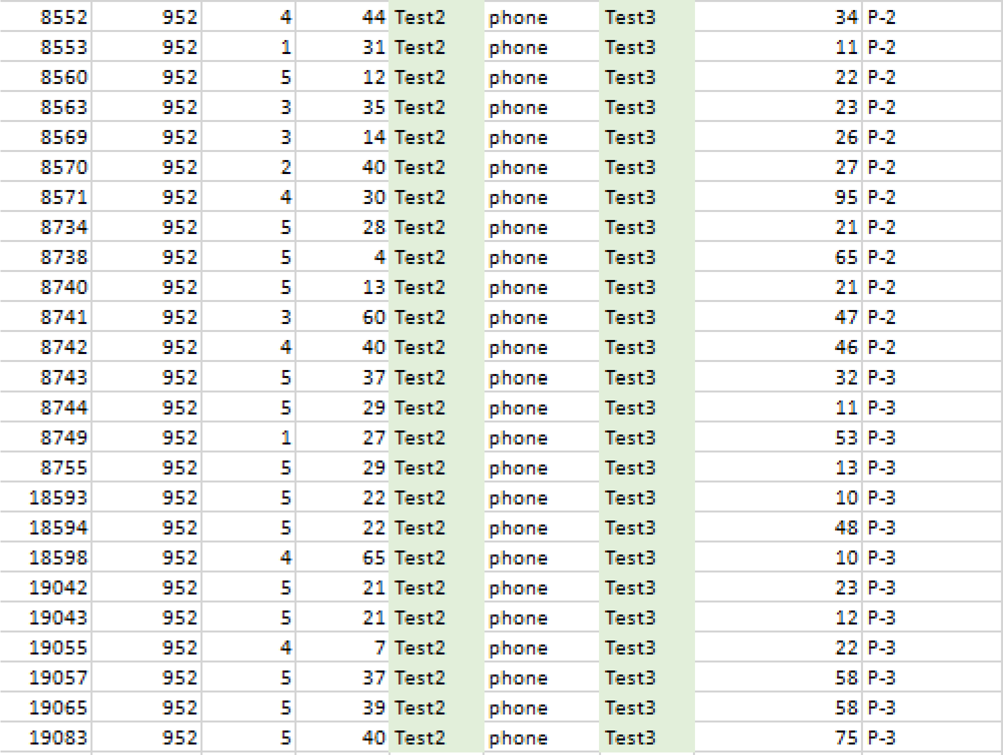
Before making use of the dataset obtained from Kaggle, we modified some of the columns to fit in to our requirements. However, this modification was done without changing any of the rating values. In particular, we added categories that do not exist in the dataset and assigned them to specific products. The purpose of doing this was to allow reputation computation in a context.

Table 5.1 and 5.2 show a sample of the dataset used for the implementation.



Column1	Product ID	Rating	Latency	Seller	Category	Buyer	purchase latency	Product-Name
22742	1	5	60	Test2	Cloth	Test3	16	P-1
22743	1	5	80	Test2	Cloth	Test3	17	P-1
22749	1	2	10	Test2	Cloth	Test3	14	P-1
14479	109	5	18	Test2	Car	Test3	2	P-1
14480	109	4	23	Test2	Car	Test3	45	P-1
14486	109	5	27	Test2	Car	Test3	12	P-1
14507	109	5	15	Test2	Car	Test3	18	P-1
6204	1202	5	29	Test2	Computer	Test3	13	P-1
6208	1202	3	25	Test2	Computer	Test3	19	P-1
6214	1202	5	22	Test2	Computer	Test3	30	P-1
6215	1202	4	9	Test2	Computer	Test3	16	P-1
6217	1202	3	33	Test2	Computer	Test3	36	P-1
6244	1202	4	21	Test2	Computer	Test3	17	P-1
6251	1202	5	24	Test2	Computer	Test3	50	P-2
6254	1202	5	65	Test2	Computer	Test3	5	P-2
6257	1202	5	17	Test2	Computer	Test3	42	P-2
894	952	5	19	Test2	phone	Test3	18	P-1
895	952	3	33	Test2	phone	Test3	28	P-1
898	952	4	42	Test2	phone	Test3	22	P-1
910	952	5	58	Test2	phone	Test3	34	P-1
928	952	5	40	Test2	phone	Test3	17	P-1
929	952	5	53	Test2	phone	Test3	21	P-1
1453	952	5	55	Test2	phone	Test3	22	P-1
1458	952	5	30	Test2	phone	Test3	22	P-1
1466	952	4	30	Test2	phone	Test3	22	P-1
1469	952	5	14	Test2	phone	Test3	23	P-1
8143	952	5	19	Test2	phone	Test3	26	P-1
8543	952	5	11	Test2	phone	Test3	45	P-1
8546	952	4	17	Test2	phone	Test3	50	P-2
8548	952	5	18	Test2	phone	Test3	46	P-2

Table 5.1 (See description below)



8552	952	4	44	Test2	phone	Test3	34	P-2
8553	952	1	31	Test2	phone	Test3	11	P-2
8560	952	5	12	Test2	phone	Test3	22	P-2
8563	952	3	35	Test2	phone	Test3	23	P-2
8569	952	3	14	Test2	phone	Test3	26	P-2
8570	952	2	40	Test2	phone	Test3	27	P-2
8571	952	4	30	Test2	phone	Test3	95	P-2
8734	952	5	28	Test2	phone	Test3	21	P-2
8738	952	5	4	Test2	phone	Test3	65	P-2
8740	952	5	13	Test2	phone	Test3	21	P-2
8741	952	3	60	Test2	phone	Test3	47	P-2
8742	952	4	40	Test2	phone	Test3	46	P-2
8743	952	5	37	Test2	phone	Test3	32	P-3
8744	952	5	29	Test2	phone	Test3	11	P-3
8749	952	1	27	Test2	phone	Test3	53	P-3
8755	952	5	29	Test2	phone	Test3	13	P-3
18593	952	5	22	Test2	phone	Test3	10	P-3
18594	952	5	22	Test2	phone	Test3	48	P-3
18598	952	4	65	Test2	phone	Test3	10	P-3
19042	952	5	21	Test2	phone	Test3	23	P-3
19043	952	5	21	Test2	phone	Test3	12	P-3
19055	952	4	7	Test2	phone	Test3	22	P-3
19057	952	5	37	Test2	phone	Test3	58	P-3
19065	952	5	39	Test2	phone	Test3	58	P-3
19083	952	5	40	Test2	phone	Test3	75	P-3

Table 5.2 (See description below)

Tables 5.1 and 5.2 are showing the dataset used for the implementation of the solutions.

5.6 Steps in the working of the prototype system for reputation computation

In order to achieve its job, the *FarMed* service carries out computations at two different levels, namely local computations and Blockchain computations. In local computations, all the reputation computations are executed locally (i.e., in the local time). In our proposed method the reputation computations processing happens at the local level. However, the reputation scores are stored and retrieved in the Blockchain. The steps for both the local level and Blockchain level computation are explained below in Section 5.6.1 and Section 5.6.2 respectively.

5.6.1 Steps in local reputation computation

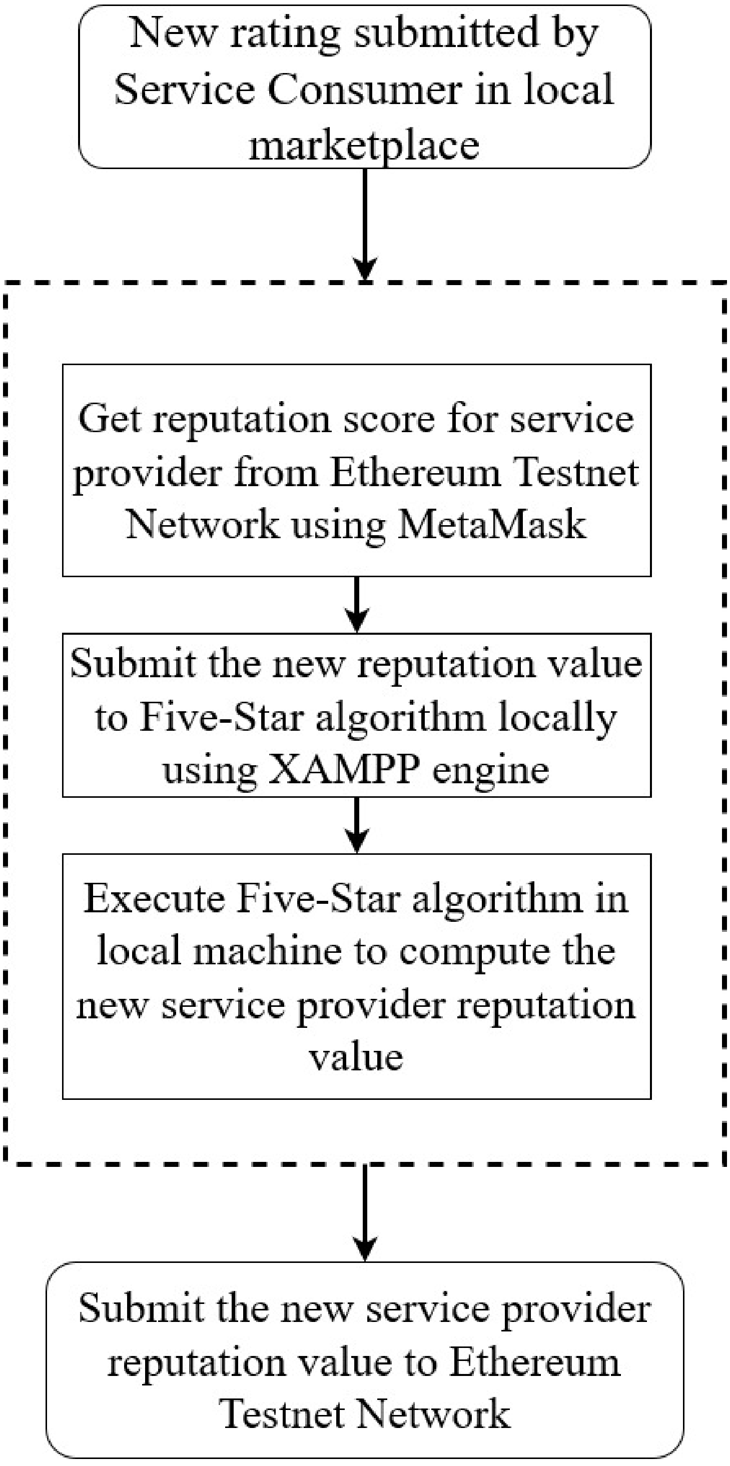

New rating submitted by Service Consumer in local marketplace
Get reputation score for service provider from Ethereum Testnet Network using MetaMask
Submit the new reputation value to Five-Star algorithm locally using XAMPP engine
Execute Five-Star algorithm in local machine to compute the new service provider reputation value
Submit the new service provider reputation value to Ethereum Testnet Network

Figure 5.6. Flow chart of local marketplace reputation computation process

After new rating is submitted by the service consumer in the local marketplace, the process goes through three steps as follows:

1. Step 1: Local marketplace acquires the service provider's current reputation score from Ethereum Testnet Network using Metamask.
2. Step 2: Run XAMPP (Cross-platform, Apache, MariaDB (Mysql), PHP and Perl) engine and prepare it for new execution. This step runs in the local machine automatically.
3. Step 3: Execute Five-Star algorithm in local machine to compute the new reputation value of service provider.
4. Step 4: After that, the new overall value of service provider reputation score is carried over to Ethereum Testnet Network to store it.

5.6.2 Steps in Blockchain-based reputation computation

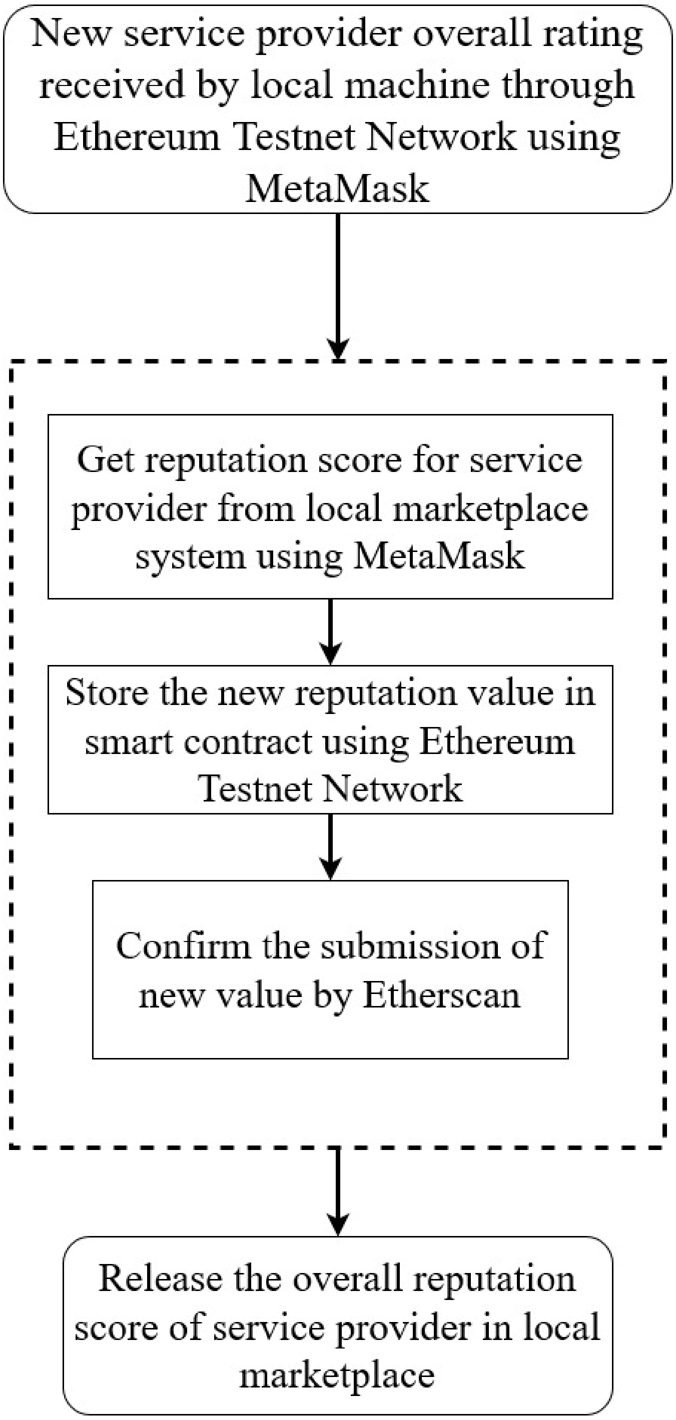

New service provider overall rating received by local machine through Ethereum Testnet Network using MetaMask
Get reputation score for service provider from local marketplace system using MetaMask
Store the new reputation value in smart contract using Ethereum Testnet Network
Confirm the submission of new value by Etherscan
Release the overall reputation score of service provider in local marketplace

Figure 5.7. Flow chart of Blockchain reputation computation process

After the new overall reputation score of the service provider has been calculated using Five-Star algorithm in local machine, the process goes through three steps as follows:

1. Step 1: Getting the new calculated overall reputation score of service provider using MetaMask.

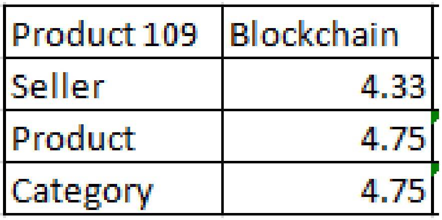
2. Step 2: Storing the reputation score in smart contract using Ethereum Testnet Network.

3. Step 3: Getting confirmation of submission from blockchain using Etherscan.

After the above steps, the new reputation value of the service provider is released in the marketplace.

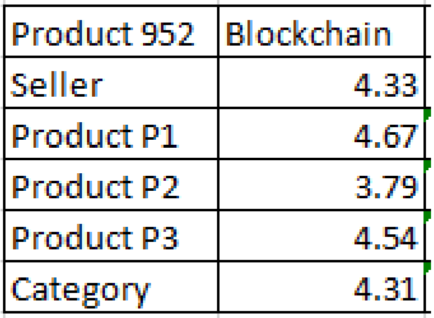
5.7 Prototype Evaluation and Discussion

Objective two involves the modelling and computation of the reputation value of service providers. The five-star algorithm was used for the computation. The results of the implementation carried out to validate the solution to this objective are shown in Tables 5.3 and Table 5.4. Tables 5.3 and 5.4 below show the computed values for two different products (Product 109 and Product 952). Both of these products are being sold by one seller only.



Product 109	Blockchain
Seller	4.33
Product	4.75
Category	4.75

Table 5.3 (See description below)

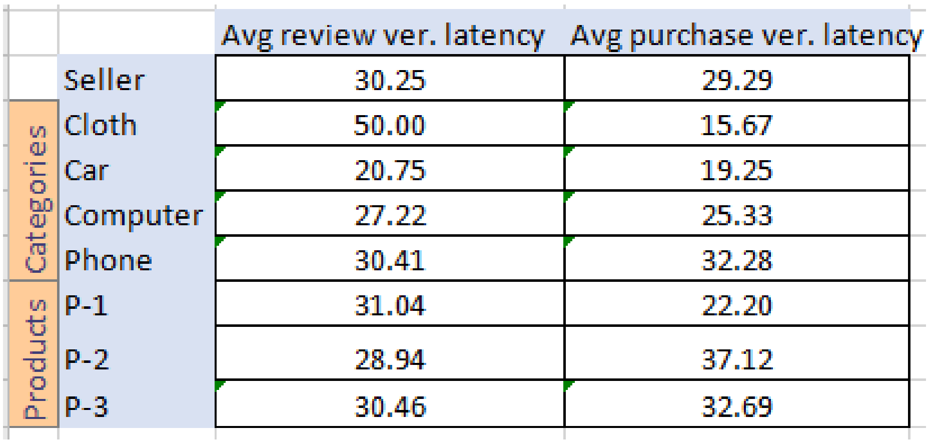


Product 952	Blockchain
Seller	4.33
Product P1	4.67
Product P2	3.79
Product P3	4.54
Category	4.31

Table 5.4 (See description below)

Tables 5.3 and Table 5.4 showing results obtained from using our Blockchain-based platform for Reputation Computation

The use of our blockchain-based platform for reputation computation has several advantages which include accuracy, reliability, and security. On the other hand, the trade-off is latency as shown in the Table 5.5.



		Avg review ver. latency	Avg purchase ver. latency
	Seller	30.25	29.29
Categories	Cloth	50.00	15.67
	Car	20.75	19.25
	Computer	27.22	25.33
	Phone	30.41	32.28
Products	P-1	31.04	22.20
	P-2	28.94	37.12
	P-3	30.46	32.69

Table 5.5. Latency values in seconds

In Table 5.5, it can be seen that it takes some time for results of computations to be returned when blockchain was used to retrieve the reputation values. For example, for Product P-1, the average review latency was 31.04 seconds. Although, this may be considered a limitation for the method, the reliability, security, and accuracy derived from using the method far outweigh this limitation. In current generation databases, for the datasets described in Table 5.1 and Table 5.2 one would expect latency of a few milliseconds in the worst-case scenario. See Figure 5.8 below for more details.

We are experiencing latency in FarMed due to consensus mechanism as there is a need to verify each transaction (Block-Supply Chain, 2020).

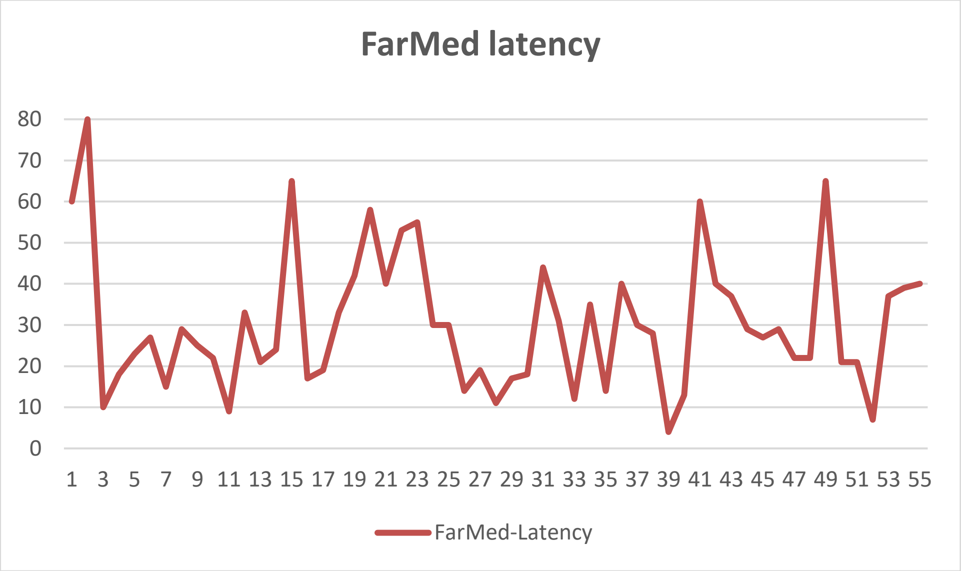


Figure 5.8. Latency experienced by *FarMed*

5.8 Conclusion

In this chapter, we discussed all the three phases that comprise the smart contract framework. In particular, we discussed in a stepwise manner the processes involved in the marketplace phase, the smart contract execution phase as well as the trust values and computations phase. In addition, we proposed a model to compute the trust value of the service provider based on the previous rating using smart contracts. This proposed a smart contract-based framework for trust computation be developed to address the second research objective of this thesis.

In this chapter we also proposed a simulation framework for validating the solution to the research objective 2. This was done using a public and real dataset. At the end of the validation and implementation process, the results show that our proposal is able to compute the overall trust value of the service provider. However, a key finding from the validation phase was latency in the computation of the trust values. The users of our proposed *FarMed* will have to be mindful of latency when using it.

The next chapter will discuss in a methodological manner the steps involved in developing a service ontology to model and deduce the trust value of service providers and service consumers within a context.

Chapter 6 : Context-driven inferencing of Trust Value for Service Providers

6.1 Introduction

In the last Chapter, we present the working of our proposed method (using *FarMed*) for computing the trust value of a service provider in a given context (say C_1) based on prior existing value for other contexts (assume $C_{2,}$ $C_{3,}C_{n,}$) . A key attribute of trust is that it is context-specific or context-driven (Sherman, 2018). Due to this, the trust value of a service provider is associated with it in a specific scenario or context (Sherman, 2018). Hence, it is important that a trust value associated with a service provider is not regarded across multiple different contexts in which it provides services.

In this chapter, building on the work presented in the last chapter, we present an intelligent method to carry out context-driven trust assessments. In our work, we use a service ontology coupled with distance-based approaches to intelligently model and infer the trust value of a service provider in a specific context.

This chapter is described as follows: In Section 6.2, we provide a detail explanation of service ontology, service metadata, transport service ontology and service knowledge base. In Section 6.3, the algorithm for semantic distance computation is discussed.

Furthermore, Section 6.4 provides the algorithm required to carry out the context-specific inferencing. The proposed solution is validated in Section 6.5 while Section 6.6 concludes in this chapter.

6.2 Service Ontology

Ontology can be defined as “an explicit specification of a shared conceptualization, readable by machine” (Gomez-Perez and Corcho, 2002). In basic terms, ontology provides common vocabularies for computers and humans to support semantics for knowledge sharing (Fensel *et al.,* 2000). Today, ontology is being applied in different fields to conceptualize specific domain knowledge as well as solve any inter-operability problems that emerge during knowledge sharing among cross-domains. Some of the fields utilizing ontology include semantic web, e-commerce, logistics and transportation, health science, and many others.

Service ontology is a domain-specific ontology which is applied to define the semantics of the queries and the services (Fensel *et al.,* 2000). Service ontology involves a knowledge-based representation that defines concepts and their semantic relations, it describes conceptual modeling between service providers and service consumers. It also supports data extraction using both query languages (SPARQL) and Web APIs (Fensel *et al.,* 2000). For example, an e-commerce platform or online marketplace that is service ontology-oriented can effectively identify groups or classes which describe a domain of knowledge along with relationships between concepts.

An ontology is used by machines to understand and model organizational knowledge. Extracting texts from various structures and different data sources and constructing ontology engineering based on this data has emerged as a promising way to reduce the cost for building and maintaining domain ontologies. This learning process is enabled to keep track of modelling choices and to connect many lexical entries (or terms) to concepts in order to associate a document with a formal interpretation of its textual content. An important step during the construction of an Ontology Knowledge Base involves all concepts and semantic relations and the model has to exclude every incoherent statement.

In this thesis, the concept of transport service ontology is used, and this followed the ontological structure proposed by Dong *et al.* (2008b). This study also follows the existing retrieval and query used in Dong *et al.* (2008a). Both the transport service metadata and transport service ontology form the Transport Service Knowledgebase.

6.2.1 Service Metadata

This section describes the methodology for semantically retrieving products reputation scores. The term "product" refers to the service offered by the service provider in the online marketplace. In this study, the product metadata is used to represent the services. The content of the services provider important information that can explain the products, for example, product description and product name.

The product metadata in this study is defined using the service structure proposed by Dong *et al.* (2011). As proposed by Dong et al (2011), the product metadata has four properties which includes product name, relevant service concepts, service provider address and service description.

The service concept refers to ideas or concepts that are provided in the specific service domain, like health service domain, mining service domain, and the transport service domain.

Dong *et al.* (2008a) utilized the concept of service metadata to build the transport service metadata. According to Dong *et al.* (2008), the primary aim of transport service metadata is to retrieve meaningful information about transport service. The retrieval of information towards the semantic query plays a vital role in finding similar conceptual terms in ontology. The transport metadata is illustrated in Figure 6.1. The figure shows that the service metadata can be represented as a tuple having complex elements defined as follows (Dong et al., 2008b):

- Linked Concepts: Make references to the semantically linked concepts
- Service Provider Name: This refers to the name of the entity that provides a service.
- Provider Address: This refers to the location address of a service provider
- Provider Contact Details: This refers to all information regarding how a service provider can be contacted
- Service Description: This is a detailed description of the content of the service.

Transport Service Metadata
Linked Concepts Service Provider Name Provider Address Provider Contact Details Service Description

Figure 6.1. Transport service metadata format (Dong *et al.,* 2008b)

6.2.2 Transport Service Ontology

Like the service metadata, the ontological structure for the Transport Services that is proposed by Dong *et al.* (2008b) will be applied to define the service ontology in this study. As shown in Figure 6.2, service ontology has a hierarchical structure. The diagram shows the arrangement of the service concepts into different levels. Any service concept in the lower level is regarded as sub-domain of the service concepts in the upper level. Going by this, it means the concept in level 0 is the main service domain while all the concepts in level 1 are sub-domains of level 0. This simple explanation applies to all the levels and it means that the service concepts in the upper level are more general while those in the lower level are more specific.

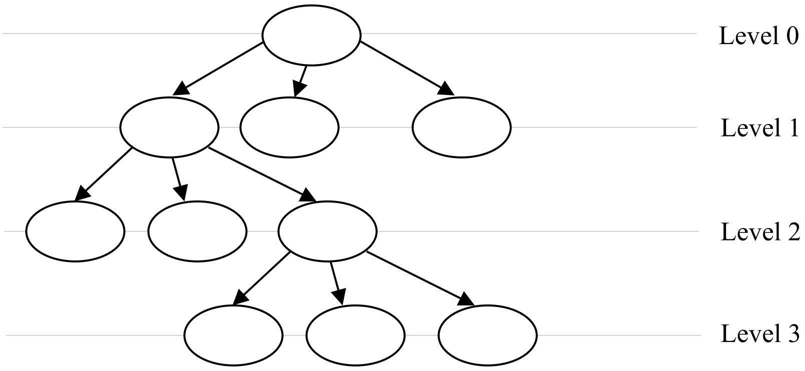


Figure 6.2. The structure of the service ontology

Applying the concept explained above, Dong *et al.* (2008b) built a transport service ontology as shown in Figure 6.3.

