Data Trading based on Seller Preferences within Blockchain Smart Contract

Thesis for the MPhil

By

Naeem Bilal

College of Engineering, Design and Physical Sciences

Department of Computer Science

Brunel University

November 2022

ABSTRACT

Online data trading has not focused on the necessary control of data selling by the data seller preferences (DSP) using blockchain technology. This research aims to explore the DSP using smart contract over blockchain within the domain of online data trading. Data trading has been carried out for several decades, but cutting-edge technologies and cloud services have grown dramatically worldwide. Industries are gaining benefits from accessing the data that enabled them to perform mission-critical tasks by performing data analysis on the massively available data and getting a higher return on investment (ROI).

This research aims to make online data trading possible only if the buyer can satisfy the conditions predefined by the seller. For example, DSP can restrict the data purchase if the participating buyer is doing business from a specific geographic location, or it can further restrict a particular type and size of business. So, data trading will be controlled by smart contract validation based on DSP hence the novel DSP artefact has been achieved and evaluated via a personal blockchain Ganache, which is always set to automatics mining. Even though the DSP Dapp artefact has been explored with a limited scope of seller preferences and data volume, future researchers may evolve the DSP Dapp artefact framework to achieve complex seller preferences such as ethical selling (e.g., green credentials). The smart contract serves as an automated contract depending on DSP,

between seller and buyer, without the involvement of any broker or third party.

After the first chapter's introduction has set up the context for chapter two to review the literature, present the research question, and set the aims and objectives. Chapter three selected the DSR methodology for this research and analysed the requirements to set the building block for chapters four and five. Chapters four and five fulfilled objective two by designing and developing the DSP artefact using a smart contract to control data trading. Chapter 6 validated the DSP trading system to confirm the novelty of this research, and finally, chapter 7 summarised the contribution and future research.

The research proposes a new approach to online data trading that controls the data selling depending on DSP within smart contract over blockchain and opens new doors for the researchers for future work in this area.

DEDICATION

I dedicated this work to God almighty, to whom all glory shall always be, for giving me the strength and motivation to finish my thesis.

I would also like to dedicate to my sweet parents, wife, friends and thesis supervisor.

ACKNOWLEDGEMENTS

First, thank my principal thesis supervisor Dr David Bell of the Department of Computer Science at Brunel University. He consistently allowed this paper to be my own work but steered me in the right direction whenever he thought I needed it. The door to Dr Bell's office was always open whenever I encountered a trouble spot or had a question about my research or writing.

I would also like to thank Dr Alan Serrano-Rico and Professor Panos Louvieris. With their passionate participation and input, the research was successfully conducted.

Finally, I express my profound gratitude to my parents and my wife for providing me with unfailing support and continuous encouragement throughout my years of study and through researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Author

Naeem Bilal

DECLARATION

I hereby declare that this thesis represents my work done after registration for the degree of MPhil at Brunel University UK and has not been included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualification.

I have read the University's current research ethics guidelines and accept responsibility for the conduct of the procedures by the University's rules and regulations. I have attempted to identify all the risks related to this research that may arise in conducting this research, obtained
The relevant ethical and/or safety approval (where applicable) acknowledged my obligations and the rights of the participants.

Naeem Bilal

Signature: ________________

Date: ____________________

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DSR	Design Science Research
ROI	Return on Investment
DSP	Data Seller Preferences
DSPM	Data Seller Preferences Model
GDRP	General Data Protection Regulation
IoT	Internet of Things
BPMNS	Business process model and notation
UML	Unified Modelling Language
Dapp	Distributed Application
ERD	Entity Relation diagram
C2B	Customer to Business
DAT	Data Token

CHAPTER 1: INTRODUCTION

1.1 Introduction

Data is a collection of behaviour related to observation, facts, and measurements. The world economy is based on business growth and how business approaches the ways to gather data from different sources to enhance the economy by utilizing that data in the right direction (Kurzer, 2018). Relevant data helps people to make the right decisions, identifies your business needs, models building in business processes, analyse organisational behaviour, speeds up productions, gives better use of resources, proven techniques of cost, time-saving and finally, methodology to improve ROI (Sheehy, 2020). Data trading is not a new concept and is widely used to achieve benefits in the economic building.

The businesses involved in global data trading focused on providing access to data among different nations and cross-industries. Data sellers will get some benefits once their data has been sold, but they cannot predefine any preferences to restrict certain buyers from purchasing their data. Hence the aim of this research is to enable the data sellers to predefine their preference to control the data selling using a smart contract. Seller preferences are like a contract which needs to be executed just before the data purchase by the buyer. For example, a seller can define their preferences not to sell their data to any buyer from Russia. If any buyers from Russia try to purchase the data, the transaction will get rejected otherwise accepted. To govern the data purchase process, the smart contract over blockchain proved to be suitable technology, as it can be programmed and executed automatically with security, reliability and efficiency (Governatori et al., 2018).

The current research will leverage online data trading by incorporating DSP within the smart contract, executed automatically without third-party or human involvement over the blockchain. This research covers the C2B (Rayport, 2000) trade in which the seller is involved in data selling, and the buyer purchases the data according to the DSP defined by the seller.

This chapter will set up the context of this research, section 1.2 covers the overview of the topic, 1.3 related work that supports the research, 1.4 approaches of this research, 1.5 contributions of research, and finally, 1.6 section describes the structure of the study.

1.2 Topic Overview

Data trading is simply buying and selling data among individuals (Duncan, 2018). Data is considered this century's Oil (Arthur, 2013). Data can be composed of personal, social (Frankenfield, 2020), financial/non-financial (Sherman, 2019) and behavioural information. Therefore, many public and private sectors will be eager to analyse and consume the data for their purposes, eventually leading to data trading.

Companies are involved in data trading by collecting consumer data. "They want to observe a customer's purchasing pattern to provide tailored experiences" (Feit, 2019). Most companies, including social media & Internet services, keep collecting and selling their customer data, which is unfair. In contrast, companies act as data brokers by providing a platform that will benefit the customer directly once they sell their data. Collecting and selling personal data has become a business worth $200 billion, which is likely to indicate the continued growth of the data-brokerage business (Tucker and Neumman, 2020). These companies formed a marketplace for buying and selling data online, termed “big data” (List of big data companies, Wikipedia 2020). To control the transparency and trust of consumer data, the general data protection regulation (GDPR) law Data have introduced by the European Union on May 25, 2018 (Eckerson, 2017). Even though sellers can sell their data directly, they are not privileged to restrict data purchases according to their preferences. DSP will enable the seller to predefine their preferences to restrict sales if, let us say, stop the data purchase if the interested buyers belong to a specific geographic location, size or type of their business. DSP artefact is the leading research focus; hence the seller's preferences parameters have been kept more straightforward; however, these parameters will be open to extension for more complex use cases for future researchers.

The selling of personalised data has become a concern because of the increasing data exploitation and insecurity issues against the usefulness of the value-added services provided through Internet advancement (Tian, 2019). In realising the same concerns, blockchain technology by design greatly supports data trading in many ways, including rewarding cryptocurrencies to the seller after selling their data, data security and auditing. Most importantly, the smart contracts used for agreements between the parties involved in data trading are essential in this scenario.

This research focuses on restricting the data trade between a seller and buyer, keeping the fact that the data can only be sold to the buyer if they fulfil specific criteria based on DSP. The seller defines these criteria and is further executed by the smart contract, a programable contract based on blockchain technology. A smart contract is an innovative way of instigating trade agreements within a blockchain to ensure the validity of seller preferences predefined by the seller. An agreement is required to validate the preferences against the buyer's profile, executed automatically during the data trading process without human or third-party involvement. The smart contract over a blockchain is an exciting technology. It has many reference implementations, such as real estate, rent management, electric vehicle charging stations, flight insurance, custom contracts, and invoice financing (Polyswarm, 2018). However, the concept of DSP will be explored, implemented and validated in this research using smart contract over a blockchain.

The study conducted (Feng, 2018) has also attempted to explore the challenges that have questioned the adoption and use of technology for data trading and exchange. The technology has been growing until it needs to rectify its weaknesses, providing a more significant customer experience. Notably, big data technology through data trading has provided a platform for businesses to experience sharing and exchanging data with the enhanced utility of data (Gao, 2018). Furthermore, with the advancement of the Internet and communication technologies, big data has significantly contributed to online data trading. The emergence of high-speed Internet has provided new dimensions to conventional information and communication technology methods. However, more importantly, it has enabled the use of digital assets in the form of cryptocurrencies and blockchain technology, especially within the context of financial, social, e-commerce and economic sectors.

The modern concept of data trading relates to data owners, trading their data as per their desired cost and preferences. DSP artefact favour data access over data ownership to avoid data exploitations. Data will stay within the premises of DSP infrastructure; however, any commercial reporting applications within the domain of DSP infrastructure will have restricted access to raw data. It is necessary to highlight that because of the emergence of high-speed Internet connectivity, data trading still exists; nevertheless, the massive data market is yet to be considered for data consumers (Li, 2017).

1.3 Motivation and Purpose of Research

The maturity and adoption of blockchain and big data technologies are providing confidence to the researchers and becoming a close match for the questions they are looking for (Meijer, 2019).

Even though prior researchers have made efforts in elaborating the effectiveness of data trading using blockchain technology, the literature needs to provide conclusive evidence that shows the process of data trading control by the seller preference via smart contract. The central theme behind conducting this research is to comprehend how data trading is controlled by imposing the DSP with the help of cutting-edge blockchain technology using smart contracts.

Effective data trading, an emerging theme in the big data market, has also started facing considerable challenges, which have become essential to resolve (Zhao, 2019). One of the challenges the data sellers face is that as they sell their data, they need a way to control to whom their data will be sold based on predefined preferences. So, the purpose of the research is to control online data trading based on the DSP over the blockchain. Before a data purchasing process, DSP will be automatically validated using smart contract over a blockchain and will ensure that the data will be sold as per the preferences defined by the seller. Buyers must comply with the predefined seller preferences; otherwise, the transaction will be rejected. For example, the seller can define preference so that his data cannot be sold to any buyer from Russia. This way, the DSP smart contract will reject the transaction if the interested buyer's geographic location is Russia. It is notable from the example that the preferences parameters seem simpler as the main focus is to explore the DSP artefact.

1.4 Research Approach

The study is supported by qualitative measures where the evidence has been collected utilising secondary sources. To analyse data, content and data analysis technique has been followed. This involves formulating themes in which the triangulation of generalised and case-specific findings proceeds. Starting from the introduction, identifying the research problem and area will be studied by reviewing the relevant literature and formulating research hypotheses. After reviewing the literature, the focus will be to find the research method, tools and requirements artefacts. Next, the DSP design artefact will be built, followed by the implementation of the DSP artefact. Then the DSP will be evaluated, and the research will conclude with the future aspects.

1.5 Contribution or Significance

This research will make an essential contribution in minimising the research gap in a manner to which the prior researchers should have given more attention regarding data trading controlled by the seller preferences using smart contract over the blockchain. For example, sellers can set their preferences due to their beliefs or otherwise not sell their data to specific business types, such as the betting or porn industry.

The following research contributes to the data trading realising the DSP artefact incorporating smart contract technology so the sellers can define their preferences. The DSP preferences smart contract will execute automatically before the data purchase to accept or reject the transaction to ensure that the interested buyers always comply with the predefined sellers' preferences.

1.6 Structure of Study

The study comprises the following seven sections.

1- *Introduction:* The first chapter of this thesis is the introduction chapter, where the researcher intends to inform the audience regarding the generalised concepts about the area of research. This chapter will provide the rationale, motivation, aim and objective of this research.
2- *Literature Overview:* With the support of previous secondary evidence collected from journal articles and literature reviews, the researcher will form the theoretical construct for the study. After the gap analysis, the research question will be presented.
3- *Research Methodology:* The methodological section entails the methods, techniques and tools that helped the researcher formulate this research work. In this section, the researcher has described in detail the research method that has assisted the researcher in compiling this research work. This particular section mainly develops theoretical and conceptual modelling for the proposed hypothesis.
4- *Designing of DSP:* The fourth chapter of the research is devoted to the DSP design artefacts. In this section, the researcher produces the DSP design artefacts, supporting the appropriate methodology, techniques and construct discussed in chapter three.
5- *Implementation of DSP artefact with Smart contract:* The fifth chapter of this research is dedicated to implementing the seller preferences with smart contract. The design artefacts produced in the previous chapter four will be used to implement models, including the

preferences smart contract to conclude the DSP implementation artefact.

6- *Validation of DSP:* Chapter six evaluates the DSP artefact developed in chapter five. Set of evaluation test cases extracted from the requirements defined in chapter three. Then these test cases are translated to the Postman scripts to be tested automatically to evaluate the DSP artefact.

7- *Contribution and future research:* The last section of the study will present the significant contribution of the research from a theoretical and practical perspective. This section highlights the significance of the research and provides recommendations and openings for future researchers.