In the proposed ontology by Dong *et al* (2008b), the transport services which is the main domain consists of four service sub-domains, including "Shipping Transport Service", "Rail Transport Service", "Air Transport Service", and "Road Transport Service". Each sub-domain comprises abstract service concept; for instance, the "Air Transport Service" sub-domain contains different abstract concepts such as "Aircraft Charter/Rental Service" and "Air Cargo Service - Abstract". A service concept is linked to the service metadata if the relevance score is greater than the threshold.

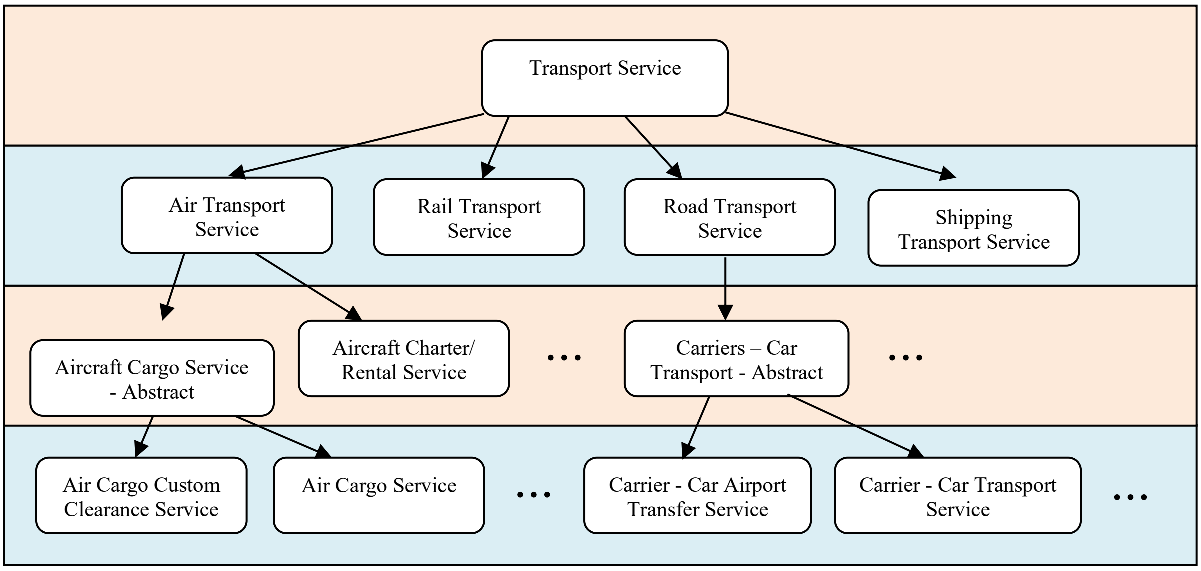

Transport Service
Air Transport Service
Rail Transport Service
Road Transport Service
Shipping Transport Service
Aircraft Cargo Service - Abstract
Aircraft Charter/ Rental Service
…
Carriers – Car Transport - Abstract
…
Air Cargo Custom Clearance Service
Air Cargo Service
…
Carrier - Car Airport Transfer Service
Carrier - Car Transport Service
…

Figure 6.3 The transport service ontology (sourced from Dong et al (2008b))

6.2.3 Service Knowledge Base

The service knowledge base comprises both service ontology and service metadata. The service knowledge base presented in this section is taken from Dong *et al.* (2011). It is used to store services and their semantics for the service retrieval methodology.

Figure 6.4 represents the structure of the service knowledge base. The structure consists of a domain-specific service ontology and an assemblage of the service metadata. A service metadata can be linked to one or more service concepts. Similarly, a service concept can be linked to one or more services. For instance, Figure 6.4 shows that the service metadata of 4 (SDE_4) is linked to two service concepts. The figure also shows that there is a particular service concept linked to both SDE_4 and SDE_1.

The links typically symbolize the relevance between the service concepts and the service metadata. In that case, service metadata can act as semantics or ideas of services, implying that one service may have different meanings and several services may mean the same thing. Also, as shown in Figure 6.4, the relatedness between the service concept and service metadata is called “service annotation”.

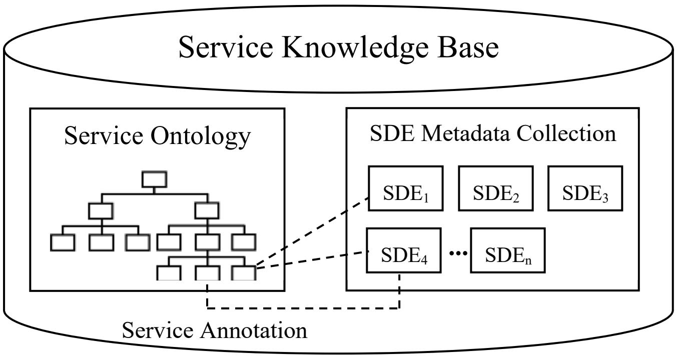


Figure 6.4. Service Knowledge base (sourced from Dong et al (2008b))

6.3 Algorithm for Semantic Distance Computation

The proposed solution to Objective 3 is heavily grounded in the use of ontology. The proposed solution for context-based trust inferencing is built on two existing approaches, Service ontology and AKTiveRank (Alani and Brewster, 2006). We use the combination of the above two approaches Ontology and AKTiveRank to compute the trust value in a related context. This model is then integrated into the *FarMed* framework.

6.3.1 AKTIVERANK

The current architecture of the AKTiveRank is shown in Figure 6.5. The main component of the architecture is a Java Servlet (No. 2 in the figure) which gets a text query from marketplace (No. 1). The text, which is product name, to be searched for are in the query. Meanwhile, it is worthy of note that product name will only be matched ontology classes rather than with comments or properties. The moment AKTiveRank (No. 2) receives a query, AKTiveRank will query Swoogle (No. 3) for all the product names provided and retrieve the ontology Uniform Resource Identifiers (URIs) from the results returned by Swoogle. Swoogle is an ontology search engine adopted a PageRank-like method to rank ontologies by analysis links and referrals between the ontologies in the hope of identifying the most relevant ones.

After gathering a list of ontology candidates from Swoogle, AKTiveRank will check whether the ontologies are stored in a Jena MySQL database back-end (No. 4). For any ontology that is not in the database, AKTiveRank will download them from the web

(No. 5) and add to the database. The Jena API is used to read the ontologies as well as handle the database.

According to Angles and Gutierrez (2005), the existing Resource Description Framework (RDF) query languages are not suitable for graph queries. Therefore, the AKTiveRank is connected to a purpose-built JUNG servlet (No. 6). This servlet gets an ontology URI and returns the results of JUNG queries in RDF. JUNG, which stands for Java Universal Network/Graph framework, is a software library that is used for visualizing and analyzing network graphs. Afterwards, AKTiveRank carries out analysis of each ontology candidate to find out which one is most relevant to the provided product names. The results of the analysis, which is a ranking of the retrieved ontologies, will be returned to the marketplace platform as a text file that contains the ontology URIs as well as their total similarity distances.

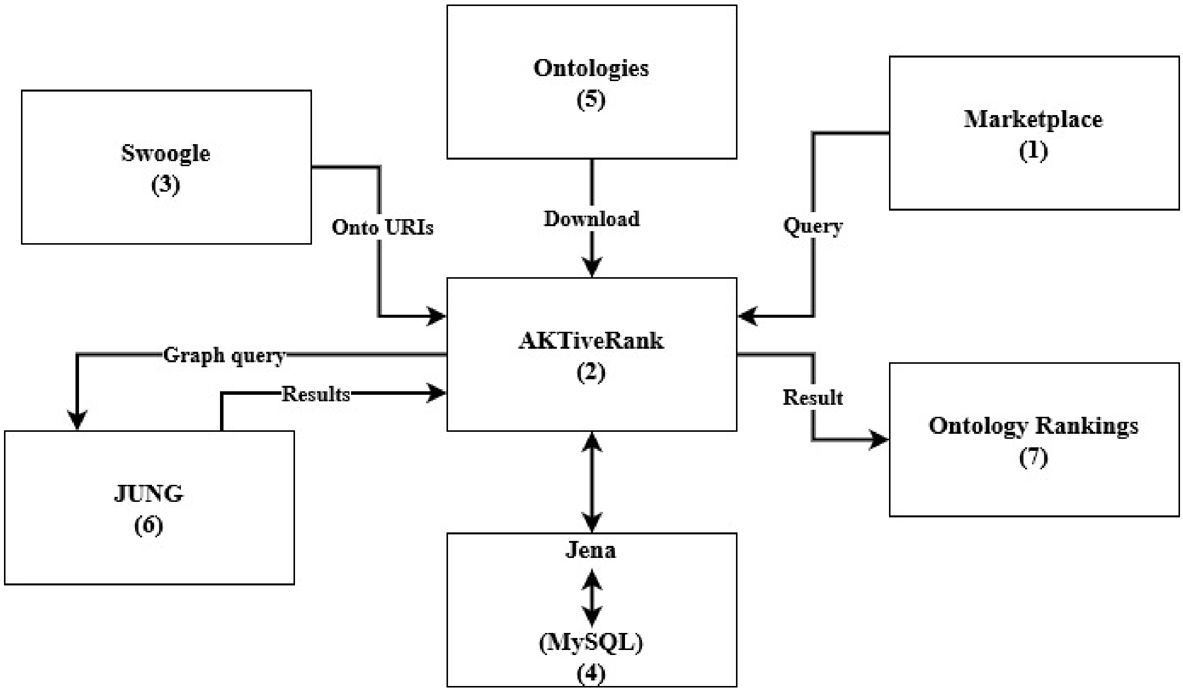


Figure 6.5. AKTiveRank Architecture (Alani and Brewster, 2006)

6.3.1.1 The Ranking Measures

The AKTiveRank uses four types of measures or assessments for each ontology to measure/assess the similarity distance (Alani and Brewster, 2006). For this reason, each measure undergoes separate calculation. After all the measures are computed for a particular ontology, the resulting values are merged to get the total rank of the ontology (Alani and Brewster, 2006).

For completeness and clarity, we reproduce the metrics and equations for capturing Class Match Measure (CMM), Semantic Similarity Measures (SSM), Betweenness Measure, Density Measure and the Total Score. It is important to note the formulations and metrices in Section 6.3.1.2 to Section 6.3.1.6 have been taken from Alani and Brewster (2006) and have been presented in these sections so that the reader can understand the working of our proposed algorithm for Context-based trust inferencing. The proposed algorithm for Context-based trust inferencing which is presented in Section 6.3.2 builds up on the Services Ontology (presented in Section 6.2) and the metrics presented in Section 6.3.1.2 to Section 6.3.1.6.

6.3.1.2 Class Match Measure (CMM)

The Class Match Measure (CMM) is used for evaluating the coverage of an ontology for the product name provided (Alani and Brewster, 2006). The AKTiveRank checks for classes in each ontology which has labels that matches a product name either partially or exactly. Any ontology that comprises of all the products names will definitely have higher scores than the others (Alani and Brewster, 2006). In addition, exact matches are better than partial matches. For instance, if a search is carried out for "Apple" and "iPhone", then an ontology that has two classes labelled exactly as the products names will have a higher score in the match measure than an ontology that containing partially matching classes like "Samsung" and "iPad"

Definition 1. *Let T be a set of products names and C[o] be a set of classes in ontology o.*

$$E(o,T) = \sum_{c \in C[o]} \sum_{t \in T} I(c,t) \qquad (1)$$

$$I(c,t) = \begin{cases} 1 & : \ if\ label(c) = t \\ 0 & : \ if\ label(c) \neq t \end{cases} \qquad (2)$$

$$P(o,T) = \sum_{c \in C[o]} \sum_{t \in T} J(c,t) \qquad (3)$$

$$J(c,t) = \begin{cases} 1 & : \ if\ label(c)\ contains\ t \\ 0 & : \ if\ label(c)\ not\ contain\ t \end{cases} \qquad (4)$$

Equation 6.1: Class match measure equation (Alani and Brewster, 2006) where *P*(*o, T*) and *E*(*o, T*) represent the number of classes of ontology o with labels matching any of the product names t partially or exactly, respectively.

$$CMM(o, \tau) = \alpha E(o, T) + \beta P(o, T) \qquad (5)$$

Equation 6.2: Class match measure equation (Alani and Brewster, 2006)

where *CMM*(*o, τ*) is the Class Match Measure for ontology o with respect to products names *τ* . *α* and *β* represent the exact matching and partial matching weight factors respectively. If *α>β,* then, exact matching is favored over partial matching.

6.3.1.3 Density Measure

While searching for a 'great' representation of a specific concept, it is expected that a certain degree of detail in the representation of the knowledge as it concerns that concept will be obtained (Alani and Brewster, 2006). It can be how well the concept is further specified (that is, the number of subclasses) or the number of siblings or the number of attributes that relates to the concept. The Density Measure (DEM) takes charge of all this. The primary role of DEM is to approximate the informational-content or representational-density of classes and subsequently, the level of knowledge detail. Density calculations are limited to the numbers of superclasses, subclasses, siblings and relations (Alani and Brewster, 2006).

Definition 2. *Let S = {S*1*, S*2*, S*3*, S*4*} = {relations*[*c*]*, superclasses*[*c*]*, subclasses*[*c*]*, siblings*[*c*]*}*

$$dem(c) = \sum_{i=1}^{4} w_i |S_i| \qquad (6)$$

$$DEM(o) = \frac{1}{n} \sum_{i=1}^{n} dem(c) \qquad (7)$$

Equation 6.3: Density measure equation (Alani and Brewster, 2006)

Where *n* = *E*(*o, T*)+*P*(*o, T*) which is the number of matched classes in ontology *o and wi* is a weight factor set to a default value of 1.

6.3.1.4 Semantic Similarity Measure (SSM)

Similarity measures are highly popularly used in information retrieval systems for the provision of better ranking for query results. Since ontologies can be visualized as semantic graphs of relations and concepts, similarity measures can be used to explore the conceptual graphs (Alani and Brewster, 2006). Resnik (1999) did this by applying a similarity measure to WordNet to resolve ambiguities. Also, Rada *et al.* (1989) introduced another common-feature based similarity which is known as the shortest-path measure. According to Rada *et al.* (1989), the more relationship that objects have in common, the closer the objects will be in an ontology. Rada *et al.* (1989) utilized the measure in the ranking of biomedical documents in a semantic knowledgebase.

SSM calculates the how close classes matching the products names are in an ontology. This is based on a simple principle, ontologies with concepts far away from each other are less likely to represent the knowledge in a compact and coherent way.

Definition 3. *Let ci, cj ∈ {classes*[*o*]*}, and* $c_i \overset{p}{\rightsquigarrow} c_j$ *is a path p ∈ P of paths between classes ci and cj*

$$ssm(c_i, c_j) = \begin{cases} \frac{1}{length(min_{p \in P}\{c_i \overset{p}{\rightsquigarrow} c_j\})} & : \ if\ i \neq j \\ 0 & : \ if\ i = j \end{cases} \qquad (8)$$

$$SSM(o) = \frac{1}{n} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ssm(c_i, c_j) \qquad (9)$$

Equation 6.4: Semantic similarity measure equation (Alani and Brewster, 2006)

where n is the number of matched classes.

6.3.1.5 Betweenness Measure

According to Freeman (1977), part of the algorithm provided by JUNG is Betweenness. It is used in calculating the number of shortest paths which pass through each node in the graph (Alani and Brewster, 2006). Nodes occurring on many shortest paths between other nodes will have higher betweenness value. It is assumed that any class that has a high betweenness value in an ontology is central to that ontology. In addition, ontologies with classes that are more central will have a higher score.

Definition 4. *Let ci, cj ∈{classes*[*o*]*}, ci and cj are any two classes in ontology o, bem*(*c*) *is the BEtweenness Measure for class c, and* C[*o*] *is the set of class in ontology o.*

$$bem(c) = \sum_{c_i \neq c_j \neq c \in C[o]} \frac{\sigma_{c_i c_j}(c)}{\sigma_{c_i c_j}} \qquad (10)$$

Equation 6.5: Betweenness measure equation (Alani and Brewster, 2006)

where *σcicj* is the shortest path from *ci* to *cj,* and *σcicj* (*c*) is the number of shortest paths from *ci* to *cj* passing through *c*.

$$BEM(o) = \frac{1}{n} \sum_{k=1}^{n} bem(c_k) \qquad (11)$$

Equation 6.6: Betweenness measure equation (Alani and Brewster, 2006)

where BEM(o) is the average Betweenness value for ontology *o, and n* is the number of matched classes in ontology *o*.

6.3.1.6 Total Score

Once the four measures discussed above are applied to all the ontologies returned, the total score of a semantic distance can be computed (Alani and Brewster, 2006). This is done by accumulating all the measures' values considering the weight of each measure, which helps to find out the degree of importance of each measure during ranking.

Definition 5. Let *M = {M*[1], .., *M*[*i*], *M*[4]*} = {CMM, DEM, SSM, BEM}*, O is the set of ontologies to rank, and *wi* is a weight factor.

$$Score(o \in O) = \sum_{i=1}^{4} w_i \frac{M[i]}{\max_{1 \leq j \leq |O|} M[j]} \qquad (12)$$

Equation 6.7: Total score equation (Alani and Brewster, 2006)

6.3.2 Algorithm for Context-Based Inference

Having found out the semantic distance between two concepts, the next procedure is to apply the following algorithm to carry out context-based trust inference. Given that we are using Transportation Ontology, the proposed algorithm for carrying out context-based inference is provided below.

As shown in the algorithm, there are three input variables which include the *source product (value), inferred product and degree of similarity*. However, there is only one variable for the output which is *inferred product (value)*.

It should be noted that the similarity value is between 0 to 1 and it is the total score obtained from Equation 6.7. Therefore, using the algorithm, if semantic similarity between *Source product* and *Inferred Product* is less than *Degree of Similarity*, it means the required degree of confidence has not been met. Therefore, the algorithm cannot compute the inferred value.

However, if semantic similarity between *Source product* and *Inferred Product* is greater than *Degree of Similarity,* then, the algorithm will go ahead to execute the computation of the *inferred value* using the *degree of similarity* and *source product* value.

Pseudocode for context inference:

```
Begin Context Inference Algorithm
Inputs:
Variable 1: Source product (value)
Variable 2: Inferred product
Variable 3: Degree of similarity
Output:
Variable 4: Inferred product (value)
If semantic similarity between Source product and Inferred Product is less than Degree of Similarity
(
      Print "Cannot compute inferred value with the required degree of confidence"
)
Exit;
Else
    (
      Inferred product (value) = Source product (value) * Degree of similarity
          Print Inferred product (value)
)
Exit
End Context Inference Algorithm
```

6.4 Validation

To validate the solution proposed for this objective. The algorithm was used to infer the reputation of iPhone 12 (a product that is yet to be released). This was done by taking the reputation values of iPhone 10 and 11 and finding their average. The average value was then used to find the semantic similarity. To carry out this validation, we already know the feature sets and capabilities such as FaceTime - making calls and messaging - and based on this, we used the AKTiveRank algorithm to compute the closeness and to predict the trust value of *iPhone 12*. Figure 6.7 to Figure 6.10 below are showing the *inferred product* process occurred in marketplace.

The steps involved in computing the new trust value in a given context of a given service provider is as below:

1. Get source and inferred products details from *FarMed*: First, all the previous ratings that are related to the service provider and are stored in *FarMed* are requested.
2. Compute degree of similarity between source and inferred products: Using AKTiveRank algorithm, we compute the similarity between the source products and the inferred product.
3. Verify that the rules regarding confidence levels have been met: During the validation purposes, we assume that the minimum level of similarity (confidence) between the *source* and *inferred products* to carry out the trust inferencing is 80%. Therefore, in the step we check if this rule has been met.
4. Compute the reputation score of inferred product: By taking the average *of source products* reputation scores that are met the rule in previous step (iii), we deduce the reputation score of the *inferred product*.
5. The updated overall trust value of the *inferred product* is then stored in *FarMed*.

The logical working of the above steps is shown pictorially in Figure 6.6 below.

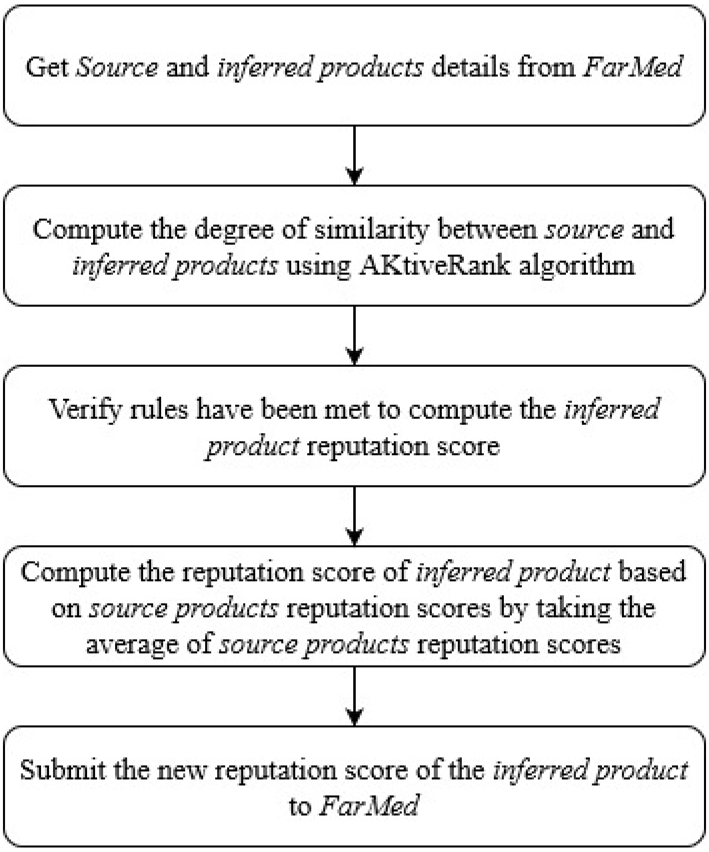
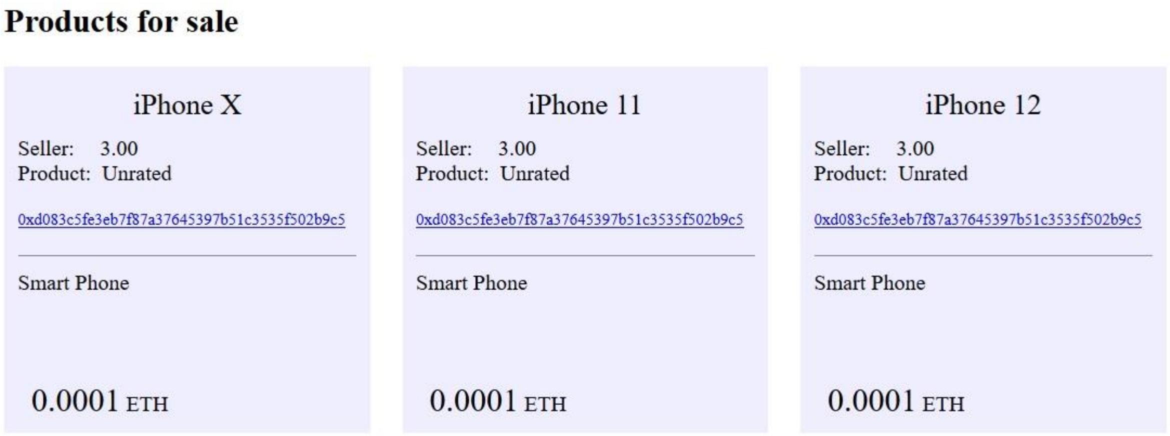

Get *Source* and *inferred products* details from *FarMed*
Compute the degree of similarity between *source* and *inferred products* using AKtiveRank algorithm
Verify rules have been met to compute the *inferred product* reputation score
Compute the reputation score of *inferred product* based on *source products* reputation scores by taking the average of *source products* reputation scores
Submit the new reputation score of the *inferred product* to *FarMed*

Figure 6.6. Overview of steps in computing the trust value of a service provider in a given context



Products for sale

iPhone X

Seller: 3.00
Product: Unrated

0xd083c5fe3eb7f87a37645397b51c3535f502b9c5

Smart Phone

0.0001 ETH

iPhone 11

Seller: 3.00
Product: Unrated

0xd083c5fe3eb7f87a37645397b51c3535f502b9c5

Smart Phone

0.0001 ETH

iPhone 12

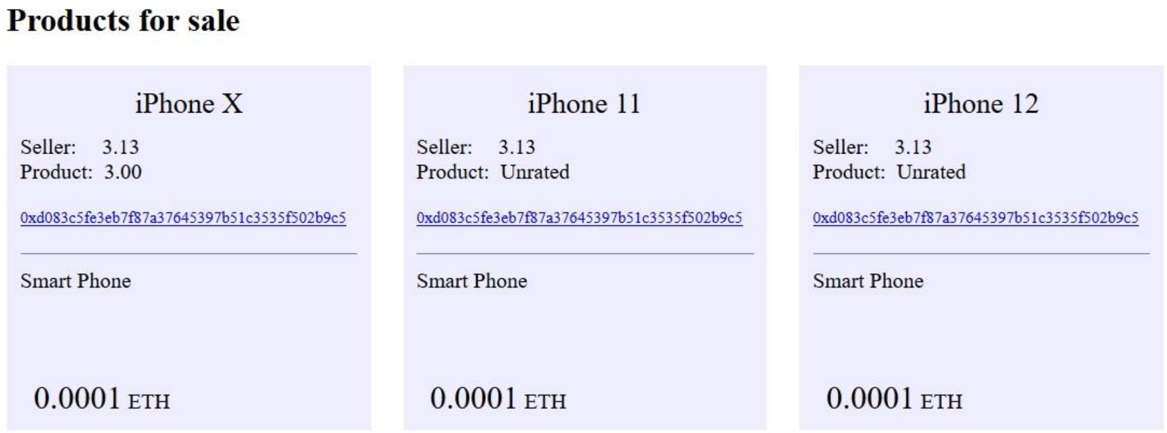
Seller: 3.00
Product: Unrated

0xd083c5fe3eb7f87a37645397b51c3535f502b9c5

Smart Phone

0.0001 ETH

Figure 6.7. Screenshot of the listed products before rating



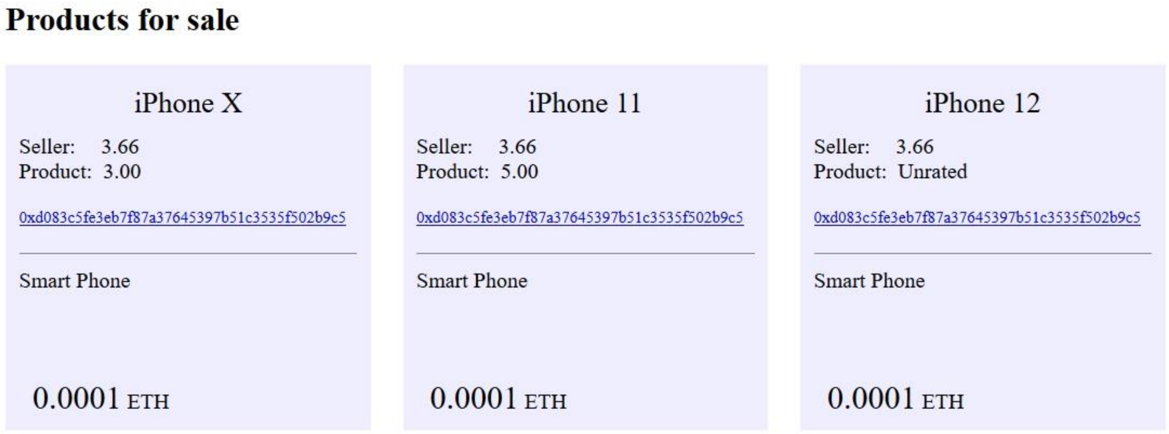
Products for sale

iPhone X
Seller: 3.13
Product: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 11
Seller: 3.13
Product: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 12
Seller: 3.13
Product: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

Figure 6.8. Screenshot of the listed products after rating *iPhone X* product



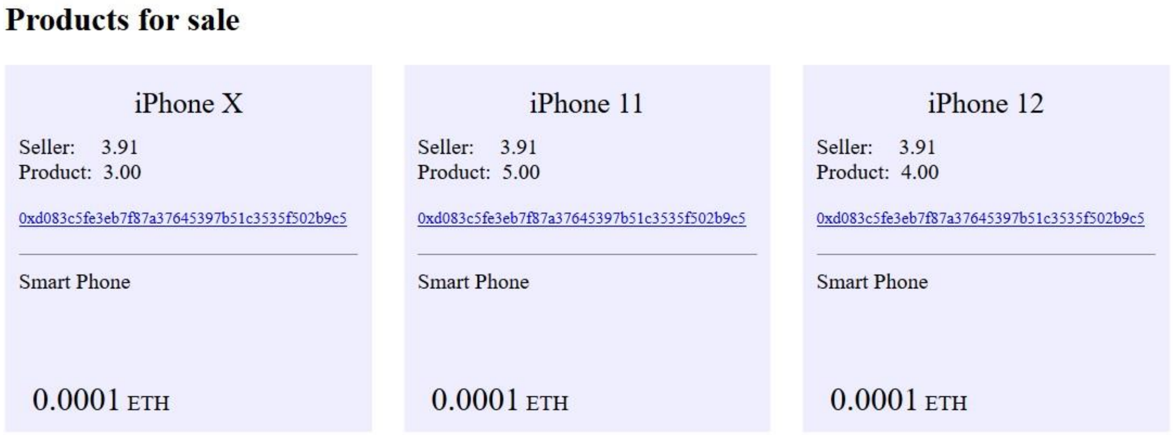
Products for sale

iPhone X
Seller: 3.66
Product: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 11
Seller: 3.66
Product: 5.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 12
Seller: 3.66
Product: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

Figure 6.9. Screenshot of the listed products after rating *iPhone 11* product



Products for sale

iPhone X
Seller: 3.91
Product: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 11
Seller: 3.91
Product: 5.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

iPhone 12
Seller: 3.91
Product: 4.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Smart Phone
0.0001 ETH

Figure 6.10. Screenshot of the listed products after inferring the reputation score for *iPhone 12* product

Figure 6.7 shows a list of products for sale before rating. In Figure 6.8, iPhone X is rated 3.00 while iPhone 11 is given a rating of 5.00 in Figure 6.9. Based on the ratings of iPhone X and iPhone 11, we used the proposed inferencing algorithm to predict the rating of iPhone 12 which is gotten to be 4.00 as shown in Figure 6.10.

6.5 Conclusion

In this paper, we used service ontology and AKTiveRank to intelligently model and infer the trust value of a service provider in a specific context. Our proposed algorithm for context-based trust inferencing is heavily grounded in the service ontology and the four similarity measures proposed in AKTiveRank.

Based on the ontology and similarity measures, we proposed an algorithm for trust inference with a degree of confidence. Using a case study, we demonstrated the working of the proposed algorithm.

In the next chapter, we propose a novel method for trading or auctioning reputation value termed as Reputation Auction Service (RAS).

Chapter 7 : RAS for Transferring of Reputation among Service Providers

7.1 Introduction

This chapter presents *Reputation Auction Service* (RAS), which is a service within *FarMed* to transfer the reputation score from one service provider to another. This service addresses the research objective four. RAS will provide intelligent mechanisms for bootstrapping of new service providers in the reputation-based economy. In addition, this chapter will demonstrate the implementation of the proposed solution for RAS.

This chapter is described as follows: in Section 7.2, we explain in detail the bootstrapping of our proposed solution or approach. In Section 7.3, the algorithms required to carry out the reputation trading and are outlined and the proposed solution is in detail in a stepwise manner.

In Section 7.4, we outline and explain the steps for testing and validation purposes. Subsequently, in Section 7.5 the implementation of the proposed solution for reputation trading is evaluated and discussed in detail. Section 7.6 concludes this chapter.

7.2 Reputation Auction Service

The central thesis of the proposed method (RAS) is that service providers with high or excess reputation value will sell part of their reputation ratings to new service providers. Subsequently, in return the new service provider will return a greater reputation value than the one lent in instalments. All of this is carried out through RAS.

Reputation Auction Service (RAS) can be used by new service providers (S_n) with little (or no reputation value) to purchase reputation value from other service providers (S_l) and bootstrap themselves in the reputation-based economy. The new service providers who have little or zero reputation score can buy *reputation* from other service providers who have a high reputation score Using the concept of *reputation trading* the new or bootstrapped service providers (S_n) who have no reputation value or little reputation value will be able to gather a reputation score. These service providers will transact using that reputation score. Subsequently, using the RAS framework the bootstrapped service provider (S_n) will return the reputation score to reputation lending service provider (S_l).

The benefit for the reputation lending service providers, who sell part of their reputation score is commercial. The reputation score that they receive back from the bootstrapped service provider (R_s) is greater than the original reputation score that they had lent R. In other words, $R_s > R$. This provides the reputation lending service provider the motivation to lend to others. Figure 7.1 shows an example of reputation scores before and after the auction.

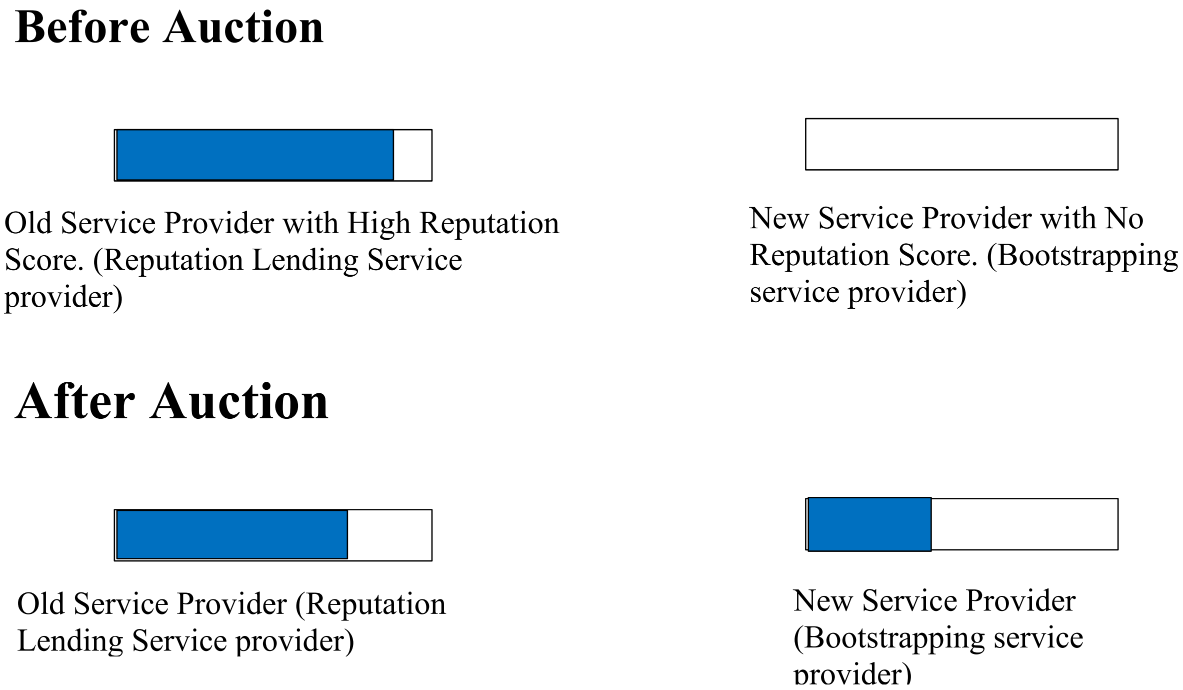

Before Auction
Old Service Provider with High Reputation Score. (Reputation Lending Service provider)
New Service Provider with No Reputation Score. (Bootstrapping service provider)
After Auction
Old Service Provider (Reputation Lending Service provider)
New Service Provider (Bootstrapping service provider)

Figure 7.1. Example of reputation scores before and after the auction

7.3 Algorithm for Reputation Computation

The working of the Reputation Auction Service is based on an algorithm built following on similar procedures as in Chapters 5 and 6.

The formulations used in the Fire-Star algorithm (CMS, 2019) are as follows:

$$Rating = {(P_1 n_1 + P_2 n_2 + P_3 n_3 + \cdots + P_f n_f)} / {(n_1 + n_2 + n_3 + \cdots + n_f)}$$

Equation: 7.1

$$Rating = {\sum Tn} / {\sum n}$$

……Equation: 7.2

where P is the rating value of product which can be 1, 2, 3, 4 or 5 only due to smart contracts structure limitations.

n is the number of rating(s) having the same rating value

In Figure 7.2, we present an overview of the process followed by the Reputation Auction Service. First, the request for reputation purchase is submitted by the bootstrapping service provider. After this, once a reputation lending service provider has been identified, an agreement is made between the transacting service providers. Then, the new reputation score for each service provider is computed and published in the marketplace.

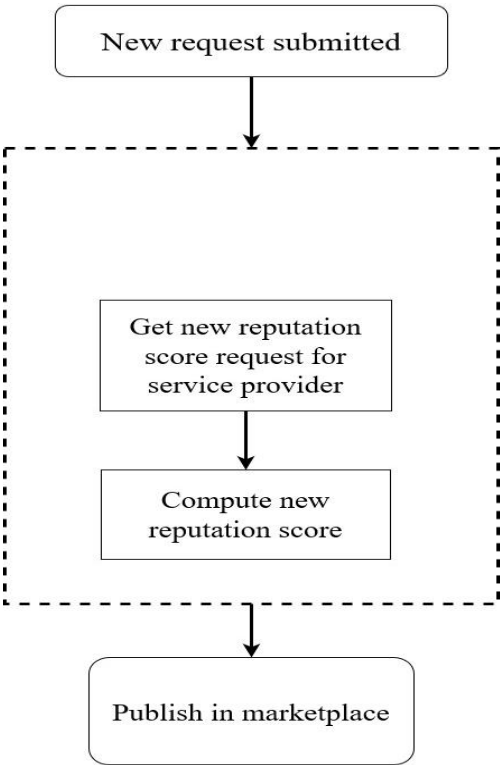

New request submitted
Get new reputation score request for service provider
Compute new reputation score
Publish in marketplace

Figure 7.2. Overview of Reputation Auction Service

7.3.1 Submission of New Reputation Request

The way the Reputation Auction Service works starts with the bootstrapping service provider making a request for reputation score from RAS. The RAS will in turn identify a reputation lending service provider to be involved in this transaction. Thereafter, the following steps are carried out to verify and approve the request. The overview of the entire process is also outlined in Figure 7.3.

i. <u>Step 1: Request to RAS for reputation score</u>: The bootstrapping service provider will contact RAS and request a reputation score from another service provider. After the notification, this new request will be stored in a smart contract.

ii. <u>Step 2: Decision on reputation trading</u>: The reputation lending service provider has the right to take a decision on whether to accept the request or otherwise. The reputation lending service provider will inform the bootstrapping party about its decision. If the reputation lending service provider decision is No, then the bootstrapping service provider involved in this reputation trading service will get notification of the rejection, thereby bringing the process to an end. However, if the reputation lending service provider's decision is Yes, the process of trading reputation value continues. Either way, the bootstrapping party is notified.

iii. <u>Step 3: Agreement between the bootstrapping party and reputation lending party:</u> The reputation lending party provides the desired payback score and the schedule of installments expected from the bootstrapping party. The bootstrapping party needs to take a decision on this as both parties need to agree. This is discussed further in Section 7.3.2.

iv. <u>Step 4: Storing the agreement in Smart Contract</u>: The terms agreed on by both the parties are stored in a smart contract and an acknowledgement is received.

v. <u>Step 5: Execution of reputation score computation</u>: Finally, the computation of the updated reputation scores for each of the service providers (i.e. for both the bootstrapping party and reputation lending party) is carried out

Figure 7.3 below outlines the stepwise working of the reputation trading in RAS in a methodological manner.

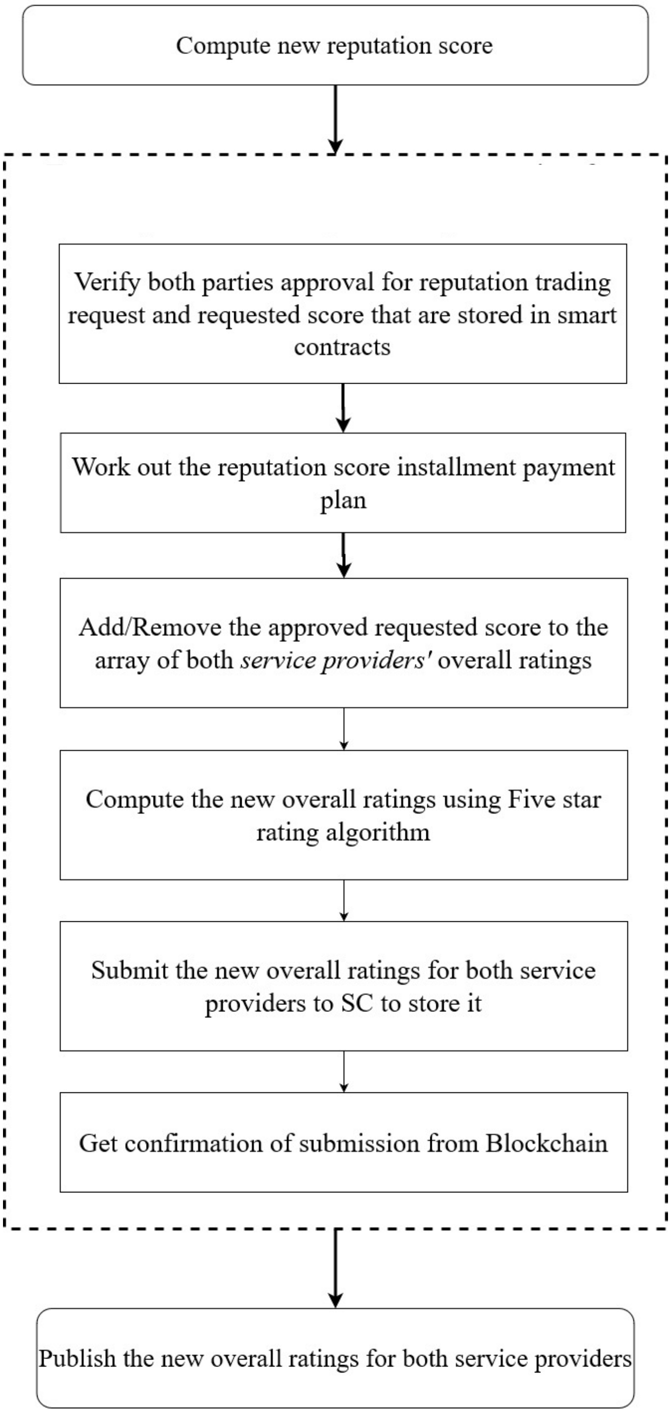

Compute new reputation score
Verify both parties approval for reputation trading request and requested score that are stored in smart contracts
Work out the reputation score installment payment plan
Add/Remove the approved requested score to the array of both *service providers'* overall ratings
Compute the new overall ratings using Five star rating algorithm
Submit the new overall ratings for both service providers to SC to store it
Get confirmation of submission from Blockchain
Publish the new overall ratings for both service providers

Figure 7.3. Overview of the submission process for submitting new lending request

7.3.2 Dividing and Storing the Score Installments in Smart Contracts

7.3.2.1 Verification of Payback Score and Schedule for Score Installments

In this section we explain in detail how the reputation score payback is computed. The payback score and schedule for score installments need to be verified by Blockchain. The following process outlines the mechanism for determining the payback score and schedule. The overview of the entire process is outlined in Figure 7.4.

i. <u>Step 1: Get the plan from the reputation score lending party</u>: The reputation lending party can specify and outline its desired payback score and the schedule for the score payback installments. In this step, the reputation score lending party can specify its reputation score payback plan.

ii. <u>Step 2: Validation for eligibility of both the parties</u>: In this step, the RAS carried our verification on both the parties to ensure that they are eligible for reputation trading. In our implementation, we have set the following eligibilities:
   a. <u>Eligibility 1</u>: The reputation requesting party should be a new party and either have zero or minimum reputation score
   b. <u>Eligibility 2</u>: The reputation lending party should be a well-established party whose reputation score should be greater than that of the reputation requesting party.

Other eligibilities can be added depending on the usage context. RAS is modular to be able to allow other eligibilities to be added.

iii. <u>Step 3: Storing the agreement</u>: Once it has been confirmed that both the parties meet the eligibilities (Step 2) then an agreement is submitted and stored in the smart contract. The submission of the agreement is confirmed by the blockchain. Finally, the computation of the reputation score can be executed.

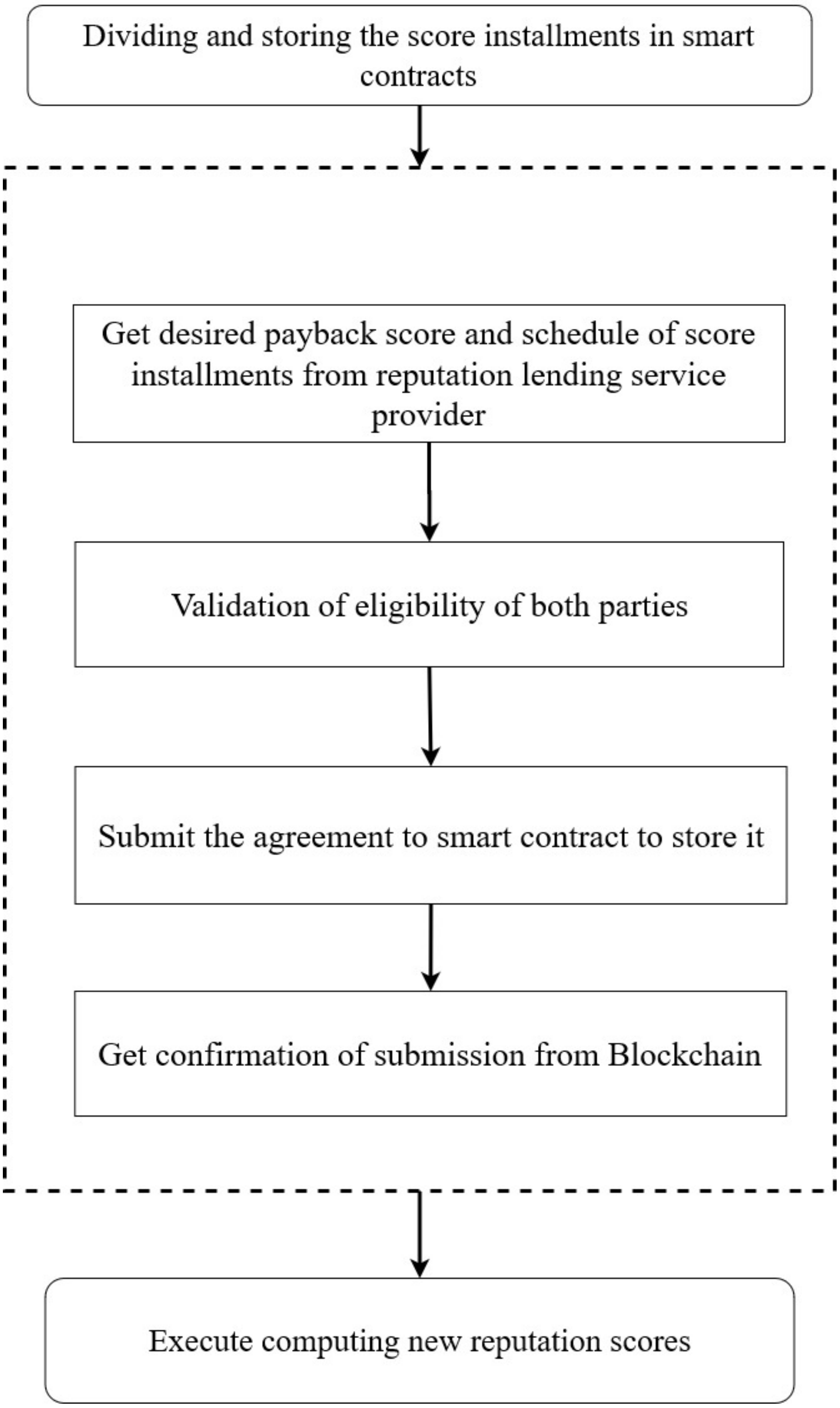

Dividing and storing the score installments in smart contracts
Get desired payback score and schedule of score installments from reputation lending service provider
Validation of eligibility of both parties
Submit the agreement to smart contract to store it
Get confirmation of submission from Blockchain
Execute computing new reputation scores

Figure 7.4. Verification of Payback Score and Score Instalments

7.3.2.2 Execution and payment of Reputation Scores in Installments

In this phase, the reputation scores are paid back in instalments. The following steps are carried out to execute this phase. They have also been outlined in Figure 7.5:

i. Step 1: Payment of reputation score by the bootstrapped entity: The bootstrapped entity will be paying back the reputation score to the reputation lending entity through RAS. For each instalment payment from the bootstrapped entity, a smart contract will be executed.

ii. Step 2: Confirm submission and update scores: Once the reputation score (in the form of instalments) is paid by the bootstrapped entity, a smart contract is executed to store the values in *FarMed*. Subsequently, updated reputation ratings are computed using the five-star algorithm for both parties.

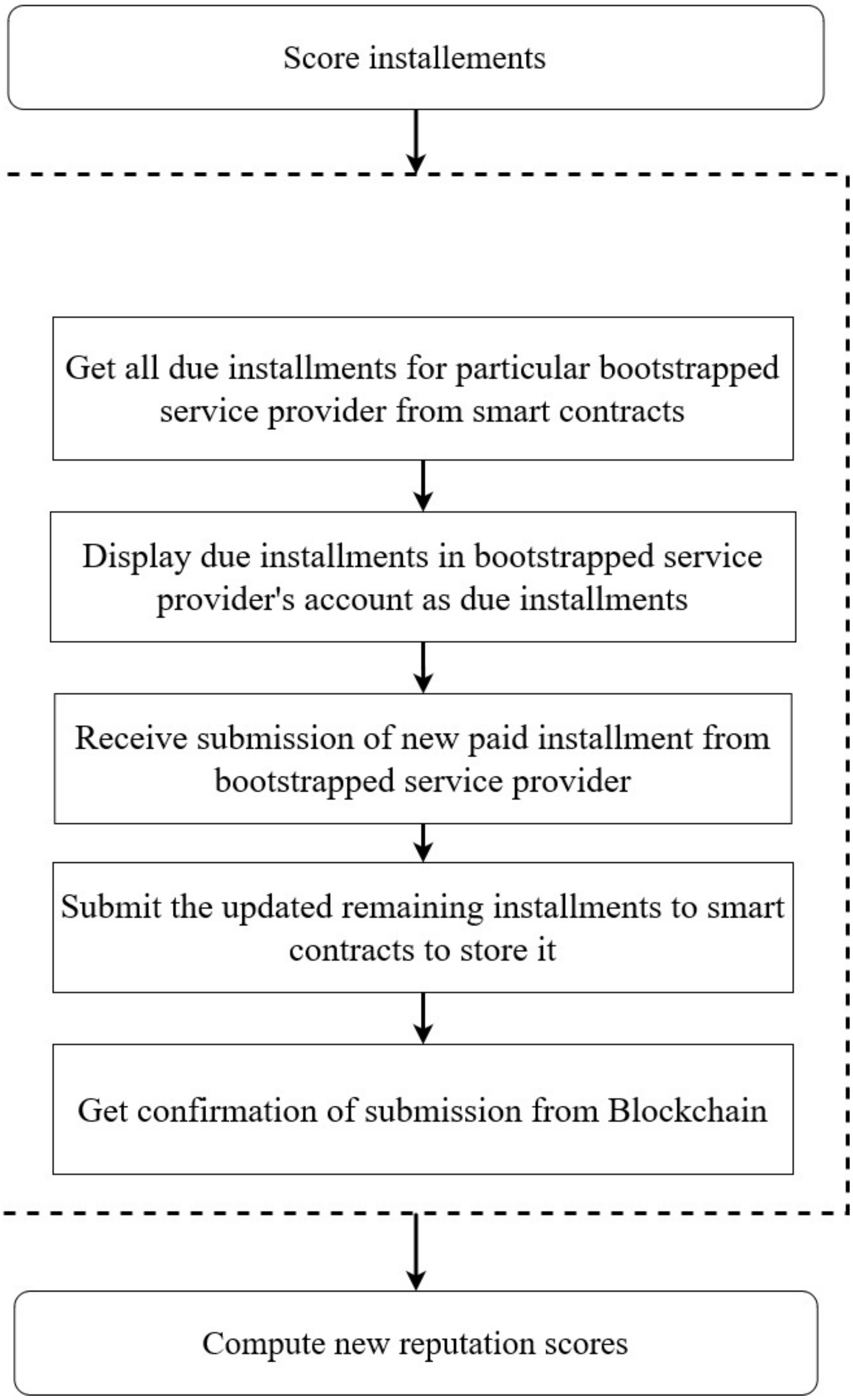

Score installements
Get all due installments for particular bootstrapped service provider from smart contracts
Display due installments in bootstrapped service provider's account as due installments
Receive submission of new paid installment from bootstrapped service provider
Submit the updated remaining installments to smart contracts to store it
Get confirmation of submission from Blockchain
Compute new reputation scores

Figure 7.5. Execution of due installments

7.4 Testing and Validation of the Solution

In this section we present an overview of the prototype system used to validate the working of RAS. Furthermore, in this section we can test the prototype system and validate it as well.

In the developed prototype for RAS, the reputation scores for the service providers who have recently joined *FarMed* is zero. These service providers are seeking to purchase reputation scores from other existing service providers within *FarMed*. The bootstrapping service providers post requests for reputation scores in RAS along with details such as the service provider's name and ID. When a reputation lending entity decides to accept one of the offers, it will enter into an agreement with the bootstrapping service provider on the pay back schedule. Once an agreement is reached and the reputation lending entity has approved the request, then it's reputation score will be deducted by an amount equal to the agreed value. Similarly, the bootstrapping service provider is credited with the reputation score by RAS. This reputation trading process is registered in the Blockchain using smart contracts.