1.7 Summary

The context has been set up by introducing and history of data trading to begin this research. This chapter then highlights the DSP artefact's relevance as the topic overview's leading research focus. The motivation and purpose of the research mentioned as controlling online data trading based on the DSP over the blockchain. Furthermore, the research approach and contribution are discussed in sections 1.4 and 1.5. Finally, to section 1.6 presented the structure of the study. The next chapter is related to the literature review to analyse the existing data trading over blockchain research and find the research question.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter involves critical findings based on literature, identifying gaps and specifying the research question. The review aims to explore the related work carried out in online data trading using smart contracts over the blockchain. The literature review comprises of blockchain-based trading platform for big data — a secure data trading system based on blockchain, trust, reputation, pricing, trading and protection. The smart contract-based data trading mode using blockchain and machine learning. After reviewing, the gap has been identified to incorporate the DSP smart contract to give back control to the data seller to predefine their preferences using the smart contract. This smart contract will execute automatically before the data purchase and either accept or reject the transaction. For instance, the seller can set up a preference not to sell their data if the buyer is from Russia. In this case, if any buyer from Russia tries to purchase the data, then the smart contract will reject the transaction. It is noted that the preferences parameters have been kept more straightforward to increase the focus to contribute to the DSP artefact. Future researchers will extend these parameters to more complex ones.

Chapter one has provided the domain of this research and this chapter will define the research question with aims and objectives. The introduction is covered in section 2.1 section 2.2 illustrated the background of online data trading, section 2.3 is about Ethereum smart contract blockchain in data trading, section 2.4 focused on related research of data trading using smart contract over blockchain, section 2.5 focused on the gap analysis of the related research, 2.6 describes research question, 2.7 discussed the aims & objectives, and finally a summary of the chapter is discussed in section 2.8.

2.2 Background of online Data Trading

Organisations have been buying and selling consumer data for decades. World Wide Web, launched in 1989, is the platform on which all the content available on the Internet is accessible. By 2016, more than 3.4 billion people worldwide were connected online (Max et al., 2016). Most of these people are the providers of the vast amount of consumer data that is being traded worldwide. This massive data generation could be categorised into five categories (Enjolras, 2014), i.e. web & social media data, machine-to-machine data (IoT), transaction data, biometric data, and human-generated data. All this data combined make up the idea of "big data", which can be defined as "the capacity to search, aggregate and cross-reference large data sets" (Lyon, 2014). Knowing as much as possible about what a consumer is interested in, their lifestyle, living situation, income, buying habits and even their feelings allow firms to increase marketing and advertising efficiency (Enjolras, 2014). Companies that do not directly process and analyse big data could offer their collected data for sale and their online platforms to advertisers. Then later, in the 1990s, due to the popularity of the Internet worldwide, consumer data collection was made by digital advertisement companies, mainly in the area of commerce, when advertising companies get an advantage through email marketing (Kurzer, 2018). Loyalty marketing played a key role in grabbing customer data for marketing purposes. As mentioned earlier, big data has a major contribution to online data trading which can be clearly seen in figure 2.1.

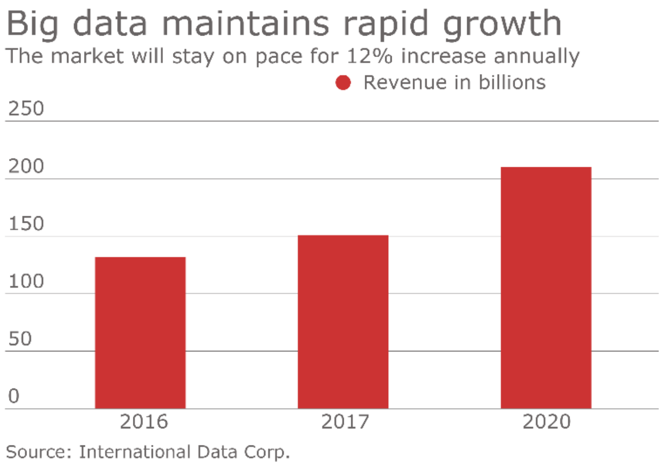

Figure 2.1: Big data rapid growth

Over time, accessible online services such as Google, Facebook, Youtube and Instagram have collected more than a petabyte of personal data. Later in the mid of 20th century, Data brokers have begun building databases to classify people and their habits for marketing, fraud detection or credit scoring purposes (Marr, 2020). The data broker industry is worth about $200 billion, including Credit Bureaus Experian, Equifax and TransUnion, which maintain files on millions of Americans (Lazarus, 2019).

Consequently, the journey is limitless; that was started by collecting data from online commerce sources and email and loyalty marketing and then involved data capturing from users on social media, infotainment and search engines. Hence organisations have been earning billions of dollars of profit from the consumer's data, so there was a need for an online data trading platform which could directly benefit data owners. Next, sections 2.3 and 2.4 will discuss blockchain technology and its suitability for data trading platforms.

2.3 Ethereum Smart Contract Blockchain in Data Trading

The traditional method of a data trading channel involves the participation of three parties: buyer, seller and middleman, which facilitates the exchange of data (Liang et al., 2018). In the traditional method of data trading channel, the seller who wishes to sell data initially sends the data to a reliable third-party data exchange platform that is trusted by many users (Dai et al., 2019). After selecting an appropriate selling price, the buyer will choose an exciting data product and place the order to purchase the data, which is commonly done on standard e-commerce websites (Feng et al., 2018). Figure 2.3 shows the process of traditional data trading.

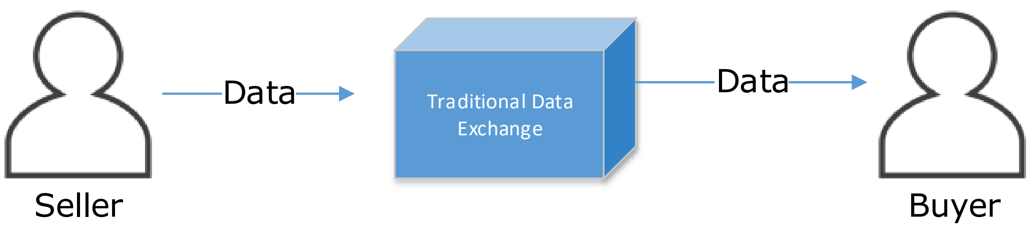

Figure 2.2: Transactions on Information Forensics and Security (Dai et al., 2019)

In traditional data trading channels, some issues are concerned with accountability and lack of transparency (Zuo et al., 2017). In the traditional channel, there are chances that the data may be tampered with without the buyer being knowledgeable of the fact. Since it has opaque transparency, the buyer has no method to know if the data has been tampered with or not. Furthermore, another issue can arise because the buyer may further sell raw data to anyone else without knowing the original seller. As stated, (Nguyen, 2016), "a dishonest buyer will further tamper the data and then resell it to another party".

"Nakamoto conceptualised the first blockchain, which states the framework for creating an online payment network that would transfer payments directly between the participants without needing an authorised financial institution (Nakamoto, 2008), from where the technology has evolved and found its way into many applications beyond cryptocurrencies" (Iredale, 2020). The blockchain is a mature decentralised technology which is the backbone of many cryptocurrencies and financial and non-financial industries. Blockchain is immutable, distributed, transparent, decentralised, secure, supports consensus, and is unanimous with faster settlement. These features make blockchain ideal for data trading over the traditional channel.

We may not go into the details of the blockchain feature list as the blockchain is an existing technology and so much technical and theoretical literature is already available. Using blockchain technology for a data trading channel, users can upload the data directly into the network and award valuable tokens. Buyers can then view which data is available for trade, and as per the buyer's interest, they can choose their relevant data

(Kiyomoto et al., 2017). Furthermore, it will allow the buyer to view the history of the data to find out if the data has been tampered with or not, thus, authenticating the audit process and making every detail transparent to each user within the network (Cong *et al.*, 2019).

In 2013, a young developer who was the co-founder of Bitcoin Magazine, named Vitalik Buterin, envisioned an entirely new form of cryptocurrency, Ethereum, which had more functions than just the transfer of payments. Vitalik and his team employed the use of a crowdfunding campaign to kick-start Ethereum (Gichigi, 2018). After Ethereum succeeded in the public domain, it created its blockchain network. Further, it introduced the concept of the smart contract, which, within the Ethereum blockchain network, had program blocks that were programmed to automatically execute a smart contract set by trading parties (ICAEW, 2019). Ethereum provides the programable interface to extend blockchain technology, making it the best suitable technology for data trading. Ethereum smart contract can be used to award valuable tokens to the data seller after successful data sales. These tokens provide more control and internal branding to the new data trading platform and act as an extension of the world-famous cryptocurrency Ether (ETH). Ethereum blockchain users have Ethereum addresses which are used to deploy smart contracts, execute the related functions, perform transactions and contribute to the communication of the chain through smart contracts (Xiong *et al.*, 2019). Smart contracts allow the interested parties to set the terms and conditions. Then these smart contracts will be able to execute automatically, cutting down the middle man during the lifecycle of the purchase process (Liang et al., 2018). Therefore, the data trading channel, by employing the use of blockchain technology, is preferred by many organisations and services as it allows them to trade data between the concerned parties seamlessly and makes the technology of blockchain preferable for use in these services.

2.4 Related Research of Data Trading Using Smart Contract

Zheng et al. (2020) found that data is an extremely important asset. Governments around the world encourage big data sharing and trading to flourish big data economics (Jung et al., 2017). However, existing data trading platforms are not fully trusted. Data providers lack control over data trading rights. Meanwhile, the data price model needs further justification. These issues hinder the development of big data trading. They proposed a blockchain-based decentralised data trading platform, on which data providers can better control data trading. Accordingly, they designed smart contracts for distributed data trading and set rules to assign data rewards in trading (Zheng et al., 2020). The contract guarantees the efficiency of data matching and fair reward distribution. They also adopted proxy re-encryption to ensure the security of data transmission. They verified the security, availability, and efficiency of the proposed big data trading platform.

Camilo et al. (2020) explored that every citizen has the right to privacy and, therefore, the right to control their personal information, deciding to whom, when, and where their information is available. They proposed a secure, agile, and effective system for distributed, automatic, and transparent data trading between domains using permissioned blockchain, smart contracts, trust, and reputation. They developed and implemented a prototype of a trust and reputation system based on real-life interactions (Putra et al., 2020). The results show that the proposed system provides security and privacy in a quick and distributed way, performing hundreds of transactions per second, and effectively punishing malicious behaviour.

Liang et al. (2018) considered big data to be the key to unlocking the next great waves of growth in productivity. The amount of collected data in our world has been exploding due to a number of new applications and technologies that permeate our daily lives, including mobile and social networking applications, and Internet of Thing-based smart-world systems (smart grid, smart transportation, smart cities, and so on) (Kim et al., 2014). With the exponential growth of data, how to efficiently utilise the data becomes a critical issue. This calls for the development of a big data market that enables efficient data trading. Via pushing data as a kind of commodity into a digital market, the data owners and consumers are able to connect with each other, sharing and further increasing the utility of data (Mussa and Rosen, 1978). Nonetheless, to enable such an effective market for data trading, several challenges need to be addressed, such as determining proper pricing for the data to be sold or purchased, designing a trading platform and schemes to enable the maximisation of the social welfare of trading participants with efficiency and privacy preservation, and protecting the traded data from being resold to maintain the value of the data. They conducted a comprehensive survey on the lifecycle of data and data trading. To be specific, we first study a variety of data pricing models, categorise them into different groups, and conduct a comprehensive comparison of the pros and cons of these models. Then, they focused on the design of data trading platforms and schemes, supporting efficient, secure, and privacy-preserving data trading (Brakerski and Vaikuntanathan, 2014). Finally, they reviewed digital copyright protection mechanisms, including digital copyright identifier, digital rights management, digital encryption, watermarking, and others, and outlined challenges in data protection in the data trading lifecycle.

Dai et al. (2020) found that Data, a key asset in our data-driven economy, has fuelled the emergence of a new data trading industry. However, there are a number of limitations in conventional data trading platforms due to the existence of dishonest buyers/data brokers (Zuo et al., 2018). To mitigate these limitations, they posited the importance of a data processing-as-a-service model, which complements the conventional data hosting/exchange-as-a-service model. Specifically, they introduced a secure data trading ecosystem and presented a new blockchain-based data trading ecosystem (hereafter referred to as SDTE). In the ecosystem, both data brokers and buyers are not able to obtain access to the seller's raw data, as they are only getting access to the analysis findings that they require. In other words, they reduced the challenge of securing the dataset to the challenge of securing the data processing. They also build a security model to analyse the data trading market and describe a new set of trading protocols for the entire data trading market. To demonstrate utility, they implemented their proposed secure data trading platform (SDTP) on Ethereum & Intel's Software Guard Extensions (SGX) and performed an in-depth analysis (McKeen et al., 2013).