7.4.1 Steps in the prototype system

In order to achieve its job, the *FarMed* service carries out computations of reputation trading at two different levels, namely local computations and Blockchain computations. In local computations, all the reputation trading computations are executed locally (i.e., in the local time). In our proposed method the reputation trading computations processing happens at the local level. However, the reputation scores are stored and retrieved in the Blockchain. The steps for both the local level and Blockchain level computation are explained below in Section 7.4.1.1 and Section 7.4.1.2 respectively

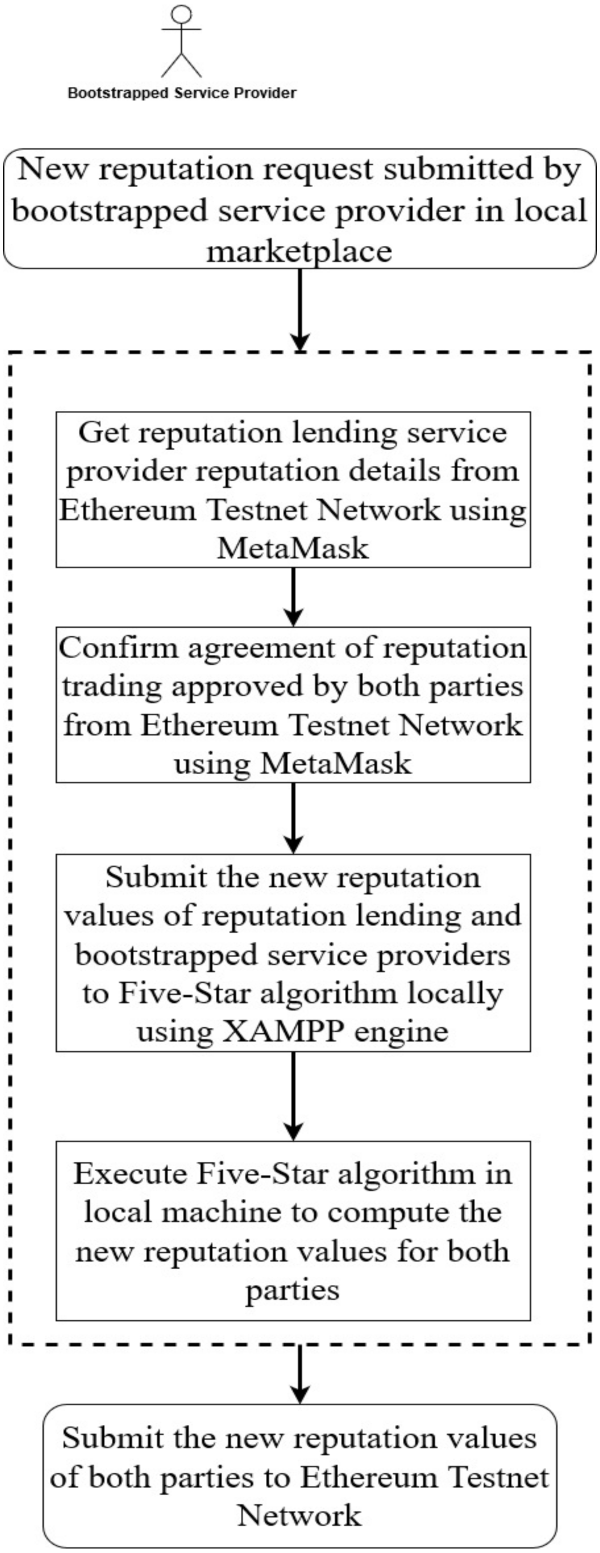

Bootstrapped Service Provider
New reputation request submitted by bootstrapped service provider in local marketplace
Get reputation lending service provider reputation details from Ethereum Testnet Network using MetaMask
Confirm agreement of reputation trading approved by both parties from Ethereum Testnet Network using MetaMask
Submit the new reputation values of reputation lending and bootstrapped service providers to Five-Star algorithm locally using XAMPP engine
Execute Five-Star algorithm in local machine to compute the new reputation values for both parties
Submit the new reputation values of both parties to Ethereum Testnet Network

Figure 7.6. Flow chart of reputation trading process in local marketplace

7.4.1.1 Steps in local-based reputation trading

After the request for reputation score is submitted, the following four steps are carried out:

- Step 1: Local marketplace acquires the reputation score of the reputation lending entity from Ethereum Testnet Network using Metamask.
- Step 2: Local marketplace acquires the reputation trading agreement from Ethereum Testnet Network using Metamask to verify that both parties are agreed.
- Step 3: Run XAMPP engine and prepare it for new execution. This step runs in local machine automatically.
- Step 4: Execute the Five-Star algorithm in local machine to compute the new reputation values of both parties. After that, the new values of both the transacting service providers are stored in the Ethereum Testnet Network.

7.4.1.2 Steps in Blockchain-based reputation trading

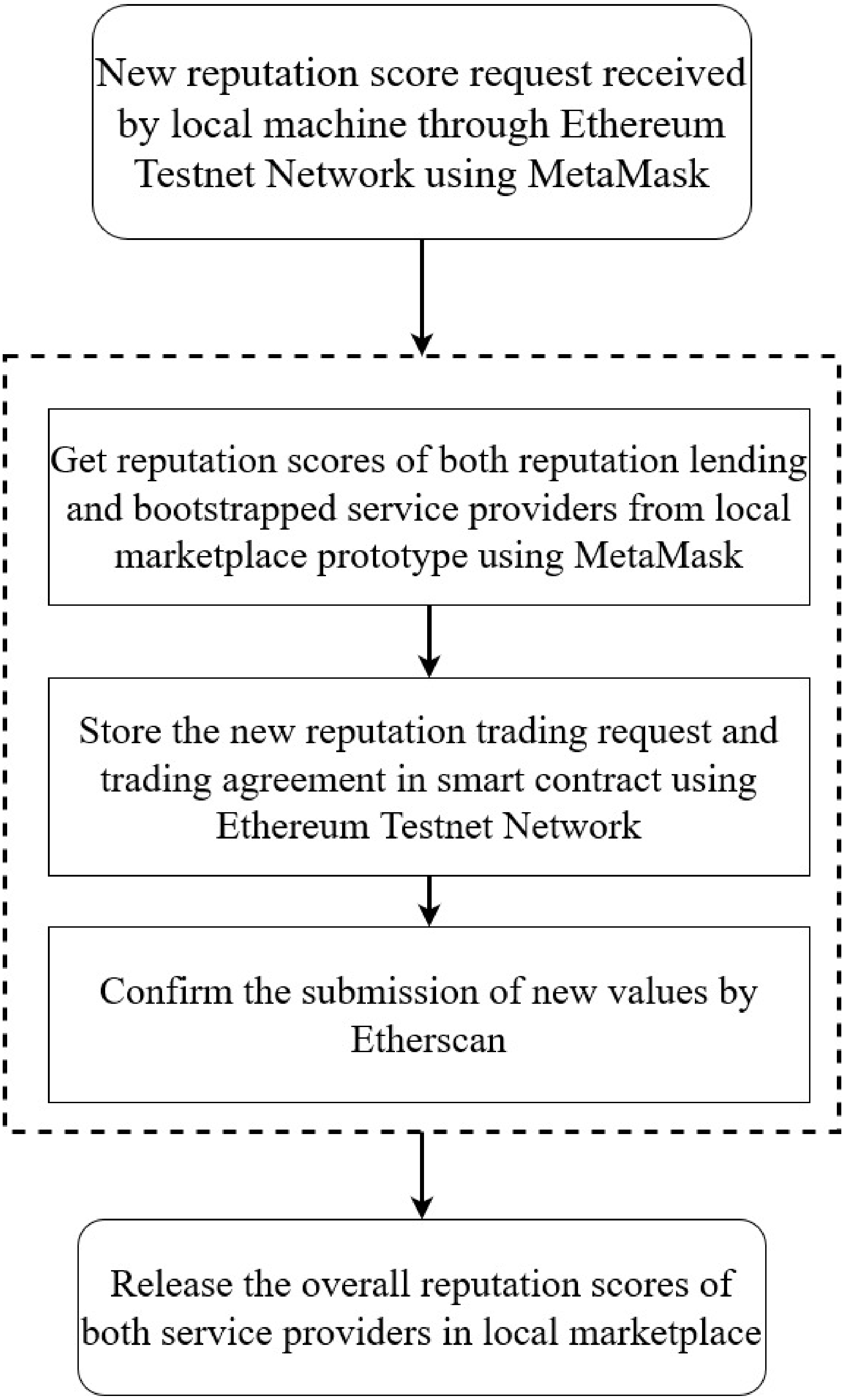

New reputation score request received by local machine through Ethereum Testnet Network using MetaMask
Get reputation scores of both reputation lending and bootstrapped service providers from local marketplace prototype using MetaMask
Store the new reputation trading request and trading agreement in smart contract using Ethereum Testnet Network
Confirm the submission of new values by Etherscan
Release the overall reputation scores of both service providers in local marketplace

Figure 7.7. Flow chart of reputation trading process in blockchain

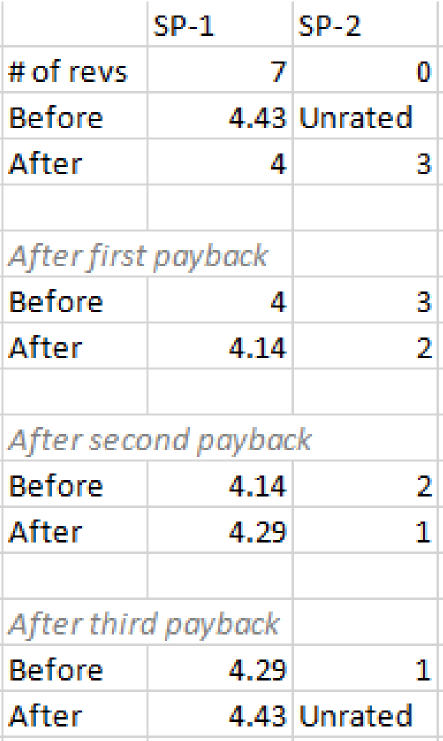
After new reputation scores of both service providers have been calculated using Five-Star algorithm in local machine, the process goes through the following three steps at the Blockchain layer as follows:

- Step 1: Get the new or updated calculated reputation scores of both service providers using MetaMask.
- Step 2: Store the reputation scores and reputation trading agreement in smart contract using Ethereum Testnet Network.
- Step 3: Getting confirmation of submission from blockchain using Etherscan.

After the above steps, the new reputation values of both service providers are released in the local marketplace. The overview of the above steps is also shown in Figure 7.7.

7.5 Evaluation Results and Discussion

In this section we present the results obtained using our prototype and discuss the findings. The results of the implementation are presented as follows:



	SP-1	SP-2
# of revs	7	0
Before	4.43	Unrated
After	4	3
After first payback		
Before	4	3
After	4.14	2
After second payback		
Before	4.14	2
After	4.29	1
After third payback		
Before	4.29	1
After	4.43	Unrated

Figure 7.8. Results obtained from implementing the Reputation Auction Service

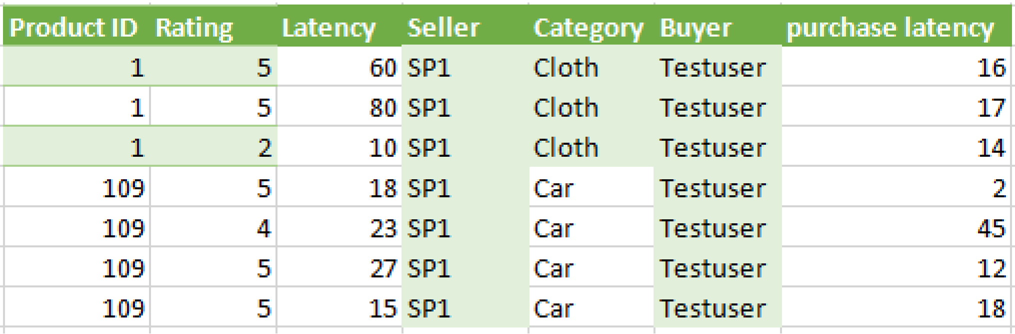
As can be seen in the above Figure 7.8, SP-1 and SP-2 are two service providers. However, SP-1 is an old service provider who has enough reputation score while SP-2 is a new service provider who is willing to get started in the marketplace by purchasing some reputation score from SP-1. At the moment SP-2 has a score of 0.

Furthermore, as shown in the Figure 7.8, SP-1 requested a reputation score of 3 from SP-2 and after the request was approved, SP-1's gets a rating of 3 while SP-2's rating reduced from 4.43 to 4. Figure 7.8 further illustrates the installments process involved in SP-1 paying back the reputation score. This was done to verify the accuracy of our solution and it was based on the assumption that both parties have no new rating from a client for the entire period.

Moreover, as can be seen in the Figure 7.8, after the first payback, SP-2's reputation value reduced from 3 to 2 while SP-1's reputation value increased from 4 to 4.14. After the second payback, SP-2's reputation value reduced from 2 to 1 while SP-1's reputation value increased from 4.14 to 4.29. Finally, after the third payback, SP-2's reputation value reduced from 1 to 0 (unrated) while SP-1's reputation value increased from 4.29 to 4.43.

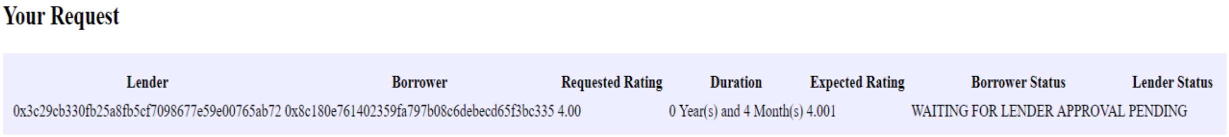
The values after the complete payback to SP-1 by SP-2 is shown under the section of "*After third payback*". It needs to be noticed that the values are exactly the same as the ones prior to the reputation trading. This confirms the accuracy and effectiveness of the Reputation Auction Service that we have built. An important point to note is that in Figure 7.8, other transactions other than reputation lending and payback SP-1 and SP-2 have not been carried out. When SP-2 is able to carry out transactions other than reputation lending, it would be able to pay additional reputation value in the form of profit to the SP-1. Furthermore, SP-1 would be able to carry out interactions with other service providers.

Figure 7.9 shows the latency of the reputation auction service. As in the case of solutions to Objective 2 and 3, the results obtained during the implementation of RAS also took a few seconds. Figures: 7.10 and 7.11 show the reputation trading dashboard.



Product ID	Rating	Latency	Seller	Category	Buyer	purchase latency
1	5	60	SP1	Cloth	Testuser	16
1	5	80	SP1	Cloth	Testuser	17
1	2	10	SP1	Cloth	Testuser	14
109	5	18	SP1	Car	Testuser	2
109	4	23	SP1	Car	Testuser	45
109	5	27	SP1	Car	Testuser	12
109	5	15	SP1	Car	Testuser	18

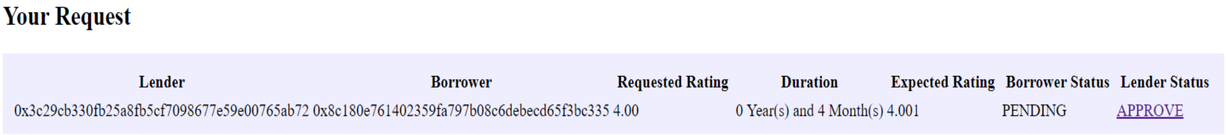
Figure 7.9. Latency for the Reputation Auction Service



Your Request

Lender	Borrower	Requested Rating	Duration	Expected Rating	Borrower Status	Lender Status
0x3c29cb330fb25a8fb5cf7098677e59e00765ab72	0x8c180e761402359fa797b08c6debecd65f3bc335	4.00	0 Year(s) and 4 Month(s)	4.001	WAITING FOR LENDER APPROVAL	PENDING

Figure 7.10 (See description below)



Your Request

Lender	Borrower	Requested Rating	Duration	Expected Rating	Borrower Status	Lender Status
0x3c29cb330fb25a8fb5cf7098677e59e00765ab72	0x8c180e761402359fa797b08c6debecd65f3bc335	4.00	0 Year(s) and 4 Month(s)	4.001	PENDING	APPROVE

Figure 7.11. (See description below)

Figures 7.10 and Figure 7.11 are showing the dashboard of Reputation Auction Service

Figures 7.11 and 7.12 show the dashboard of the Reputation Auction Service which provides details like Lender ID, Borrower's ID, Requested Rating, Duration, Expected Rating, Borrower Status and Lender Status. As shown in the two figures, the amount of Requested Rating is 4.00 while the expected rating after the request has been approved is 4.001. The difference between the two figures is that in Figure 7.11, the request is still pending (yet to be approved) while in Figure 7.12, the request has been approved by the lender.

7.6 Conclusion

In this chapter, we discussed the working of RAS in detail. This corresponds to the objective 4 of this research and will provide a mechanism to transfer reputation among service providers. The detailed working of RAS was explained in a methodological manner.

To validate the working of RAS, we set up a small prototype for RAS. Based on the results obtained, we were able to conclude that RAS is able to provide a framework for service providers to exchange reputation values with each other in a reliable manner. The Blockchain layer of *FarMed* enables and provides reliability to reputation lending transactions in RAS.

In the next chapter, we explain the working of our developed prototype for all the objectives in this thesis.

Chapter 8 : Prototype Working and Demonstration

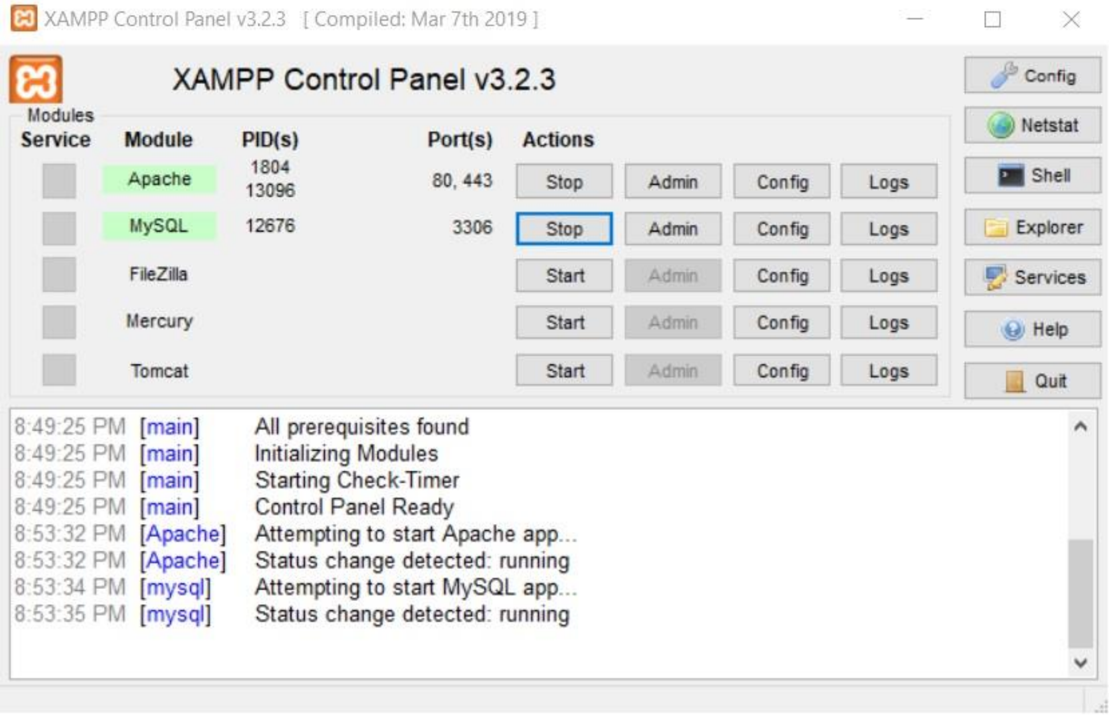
8.1 Introduction

In the last chapter, we proposed and discussed an overview of our proposed solution for reputation trading between the members of *FarMed* with a view to bootstrap new service providers. In this chapter, building on the research solutions presented in Chapter 5, Chapter 6 and Chapter 7, we present the working of our prototype. Using screenshots and pictures we demonstrate in a step-by-step manner the prototype setup which includes the local setup and blockchain setup. Furthermore, we also demonstrate the working of the proposed solutions for reputation computation (in section 8.3) and reputation transfer (in section 8.4) using the developed prototype.

8.2 Prototype Setup

In order to achieve its job, the *FarMed* service carries out computations at two different levels, namely local computations and Blockchain computations. In local computations, all the reputation computations are executed locally (i.e., in the local time). In our proposed method all the reputation computations and reputation processing happen at the local level. However, the reputation scores are stored and retrieved in the Blockchain. The steps for both the local level and Blockchain level computation are explained below in Section 8.2.1 and Section 8.2.2 respectively.

8.2.1 Description of the Local machine Setup

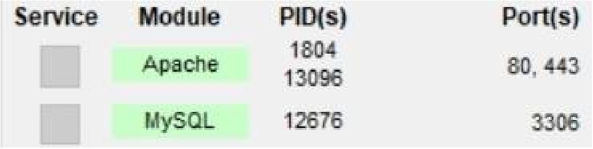


Service	Module	PID(s)	Port(s)
☐	Apache	1804 13096	80, 443
☐	MySQL	12676	3306
☐	FileZilla		
☐	Mercury		
☐	Tomcat		

```
8:49:25 PM  [main]      All prerequisites found
8:49:25 PM  [main]      Initializing Modules
8:49:25 PM  [main]      Starting Check-Timer
8:49:25 PM  [main]      Control Panel Ready
8:53:32 PM  [Apache]    Attempting to start Apache app...
8:53:32 PM  [Apache]    Status change detected: running
8:53:34 PM  [mysql]     Attempting to start MySQL app...
8:53:35 PM  [mysql]     Status change detected: running
```

Figure 8.1. XAMPP Control Panel interface

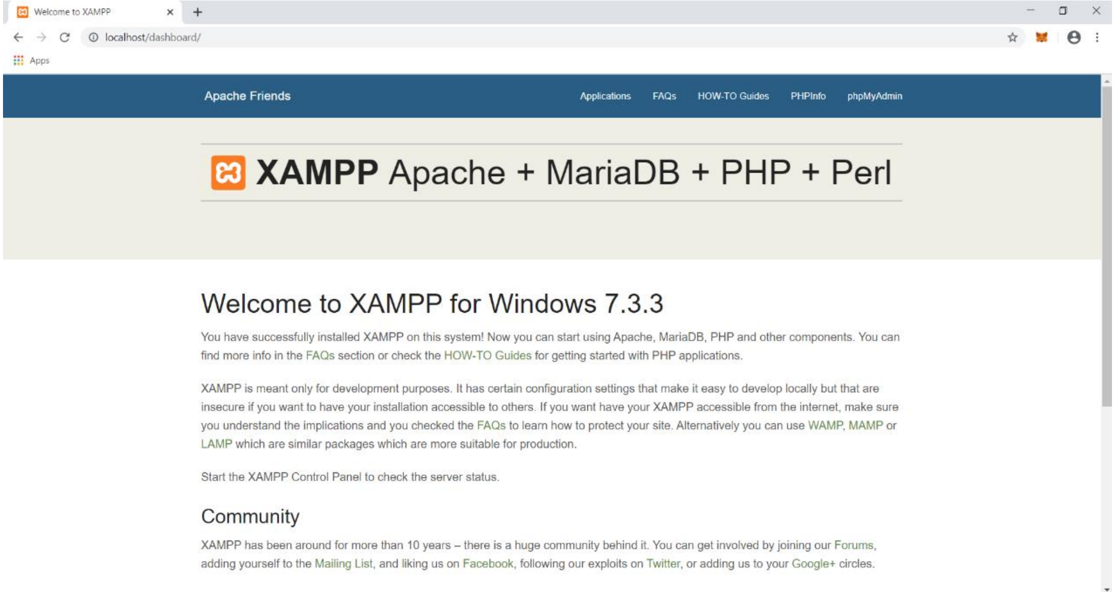
XAMPP software (shown above) [https://www.apachefriends.org] is used to run the machine as localhost.



Service	Module	PID(s)	Port(s)
☐	Apache	1804 13096	80, 443
☐	MySQL	12676	3306

Figure 8.2. Apache and MySQL services activated

XAMPP uses Apache [https://www.apachefriends.org] to run the localhost by opening port 80 and 443, and MySQL [https://www.apachefriends.org] is used to run MySQL database to handle storing the data in Apache by opening port 3306.



Welcome to XAMPP for Windows 7.3.3

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

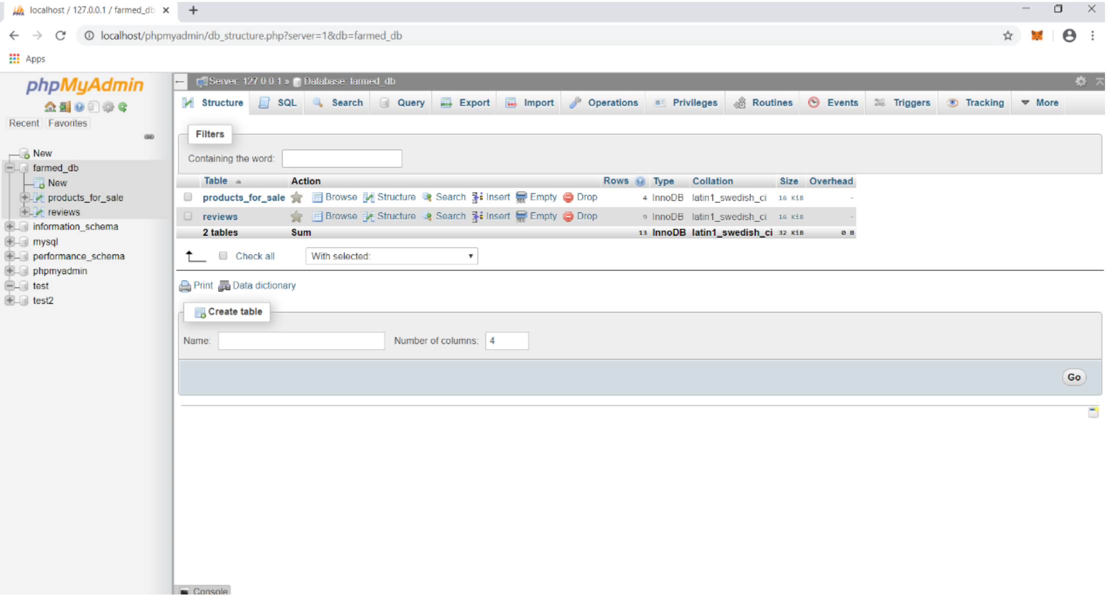
Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

Figure 8.3. Machine is running localhost successfully

We interact with XAMPP software by opening Chrome browser and entering either localhost or 127.0.0.1 to open the home page directory of localhost



Containing the word:

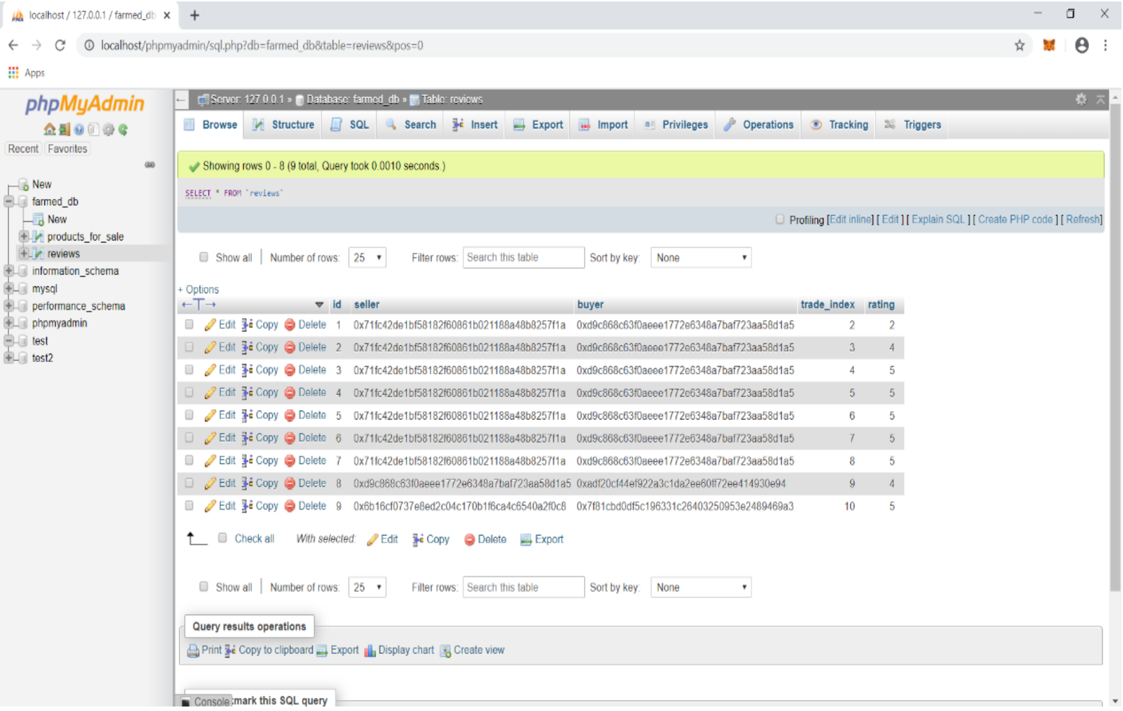
Table	Action	Rows	Type	Collation	Size	Overhead
products_for_sale	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
reviews	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 KiB	-
2 tables	Sum	13	InnoDB	latin1_swedish_ci	32 KiB	0 B

With selected:

Name: Number of columns: 4

Figure 8.4. phpMyAdmin interface

In the next step, using phpMyAdmin we built a new database named "farmed_db" with two tables, named Reviews and Products_for_sale. The purpose of the database is to simulate the local marketplace and prepare it to be linked with blockchain to be in the top of it to store reputation values.


localhost / 127.0.0.1 / farmed_db
localhost/phpmyadmin/sql.php?db=farmed_db&table=reviews&pos=0
Apps
phpMyAdmin
Recent Favorites
New
farmed_db
New
products_for_sale
reviews
information_schema
mysql
performance_schema
phpmyadmin
test
test2
Server: 127.0.0.1 » Database: farmed_db » Table: reviews
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Console
Bookmark this SQL query

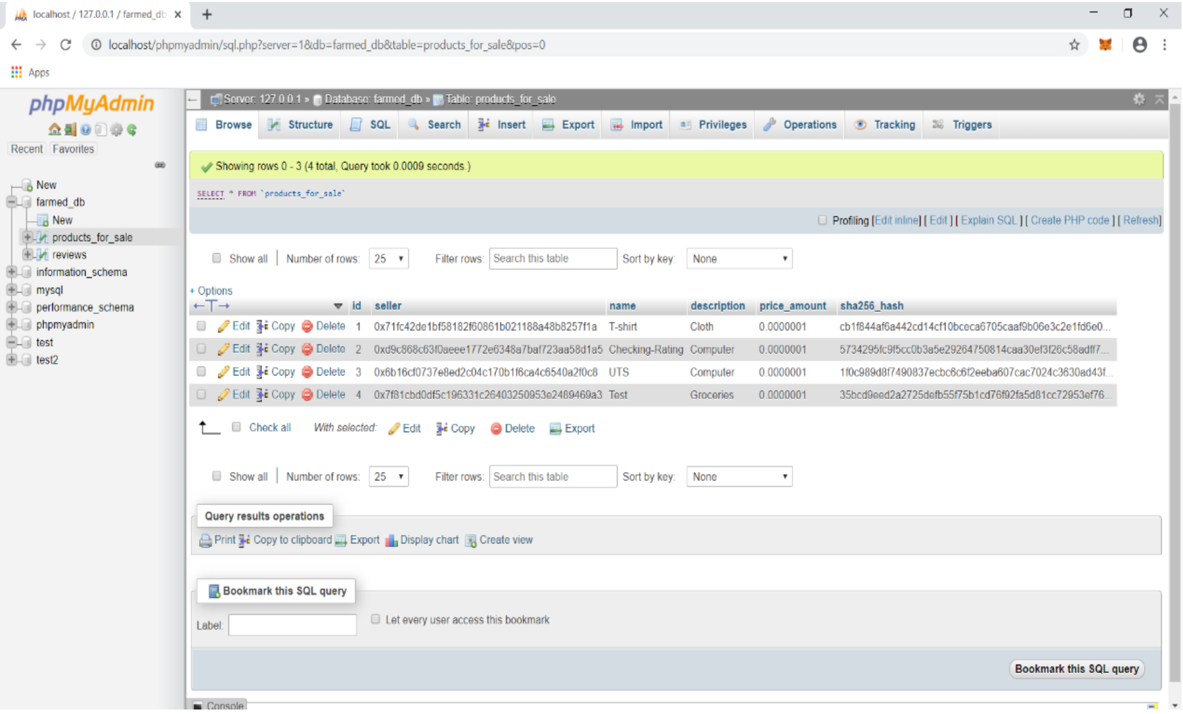
Showing rows 0 - 8 (9 total, Query took 0.0010 seconds.)

```
SELECT * FROM `reviews`
```

id	seller	buyer	trade_index	rating
1	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	2	2
2	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	3	4
3	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	4	5
4	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	5	5
5	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	6	5
6	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	7	5
7	0x71fc42de1bf58182f60861b021188a48b8257f1a	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	8	5
8	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	0xadf20cf44ef922a3c1da2ee60ff72ee414930e94	9	4
9	0x6b16cf0737e8ed2c04c170b1f6ca4c6540a2f0c8	0x7f81cbd0df5c196331c26403250953e2489469a3	10	5

Figure 8.5. Review table in database

As can be seen in Figure 8.5 above, the Review table consists of ID, Seller, Buyer, Trade_index, and Rating fields. Seller and Buyer fields are to be populated based on addresses from the blockchain.


localhost / 127.0.0.1 / farmed_db
localhost/phpmyadmin/sql.php?server=1&db=farmed_db&table=products_for_sale&pos=0
Apps
phpMyAdmin
Recent Favorites
New
farmed_db
New
products_for_sale
reviews
information_schema
mysql
performance_schema
phpmyadmin
test
test2
Server: 127.0.0.1 » Database: farmed_db » Table: products_for_sale
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query
Console

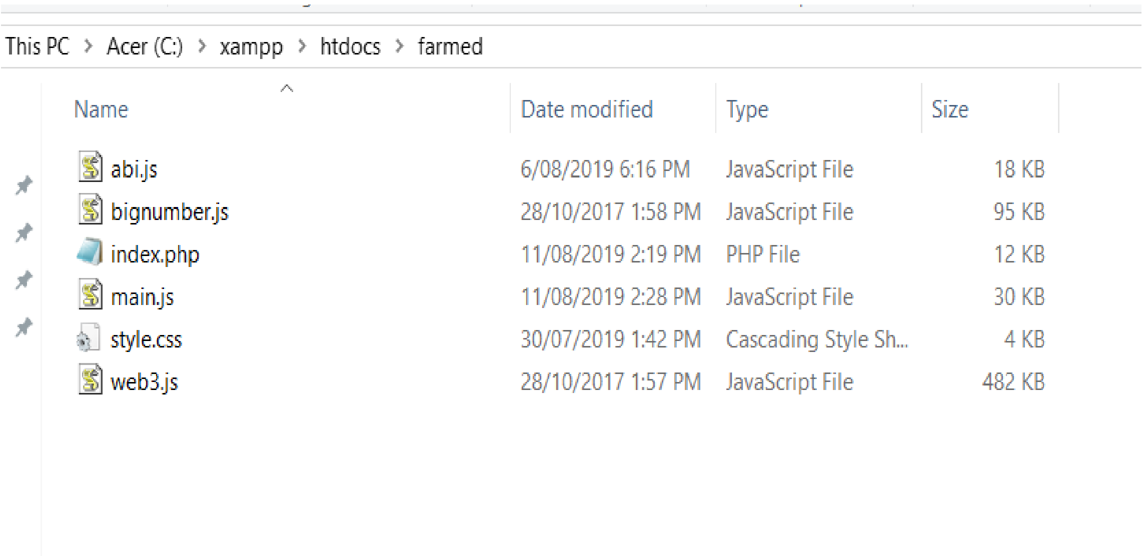
Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT * FROM `products_for_sale`
```

id	seller	name	description	price_amount	sha256_hash
1	0x71fc42de1bf58182f60861b021188a48b8257f1a	T-shirt	Cloth	0.0000001	cb1f844af6a442cd14cf10bceca6705caaf9b06e3c2e1fd6e0...
2	0xd9c868c63f0aeee1772e6348a7baf723aa58d1a5	Checking-Rating	Computer	0.0000001	5734295fc9f5cc0b3a5e29264750814caa30ef3f26c58adff7...
3	0x6b16cf0737e8ed2c04c170b1f6ca4c6540a2f0c8	UTS	Computer	0.0000001	1f0c989d8f7490837ecbc6c6f2eeba607cac7024c3630ad43f...
4	0x7f81cbd0df5c196331c26403250953e2489469a3	Test	Groceries	0.0000001	35bcd9eed2a2725defb55f75b1cd76f92fa5d81cc72953ef76...

Figure 8.6. Product_for_sale table in database

The Product_for_sale table lists all the products that are currently available for sale on the marketplace. It is comprised of the following attributes: Product ID, Seller, name, description, Price_amount, and sha256_hash fields. The Sha256_hash field is used to store the hash of each product in the local database.



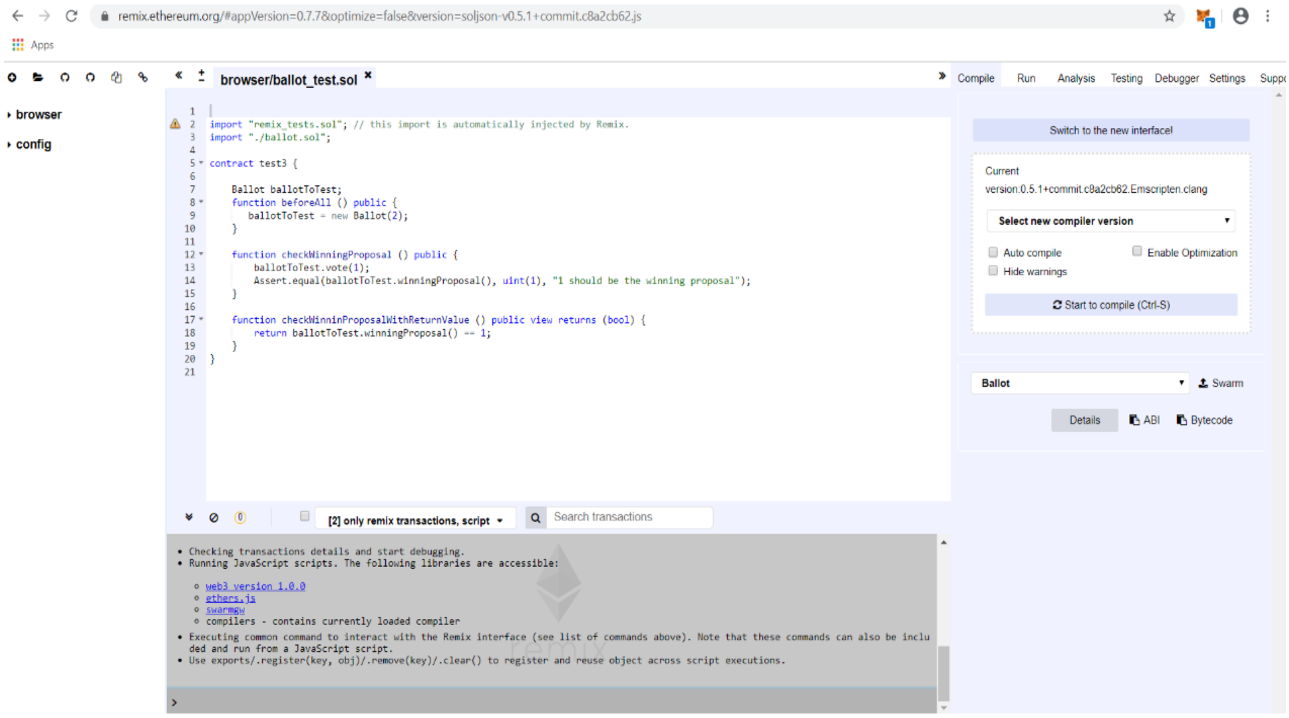
This PC › Acer (C:) › xampp › htdocs › farmed

Name	Date modified	Type	Size
abi.js	6/08/2019 6:16 PM	JavaScript File	18 KB
bignumber.js	28/10/2017 1:58 PM	JavaScript File	95 KB
index.php	11/08/2019 2:19 PM	PHP File	12 KB
main.js	11/08/2019 2:28 PM	JavaScript File	30 KB
style.css	30/07/2019 1:42 PM	Cascading Style Sh...	4 KB
web3.js	28/10/2017 1:57 PM	JavaScript File	482 KB

Figure 8.7. Marketplace files

We stored all the marketplace files into localhost home directory to execute it. We built a simulated marketplace using PHP and JavaScript for the system and MySQL for the database.

8.2.2 Description of the Blockchain Setup



remix.ethereum.org/#appVersion=0.7.7&optimize=false&version=soljson-v0.5.1+commit.c8a2cb62.js
Apps
browser
config
browser/ballot_test.sol
Compile
Run
Analysis
Testing
Debugger
Settings
Switch to the new interface!
Select new compiler version
□ Auto compile
□ Enable Optimization
□ Hide warnings
Start to compile (Ctrl-S)
Ballot
Swarm
Details
ABI
Bytecode
[2] only remix transactions, script
Search transactions

Current
version:0.5.1+commit.c8a2cb62.Emscripten.clang

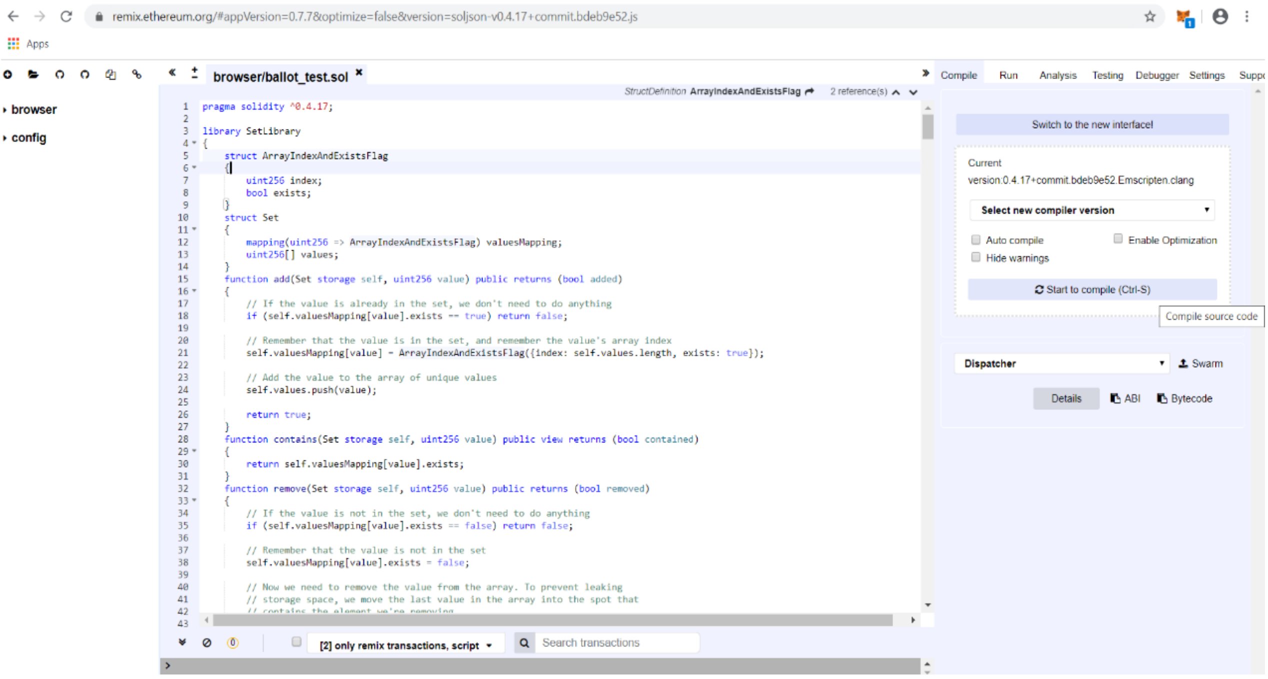
```

import "remix_tests.sol"; // this import is automatically injected by Remix.
import "./ballot.sol";

contract test3 {

    Ballot ballotToTest;
    function beforeAll () public {
       ballotToTest = new Ballot(2);
    }

    function checkWinningProposal () public {
        ballotToTest.vote(1);
        Assert.equal(ballotToTest.winningProposal(), uint(1), "1 should be the winning proposal");
    }

    function checkWinninProposalWithReturnValue () public view returns (bool) {
        return ballotToTest.winningProposal() == 1;
    }
}

```

```
• Checking transactions details and start debugging.
• Running JavaScript scripts. The following libraries are accessible:

   ○ web3 version 1.0.0
   ○ ethers.js
   ○ swarmgw
   ○ compilers - contains currently loaded compiler
• Executing common command to interact with the Remix interface (see list of commands above). Note that these commands can also be inclu
  ded and run from a JavaScript script.
• Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.

>
```

Figure 1.8. Remix – Ethereum IDE in browser interface

In order to write smart contract code using Solidity language and compile it, we used Remix – Ethereum IDE editor and compile [https://remix.ethereum.org/].

Remix is used to write the smart contract and to effect deployment on the Ethereum Virtual Machine (EVM). After deployment with Remix, we went to Etherscan to check the status of deployment, if the deployment is successful or if it failed. Refer to Figure 8.9 below for more details.


remix.ethereum.org/#appVersion=0.7.7&optimize=false&version=soljson-v0.4.17+commit.bdeb9e52.js
Apps
browser
config
browser/ballot_test.sol
StructDefinition ArrayIndexAndExistsFlag
2 reference(s)
Compile
Run
Analysis
Testing
Debugger
Settings
Switch to the new interface!
Select new compiler version
Start to compile (Ctrl-S)
Compile source code
Dispatcher
Swarm
Details
ABI
Bytecode
[2] only remix transactions, script
Search transactions

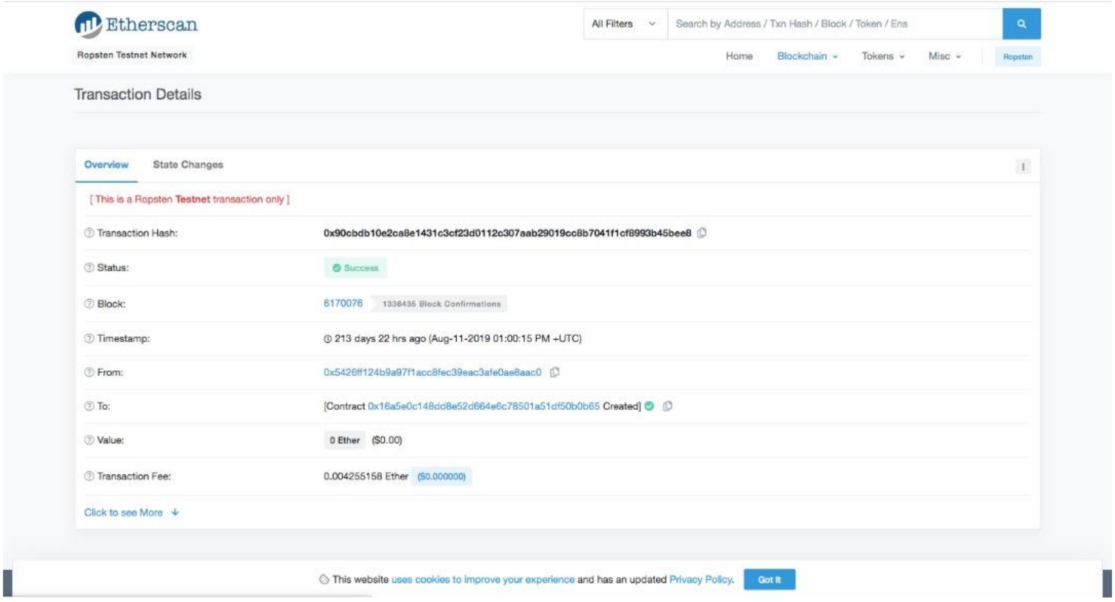
Current
version:0.4.17+commit.bdeb9e52.Emscripten.clang

□ Auto compile
□ Enable Optimization
□ Hide warnings

```
pragma solidity ^0.4.17;

library SetLibrary
{
    struct ArrayIndexAndExistsFlag
    {
        uint256 index;
        bool exists;
    }
    struct Set
    {
        mapping(uint256 => ArrayIndexAndExistsFlag) valuesMapping;
        uint256[] values;
    }
    function add(Set storage self, uint256 value) public returns (bool added)
    {
        // If the value is already in the set, we don't need to do anything
        if (self.valuesMapping[value].exists == true) return false;

        // Remember that the value is in the set, and remember the value's array index
        self.valuesMapping[value] = ArrayIndexAndExistsFlag({index: self.values.length, exists: true});

        // Add the value to the array of unique values
        self.values.push(value);

        return true;
    }
    function contains(Set storage self, uint256 value) public view returns (bool contained)
    {
        return self.valuesMapping[value].exists;
    }
    function remove(Set storage self, uint256 value) public returns (bool removed)
    {
        // If the value is not in the set, we don't need to do anything
        if (self.valuesMapping[value].exists == false) return false;

        // Remember that the value is not in the set
        self.valuesMapping[value].exists = false;

        // Now we need to remove the value from the array. To prevent leaking
        // storage space, we move the last value in the array into the spot that
        // contains the element we're removing
```

Figure 8.9. Remix IDE compiler

In Figure 8.9, we compiled the smart contract code to execute the code on the EVM.


Etherscan
Ropsten Testnet Network
All Filters
Search by Address / Txn Hash / Block / Token / Ens
Home
Blockchain
Tokens
Misc
Ropsten
Overview
State Changes
Click to see More
This website uses cookies to improve your experience and has an updated Privacy Policy.
Got It

Transaction Details

[This is a Ropsten **Testnet** transaction only]

Transaction Hash:	0x90cbdb10e2ca8e1431c3cf23d0112c307aab29019cc8b7041f1cf8993b45bee8
Status:	Success
Block:	6170076 1336435 Block Confirmations
Timestamp:	213 days 22 hrs ago (Aug-11-2019 01:00:15 PM +UTC)
From:	0x5426ff124b9a97f1acc8fec39eac3afe0ae8aac0
To:	[Contract 0x16a5e0c148dd8e52d664e6c78501a51df50b0b65 Created]
Value:	0 Ether ($0.00)
Transaction Fee:	0.004255158 Ether ($0.000000)

Figure 8.10. Etherscan - EVM public web interface - Etherscan

In Figure 8.9, we successfully created smart contract using Ethereum Remix IDE. In Figure 8.10, Etherscan [https://etherscan.io/] shows details of our deployment on the EVM. It also shows details like smart contract address and the transaction hash of the transaction registered on the blockchain. As shown in Figure 8.10, Etherscan showed that the deployment is successful, and it also shows the details of the deployment includes the smart contract address that is needed to interact with the smart contract on the EVM.

Eventually, Etherscan and Remix communicate with the EVM, Remix sends transactions to the EVM, while Etherscan reads the status of the transaction from the EVM.

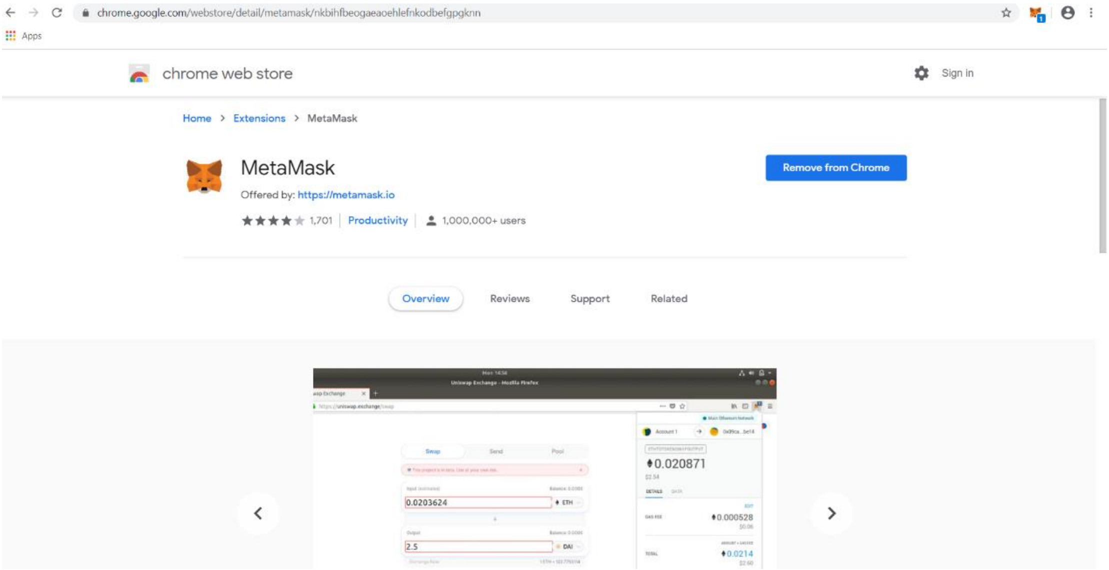
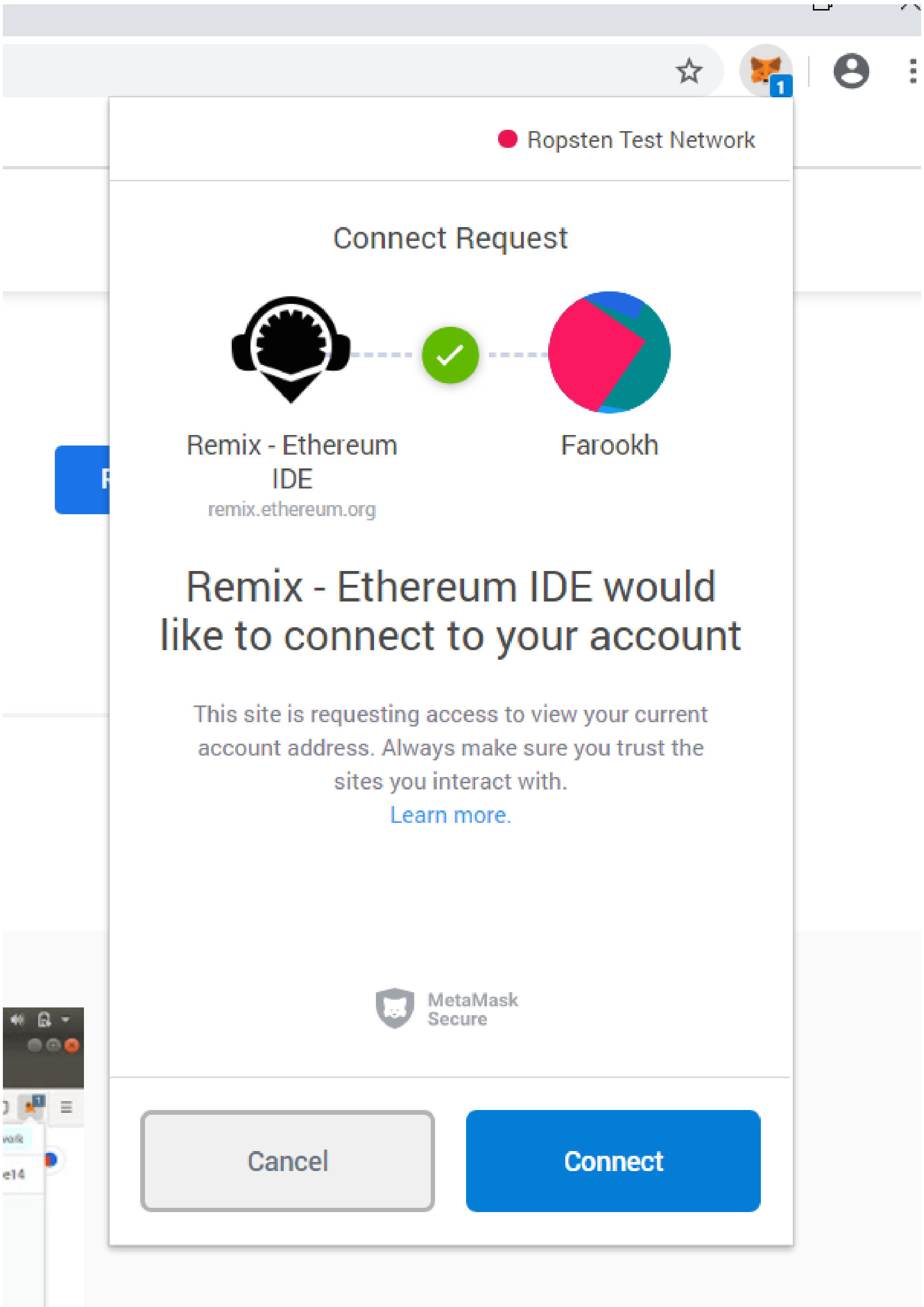


Figure 8.11. MetaMask plugin installation

We installed MetaMask plugin [https://metamask.io] on Chrome browser. The purpose of installing the MetaMask provider is that it securely stores the private key and the public Ethereum address. The function of MetaMask is to help securely sign transactions on the blockchain



● Ropsten Test Network

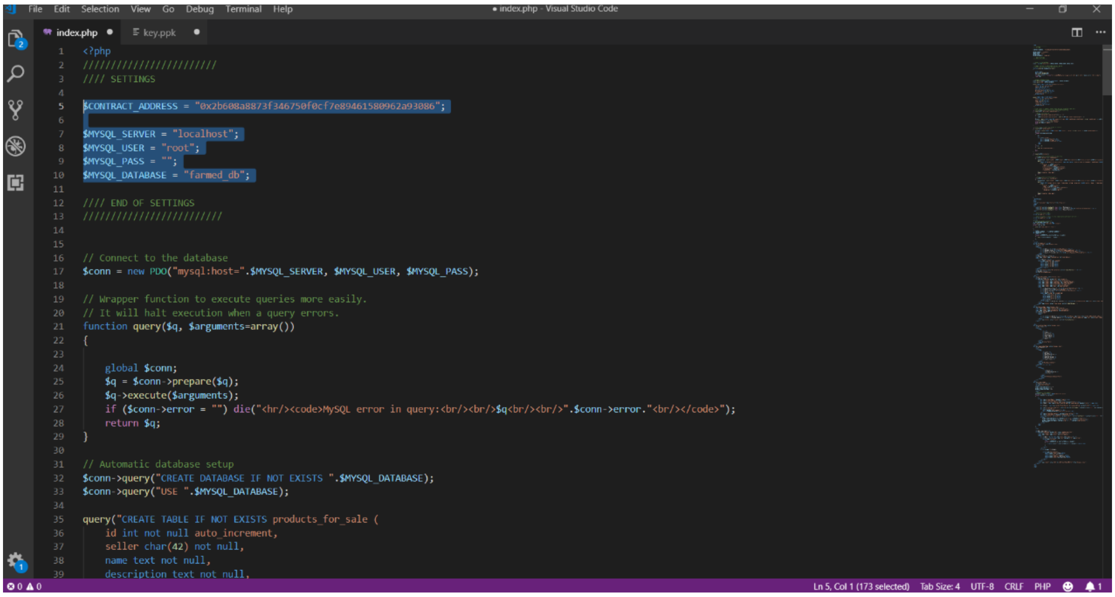
Connect Request

Remix - Ethereum IDE would like to connect to your account

This site is requesting access to view your current account address. Always make sure you trust the sites you interact with.
Learn more.

Figure 8.12. Linking Ethereum account with Ropsten Test Network connection

In Figure 8.12, we linked our "Farookh" Ethereum account with Ropsten Test Network.


File Edit Selection View Go Debug Terminal Help
• index.php - Visual Studio Code
index.php
key.ppk
Ln 5, Col 1 (173 selected) Tab Size: 4 UTF-8 CRLF PHP
0 0

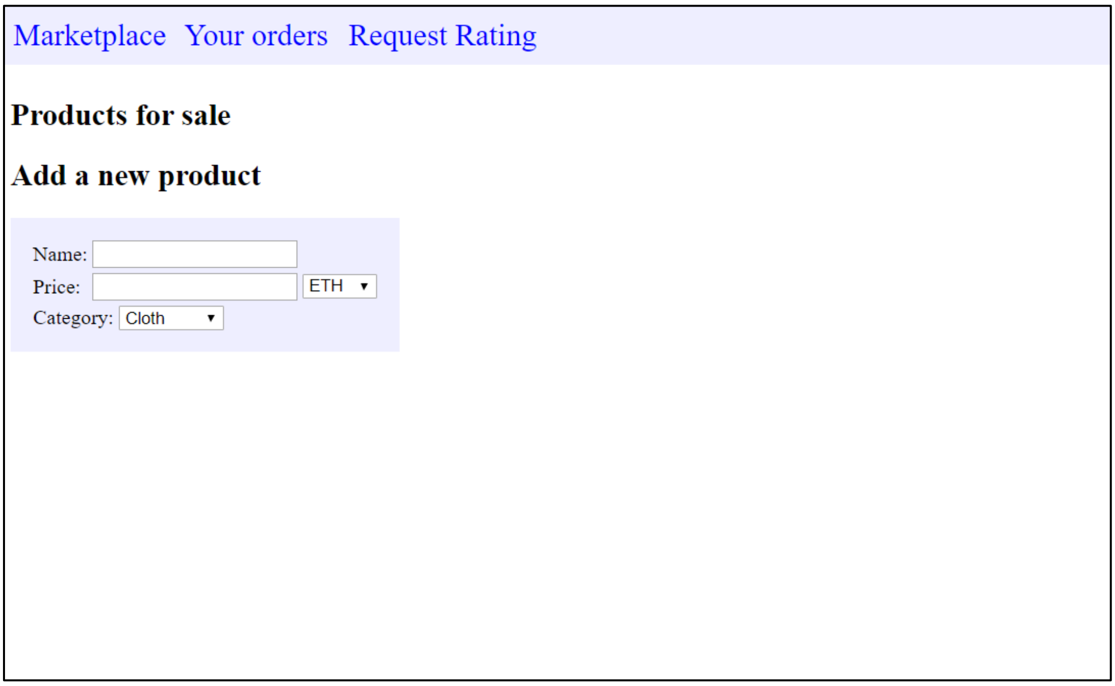
```php
<?php
////////////////////////
//// SETTINGS