Xiong (2019) learned that there are two traditional data trading modes, the hosting mode, and the aggregation mode, which depend on trusted third parties to a large extent. The hosting mode is that the data are completely hosted in the data trading centre, so the data trading centre retains the data. On the surface, the aggregation mode is that the data trading centre does not retain the data of trading, but actually, it has the ability to retain the data. There is a fundamental difference between the ability to retain the data and the inability to retain the data. These two trading modes cause the data owners to be afraid to share data trading (Yang, 2016). In this paper, they proposed a solution to the data trading mode based on the smart contract using blockchain and machine learning. Their solution took advantage of the immutability, tamper-proof and traceability of blockchain, the programmability of smart contract, and the verification of data availability by the similarity learning to propose a challenge-response mechanism between the data purchaser and the data owner, an off-chain download mechanism between the data purchaser and the data storage service provider, and an arbitration mechanism for the controversy resolution of the data trading. The challenge-response mechanism is used to authenticate and authorise the data owner, the off-chain download mechanism is used to authenticate and authorise the data purchaser to download the purchased data, and similarity learning is used to deal with the controversy over the data availability in the data trading. The design and implementation of the data trading smart contract successfully achieved the goal of removing the trusted third party in the data trading, and thus, the problem that the data trading centre has the ability to retain the data in the process of the data trading is solved, as well as the automatic payment by using the Ethereum encrypted currency among the trading participants is realised (Cao, Chen and Liu, 2017). They presented the whole process of smart contract from the design and implementation to the test completion and provided the security analysis and performance evaluation.

Weber and Prinz (2019) considered that user data has become a valuable economic resource for many companies. While on the one hand user data may help companies improve their services and gain a better understanding of potential customers, on the other hand, the proliferation of user data is usually accompanied by a loss of user privacy. In the case of platforms like Facebook, Twitter, and Google, users are essentially trading their personal information for services and are often unaware of the potential consequences this trade may have on their private life (Markovikj et al., 2013). They presented an approach that allows users to store their data on their own smartphones and have full control over their data, allowing them to specify which companies or organisations obtain their personal data. At the same time, blockchain technology will be used to realise a system for reliable negotiations involving user data without a central entity. Their approach made it impossible for either the data provider or data collector to perform manipulations or cheat once a data exchange contract has been signed.

2.5 Gap Analysis of the Related Research

The following table extracted the research summary from section 2.4 to provide a comparison of what already has been covered by the prior research, and a gap will be identified to build a hypothesis for this research.

S#	Research Summary	Conducted by
1	Designed the smart contracts to assign data rewards in trading. They also adopted proxy re-encryption to ensure the security of data transmission. They verified the security, availability, and efficiency of the proposed big data trading platform.	(Zheng et al., 2020)
2	They proposed a secure, agile, and effective system for distributed, automatic, and transparent data trading between domains using blockchain, smart contracts, trust, and reputation. They developed and implemented a prototype of a trust and reputation system based on real-life interactions and effectively punishing malicious behaviour.	(Camilo et al., 2020)
3	They reviewed digital copyright protection mechanisms, including digital copyright identifier, digital rights management, digital encryption, watermarking, and others, and outlined challenges in data protection in the data trading lifecycle.	(Liang et al., 2018)

4	The data broker and buyers are not able to obtain access to the seller’s raw data, as they are only getting access to the analysis findings that they require. In other words, they reduced the challenge of securing the dataset to the challenge of securing the data processing. We also build a security model to analyse the data trading market and describe a new set of trading protocols for the entire data trading market.	(Dai et al., 2020)
5	They proposed a solution to the data trading mode based on the smart contract using blockchain and machine learning. Their solution took advantage of the immutability, tamper-proof and traceability of blockchain, the programmability of smart contract, and the verification of data availability by the similarity learning to propose a challenge-response mechanism between the data purchaser and the data owner, the off-chain download mechanism is used to authenticate and authorise the data purchaser to purchased data, and similarity learning is used to deal with the controversy over the data availability in the data trading.	(Xiong, 2019)
6	They used blockchain technology to realise a system for reliable negotiations involving user data without a central entity. Their approach made it impossible for either the data provider or data collector to perform manipulations or cheat once a data exchange contract has been signed.	(Weber and Prinz, 2019)

Table 2.1. Summary of the related research area

After reviewing the gap analysis in table 2.1, prior researchers used the smart contracts over blockchain to reward the data sellers, pricing, trading data storage audit, security, tracking, avoid cheat or spoofing & data privacy.

What has already been researched will not be a part of this research; however, there is a need for the data sellers to have more control over the data selling by providing them with a framework of DSP using a smart contract over the blockchain. The DSP will allow data sellers to define their preferences before the data purchase. The DSP smart contract will execute before the data purchase to ensure interested buyers comply with the data seller's preferences. Hence DSP smart contract will accept or reject the transaction based on the predefined preferences of the seller. The seller preferences will be kept more straightforward to focus on the DSP artefact's contribution; however, future researchers can extend the preferences to the more complex ones, such as applying the green credentials. During the formation of DSP artefacts, this research will heavily extend from the related research discussed in section 2.4.

2.6 Research Question

Based on the discussion in the above sections (2.4 and 2.5), the research question is as follows:

“Can a Data Seller Preferences (DSP), using Smart Contracts (SC) over blockchain, be developed to control data selling?”

2.7 Research Aims and Objectives

Prior researchers have already explored the different aspects and suitability of data selling using smart contracts over blockchain (sections 2.4 and 2.5) The aim of this research is to control data selling via Data Seller Preferences using smart contracts over the blockchain. The data seller preferences will enable the data seller to predefine their preferences via smart contract. The seller preferences smart contract will execute automatically before the data purchase process to accept or reject the transaction based on the seller's preferences. For example, a seller can predefine their preferences not to sell their data to a betting company; in this case, if the betting company's buyer tries to purchase the data, then the transaction will be rejected.

Although the preferences model is limited to fewer parameters to validate the implementations, such as geographical location (country) and company structure (size and type), it can be extended further to more complex ones, such as green credentials. In this way, the effectiveness of data trading will enhance.

In fulfilling this aim, the following objectives will be considered:

Objective 1:

Analyse and investigate the major artefacts from the literature related to data trading over the blockchain and capture the domain-specific and novel requirements.

Objective 2:

Design and develop the requirements investigated in Objective 1. Appropriate design tools and technologies are required to build a demonstratable trading system to control the data selling based on DSP.

Objective 3:

Scenario-based integration tests are required to validate the novel or thesis-specific requirements.

2.8 Summary

After the introduction, this chapter presents the background of online data trading and then explains the role of the Ethereum smart contract blockchain in data trading. More importantly, the related research of data trading using smart contract over blockchain was discussed in detail. Chapter one provided the domain of this research, leading us to review the literature in the same domain and focus on the gap analysis to build a hypothesis or research question. So, this chapter concluded the research question along with the aims and objectives.

This research will heavily extend from the literature review and then be able to define the research objectives to drive the DSP artefacts. The next chapter will be focused on research methodology.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

Different mythologies have been used to develop simple and complex systems over the past years. A methodology is a systematic approach to conduct the phases of system development which consist of requirements, activities, tools and techniques based on concrete philosophy for the proposed system. DSRM (design science research methodology) is emphasised in two phases build and evaluate. The build phase defines the problem, objectives, design and development artefacts. Where the evaluation phase involves demonstration with the help of artefacts, evaluation and communication, the proposed data seller preference will be based on the DSR methodology followed by these phases.

Chapter two highlighted the research question with objective one to explore further to find the research methodology and requirement artefact to achieve our next objectives. In this chapter, first in section 3.2, we will select the methodology of this research after comparing related ones. Sections 3.3 & 3.4 provide details of the DSRM methodology and its process steps. Section 3.5 elaborates on DSP's DSRM approach, and section 3.6 focuses on modelling. Further, section 3.7 analyses the major artefacts and extracts the requirements from the existing literature to define the objectives of DSP. Research limitation is described in section 3.7, where 3.8 touch the ethical consideration. Finally, section 3.11 concludes this chapter summary.

3.2 Selection of Research Methodology

As this research is related to IS (Information Systems), we have evaluated and reviewed six IS methodologies to justify that the DSRM is suitable for this research.

SDRM: Systems Development Research Methodology

DSRPM: DSR Process Model

DSRM: Design Science Research Methodology

ADR: Action Design Research

SDSM: Soft Design Science Methodology

PADR: Participatory Action Design Research

	SDRM	**DSRPM**	**DSRM**	**ADR**	**SDSM**	**PADR**
1. Philosophy						
a. Paradigm	Science, Objectivist, Positivist	Science, Objectivist, Positivist	Science, Objectivist, Positivist	Systems, Subjectivist, Interpretive	Systems, Subjectivist, Interpretive	Systems, Subjectivist, Interpretive
b. Objectives	New artefact, improvement	New artefact, improvement	New artefact, improvement	New artefact, improvement, client service and relevance	New artefact, improvement, effectiveness	New artefact, improvement, effectiveness, consensus, emancipation
c. Domain	No specific client	No specific client	No specific client	Single client	Single or multiple clients	Multiple /societal clients

d. Target	CBIS (computer-based IS), IT	CBIS, IT, methods	CBIS, IT, methods		Product or Process	Product, Urban Informatics
2. Model	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified
3. Techni-ques & tools	None	None	None	None	None	None
4. Scope (DSR activities)						
a. Problem assessment	Investigate functionalities and requirements	Awareness of the problem	Identify the problem	Problem formulation	1. Learn about specific problem, 2. Inspire and create the general problem and general requirements	Diagnosing and Problem Formulation (Participative problem setting, Ethnographic study)
b. Design/ framing	Construct conceptual framework	Suggestion	Define objectives of solution	Theory ingrained artifact	2. Inspire and create the general problem and general requirements	Action Planning (Opportunity identification, Participative planning)
c. Design/ making	Architect, analyse & build the system	Development	Design & development	Building & intervening	3. Intuit and abduce general solution, 5. & 7. Design & construct	Action Taking: Design (Participative design, Prototyping & installation)

					specific solution	
d. Evaluation	Observe/ evaluate	Evaluation	Evaluation & Extensive adaptation to daily use	Intervening & Evaluation	4. Ex Ante Evaluation (General), 6. Ex Ante Evaluation (Specific), 8. Ex Post Evaluation	Action Taking: Design; Impact Evaluation: (Ethnographic study, Participative evaluation)
e. Reflection	Develop theories & models, consolidate experience	Reflection & abstraction	Communication	Reflection, learning, formalization	Each of the evaluations includes reflection	Reflection and Learning: (Participative client learning, Design theorising for UI)
5. Outputs	Artefact	Artefact, Theory	Artefact	Artefact, Design Theory	Artefact, Design Theory	Artefact, Design Theory
6. Practice						
a. Background	Academic	Academic	Academic	Academic	Academic	Academic
b. User Base (Google citations on 15/08/2017)	(JMIS 1990) 1293	(Webpage 2004) 725 (book 2015) 509	(JMIS 2008) 2561 (DESRIST 2006) 388	(MISQ 2011) 878	(DESRIST 2009) 157 (DESRIST 2007) 73 (book chapter 2014) 1	(JoCI 2011) 55
c. Participants	DSR researchers,	DSR researchers,	DSR researchers,	DSR researchers,	DSR researchers,	DSR researchers,

	users (evaluators)	users (evaluators)	users (evaluators)	clients, users (evaluators)	clients, users (evaluators)	clients, public, users (evaluators)
7. Product	Article	Website, Book	Articles	Article	Articles	Article

Table 3.1: Comparison of six DSR Methodologies (Adopted from Venable et al.*, 2017)*

In our interpretation, SDRM, DSRPM, and DSRM take a more objectivist, positivistic stance to these activities, while the other three methodologies take a more subjectivist, interpretive stance. DSRM seeks clarity of the researcher's understanding of the problem and its significance. For evaluation, DSRM suggests "observe and measure how well the artefact supports a solution to the problem" (Peffers et al., 2008), a substantially objectivist and positivist position. In contrast, ADR, SDSM, and PADR all specifically include problem formulation based on local (not literature-based) needs and working with client stakeholders in doing so, as well as in the evaluation, which demonstrates a much more subjectivist, interpretive stance (Venable et al., 2017). Hence methodology in line with the precepts of design science research (DSRM) is used for this research project. DSRM is posed as a type of research alongside behavioural, social, and other scientific approaches aiming to understand a phenomenon. By contrast, DSR aims at intervening in a phenomenon; while this may necessitate first understanding it, the practical goal of intervention always predominates in DSRM (Wieringa, 2009). DSRM thus aims at developing practical solutions that can be used by professionals in their field (Hevner et al., 2004). More concretely, solutions or design artefacts - can take the form of constructs, models, methods or instantiations (March et al., 1995).

3.3 Design Science Research Methodology (DSRM)

The design deals with creating some new artefact that does not exist. If the knowledge required for creating such an artefact already exists, then the design is routine; otherwise, it is innovative (Vaishnavi et al., 2008).

Design science research (DSR) is a relatively new approach to research (Reubens, 2016). DSR provides powerful frameworks for IS (Information Systems) studies (Hevner et al., 2004; Nunamaker et al., 1990; March et al., 1995). Design Science Research creates this missing knowledge using design, analysis, reflection, and abstraction (Vaishnavi et al., 2008).

This framework is iterative, and each cycle produces purposefully designed artefacts to address business requirements. The term 'purposefully' means the produced artefacts should offer a 'utility' that addresses unsolved problems or offers a better solution that can enhance existing practices (Vaishnavi et al., 2004).

Knowledge can be generated and accumulated through an iterative process via knowledge-using and knowledge-building activities (Owen, 1997; Takeda et al., 1990).