$CONTRACT_ADDRESS = "0x2b608a8873f346750f0cf7e89461580962a93086";

$MYSQL_SERVER = "localhost";
$MYSQL_USER = "root";
$MYSQL_PASS = "";
$MYSQL_DATABASE = "farmed_db";

//// END OF SETTINGS
/////////////////////////


// Connect to the database
$conn = new PDO("mysql:host=".$MYSQL_SERVER, $MYSQL_USER, $MYSQL_PASS);

// Wrapper function to execute queries more easily.
// It will halt execution when a query errors.
function query($q, $arguments=array())
{

    global $conn;
    $q = $conn->prepare($q);
    $q->execute($arguments);
    if ($conn->error = "") die("<hr/><code>MySQL error in query:<br/><br/>$q<br/><br/>".$conn->error."<br/></code>");
    return $q;
}

// Automatic database setup
$conn->query("CREATE DATABASE IF NOT EXISTS ".$MYSQL_DATABASE);
$conn->query("USE ".$MYSQL_DATABASE);

query("CREATE TABLE IF NOT EXISTS products_for_sale (
    id int not null auto_increment,
    seller char(42) not null,
    name text not null,
    description text not null,
```

Figure 8.13: Index file in editor

Figure 8.13 shows the index.php file that we placed in the localhost home directory. We specified the values for settings variables as follows:

Variable	Value
$CONTRACT_ADDRESS	0x2b608a8873f346750f0cf7e89461580962a93086
$MYSQL_SERVER	Localhost
$MYSQL_USER	root
$MYSQL_PASS	*NULL*
$MYSQL_DATABASE	Farmed_db

Table 8.1: Local database connection variables



Products for sale

Add a new product

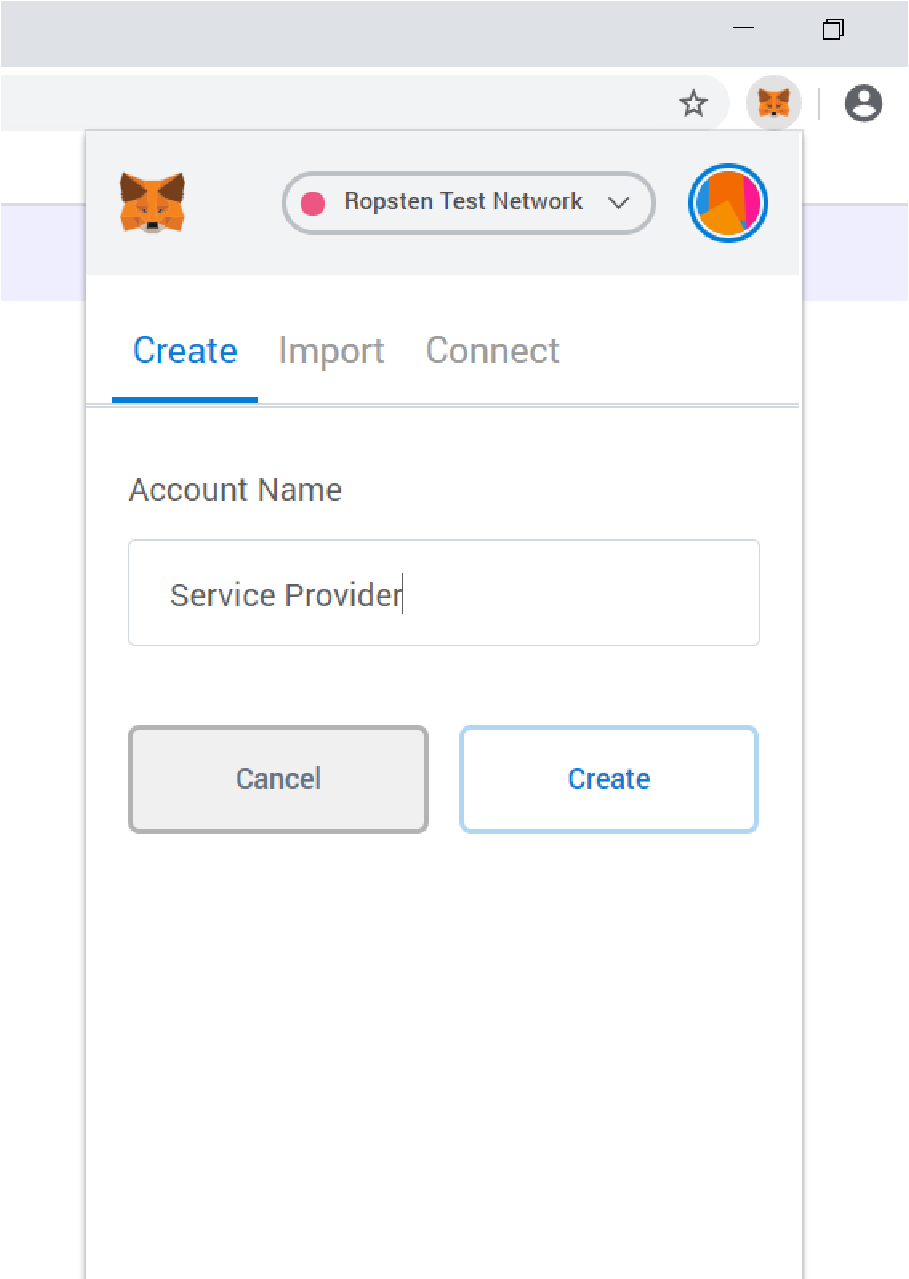
Name:

Price: ETH

Category: Cloth

Figure 8.14. The simulated marketplace interface

In Figure 8.14, we can see the screenshot for the *FarMed* marketplace interface with option to add a new product as service provider, Your orders, and Request ratings options.

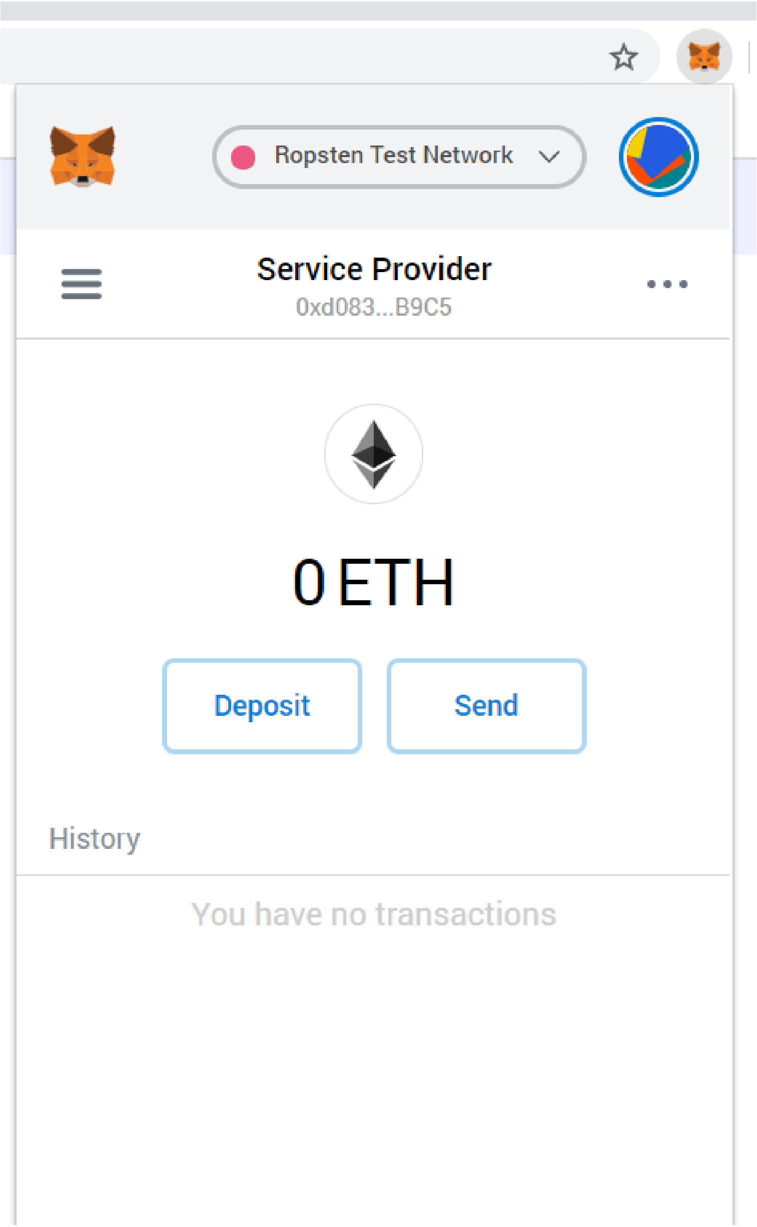


Account Name

Service Provider

Figure 8.15. Adding new service provider

We added new service provider account through MetaMask plugin to enable "Add product" option.



Ropsten Test Network

Service Provider

0xd083...B9C5

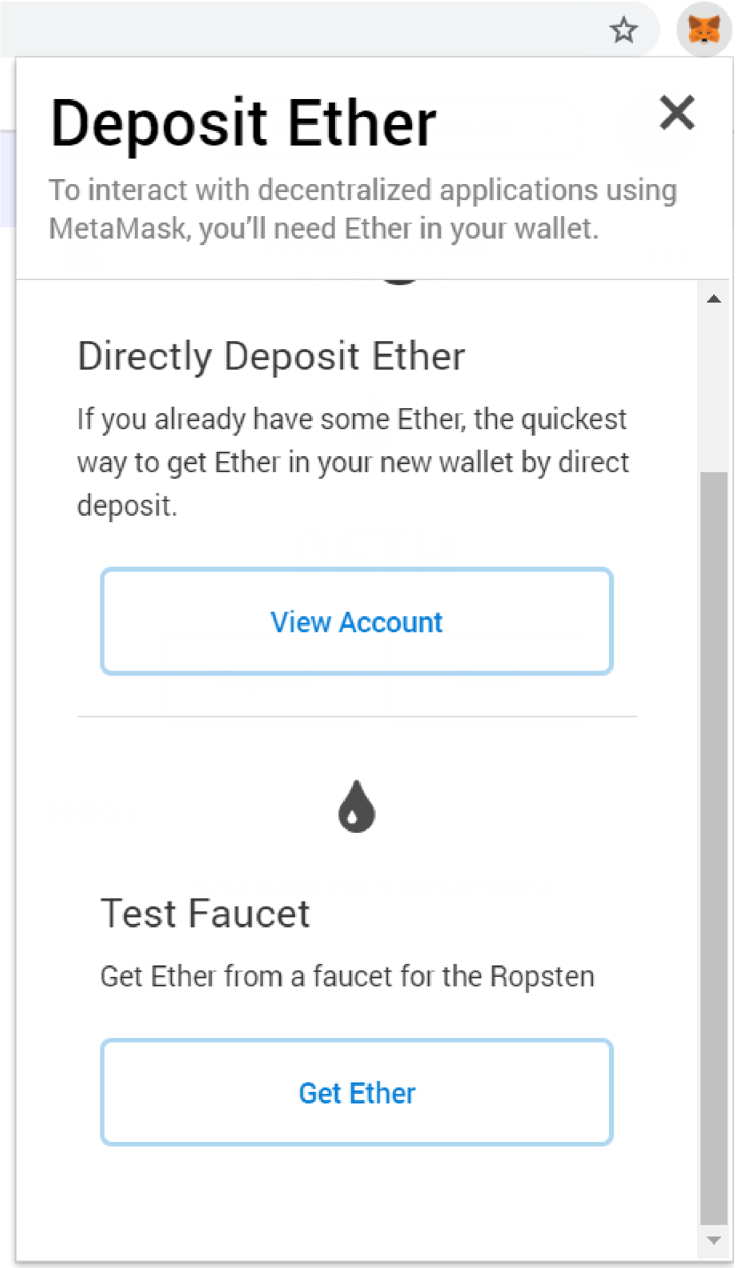
0 ETH

History

You have no transactions

Figure 8.16. New service provider account created

Figure 8.16 shows the account was created successfully. As it appears in the above screenshot, the Ether is 0 in the account wallet, and we needed Ether (a digital bearer asset) to execute the transactions through blockchain. To get Ether (also known as gas), we clicked on the "Deposit" button.



Deposit Ether

To interact with decentralized applications using MetaMask, you'll need Ether in your wallet.

Directly Deposit Ether

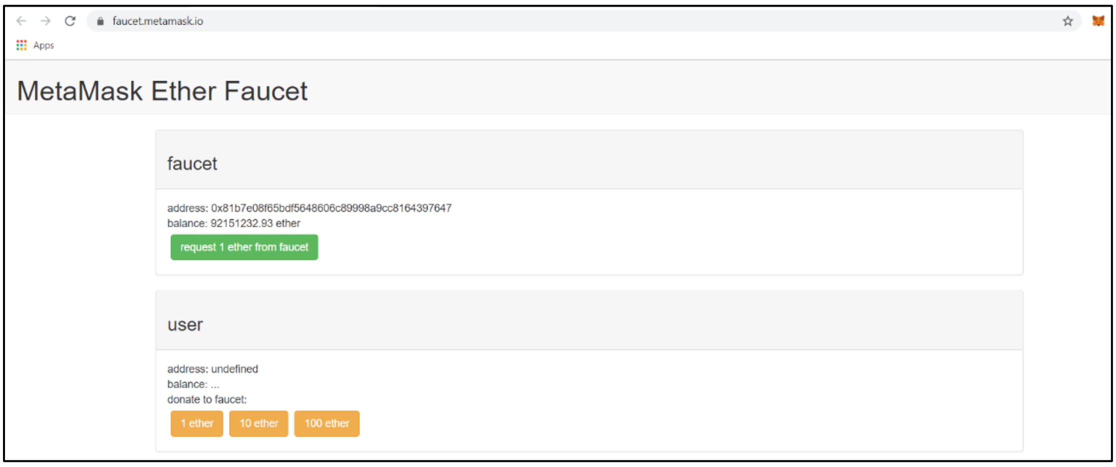
If you already have some Ether, the quickest way to get Ether in your new wallet by direct deposit.

Test Faucet

Get Ether from a faucet for the Ropsten

Figure 8.17. Ether deposit option

In Figure 8.17, we clicked on "Get Ether" button to deposit Ether for testing.



MetaMask Ether Faucet

faucet

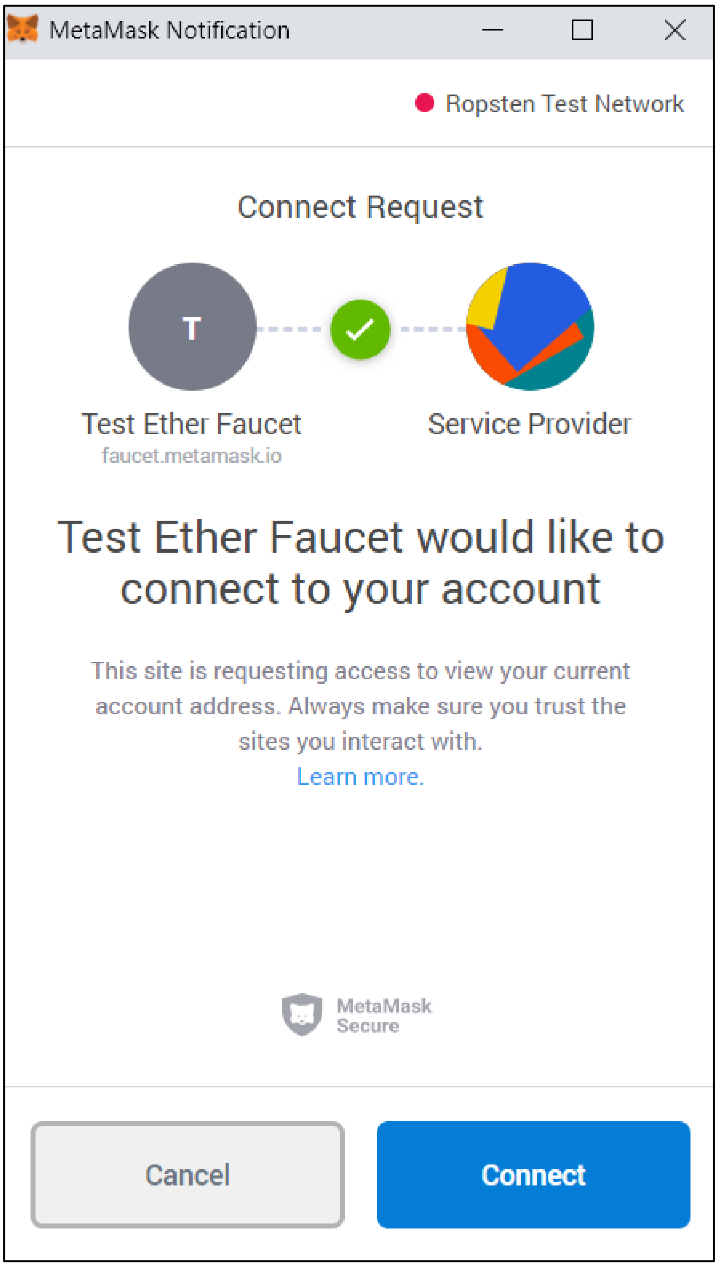
address: 0x81b7e08f65bdf5648606c89998a9cc8164397647
balance: 92151232.93 ether

user

address: undefined
balance: ...
donate to faucet:

Figure 8.18. MetaMask Ether Faucet

After clicking on "Deposit Ether", we were directed to MetaMask Ether Faucet page to request 1 Ether. See Figure 8.18.



MetaMask Notification

● Ropsten Test Network

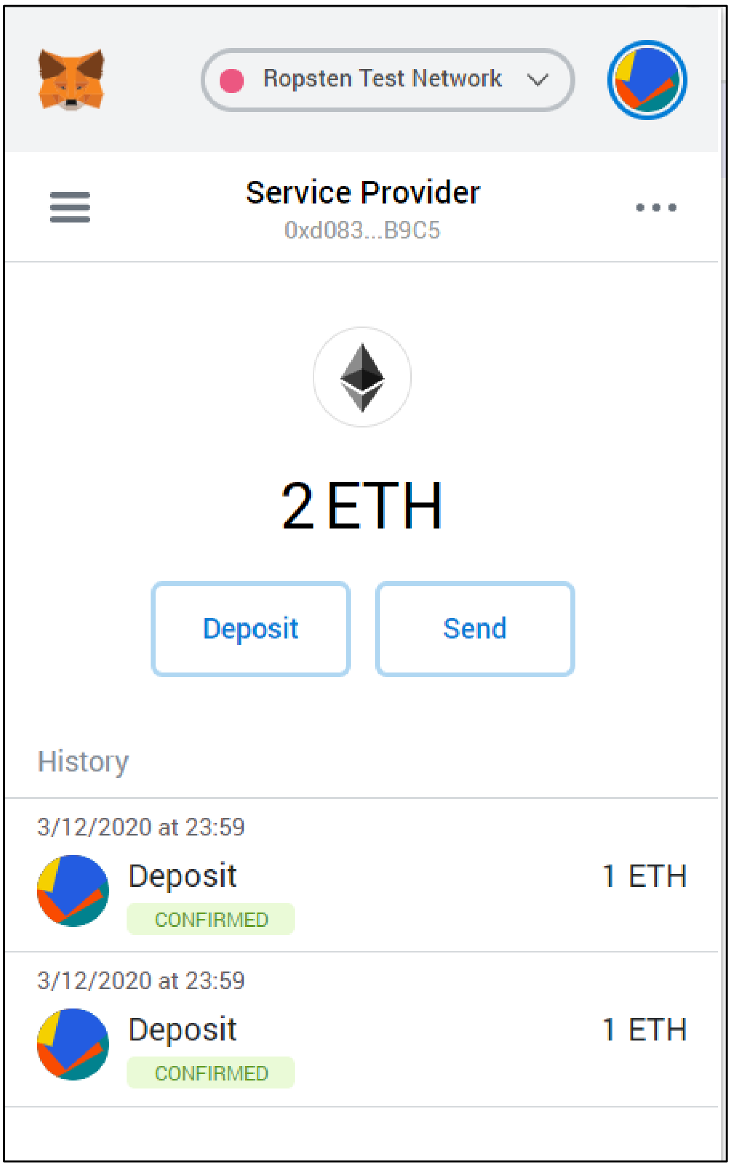
Connect Request

Test Ether Faucet would like to connect to your account

This site is requesting access to view your current account address. Always make sure you trust the sites you interact with.
Learn more.

Figure 8.19 Ether request confirmation

In Figure 8.19, we clicked on connect button to send the request to Test Ether Faucet.



Ropsten Test Network

Service Provider

0xd083...B9C5

2 ETH

History

3/12/2020 at 23:59

Deposit 1 ETH

CONFIRMED

3/12/2020 at 23:59

Deposit 1 ETH

CONFIRMED

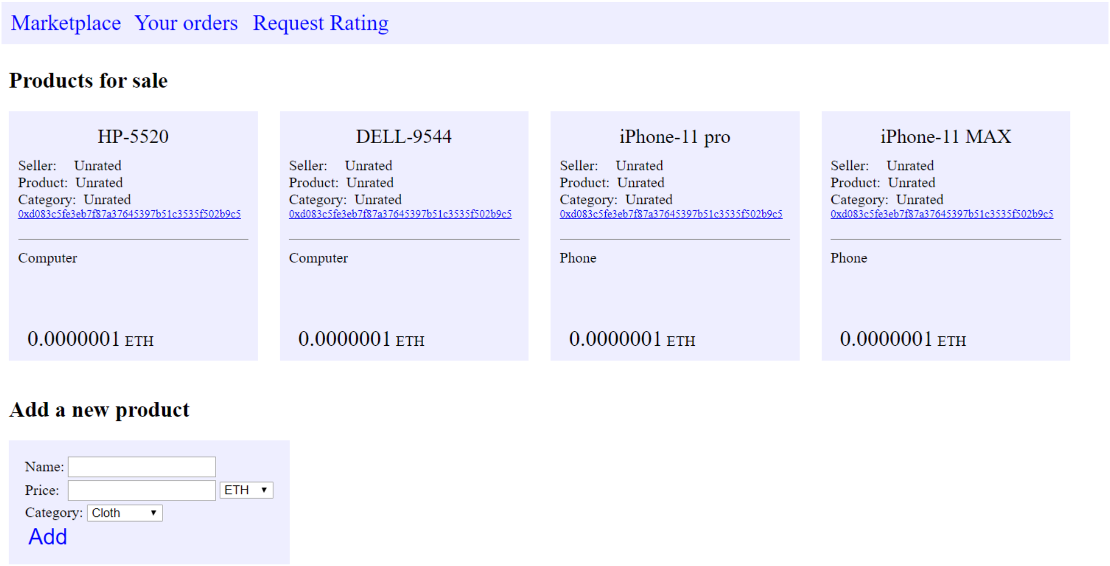
Figure 8.20. Ether is deposited

In Figure 8.20, you can see that we sent two requests; as a result, 2 Ethers were added to service provider wallet successfully.

In this section, we described the set up and working of our prototype. Building on that, in the next section we explain the working of the prototype for reputation computation. Subsequently, we explain the working of the prototype for reputation transfer in Section 8.4.

8.3 Prototype Working for Reputation Computation

In this section, we explain the working of the prototype (using screenshots) for the reputation computation process.



Products for sale

HP-5520
Seller: Unrated
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer
0.0000001 ETH

DELL-9544
Seller: Unrated
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer
0.0000001 ETH

iPhone-11 pro
Seller: Unrated
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone
0.0000001 ETH

iPhone-11 MAX
Seller: Unrated
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone
0.0000001 ETH

Add a new product

Name:
Price: ETH
Category: Cloth

Figure 8.21. Service provider's product listed in prototype

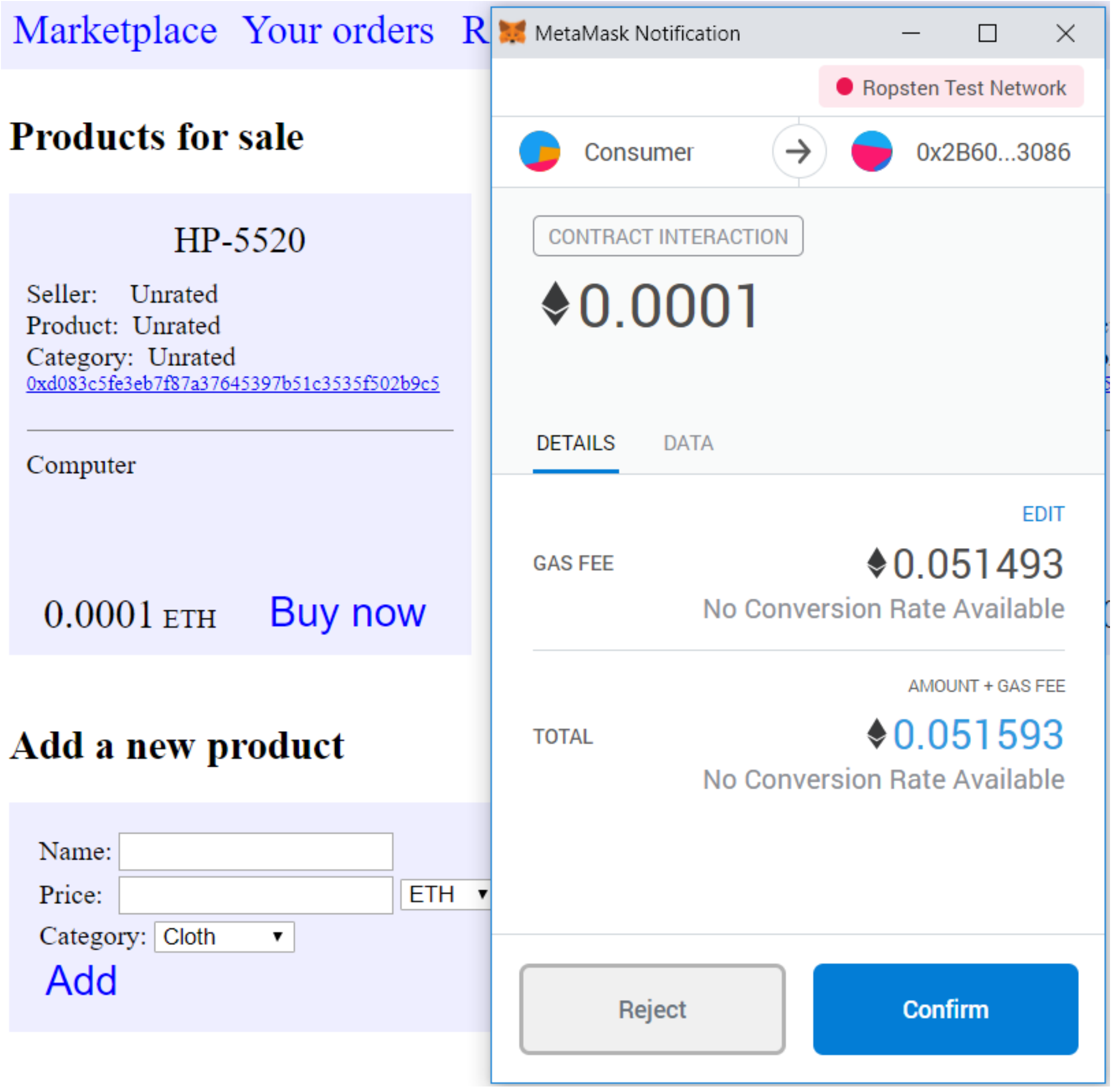
Figure 8.21 above shows the listing of the service provider's products for sale. Using "Service Provider" account, we have listed four different products in two different categories to FarMed as follows:

Product name	Product category
HP-5520	Computer
DELL-9544	Computer
iPhone-11 pro	Phone
iPhone-11 MAX	Phone

Table 8.2: Products details inserted in prototype

We created a new account named "Consumer" using the same steps as outlined in Figure 8.15 to Figure 8.16 and deposited Ether using the same steps as outlined in Figure 8.17 to Figure 8.20.

After that, we changed the account to "Consumer" from MetaMask to enable the system to activate purchase and review options and use them against the Service provider's products. See Figure 8.22 for more details.



Products for sale

HP-5520

Seller: Unrated
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5

Computer

0.0001 ETH

Add a new product

Name:
Price: ETH
Category: Cloth

Ropsten Test Network

Consumer → 0x2B60...3086

CONTRACT INTERACTION

0.0001

GAS FEE 0.051493
No Conversion Rate Available

AMOUNT + GAS FEE
TOTAL 0.051593
No Conversion Rate Available

Figure 8.22. Confirmation Process

Using the "Consumer" account, we purchase "HP-5520" product from "Service provider" products and subsequently confirmed the transaction through Ropsten Test Network to get registered and verified.

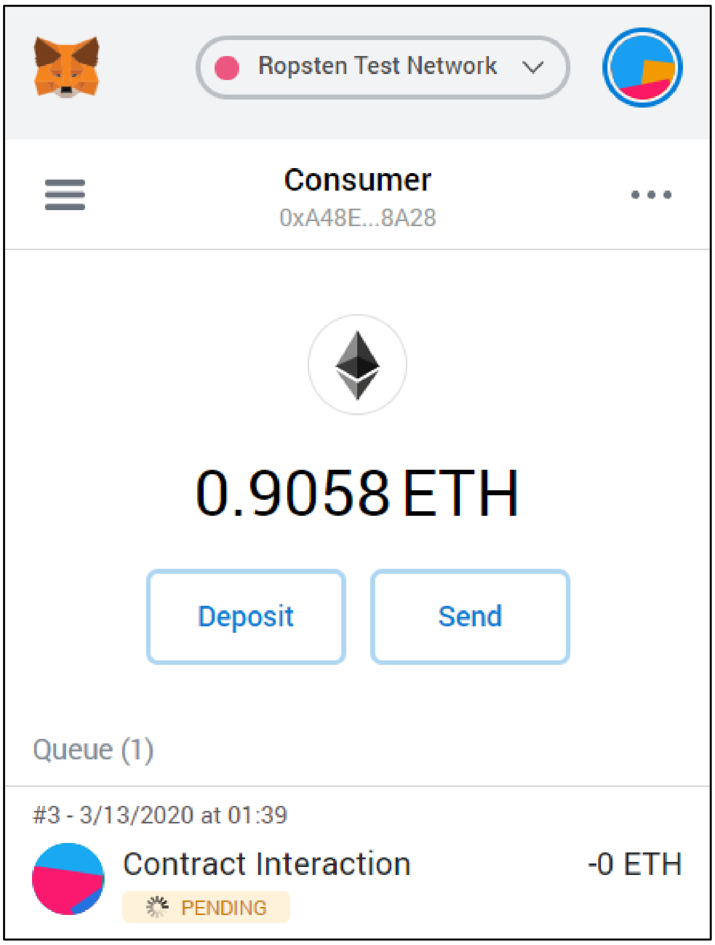
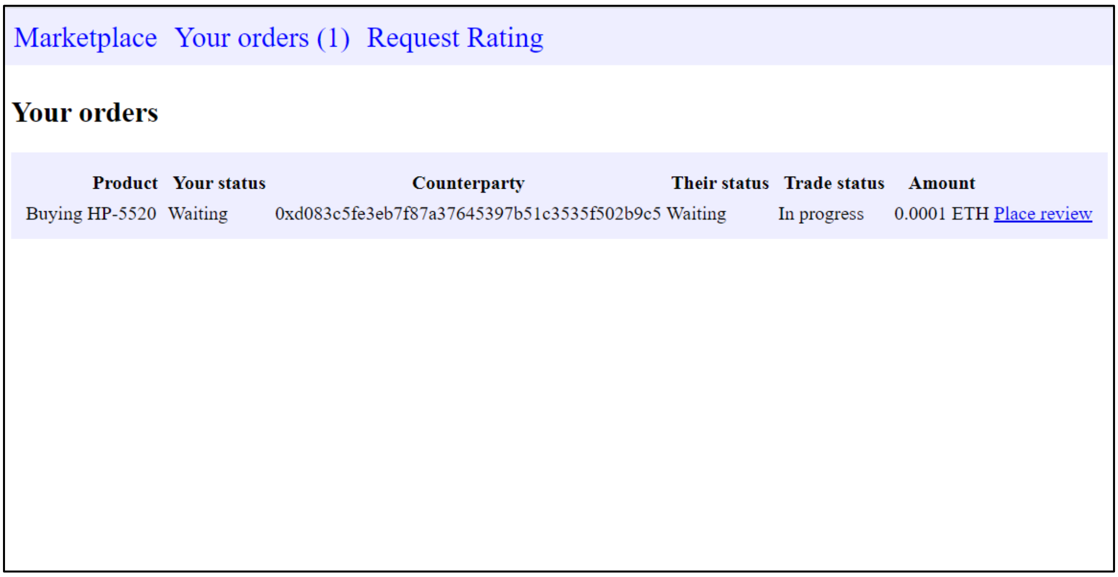


Figure 8.23. Contract Interaction pending

It needs to be noted that every time we make an action in the marketplace such as purchasing or placing a review, we had to wait some time until Ropsten Test Network confirms the transaction and we can see the new results appear in the marketplace.

This is due to the consensus mechanism that needs to happen in Blockchain before the confirmation can be presented to the end-user.

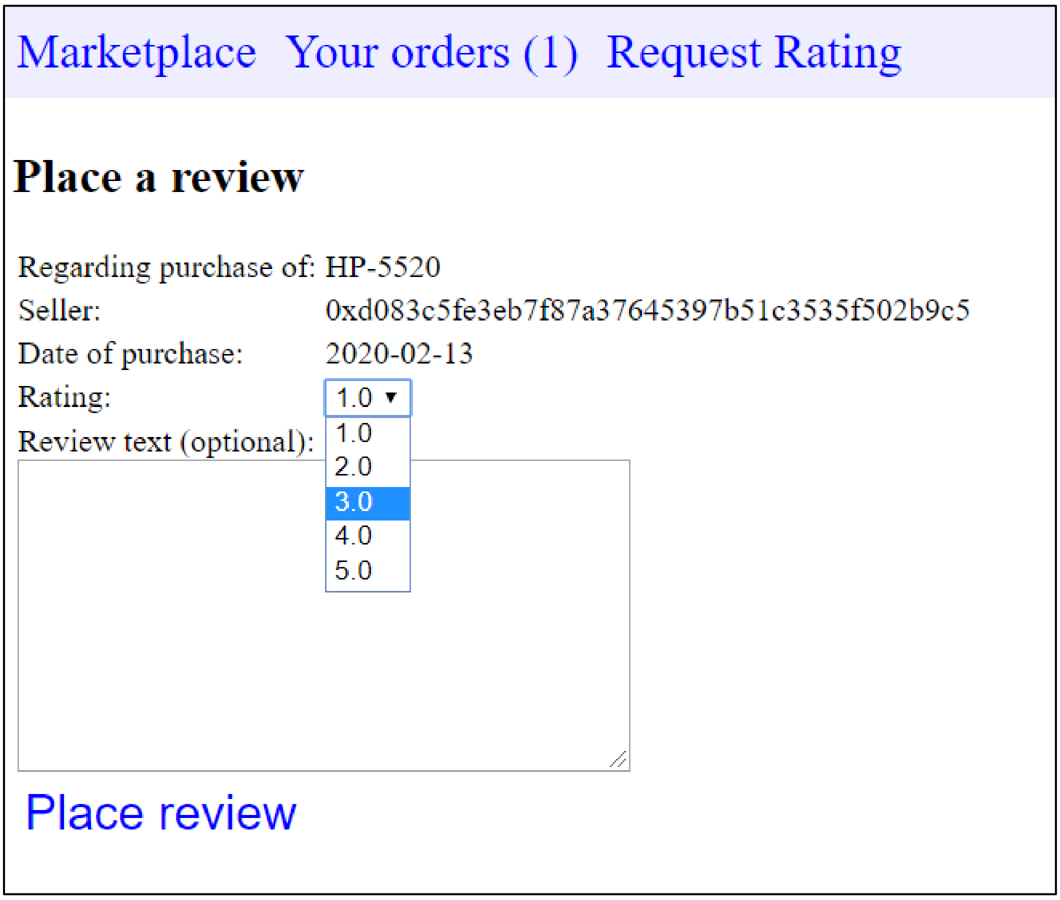


Your orders

Product	Your status	Counterparty	Their status	Trade status	Amount	
Buying HP-5520	Waiting	0xd083c5fe3eb7f87a37645397b51c3535f502b9c5	Waiting	In progress	0.0001 ETH	Place review

Figure 8.24. Review the order

Section 8.24 shows that from the "Your orders" tab, we reviewed the purchase that had been made and we were ready to place a review.

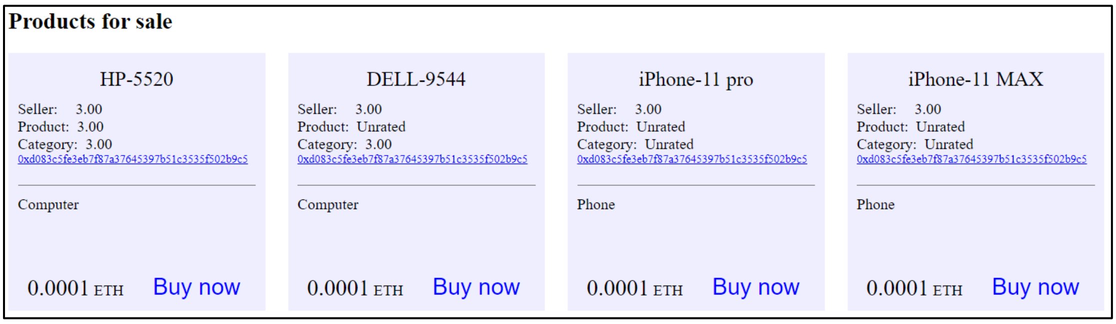


Place a review

Regarding purchase of:	HP-5520
Seller:	0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Date of purchase:	2020-02-13
Rating:	1.0
Review text (optional):	

Figure 8.25. Place a review

In Figure 8.25, we assigned a score of 3 out of 5 from the rating list and we clicked on "Place review" option. Then the rating went through the confirmation process to register the transaction through Ropsten Test Network. As stated previously, because the ratings are stored on the Blockchain, there would be a time delay experienced by the end-user. See Figure 8.23 for more details.



Products for sale

HP-5520
Seller: 3.00
Product: 3.00
Category: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer
0.0001 ETH Buy now

DELL-9544
Seller: 3.00
Product: Unrated
Category: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer
0.0001 ETH Buy now

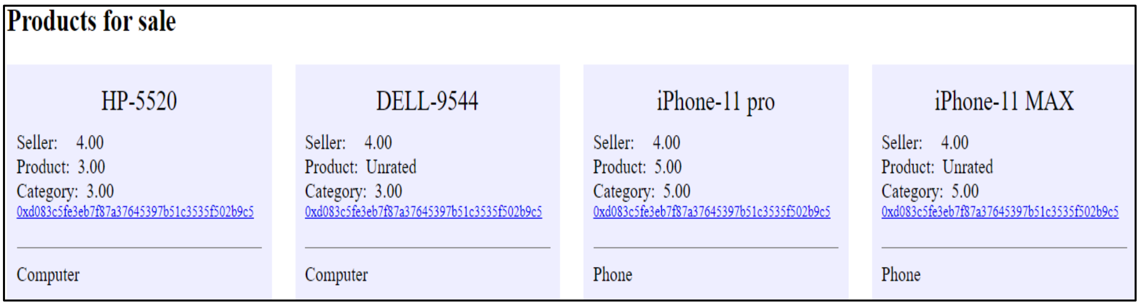
iPhone-11 pro
Seller: 3.00
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone
0.0001 ETH Buy now

iPhone-11 MAX
Seller: 3.00
Product: Unrated
Category: Unrated
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone
0.0001 ETH Buy now

Figure 8.26. HP-5520 product rating

In this case, the overall reputation rating of "Service Provider" is 3 and this overall score is reflected in all the products that he/she is selling. HP-5520 product reputation score, however, is 3 while the other products are "Unrated". Also, the "Computer" category regardless of the number of products under it is 3, while "Phone" category is still "Unrated".

We purchased "iPhone-11 pro" using the same steps mentioned previously and placed a review of 5 into the "Service Provider" account from the "Consumer" account.



Products for sale

HP-5520
Seller: 4.00
Product: 3.00
Category: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer

DELL-9544
Seller: 4.00
Product: Unrated
Category: 3.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Computer

iPhone-11 pro
Seller: 4.00
Product: 5.00
Category: 5.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone

iPhone-11 MAX
Seller: 4.00
Product: Unrated
Category: 5.00
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Phone

Figure 8.27. New values in marketplace

It is clear that after submitting another review to another product for the same service provider, the new overall reputation score is 4, "HP-5520" and "iPhone-11 pro" have different product reputation scores, and lastly "Computer" and "Phone" categories have different category reputation scores.

8.4 Prototype Working for Reputation Transfer

We created a new account named "Service provider 2" using the steps in Figure 8.15- Figure 8.16 and deposited Ether using steps in Figure 8.17- Figure 8.20 to test the reputation transfer service.

The new service provider reputation score is "Unrated" because he still has not received any reviews, while the other service provider named "Service provider" has total overall reputation score of 4.00.

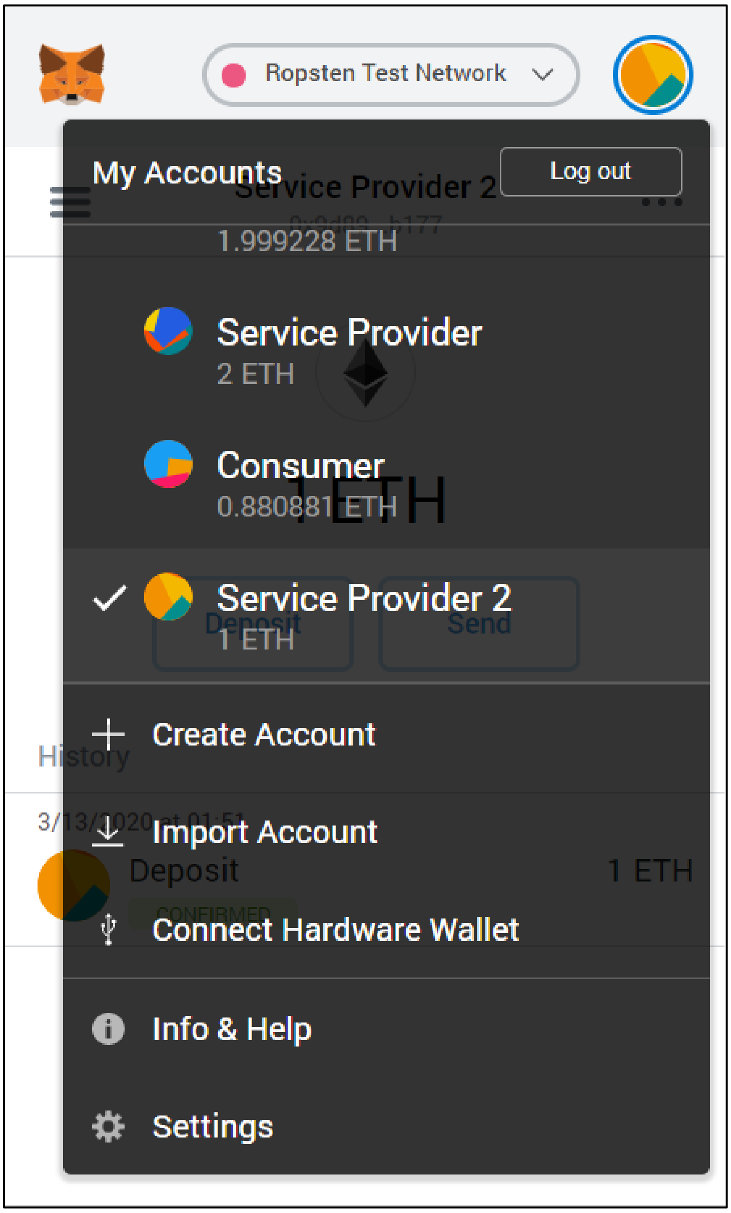
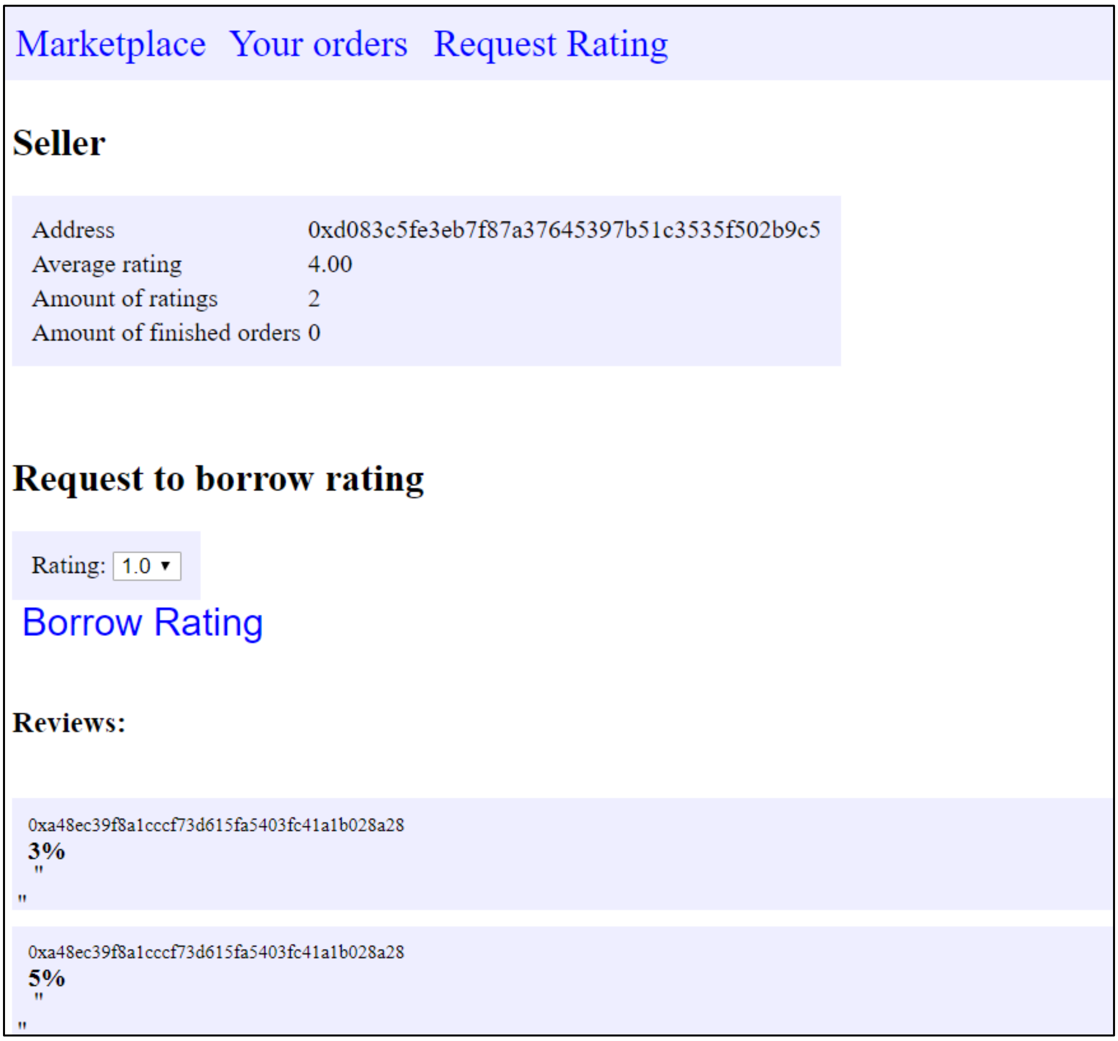


Figure 8.28. Changing the user account from MetaMask

We changed the service provider to "Service provider-2" account from MetaMask so we act on behalf of the service provider in the marketplace. This step is needed so "Service provider-2" can request a reputation score from any other service provider in the market.



Seller

Address	0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Average rating	4.00
Amount of ratings	2
Amount of finished orders	0

Request to borrow rating

Rating: 1.0 ▾

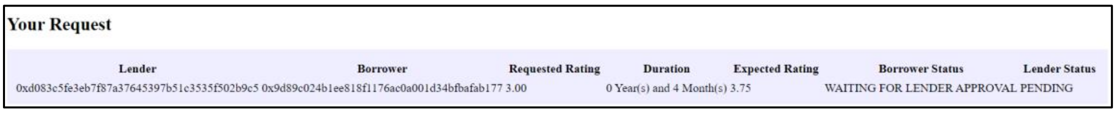
Reviews:

0xa48ec39f8a1cccf73d615fa5403fc41a1b028a28
3%
"
"

0xa48ec39f8a1cccf73d615fa5403fc41a1b028a28
5%
"
"

Figure 8.29. Request rating interface

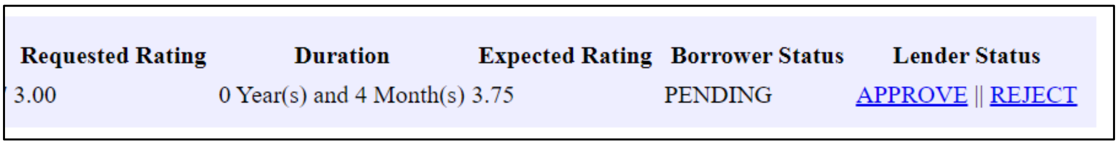
As "Service provider-2", we went to "Service provider" page to request the rating to be transferred, and we requested reputation value of 3 from "Service Provider" account. The request went through the steps above for verification through *FarMed*. See Figure 8.22 for more details.


Your Request

Lender	Borrower	Requested Rating	Duration	Expected Rating	Borrower Status	Lender Status
0xd083c5fe3eb7f87a37645397b51c3535f502b9c5	0x9d89c024b1ee818f1176ac0a001d34bfbafab177	3.00	0 Year(s) and 4 Month(s)	3.75	WAITING FOR LENDER APPROVAL	PENDING

Figure 8.30. Pending waiting for approval

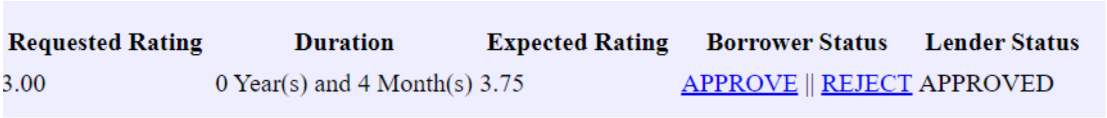
After the request had been sent and been verified by Ropsten Test Network, the approval from both parties had to be received, first step should be taken by the lender who is the "Service provider"



Requested Rating	Duration	Expected Rating	Borrower Status	Lender Status
3.00	0 Year(s) and 4 Month(s)	3.75	PENDING	APPROVE || REJECT

Figure 8.31. Request was approved by lender

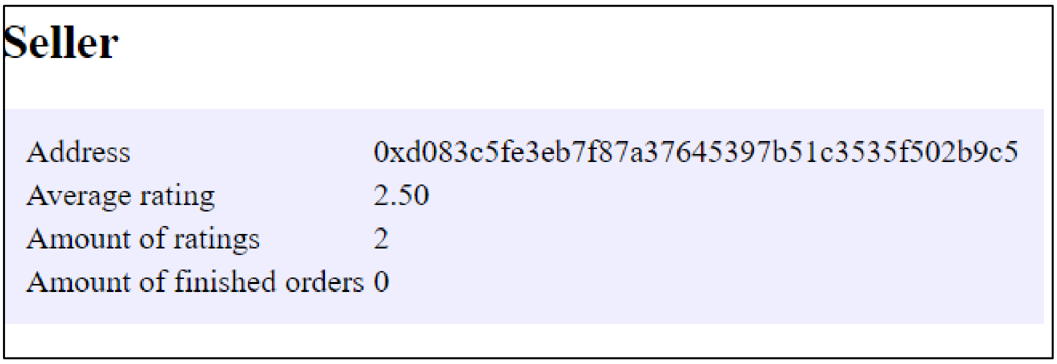
As the lender, "Service provider", we had approved the request and went through verification steps in Figure 8.22, then we had to wait for the borrower "Service provider-2" approval.



Requested Rating	Duration	Expected Rating	Borrower Status	Lender Status
3.00	0 Year(s) and 4 Month(s)	3.75	APPROVE || REJECT	APPROVED

Figure 8.32. Request was approved by borrower

As Borrower, "Service provider-2", we had approved the request and went through verification steps in Figure 8.22. Then we waited for the new reputation score to be transferred.

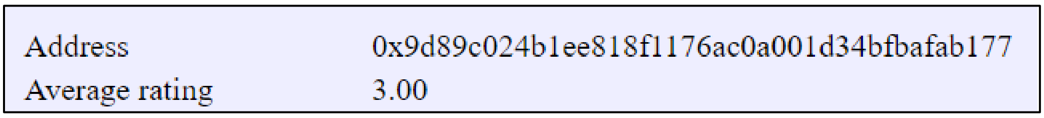


Seller

Address	0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Average rating	2.50
Amount of ratings	2
Amount of finished orders	0

Figure 8.33. New reputation score after trading for lender

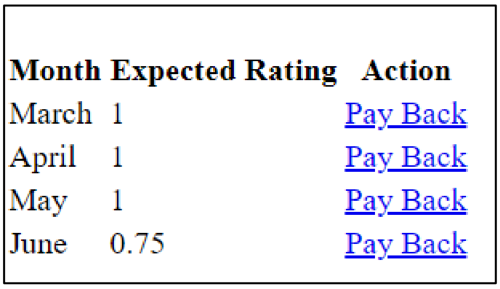
In Figure 8.33, we can see that the new reputation score for lender "Service provider", after transferring 3 points from his overall reputation score to the borrower "Service provider-2", dropped from 4 to 2.50.



Address	0x9d89c024b1ee818f1176ac0a001d34bfbafab177
Average rating	3.00

Figure 8.34. New reputation score after trading for borrower

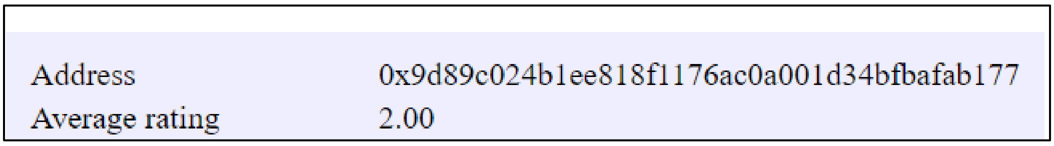
In Figure 8.34, we can see that the new reputation score for borrower "Service provider-2" after receiving the approved 3 points from "Service provider" increased from "Unrated" to 3.00.



Month	Expected Rating	Action
March	1	Pay Back
April	1	Pay Back
May	1	Pay Back
June	0.75	Pay Back

Figure 8.35. Instalments to be made

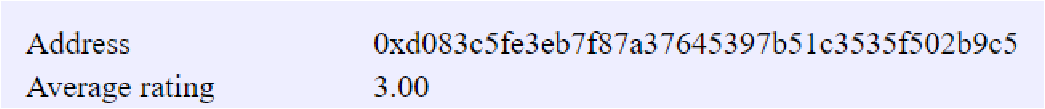
In Figure 8.35, we can see the instalments schedule that borrower "Service provider-2" must follow to pay back the lender "Service provider". Notice that the total payback score is 3.75 where 3 is the capital of the reputation loan and 0.75 is the interest. Instalments period was set to pay back month by month.



Address	0x9d89c024b1ee818f1176ac0a001d34bfbafab177
Average rating	2.00

Figure 8.36. After first payback instalment – Borrower

In Figure 8.36, we can see that the reputation score of the borrower "Service provider-2" was affected after the first instalment was made and his score was reduced from 3 to 2.



Address	0xd083c5fe3eb7f87a37645397b51c3535f502b9c5
Average rating	3.00

Figure 8.37. After first payback instalment – Lender

In Figure 8.37, we can see that the reputation score of the lender "Service provider" was affected after the first instalment was made and his scored was increased from 2.5 to 3.

8.5 Conclusion

In this chapter we presented the working of the prototype developed for this research work. In doing so, we presented the methodological progression and working of the developed prototype in achieving the objectives of this thesis. We explained in detail the prototype setup which included the local set up and the blockchain setup.

The next chapter will conclude the whole thesis and provide a background for future research work

Chapter 9 : Conclusion and Future Work

9.1 Introduction

This chapter concludes the thesis by presenting an overview of the research results and making suggestions for future work. While several researchers have worked on blockchain and reputation systems, this research work represents a pioneering effort in using smart contract for blockchain-based reputation systems. This is evident in Chapter 2 where the outcomes of a systematic literature review and thorough investigation into previous work are presented. Based on the investigations, research gaps were identified, and this research created a novel solution called *FarMed* framework to address the gaps.

9.2 Problems Addressed in this Thesis

The primary aim of this thesis is to fill critical gaps related to the use of smart contracts for blockchain-based reputation systems in the existing literature body. Based on the literature review in Chapter 2, the research issues that were identified and subsequently addressed in the thesis include the following:

1. There is no existing method for deriving the reputation value of service providers based on the values (or ratings) present in the smart contracts. In addition, there is no means of deducing the trust value of a service provider based on the trust values in the blocks.
2. There are no methods to intelligently infer the reputation value of a service provider in a specific context based on existing trust values in various contexts.
3. There is no existing framework by which reputation is regarded as a digital asset and can be moved across platforms or from one service provider to another.
4. There is no mechanism by which new service providers can be bootstrapped into the reputation-based economy.

9.3 Contributions to the Existing Literature

With respect to the research issues identified, the major contribution of this thesis to the existing literature is the creation of a novel solution called *FarMed* framework. A brief overview of the research contributions is as follows:

9.3.1 Systematic Literature Review

The thesis presented an extensive and systematic state-of-the art review of the existing literature in the areas of blockchain, smart contracts and reputation systems. This is properly documented in Chapter 2. For the SLR, specific search terms were inputted in three databases including Elsevier ScienceDirect (www.sciencedirect.com/), IEEE Xplore (www.ieexplore.ieee.org/Xplore/), and Google Scholar (www.scholar.google.com.au/). The results of the search process were subjected to inclusion and exclusion criteria as well as evaluation for relevance. At the end of the evaluation process, 30 relevant papers were identified and critically reviewed. The outcome of the review showed that the existing literature has not proposed a framework that facilitates the interchangeable use of smart contracts for blockchain-based reputation systems. At the time of writing this thesis, we have written a journal paper capturing the Systematic Literature Review. It has been submitted to the Journal of Network and Computer Applications and is currently under review.

9.3.2 Creation of a Novel Solution: The FarMed Framework

This study proposed and developed an intelligent framework termed as *FarMed*. The framework can execute Ethereum smart contact-based reputation systems and develop reliable blockchain-based protocols for computing reputation values, deducing trust values and transferring reputation values from one provider to another. In the *FarMed* framework, there are three phases. The marketplace represents the first phase while the smart contract execution is the second phase. After the second phase comes the third phase which involves blockchain-based trust values and computations. The intelligence that has been built into the Farmed service provides automated and reliable mechanisms that ensures the following:

9.3.2.1 Determination of the current reputation value of a service provider based on its previous ratings being stored in a Blockchain

For this research objective, the Ropstein Ethereum network is used. The network validates the seller's rating through a signature and adds the rating to the network. The information stored in the network includes the raw rating value, the seller's ID and the buyer's ID. The five-star algorithm is used to compute the overall reputation value of the service provider. This algorithm is linked to the electronic marketplace.

This system is intelligent because it aggregates the ratings for a seller and uses an algorithm to derive the trust value from the overall rating of that seller. The ratings are reliable because it is only buyers that have patronized a seller that are allowed to rate the seller. All the ratings are stored in the blockchain. Since records on blockchain are immutable, it means the rating can never be altered.

9.3.2.2 Intelligent mechanisms for trust-based inferencing in different contexts

The ontology-based method was used to model the context-specific nature of trust and reputation. To address the research objective, the semantic similarity for each unrated product was computed based on the rated product. We used transport service ontology and AKTiveRank to compute the semantic similarity. Once the semantic distance has been determined, an algorithm was applied to model a context-based inference. The algorithm is only used when semantic distance between the existing product(s) and unrated products is greater than a certain threshold.

9.3.2.3 Intelligently preventing people from manipulating reviews in the reputation system

The smart contract needs to have validated methods of detecting and dealing with any reputation fraud. This is the most critical part of the smart contract. To achieve this, the smart contract was evaluated from time to time by deploying the Ethereum Ropsten network. The Metamask plugin was installed first so that it communicates with nodes on the remote server. After switching to the Ropsten network, a new account was created, and the solidity compiler was used to deploy the contract. The contract was then tested to detect any non-compliant behavior using the provided interface.

9.3.2.4 Providing a platform for transferring the reputation value of a service provider to other service providers.

To address this objective, a Reputation Auction Service (RAS) is developed. This service provides extensive opportunity for the bootstrapping of new service providers in the reputation-based economy. Service providers with high or excess reputation value will auction part of their reputation ratings to other service providers. The new sellers who have zero reputation score can buy from this auction to build their reputation in the market fast while the offers in the auction will be made by sellers who have a high reputation score. The benefit for the sellers is completely commercial.

The working of the Reputation Auction Service is based on modified five-star algorithm. First, the request for reputation purchase is submitted. After this, an agreement is made between the service providers. Then, the new reputation score for each service provider is computed using the modified five-star algorithm and then is published in the marketplace.

9.3.3 Evaluation, Validation, and Implementation of the proposed solutions

In order to validate the performance and accuracy of the proposed framework in this thesis, software prototyping was the model of choice. The working of the prototype, both for reputation computation as well as reputation inference and transfer, are demonstrated in Chapter 8. Furthermore, Chapter 5-7 also demonstrate the working of the developed prototype corresponding to each objective.

9.4 Future Work

Although this study has carried out a lot of research on using smart contracts for blockchain-based reputation system, there are still other issues that can be explored in the future. Our future work will be along the following dimensions:

a) Implementing and comparing with other Blockchain platforms like Dfinity and Ethereum 2.0. Dfinity has just released an initial alpha version to enable developers to get familiar with the technology. Ethereum 2.0 is faster than the traditional Ethereum and it can run the Ethereum smart contract. However, it is also at the development stage. We hope to implement our solution on these new blockchain platforms in the future and make comparisons with the traditional Ethereum technology.

b) Predicting future reputation values of service providers: Future research work should build a model that can allow users in an online marketplace to predict the future reputation values of service providers based on the trust values in the smart contract.

c) Addressing the latency issue faced by the *FarMed* framework: As I demonstrated the working of the prototype in Chapters 5, 6, 7, and 8, a key shortcoming associated with my work is latency. Due to this latency, it is not possible to implement the solution in a commercial application. In future, the potential way to solve this problem is to build intelligent algorithms with proof-of-stake.

d) Implementation of the *FarMed* framework in a real marketplace: In this research, I developed the *FarMed* framework, conceptualized it and built a prototype for it. In future, this can be made into commercial reality by building a commercial system that uses the *FarMed* framework.

List of References

Alani, H. and Brewster, C. (2006). Metrics for Ranking Ontologies. *Intelligence, Agents, Multimedia Group*, School of Electronics and Computer Science University of Southampton Southampton, UK.

Al-Bassam, M. 2017, ‘SCPKI: a smart contract-based PKI and identity system’, *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts,* ACM, pp. 35-40.

Angles, R. and Gutierrez, C. (2005). Querying rdf data from a graph database perspective. In *Proc. of 2nd European Semantic Web conference, ESWC*, pages 346–360, Crete.

Audun, J., Roslan, I., and Colin, B. 2007, ‘A Survey of Trust and Reputation Systems for Online Service Provision’, *Information Security Research Centre Queensland University of Technology*, Brisbane, Australia.

Bigi, G., Bracciali, A., Meacci, G., & Tuosto, E. 2015, ‘Validation of decentralised smart contracts through game theory and formal methods’, *Programming Languages with Applications to Biology and Security*, Springer, pp. 142-61.

Bogner, A., Chanson, M., & Meeuw, A. 2016, ‘A decentralised sharing app running a smart contract on the ethereum blockchain’, *Proceedings of the 6th International Conference on the Internet of Things,* ACM, pp. 177-78.

Buechler, M. Earabathini, M. Hockenbrocht, C. Wan, D. 2015, *Decentralized Reputation System for Transaction Networks,* University of Pennsylvania.

Butcher, M. and Lunden, I. (2020). Dfinity launches an open-source platform aimed at the social networking giants.

Buterin, V. 2014, ‘A next-generation smart contract and decentralized application platform’, *White Paper*.