Figure 3.1 demonstrates the design disciplines: "Knowledge is generated and accumulated through action. The process is shown as a cycle in which knowledge is used creatively to construct (create) works, and works are evaluated to build knowledge" (Vaishnavi et al., 2015).

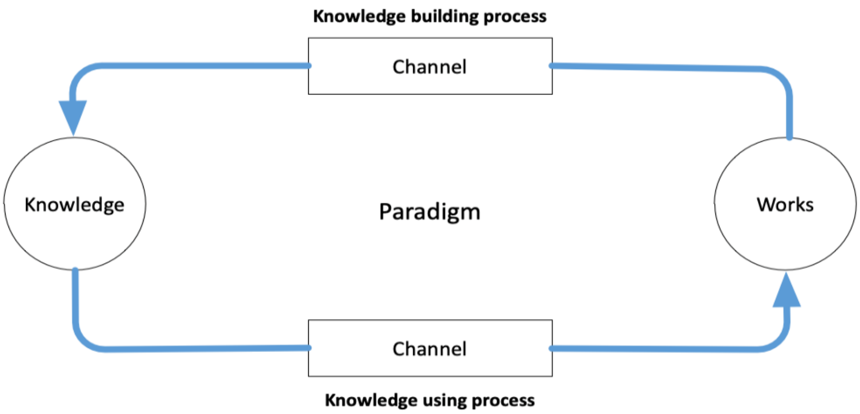

Figure 3.1: Generating and accumulating knowledge (Owen, 1997)

As shown in figure 3.1, knowledge is generated and accumulated through actions. So, the DSR's primary goal is to build an understanding of where constructing artefacts plays an important role. This is the primary differentiation of DSR from pure design, as it is a science of realisation through construction.

DSR process model is shown in figure 3.2. This model adapts a computable design process developed by (Vaishnavi et al., 2015). Various artefacts have been built on understanding a problem to satisfy functional requirements. Further, the evaluation of artefacts contributes to the knowledge area iteratively.

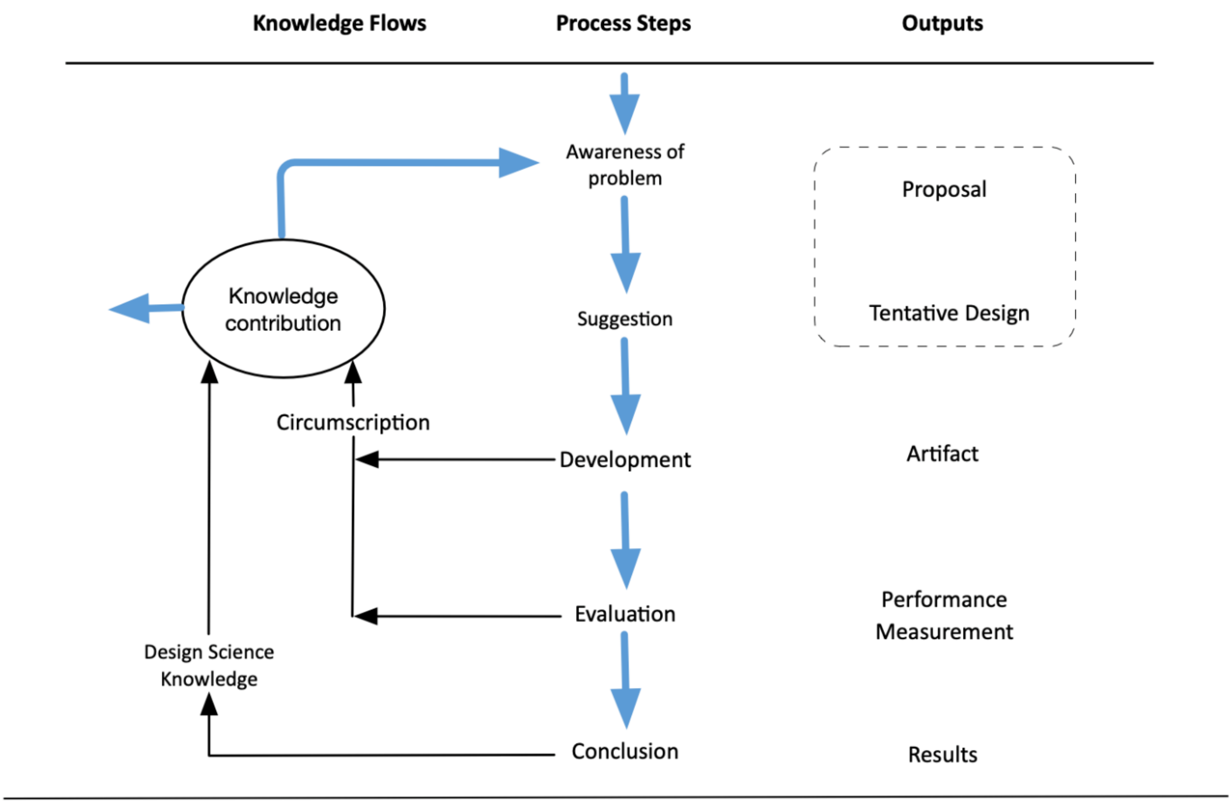

Figure 3.2: Design science research process model (Vaishnavi et al., 2015)

3.4 DSRM Process Steps

The five DSRM processes are followed, from awareness of the problem to the conclusion.

3.4.1 Awareness of Problem

According to figure 3.2, the first process step is the awareness of the problem. The awareness of the problem comes from multiple sources, such as literature or new developments in the industry. New findings in the researcher's field can be found during this process. The outcomes of this step can be formal or informal research proposals for new research.

3.4.2 Suggestion

The next step is the Suggestion. This is a creative step to find new functionality based on the novel configuration of new or existing elements. The output of this phase is a tentative design in which specifications of the candidate solution of the research problem are defined.

3.4.3 Development

Immediately after the suggestion, there is a development phase as per figure 3.2. In this step, the tentative design is further developed and implemented. The artefacts are produced as an output of this phase.

3.4.4 Evaluation

Once developed, the artefacts are evaluated according to the criteria defined in the Proposal (Awareness of Problem phase); therefore, it is critical to develop appropriate evaluation metrics to assess the performance of an artefact and prove the suitability/validity of the evaluation criteria (March et al., 1995). Any new information gained or deviation from the original objective during development and evaluation is fed back to another round of suggestions, as shown in figure 3.2 circumscription arrow. Circumscription is a discovery of constraints and knowledge about theories gained through the detection and analysis of contradictions when things do not work according to theory (McCarthy, 1980). This is an iterative process to meet the solution requirements, whereas the performance measurement is the output of this phase to improve the efficiency and effectiveness of the artefact.

3.4.5 Conclusion

Communication is essential in research (Hevner et al., 2004). Therefore, as a conclusion of a research effort indicated by the small leftward arrow coming out of Knowledge Contribution in figure 3.2, this phase needs to appropriately position the research being reported and make a strong case for its knowledge contribution (Gregor et al., 2013). The effort results are consolidated and "written up" at this phase. The knowledge gained in the effort is frequently categorised as either "firm"—facts and can be repeatedly applied, and may serve as the subject of further research (Vaishnavi et al., 2015).

At the end of iterations, the DSR methodology presented in this research is evaluated according to the following guidelines to understand the effectiveness of the approach followed in this study (Hevner et al., 2004).

Guideline	Description
1: Design an Artefact	DSR must produce a viable artefact in the form of constructs, models, methods and instantiations.
2: Problem Relevance	The objective of DSR is to develop technology-based solutions to important and relevant business problems.
3: Design Evaluation	The utility, quality, and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods.
4: Research Contributions	Effective DSR must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.

5: Research Rigor	DSR relies upon the application of rigorous methods in both the construction and evaluation of the design artefact.
6: Design as a Search Process	The search for an effective artefact requires the utilisation of available means to reach the desired end while satisfying laws in the problem environment.
7: Communication of Research	DSR must be presented effectively both to technology- oriented as well as management-oriented audiences.

Table 3.2: DSRM guidelines (Adopted from Hevner et al., 2004)

3.5 DSRM Approach for DSP

Design Science Research Methodology (DSRM) supports a pragmatic research paradigm promoting the creation of artefacts to solve real-life problems. The DSRM approach for DSP is shown in figure 3.3 adapted from figure 3.2 (Vaishnavi *et al.*, 2015).

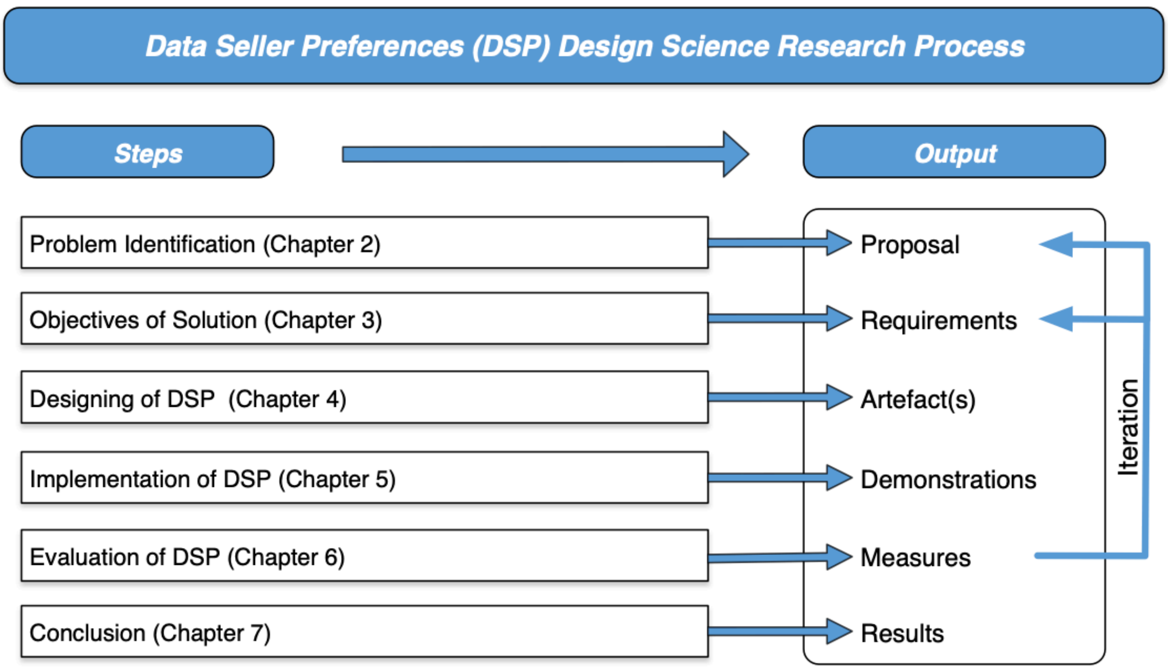

Figure 3.3: DSP Design Science Research Methodology (Adapted from Figure 3.2)

3.5.1 Proposal

In chapter 2, the problem identification is described after reviewing the literature. Section 2.5 of chapter 2 compared the existing literature and found the gap drawing attention to this research proposal.

3.5.2 Requirements

In this chapter, we will go into the low-level details of existing literature (2.4) and prepare the list of requirements in section 3.7 by keeping the DSP smart contract over the blockchain in focus. This will be a creative step to explore the new functionalities based on the novel configurations of new or existing elements.

3.5.3 Artefacts

DSR must produce a viable artefact from constructs, models, methods and instantiations. Different techniques/tools will be used, such as business process model and notation (BPMN) and unified modelling language (UML). The BPMN is used for business process modelling based on flowcharting techniques and UML (unified modelling language) to model the software system and document the artefacts. Constructs consist of requirements (table 3.4) used to support the proposed model. A model consists of a graphical representation of key concepts, an architecture blueprint, and system diagrams. The model will be used to represent the system design with the help of sequence and UML diagrams. UML will be used to design the methods depicted in sequence diagrams. The same technique will be adopted for instantiation by defining class diagrams and entity relationship diagrams for software implementation.

3.5.4 Demonstration

At this stage, the DSP artefact Rest web services will be developed based on the design artefacts and be ready for demonstration. Software prototyping is used to develop incomplete software program versions, focusing on a few aspects of the software. Apart from the Rest web services, we need MongoDB persistence for data storage, middleware technology to integrate with other components and a programable smart contract over the blockchain development server with auto mining feature.

3.5.5 Measure

The DSP artefacts will be evaluated against the novel requirements defined in table 3.4. The list of test cases will be defined at this stage to validate the novel requirements using the Postman automation. Postman will be configured with the list of DSP Dapp web services; each Postman test case will have a test assertion to either pass or fail the test case. Randomly generated data will be used during these test cases to evaluate the DSP artefact.

3.5.6 Results

The final part of the research will cover the critical contribution of DSP using smart contracts over blockchain with research limitations and recommendations for future researchers. The conclusion will be based on the results and knowledge captured within the area of the domain.

3.6 Modelling

In almost all cases, a modelling language is used to prepare the design. This allows the designer to try different designs and decide which will be best for the final solution. Software models are ways of expressing software design. Usually, some abstract pictures or language is used to express the software design (Vlissides et al., 1995). The following subsections elaborate on the modelling techniques used to design and visualise the solution.

3.6.1 Business Process Model and Notation

A BPMN model is designed to be readily understandable by all business stakeholders and serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation (Rosing *et al.*, 2014). By formalizing the transaction process of Data Trading using blockchain in BPMN, a simple and understandable overview of the business process is created while providing the semantics and underlying mechanisms to handle the complexity inherent of the process (Chinosi *et al.*, 2012; Von *et al.*, 2014).

Business Process Modelling Notation (BPMN) illustrates business processes in the form of a diagram. The BPMN process model has four core elements that describe the business process flow.

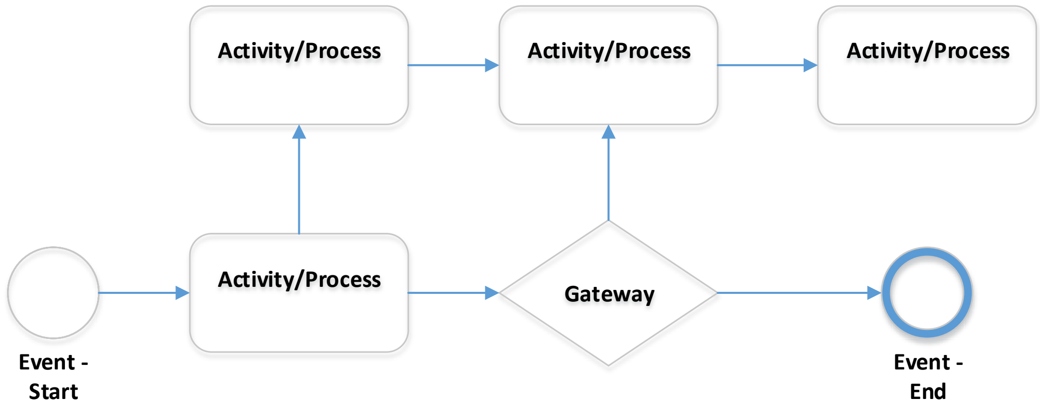

Figure 3.4: Business process modelling and notation

Figure 3.5 represents the BPMN, where the end-to-end business processes are shown with events. Activity or process is a task representing actions that how goals will be achieved by taking actions. Gateways allow controlling, merging, and splitting the process flow based on some conditions. Connectors show the flow of processes and link the tasks together.

3.6.2 Unified Modelling Language

So, in combination with literature review, design science research and the BPMN modelling, the problem has identified the gap in the existing research and the high-level business process model for us to understand the solution better.

To analyse and design a software solution in detail, this research will use Unified Modelling Language (UML). UML is a software modelling standard for an actual or planned process. It can be examined to determine planned systems features and characteristics, but also the structure, behaviour, relationship of systems elements and the purposes, architecture, and design decisions of the system in general (Chonoles, 2018).

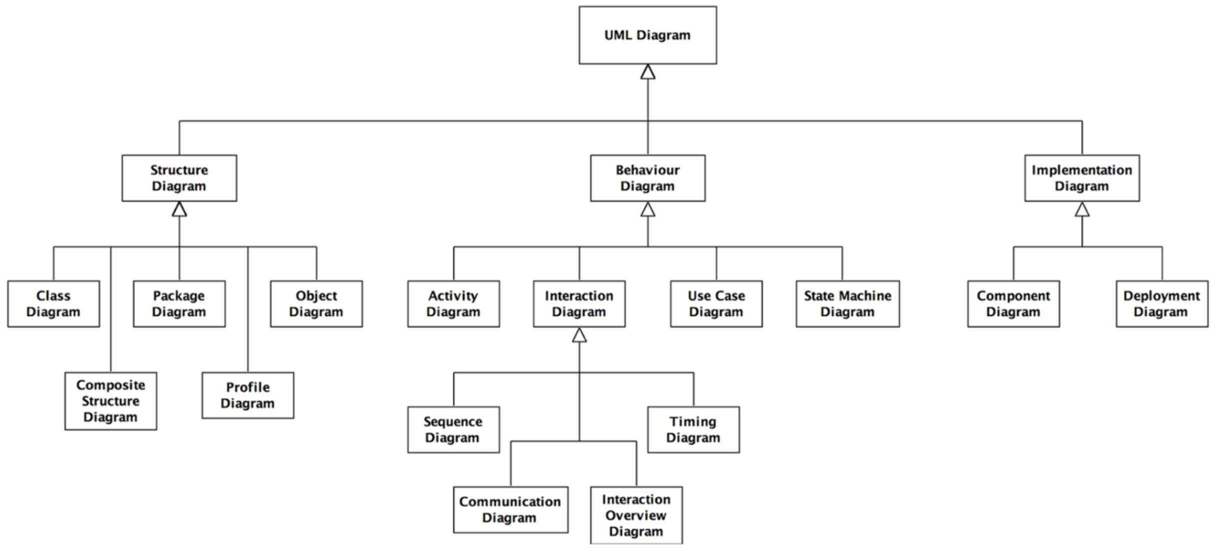

Figure 3.5: Taxonomy of UML diagramming notations (Object Management Group, 2011)

Structure diagrams show the static structure of the components within a system and comprise: Class, Composite Structure, Package, Profile and Object diagrams. Behaviour diagrams show the dynamic behaviour of the components within a system and comprise Activity, Sequence, Communication, State Machine, Use Case, Interaction Overview, and Timing diagrams. Finally, implementation diagrams comprise Component and Deployment diagrams (Timmis, 2016).

Chapters 4 and 5 of this research will use a mix of static and dynamic UML diagrams, including User Case, Activity, Sequence and Class Diagrams, to provide high and low-level solution diagrams.

3.7 DSP Requirement Artefact

This section will analyse the major artefacts and extract the requirements from the existing literature to define the building blocks of DSP. This is a creative step to explore the new functionalities based on the novel configurations of new or existing elements.

Citation	Literature Reference	Major Artefacts
Zheng et al. (2020)	A Blockchain-Based Trading Platform for Big Data	Data provider, Data requester, Data agent, Reward system, Smart contract Blockchain trading platform, Security related to the dishonest data provider and the requester.
Camilo et al. (2020)	A Secure Personal-Data Trading System Based on Permissioned Blockchain, Trust, and Reputation	Seller client, Buyer client, Storage server, Smart contract over Blockchain, Reputation system.
Liang et al. (2018)	A Survey on Big Data Market: Pricing, Trading and Protection	Data auction, Content server, Data owners, Data consumers, Big data trading, Data pricing, Data protection and collection.
Dai et al. (2020)	SDTE: A Secure Blockchain-Based Data Trading Ecosystem	Seller, Buyer, Data trading platform, SDTP (Secure Blockchain-Based Data

		Trading Ecosystem), Malicious Seller & Buyer.
Xiong (2019)	Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning	Data owner, Data purchases, Data hosting centre, Data trading modes & smart contract, tamper-proof and traceability blockchain.
Weber and Prinz (2019)	Trading User Data: A Blockchain Based Approach	Data buyer, Data seller, Data exchange, Bidding & purchase blockchain management.

Table 3.3: Major artefacts from the literature

Table 3.3 has the existing literature references (section 2.4) and analyses the significant artefacts used in their research, and this table will help to construct the building blocks of this research. The following table 3.4 will further extend the requirements from the existing literature revolving around the data seller, buyer and system admin activities with background processes which are the key artefacts from table 3.3. Simply put, data sellers are willing to sell their data to earn digital tokens as a reward (Zheng et al., 2020). The data buyers are interested in accessing the data, already available in the trading platform (Dai et al., 2020). The trading platform will have its background processes and be managed by the system admin to enable trading activities (Weber and Prinz, 2019). The trading platform will let the seller define their preferences to control the purchase to reject the transaction if the buyer doesn’t comply with the seller's predefined preferences otherwise, accept it. The following table 3.4 will list the domain-specific and novel requirements with literature references.

R. No.	Requirement / Literature Reference
	Seller activities
3.1	**As a seller, I need to register my personal details including username and password so that I can create my account on the system.** Data providers include private users or organizations that can decide whether to share their own data based on the request of the data requester (Zheng et al., 2020).
3.2	**As a user (buyer, seller, admin), I need to provide my credentials i.e. username and password so that I can access the system securely.** An overview of the architecture of the system shows the secure interaction of the data provider, requestor and agent (Zheng et al., 2020).
3.3	**As a seller, I want to define my selling preferences so that my selling rights will be protected.** Our contribution is a classification of emotion-related phenomena and an emotion-based account of how anticipatory and anticipated emotions interact to determine investors’ buy and sell preferences in asset markets. (Darren et al., 2020) *Note: this is a thesis-specific requirement.*

3.4	**As a seller, I need to sell my data so that I can collect tokens.** The complete marketplace system is secure, agile, and automatic, rewarding the sellers who sell their private data (Camilo et al., 2020)
	Buyer activities
3.5	**As a buyer, I need to provide my profile such as company details including username and password so that I can create my account on the system.** An overview of the architecture of the system shows the secure interaction of the data provider, requestor and agent (Zheng et al., 2020).
3.6	**As a buyer, I want to access the system securely so that to view the available data.** The buyer will select a data product of interest and place an online order, similar to other e-commerce transactions (Dai et al., 2020).
3.7	**As a buyer, I need to access the system securely so that to purchase the data.** The Data Purchaser 1 selects the data to be purchased from the ‘‘X Data’’ summary information, and obtains the corresponding data from the corresponding API interface after purchase (Xiong, 2019).
	System admin activities & background processes

3.8	**As system admin, I need to deploy smart contract on the blockchain to take advantage of smart contract functionality.** It pioneered the development of smart contract (Szabo, 1994), which is a decentralised application that can be programmed and deployed as an ''automation program'' in a blockchain (Xiong, 2019).
3.9	**As a system, I need to capture all the blockchain address provided by the blockchain network, so that I can assign them to the new account holders.** In Ethereum (Buterin and Vitalik, 2017), a smart contract contains a contract account, a 160-bit blockchain address, runtime bytecode and some related transactions (Dai et al., 2020).
3.10	**As a system admin, I need to add company types and countries so that sellers and buyers can reference them for their account helping to define the seller preferences.** Reference data sets are constructed, using a data generator, to have known solutions, i.e., solutions specified a priori (Cox and Harris, 1999) *Note: this is a thesis specific requirement.*
3.11	**As a process, I need to allocate the available blockchain address so that to identify the buyer/seller on the blockchain network.** The data trading centres, data owners, data purchasers, and data storage service providers are all the Ethereum users who have the Ethereum addresses and know how to create and publish the smart

	contracts, perform the functions of smart contracts and execute the transactions (Xiong, 2019).
3.12	**As a process, I want to store the seller preferences on the smart contract over the blockchain, so that data selling is controlled.** A firm's sales control or coordination system is often recognized as an important management tool that affects salesperson behaviours and outcomes (Cravens et al., 1993). *Note: this is a thesis specific requirement.*
3.13	**As a process, I need to award the tokens (Ethereum cryptocurrency based) so that sellers will encourage to sell their data.** If the transaction is valid, the contract deduces the corresponding number of tokens paid (Camilo et al., 2020).
3.14	**As a process, I want to store the buyer's preference related attributes on the blockchain, so that these details will be used to evaluate with the help of smart contract before data purchase to protect their selling rights.** Sell and buy preferences are proposed to be influenced by differences in strength between the anticipatory emotions (the hope-fear balance) and the anticipated emotions (the elation-disappointment balance) (Darren et al., 2020) *Note: this is a thesis specific requirement.*

3.15	**As a process, I need to capture the user (buyer, seller, admin) privacy consent, so to comply with GDPR.** The broker is trustable which results in privacy disclosure and risk in the scenario of practical applications (Dai et al., 2020).
3.16	**As a process, I need to execute the DSP smart contract before purchase to control the data purchase.** To describe how sales control systems may best be utilized across sales situations, we must first understand the patterns created by the combination of as many of these dimensions as feasible. (Karen et al., 2007). *Note: this is a thesis specific requirement.*

Table 3.4: List of requirements referenced back to the literature

3.8 Research Limitation

The DSP artefact covers the objectives defined in table 3.4. Outside the scope of table 3.4 will form a limitation of this research and will not be covered by this research. In a few words, the DSP will allow the data seller to access the system, define the selling preferences and sell their data to get awards as Ethereum-based tokens. On the other hand, buyers register with their details, search for the data, and purchase it to have full access. Some system and admin activities have also been covered during DSP implementations.

DSP artefact's primary responsibility is demonstrating control of the data purchase based on the seller's preferences. The preferences are kept straightforward to increase the focus key contribution, which is the DSP artefact. Future researchers can extend the preferences parameters to complex ones such as green credentials.

3.9 Ethical Consideration

As research is developing new artefacts and prototyping, it has worked within the ethical boundaries of the DSP. With regards to the moral effects of artefacts, there are, until now, no settled accreditation models and methodologies that could explicitly address moral contemplations, including inclination and straightforwardness, in the area of calculations. Creating moral standards and codes for calculations implies distinguishing the decision-making standards and allotting jobs and duties to the chosen framework. There are no benchmarks related to these issues to affirm or provide a framework against these.