Caesar, C. 2018, ‘How to build a Reputation System on Blockchain?’ *Bitconch White Paper gives out a great answer.*

Cai,Y. Zhu, D. 2016, ‘Fraud detections for online businesses: a perspective from blockchain technology’, *Financial Innovation*.

Carboni, D. 2015, ‘Feedback based Reputation on top of the Bitcoin Blockchain’, *Information Society Department,* Italy.

Casassa, M., Tomasi, M., and Montanari, L. 2001, ‘An Adaptive System Responsive to Trust Assessment Based on Peer-to-Peer Evidence Replication and Storage’, *Technical report HPL-2001-133*, Hewlett Packard Laboratories.

Chakravorty, A. Rong, C. 2017, ‘Ushare: user controlled social media based on blockchain’, *ACM,* ISBN 978-1-4503-4888-1/17/0.

Chanana, N. and Goele, S. (2012) “Future of e-commerce in India”, *International Journal of Computing & Business Research*, ISSN (Online): 2229-6166.

Chatfield, C., 2000, *Time-series forecasting*, CRC Press.

Chen, T., Li, X., Luo, X., & Zhang, X. 2017, ‘Under-optimized smart contracts devour your money’, *IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER),* pp. 442-46.

Christidis, K., & Devetsikiotis, M. 2016, ‘Blockchains and smart contracts for the internet of things’, *IEEE Access*, vol. 4, pp. 2292-303.

CMS (2019). Design for Nursing Home Compare Five-Star Quality Rating System: Technical Users’ Guide. *Centers for Medicare and Medicaid Services*.

Cong, L. W., & He, Z. 2018, ‘Blockchain disruption and smart contracts (No. w24399)’, *National Bureau of Economic Research.*

Decker, C. and Wattenhofer, R. 2013 ‘Information propagation in the bitcoin network’ *In: Proceedings of the IEEE Internation Conference on Peer-to-Peer Computing (P2P),* Trento, Italy.

Dellarocas, C. 2012, ‘Goodwill hunting: An economically efficient online feedback mechanism for environments with variable product quality’, in *Workshop on Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, July 2012, pp. 93–112.

Dellarocas, C. 2003, ‘The digitization of word-of-mouth: Promise and challenges of online reputation systems’, *Management Science*, Vol. 49, no. 10, pp. 1407–1424.

Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. 2016, 'Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab', *International Conference on Financial Cryptography and Data Security,* Springer, Berlin, pp. 79-94.

Dennis, R., Owenson, G., & Aziz, B. 2016, 'A temporal blockchain: a formal analysis', *International Conference on Collaboration Technologies and Systems (CTS),* IEEE, pp. 430-37.

Dong, H., Hussain, F., and Chang, E. (2008). Transport Service Ontology and Its Application in the Field of Semantic Search. *Digital Ecosystems and Business Intelligence Institute.* Curtin University of Technology Perth, Australia.

Egbertsen, W., Hardeman, G., van den Hoven, M., van der Kolk, G., & van Rijsewijk, A. 2016, 'Replacing Paper Contracts with Ethereum Smart Contracts'.

English, M., Auer, S., & Domingue, J. 2016, 'Block chain technologies & the semantic web: A framework for symbiotic development', *Computer Science Conference for University of Bonn Students, J. Lehmann, H. Thakkar, L. Halilaj, and R. Asmat, (eds),* pp. 47-61.

Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R., and Musen, M. A. 1995, 'Task Modelling with Reusable Problem-Solving Methods', *Artificial Intelligence*, vol. 79, pp. 293–326.

Everts, M., & Muller, F. 2018, 'Will that smart contract really do what you expect it to do?' *TNO*.

Fensel, D., Harmelen, F., Klein, M., and Akkermans, H. (2000). "On-ToKnowledge: Ontology-based tools for knowledge management," in *The eBusiness and eWork 2000 Conference (EMMSEC 2000),* P. I. Kidd, Ed. Madrid: Cheshire Henbury.

Fensel, D. and Motta, E. (2001), 'Structured Development of Problem Solving Methods', *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 913–32.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

Frantz, C. K., & Nowostawski, M. 2016, 'From institutions to code: Towards automated generation of smart contracts', *IEEE International Workshops on Foundations and Applications of Self Systems,* pp. 210-15.

Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaría, V. 2018, 'Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?', *Future Internet*, vol. 10, no. 2, p. 20.

Gennari, J.H., Tu, S.W., Rothenfluh, T.E., & Musen, M.A. 1994, 'Mapping Domains to Methods in Support of Reuse', *International Journal of Human-Computer Studies,* vol. 41, pp. 399–424.

Gomez-Perez, A. and Corcho, O. (2002). "Ontology specification languages for the semantic web," *IEEE Intelligent Systems,* vol. 17, pp. 54-60.

Gopalan, R., Nanda, V. and Yerramilli, V. 2011, 'Does Poor Performance Damage the Reputation of Financial Intermediaries? Evidence from the Loan Syndication Market', *The Journal of Finance*, 66, pp 2083–2120.

Gruber, T.R. 1993, 'A translation approach to portable ontology specifications', Knowledge Acquisition.

Hendrikx, F., Bubendorfer, K., & Chard, R. 2015. 'Reputation systems: A survey and taxonomy', *Journal of Parallel and Distributed Computing*, vol. 75, pp. 184-97.

Han, E. 2018, 'Very poor': GP booking service HealthEngine sanitises patient reviews', *The Sydney Morning Herald,* June 9.

Hoffman, K., Zage, D., and Nita-Rotaru, C. 2009, 'A survey of attack and defense techniques for reputation systems', *ACM Computing Surveys*, vol. 42, no. 1, pp. 1–31.

Holt, C. C. 2004, 'Forecasting seasonals and trends by exponentially weighted moving averages', *International journal of forecasting*, vol. 20, no. 1, pp. 5-10.

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. 2016, 'Evaluation of logic-based smart contracts for blockchain systems', *International Symposium on Rules and Rule Markup Languages for the Semantic Web,* Springer, pp. 167-83.

Josang, A. 2007, ‘Trust and Reputation Systems’, *Foundations of Security Analysis and Design.*

Josang, A. 2009, ‘Challenges for Robust Trust and Reputation Systems’, *Elsevier Science*.

Josang, A., & Pope, S. 2005, ‘Semantic constraints for trust transitivity’, *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling,* vol. 43, pp. 59-68.

Josang, A., & Presti, S. L. 2004, ‘Analysing the relationship between risk and trust’, *International Conference on Trust Management*, Springer, Berlin, pp. 135-45.

Josang, A. Hayward, R. Pope, S. 2006, ‘Trust Network Analysis with Subjective Logic’, *Proceedings of the 29th Australasian Computer Science Conference*, vol. 48, pp. 85-94.

Josang, A. Ismail, R. Boyd, C.   2007, ‘A survey of trust and reputation systems for online service provision’, *Decision Support Systems,* vol. 43, no. 2, March, pp. 618-44.

Juels, A., Kosba, A., & Shi, E. 2016, ‘The ring of gyges: Investigating the future of criminal smart contracts’, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, pp. 283-95.

Khaqqi, K. N., Sikorski,J. J., Hadinoto, K., & Kraft, M. 2018, ‘Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application’, *Applied Energy,* vol. 209, January, pp. 8-19.

Kerr, R. and Cohen, R. 2016, ‘Modeling trust using transactional, numerical units’, *in Proceedings of the International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, pp. 21:1–21:11.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. 2016, ‘Hawk: The blockchain model of cryptography and privacy-preserving smart contracts’, *IEEE Symposium on Security and Privacy (SP),* pp. 839-58.

Lauslahti, K., Mattila, J., & Seppala, T. 2017, ‘Smart contracts–How will blockchain technology affect contractual practices?’, *ETLA Reports No. 68*.

Magazzeni, D., McBurney, P., & Nash, W. 2017, ‘Validation and Verification of Smart Contracts: A Research Agenda’, *Computer*, vol. 50, no. 9, pp. 50-7.

Malaga, R. 2001, ‘Web-based reputation management systems: Problems and suggested solution” *Electronic Commerce Research*, vol. 1, no. 4, pp. 403–417, 2001.

Manchala, D. W. 1998, ‘Trust metrics, models and protocols for electronic commerce transactions’, *Proceedings of the 18th International Conference on Distributed Computing Systems,* IEEE, pp. 312-321.

Marcus, S., Stout, J., & McDermott, J. 1988, ‘VT: An Expert Elevator Designer that Uses Knowledge-Directed Backtracking’, *AI Magazine*, vol. 9, no. 1, pp. 95–112.

Marino, B., & Juels, A. 2016, ‘Setting standards for altering and undoing smart contracts’, *International Symposium on Rules and Rule Markup Languages for the Semantic Web,* Springer, Cham, pp. 151-66.

Mark, G. (2017). Is a ‘smart contract’ really a smart idea? Insights from a legal perspective. Law School, University of Adelaide, Adelaide, SA, Australia. Available online 5 June 2017.

Nicapotato (2018) Women’s e-commerce Clothing Review. https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews.

Hendrikx, F., Bubendorfer, K., and Chard, R. (2014). Reputation systems: A survey and taxonomy. Journal of Parallel and Distributed Computing. 75. 10.1016/j.jpdc.2014.08.004.

Papazoglou, M. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering. 10.1109/WISE.2003.1254461.

Park, J.Y., Gennari, J.H., & Musen, M.A. 1998, ‘Mappings for Reuse in Knowledge-Based Systems’, In *Eleventh Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Alberta.

Philip, J., Kevin, T., and Delvin, D. 2017, *‘A framework for building reputation system’*

Rachel, W. (2020). Vitalik Buterin Reveals Ethereum 2.0 Roadmap to Cointelegraph. https://cointelegraph.com/news/vitalik-buterin-reveals-ethereum-20-roadmap-to-cointelegraph

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.

Resnick, P. Zeckhauser, R. 2015, 'Trust among strangers in internet transactions: Empirical analysis of eBay' s reputation system', *Economics of the Internet and E-commerce.*

Resnik, R. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

Schaub, A., Bazin, R., Hasan, O., & Brunie, L. 2016, 'A trustless privacy-preserving reputation system', *IFIP International Information Security and Privacy Conference,* Springer, Cham, pp. 398-411.

Sebastian, B. and Dubravka, C. (2015). On being 'systematic' in literature reviews in IS. *Journal of Information Technology.* 30. 10.1057/jit.2014.26.

Sergi, D., Cristian, T., and Jordi, H. 2016, 'Reputation and Reward: Two Sides of the Same Bitcoin', *National Center for Biotechnology Information, U.S. National Library of Medicine,* 8600 Rockville Pike, Bethesda MD, 20894 USA

Sherman, L. 2018, 'A Decentralized Reputation System', *How Blockchain Can Restore Trust In Online Markets.*

Soska, K., Kwon, A., Christin, N., & Devadas, S. 2016, 'Beaver: A Decentralized Anonymous Marketplace with Secure Reputation', *IACR Cryptology ePrint Archive*, p. 464.

Tadelis, S. 2016, 'Reputation and feedback systems in online platform markets', *Annual Review of Economics*, vol. 8, pp. 321-40.

Tonelli, R., Destefanis, G., Marchesi, M., & Ortu, M. 2018, 'Smart contracts software metrics: a first study', *arXiv preprint arXiv:1802.01517*, Elsevier.

Vandervort, D. 2014, ‘Challenges and opportunities associated with a bitcoin-based transaction rating system’, *International Conference on Financial Cryptography and Data Security,* Springer, Berlin, pp. 33-42.

Vukolić, M. 2017, ‘Rethinking permissioned blockchains’, *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pp. 3-7.

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. 2016, ‘Blockchain contract: Securing a blockchain applied to smart contracts’, *IEEE International Conference on Consumer Electronics (ICCE),* pp. 467-68.

Wiederhold, G. 1992, ‘Mediators in the Architecture of Future Information Systems’, *IEEE Computer*, vol. 25, no. 3, pp. 38–49.

Wohrer, M., & Zdun, U. 2018, ‘Smart contracts: security patterns in the ethereum ecosystem and solidity’, *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE),* IEEE, pp. 2-8.

Wood, G. 2014, ‘Ethereum: A secure decentralised generalised transaction ledger’, *Ethereum Project Yellow Paper*, vol. 151, pp. 1-32.

Xu X., Pautasso C., Zhu L., Gramoli V., Ponomarev A., Tran A., and Chen S. 2016, ‘The blockchain as a software connector’, *In: 13th Working IEEE/IFIP Conference on Software Architecture (WICSA),* pp. 182-191, IEEE.

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., ... & Rimba, P. 2017, ‘A taxonomy of blockchain-based systems for architecture design’, *2017 IEEE International Conference on Software Architecture (ICSA),* pp. 243-52.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., and Smolander, K. (2016) Where is current research on blockchain technology?–a systematic review. PLoS One, 11 (10) (2016), pp. 1-27, 10.1371/journal.pone.0163477.

Yu, H., Kaminsky, M., Gibbons, P. B., & Flaxman, A. 2006, ‘Sybilguard: defending against sybil attacks via social networks’, *ACM SIGCOMM Computer Communication Review,* vol. 36, No. 4, pp. 267-78.

Zhang, F., Cecchetti, E., Croman, K., Juels, A., & Shi, E. 2016, 'Town crier: An authenticated data feed for smart contracts', *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security,* pp. 270-82.

Zyskind, G., & Nathan, O. 2015, 'Decentralizing privacy: Using blockchain to protect personal data', *Security and Privacy Workshops (SPW),* IEEE, pp. 180-84.