3.10 Summary

In this chapter, the methodology of this academic study is provided in light of pragmatic research philosophy. The DSRM (Design Science Research) is used for this research to introduce new artefacts based on the provided content analysis of the existing literature. DSRM is a comparatively new approach that helps explore several new artefacts. Table 3.4, with the list of requirements, will provide the baseline for chapters 4 and 5 to design and implement the DSP artefact.

DSP DSRM outputs		Steps	Description
	Artefact(s)	Chapter 4	Designing of DSP
	Demonstration	Chapter 5	Implementation of DSP
	Measure	Chapter 6	Validation of DSP
	Conclusion	Chapter 7	Results

Table 3.5: DSRM approach for DSP

Chapter 4 will use the requirements from table 3.4 and provide the DSP design artefacts with the help of sequence & UML diagrams. Based on the design artefacts, chapter 5 will provide the implementation to demonstrate the DSP smart contract controlling the data selling. Chapter 6 will validate the novel artefacts to assess if this research answered the question raised in section 2.6. Chapter 7 will summarise the essential contribution and discuss any future aspects of this research.

Chapter 4: DESIGNING OF DSP

4.1 Introduction

Chapter 4 presents design artefacts to transform the requirements mentioned in chapter three - table 3.4 into suitable forms. DSP designing is envisioning and defining software solutions to sets of requirements defined in table 3.4. Software modelling standards (section 3.6), particularly UML use-case and sequence diagrams, will be used to provide the high-level design. Think of designing software as someone designing a house. They may start by drawing a rough sketch of the floor plan and layout of the rooms and floors. The drawing is modelling language, and the resulting blueprint will be a model of their final design. We discover problems early and fix them without refactoring our implementations by designing DSP using UML modelling language.

The use case diagram 4.1 is used to support the proposed system for modelling purposes. Moreover, sequence diagrams depict the activities of actors involved in the DSP trading system per the requirements in table 3.4. Finally, DSP smart contract interfaces are designed in section 4.5.

4.2 Use Case Diagram for DSP Trading System

Let us take an overview of the proposed DSP trading system, referring back to the requirements in table 3.4 with the help of a use case diagram based on blockchain infrastructure.

The use case diagram illustrates the DSP trading system with the help of actors and the association of activities with the actors. By analysing tables

3.3 and 3.4, we can find the buyer and seller as primary actors and the system admin as a secondary actor. Moreover, the system boundary explains the case containment behaviour and interaction of processes with the smart contract and its implication for the different actors involved.

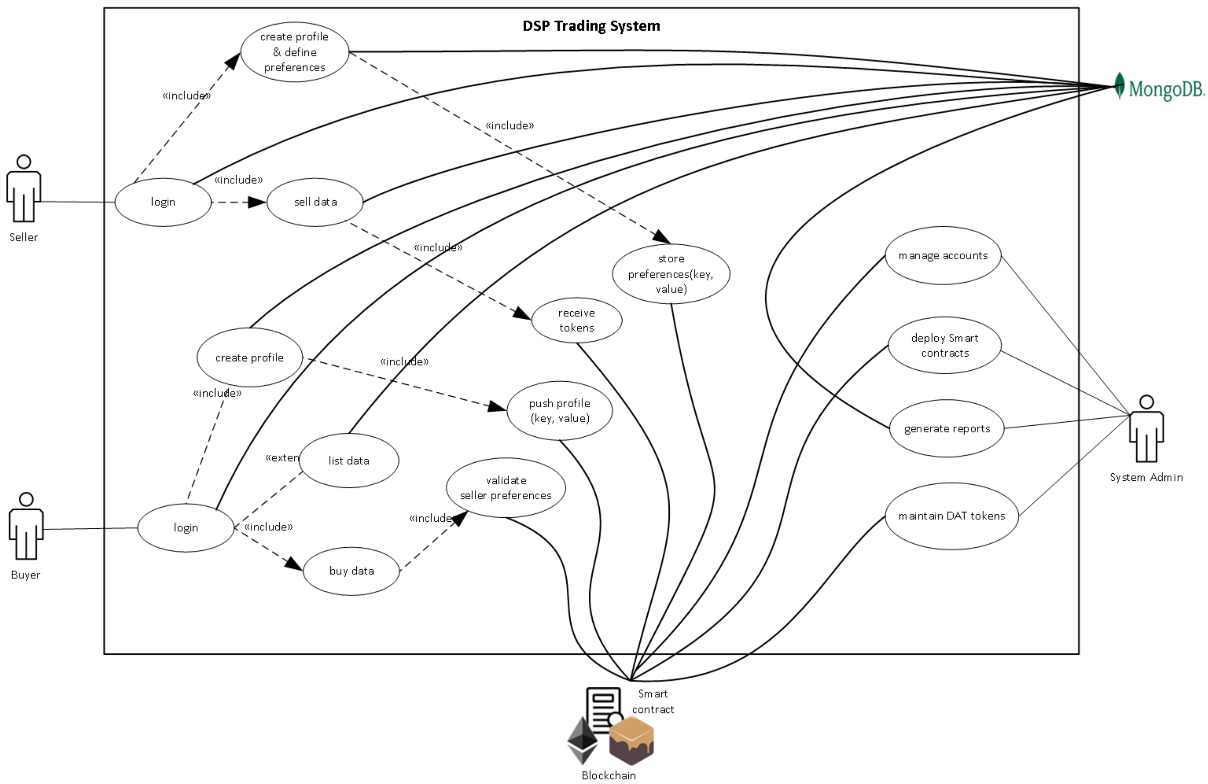

Figure 4.1: Use case system diagram of the DSP trading system

Sellers, buyers and system admin are the main actors interacting with the DSP trading system securely using OAuth 2.0 to authorization with grant type password. DSP trading system users will use the username and password to authenticate the user by utilizing a username and password (Table 3.4, req. no 3.2). In response to the authentication, the access token of type Bearer will be issued with the expiry time. The DSP trading system will expose the secure Rest web services that accept the request using JSON

(JavaScript Object Notation, a lightweight data-interchange format) and the response in the same format. The Bearer access token will be used to access the secure web services. System admin performs administrative tasks such as managing seller and buyer accounts, generating custom cryptocurrency tokens, and generating reports.

A seller can create a profile and define the preferences stored on the smart contract deployed on the Ganache blockchain server. The seller can sell their personal data sent as JSON payload via web services and get rewarded by the tokens, and data ownership will change internally to the system admin. Furthermore, data trading will take place depending on the preferences set by the seller.

A buyer can create a profile and view a data summary with price, so that he can choose a data block to buy it for detailed data view access. The data price is simply calculated as total number of data rows multiply by the system define constant in pence. The DSP trading system can extract the buyer's preferences-related parameters and store them on the same smart contract deployed on the Ganache blockchain server. A buyer triggers the data buying process by paying an amount in the traditional currency according to the data volume. However, the purchase transaction can be accepted or rejected due to the smart contract's automatic decision based on whether the buyer satisfies the seller's preferences. The buyer can only view the data using the DSP trading system but cannot download it. Ethereum blockchain and MongoDB are well-known technologies, but we may not discuss them in detail. However, the Ethereum blockchain ledger records every transaction, and MongoDB will be used to store the reference & personal data in the DSP trading system. As shown in figure 4.1, three main actors have been discovered apart from the internal processes responsible for processing the data trading. The following sections elaborate on system admin, seller and buyer.

4.2.1 Seller

Sellers play an essential role in the DSP trading system because they are the primary source of the data provider. The data provided by the seller's stored in the MongoDB database. Once the data is uploaded to the DSP trading system, the seller will be awarded the custom DAT tokens, and data ownership will transfer to the system admin. However, each data block refers to its original seller, so associated preferences can be triggered during the data-buying process.

4.2.2 Buyer

The buyers are the ones who need the data provided by the sellers. They can attempt to pay for the selected data, but the purchase process will be subjected to the preferences smart contract to accept or reject the transaction based on the seller's preferences.

4.2.3 System Admin

Seller activities sequence diagram

System administrator deals with the overall functionality of infrastructure and provides an application interface between entities, enabling them to perform their roles depending on the functionality written in smart contract and allowing them to execute their functions.

4.3 DSP Trading System Sequence Diagrams

DSP trading system defines the overall interaction of actors discussed in section 4.2 in connection with smart contract functions and their implementation within the system. These processes generalise the interaction between an individual actor and the smart contract. Each process shows the relationship linked with an actor and how those processes are associated with it. The main actors and their associated processes are defined in their respective sequence diagrams to elaborate on the business model of the DSP trading system.

4.3.1 Seller Activities Sequence Diagram

The following diagram 4.2 depicts the seller activities according to the requirements covered in the previous chapter, table 3.4 – seller activities. A seller creates a profile, including preferences which will be stored in the smart contract and sells the data on the proposed platform so that a buyer can buy the available data to view. The system provides an interface for a seller to log in and sells the data, which will then be stored in the MongoDB database.

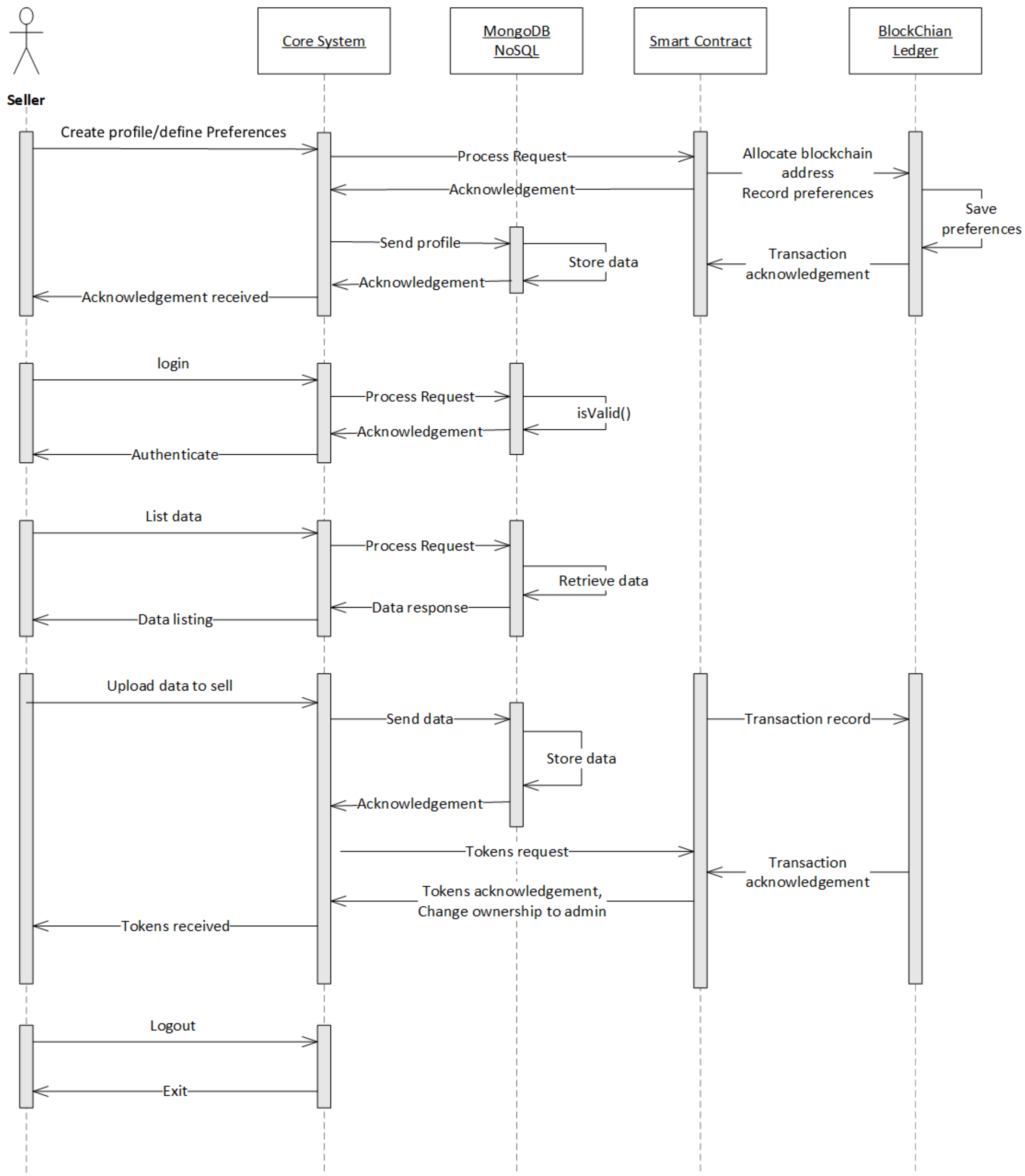

Figure 4.2: Seller activities sequence diagram

The following table 4.1 describes the details of sequence diagram steps in reference to the requirements from table 3.4.

Seller activities sequence diagram steps	**Table 3.4,** **Requirement reference**
The seller can provide access credentials, create a profile and define the preferences which are saved by the DSP trading system on the blockchain network. These preferences help DSP smart contract to govern data trading.	3.1, 3.3, 3.15
Seller accesses the secure DSP trading system after authenticating by providing username and password and using OAuth2 Bearer access token for authorization.	3.2
The DSP trading system allocates and assign available blockchain address to each seller account for recording the transaction and validation of DSP within smart contract.	3.9
The seller can sell their personal data as JSON payload via web services and get rewarded the tokens (DAT) and data ownership will change internally to the system admin by keeping the seller preferences within the relationship.	3.4

Table 4.1: Seller activities sequence diagram steps with requirement numbers.

At this stage, system behaves as a party A and seller as party B. Transaction took place according to data sell to DAT tokens conversion by the smart contract and cryptocurrency tokens transferred from party A to Party B digital wallet and data ownership internal references will be updated from Party A to Part B, by keeping the seller preferences within the relationship.

4.3.2 Buyer Activities Sequence Diagram

The following diagram 4.3 depicts the buyer activities according to the requirements covered in the previous chapter, table 3.4 – Buyer activities.

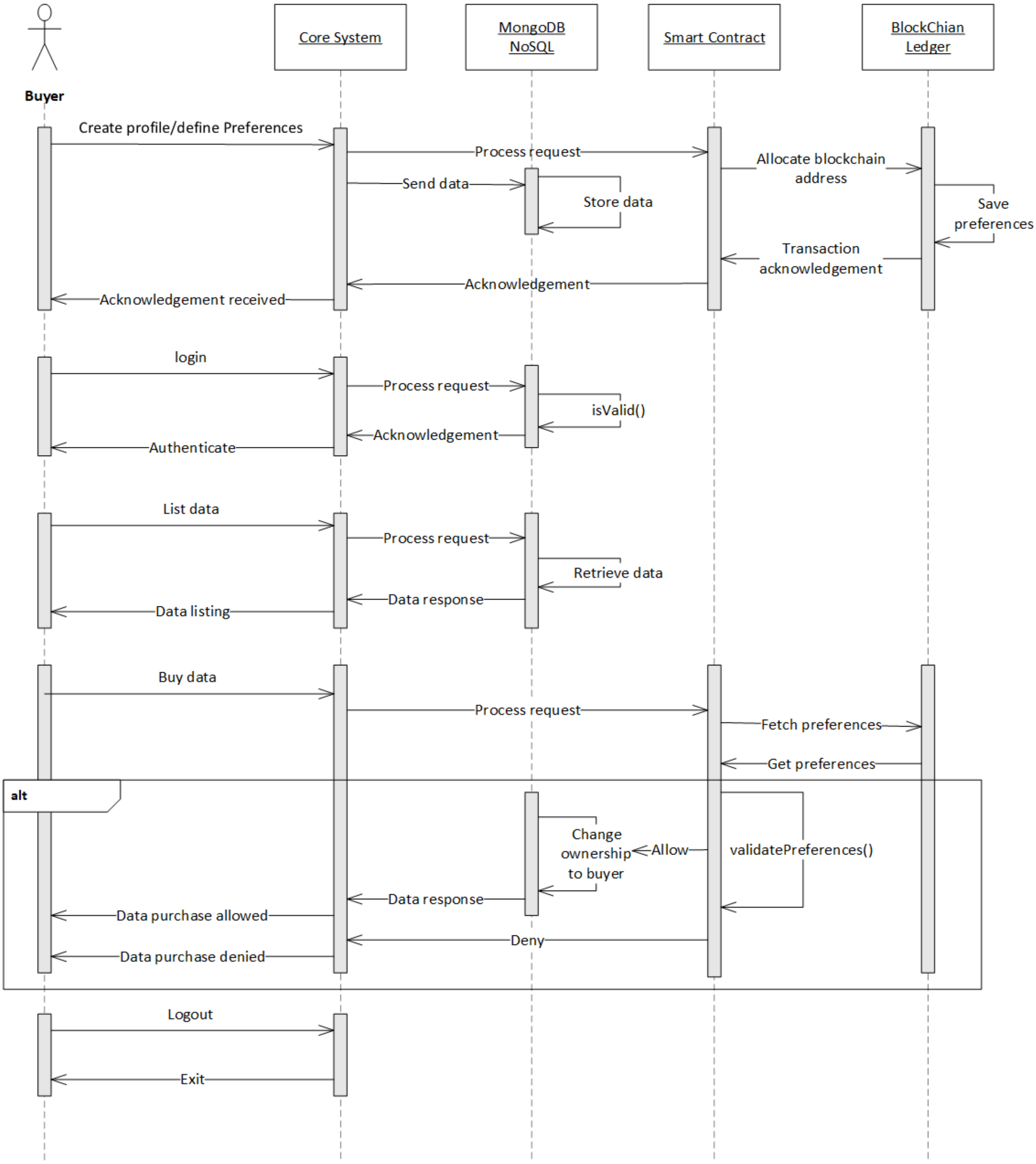

Figure 4.3: Buyer activities sequence diagram.

The following table 4.2 describes the sequence diagram steps about the requirements from table 3.4.

Buyer activities sequence diagram steps	Table 3.4, Requirement reference
The buyer can provide access credentials and creates the profile. The DSP trading system selects and stores the buyer's preferences parameters such as country, company type and size of the company to the preference smart contract deployed on the Ganache blockchain.	3.5, 3.15
Buyer list data web services of the secure DSP trading system after authenticating by providing username and password and using OAuth2 Bearer access token for authorization. The list data response will be a summary of block data with the associated price.	3.6
The buyer then attempts to pay using a credit or debit card. At this stage, the preference smart contract deployed by the Ganache blockchain server, executes automatically once called by the DSP trading system. In this instance, the preference smart contract is aware of the buyer's preference parameters	3.7, 3.16

and the seller's predefined preference. Once matched, then the purchase will be denied; otherwise allowed. For example, if the seller defines a preference parameter as (country: Russia). So, all purchases will be allowed except for the Russian buyers.	
On successful purchase, the data block ownership will be changed from system admin to buyer. It is important to note here that buyers will not be able to download the data, but they will have unlimited access to view the details data.	3.7

Table 4.2: Buyer activities sequence diagram steps with requirement numbers.

4.3.3 System Admin Activities Sequence Diagram

The following diagram 4.4 depicts the system admin activities according to the requirements covered in the previous chapter, table 3.4 – System admin activities and background processes. A system admin has access to all resources of the DSP trading system. The administrator's responsibilities are to maintain the actor’s accounts, smart contract deployment, execution logging, GUI interaction with the smart contract and generating general and analytical reports for system support and to compete for market demands.

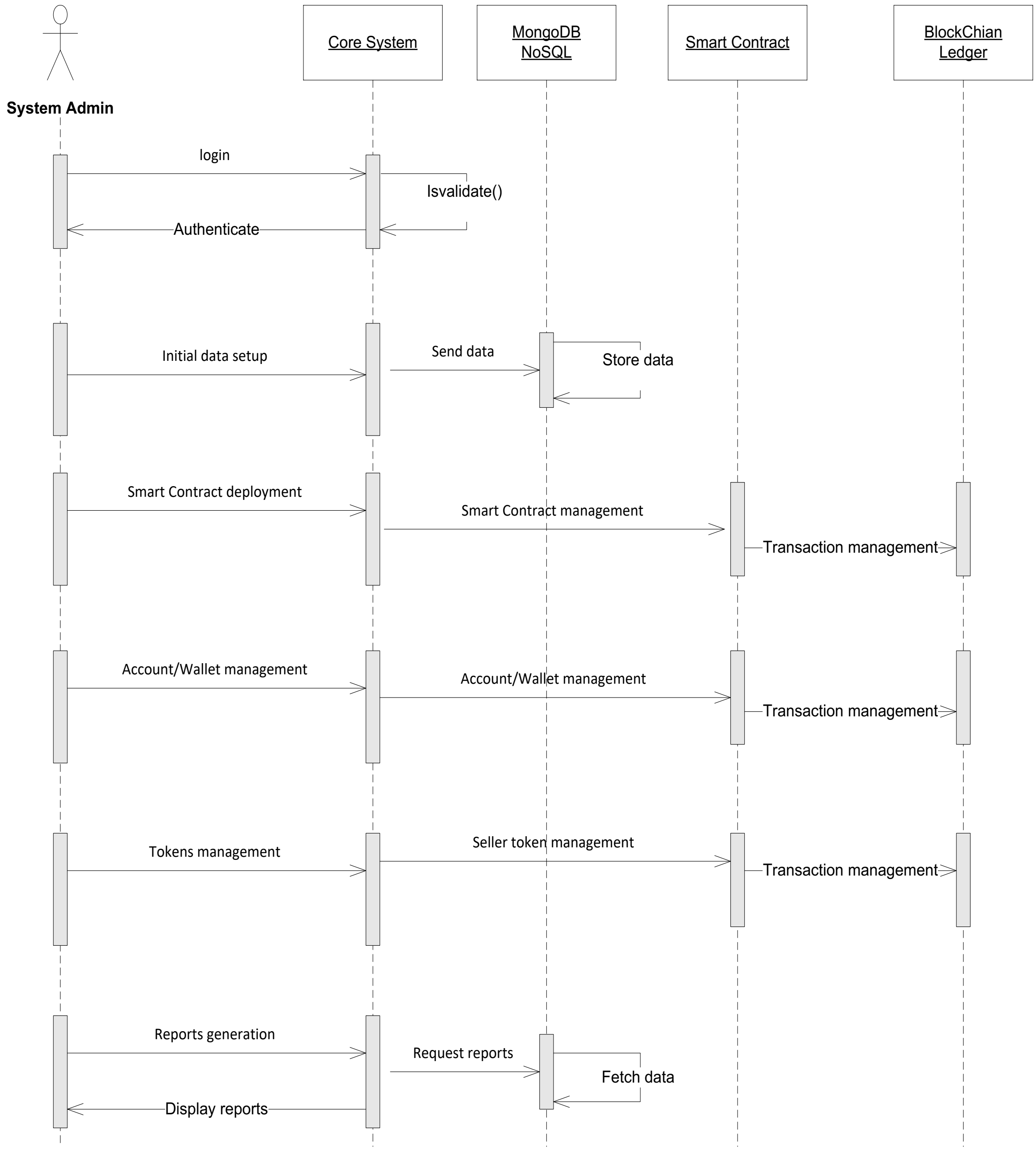

Figure 4.4: system admin activities sequence diagram.

The following table 4.3 describes the details of sequence diagram steps in reference to the requirements from table 3.4.

System admin activities sequence diagram steps	Table 3.4, Requirement reference
The system admin setup the initial data in terms of company type and countries, which are stored in MongoDB so that this data can be referenced by the seller and buyer at the time of creating profiles and defining the preferences. This includes setting up a single data row value property in the pence app.datatrading.datarowvaluepence=5. The data rows are always priced by multiplying the total number of data rows with this property value.	3.10
To use the functionality of smart contract, it must be deployed on the blockchain and once deployed.	3.8
The system admin also manages accounts and tokens along with the system processes as it is needed to allocate the available blockchain address to identify the buyer/seller Ganache blockchain server.	3.11, 3.13

Table 4.3: System admin activities sequence diagram steps with requirement numbers.

4.4 Permissioned Blockchain suitable for DSP trading system

Blockchain has two major architectures based on the design of the network, Permissionless, which is open for anyone to participate, and Permissioned, which is limited only to designated participants. However, implementation decisions must consider several other technological and market dynamics summarised below and in table 4.4.

	Permissioned Blockchain	**Permissionless Blockchain**
Operational costs	Depends on the redundancy requirements	High (Bitcoin estimate $657,000,000 per year in 2017 at $1000/BTC)
Interpretability	Poor	Excellent
Transaction Throughput	Good	Poor
Data Privacy	Good	Poor
Scalability	Poor	Good
System robustness and resilience	Fair	Good

Table 4.4: Comparison between permissioned and permissionless blockchain (Yu et al., 2018)

4.4.1 Permissionless Blockchain

Some of the advantages of a permissionless blockchain are.

1- It has an open network to enable anyone to join/quit the protocol freely.

2- The network typically has an incentivizing mechanism to encourage more participants to join the network.

3- It is suitable for cryptocurrency and applications that do not have strict privacy requirements.

4.4.2 Permissioned Blockchain

Some of the advantages of a permissioned blockchain are.

1- All blockchain participants are registered and verified by the protocol administrator, so it is easy to identify nodes that do not comply with the protocol.

2- Since the public has no access to the blockchain, privacy is preserved.

3- Since the blockchain administrator can control the network size by controlling the number of nodes involved, the permissioned blockchain usually has a high transaction throughput.

4- Mining is not required to validate transactions or execute the smart contract.

As the DSP trading system requires buyers and sellers to be registered, and no anonymity is accepted, good performance and data privacy make Permissioned Blockchain the best suitable for the DSP trading system (table 4.4).

4.4.3 Ganache Ethereum Blockchain Server

Ganache is a personal blockchain for rapid Ethereum and distributed application development. Ganache is used across the entire development cycle, enabling the development, deployment, and testing Dapps (Decentralised Apps) in a safe and deterministic environment. It is a local in-memory blockchain designed for development and testing purposes. It simulates the features of a real Ethereum network, including the availability of several accounts funded with test Ether. It is an integral part of the Truffle suit that allows deploying smart contracts and recording transactions for the test DSP development. DSP trading system will use Ganache to deploy and execute the smart contract on specific addresses. Its GUI-based blockchain is available locally. The figure shows the Ganache GUI with the list of addresses and other account-related information for testing Dapp on the local machine.

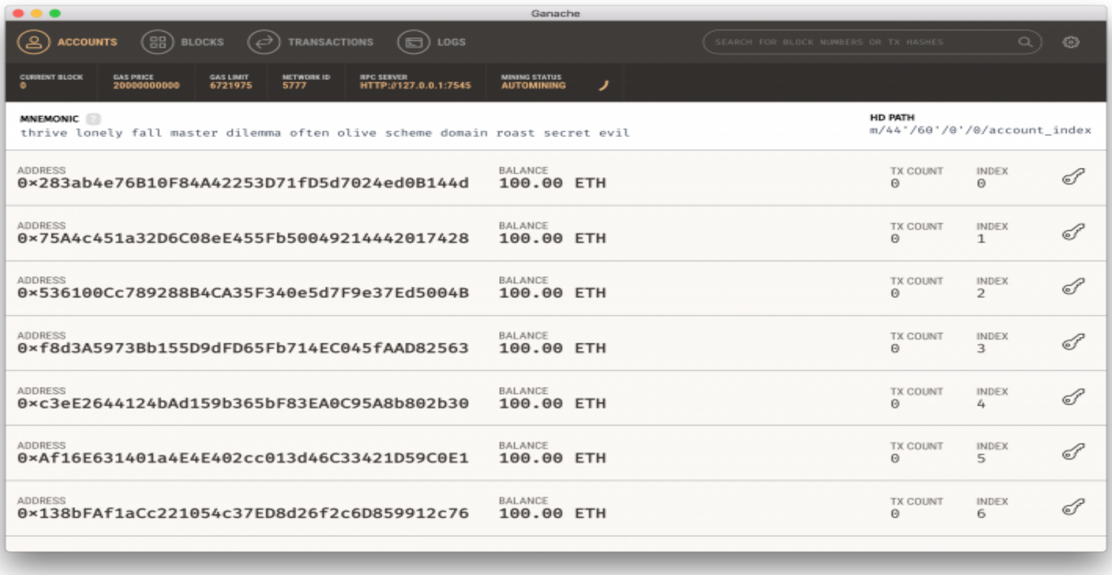

Figure 4.5: Ganache Server for DSP trading system.

4.5 DSP Smart Contracts Interface

A smart contract is a digital agreement between parties with specific stipulations that must be fulfilled for the successful transaction to take place on the blockchain.

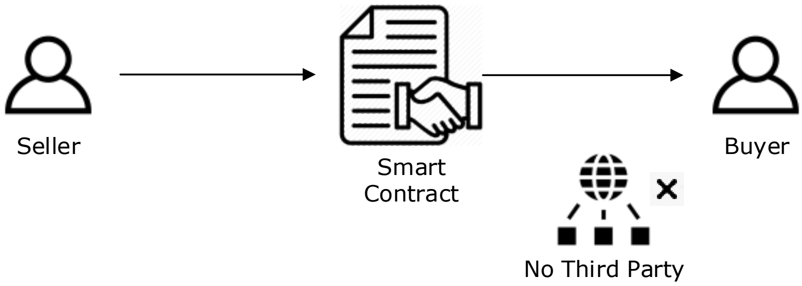

Figure 4.6: Smart Contract role.

Figure 4.7 depicts the role of smart contracts and their execution without any third-party involvement, which makes them cost-effective and self-managed in contrast to traditional contracts. DSP trading system uses Ethereum blockchain technology which provides smart contracts to be programmed in the solidity language. It is based on JavaScript-like language. The smart contract code is compiled into byte code and deployed to the Ganache blockchain server for execution. DSP trading systems use two smart contracts.

- Token management smart contract
- Preferences smart contract

4.5.1 Tokens management smart contract

The EIP20 interface is used to deal with the tokens management. These tokens will be awarded to the data sellers in response to successful data selling. EIP20 was introduced in 2015. This interface allows the development of a standard smart contract to transfer the tokens to the participants over the Ethereum network.

Furthermore, the DAT custom token will be used to exchange the data between the seller and the DSP trading system. DAT token act as Ethereum cryptocurrency for the exchange of data. EIP20 standard (Fabian, 2105) interface contains the following data items and functions are shown in Table 4.5 and 4.6, respectively.

S. No	Data Item	Description
1	Totalsupply	Total tokens count
2	Name	Name of token (e.g. DAT)
3	Decimal	Number of decimal points at the time of displaying tokens
4	Symbols	Alias of token name
5	Balances	Token balance of a participant linked with account address
6	Allowed	Authorised token list linked with sender account address

Table 4.5: Data Items – EIP20 Smart Contract

S. No	Function	Description
1	totalSupply()	Total numbers of token supplied at the time of initialization
2	balanceOf()	Evaluate Token balance of certain account
3	allowance()	Returns number of tokens that are allowed to be transferred from sender account to a recipient account

4	transfer()	Provide a functionality of transferring token from caller account
5	approve()	Manage number of tokens that are allowed to be transferred from sender account to a recipient account
6	transferform()	Transfer tokens from one account to another account

Table 4.6: Functions – EIP20 Smart Contract

4.5.2 Preferences Smart Contract

Once the seller creates their profile and preferences, the seller's preferences will be stored in the preferences smart contract as key-value pair. Similarly, the buyer preferences parameters key-value will be stored in this smart contract. Just before the data purchase, the preferences smart contract will be executed to either accept or deny the purchase process. For example, a seller can define preferences as a type of company as betting (companyType:Betting), it means that interested buyers with company type as Betting will be rejected to purchase seller's data. Following table 4.7 mentioned key functions of the preferences smart contract interface.

S. No	Function	Description
1	savePreferences ()	Saves the seller & buyer preferences
2	validateSellerRights ()	Validates the seller's preferences.

Table 4.7: Functions – Seller preferences Smart Contract

4.6 Summary

Although chapter three provided us with the foundation of requirements artefact, innovative software design has been achieved in this chapter, so the DSP trading system has designed based on the DSRM tools and requirements artefacts analysed in chapter three. The high-level use case and sequence diagrams have been covered for the DSP trading system, including the smart contract interface design. Design activity in this chapter will significantly help in chapter five, where the DSP treading system will be implemented after the realisation of low-level diagrams.

CHAPTER 5: DSP IMPLEMENTATION WITHIN SMART CONTRACT

5.1 Introduction

This chapter focuses on the implementation phase following the high-level design provided in chapter four. This chapter aims to extend domain-specific requirements and implement the thesis-specific requirements. The development phase is focused on data trading between buyer and seller depending upon DSP and implemented as a machine-executable smart contract to control the data selling. These preferences are defined by the seller consisting of country, size or type of business. The preferences smart contract will be executed based on these preferences, and validation will occur at the data purchase time.

To implement the smart contract in the blockchain system, Ethereum is used, which can execute a smart contract and perform its functions automatically to complete the data trading transactions. A smart contract consists of machine language usually written in a Solidity which has functions to implement the contract stipulations. The proposed system covers those stipulations in the form of DSP. As the DSP trading system is based on a blockchain controlled by the smart contract, therefore, architecture and technology will be discussed in this chapter to get the job done.

The chapter is divided into sections based on the factors described in the previous paragraph. After the introduction, section 5.2 describes the DSP trading system architecture then sections 5.3 to 5.7 present low-level design and implementation details with class diagrams, ERD diagrams, code snippets and database representation. Section 5.8 list the thesis-specific requirements implemented in chapter 5, which will be validated in chapter 6. Finally, section 5.9 summarises this chapter.

5.2 DSP Trading System Architecture

Referring back to figure 4.1, system architecture in figure 5.1 is proposed. This architecture covers the technology stack and its interactions.

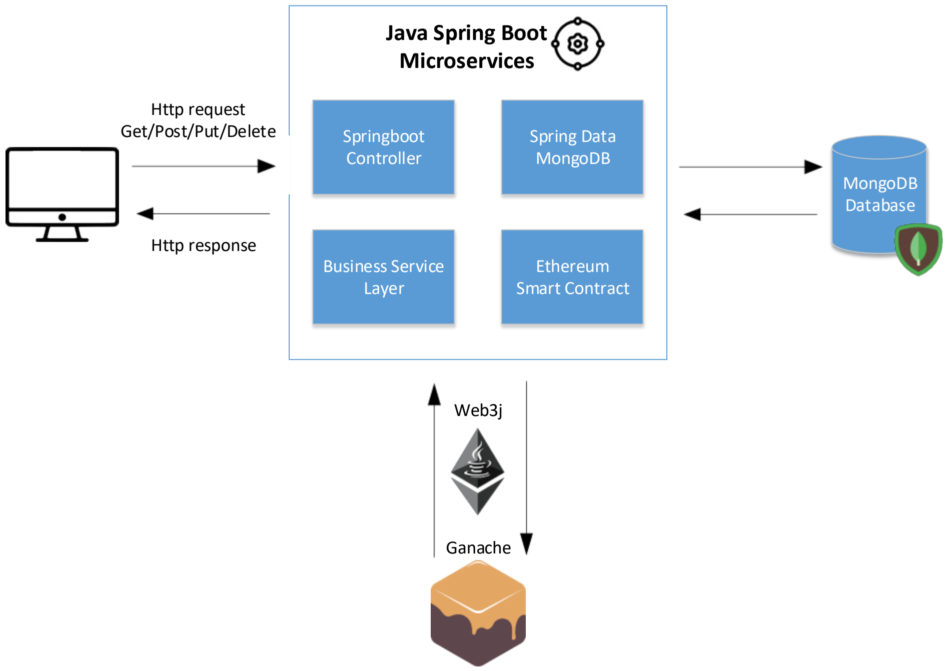

Figure 5.1: DSP trading system architecture

Figure 5.1 shows the architecture of the proposed DSP trading system artefact. The system will be a web services Dapp (distributed application). The prototype will be designed with the help of a distributed architecture.

Java spring boot-microservices will be used for rapid software development and enable the integration of other components. When a buyer/seller requests any system activity, spring boot-microservices will handle the web service request and communicate with the other components such as database MongoDB, Ethereum smart contract over blockchain ganache server with the help of Web3j for recording data trading awarding tokens, saving preferences and its validations. The seller defines DSP at the time of account registration, and the system will collect buyer preferences from their profile. During the data purchase process of data trading, DSP will be validated, and transactions will be controlled based on these parameters. The smart seller contract will be executed automatically before the data purchase process with the help of preferences already stored on the Ganache blockchain server. If a match is found, the transaction cannot take place; otherwise, it will be treated as successful.

5.2.1 Blockchain Framework

Blockchain provides a platform to conduct online transactions between parties using smart contract which executes digitally. Every transaction carries the information, which is then stored in a digital ledger, making a transaction history, and this information is stored irreversibly. When the transaction is initialised over the network, it is routed to all the nodes across the blockchain p2p network. Each node maintains the address of the next node that validates the transaction on all nodes and forms a group of chains shown in figure 5.2.

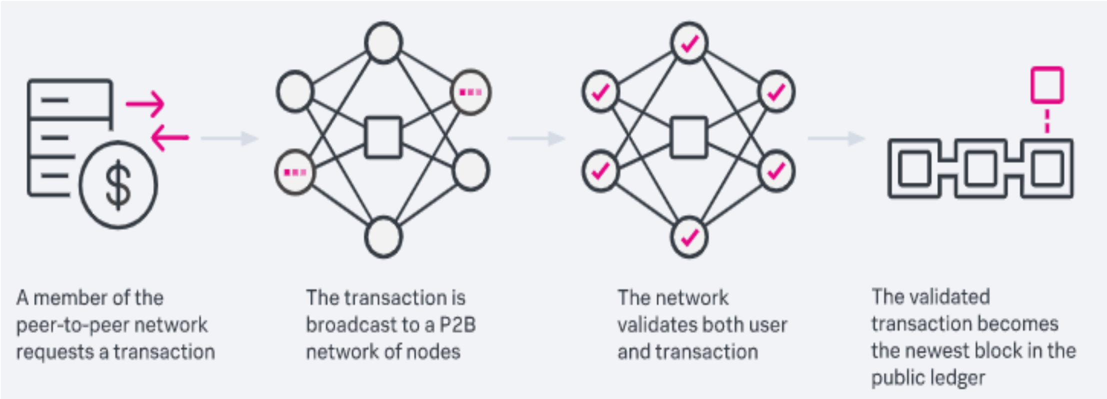

Figure 5.2: Blockchain Framework

Data trading model implemented on the above framework, where buyer and seller become members on the system and data trading occur as per the rules defined in the smart contract.

5.2.2 Smart Contract

The smart contract acts as an agreement document written in computer code with a specific blockchain address. That document automatically ensures the conditions are satisfactory between buyer and seller, thus completing the transaction over the blockchain network. In this case, the smart contract contains the code that validates DSP so that all buyers must agree to abide by the terms of the parameters for conducting data trading. In other words, the system will not allow the buyer to buy the data out of smart contract DSP validation.

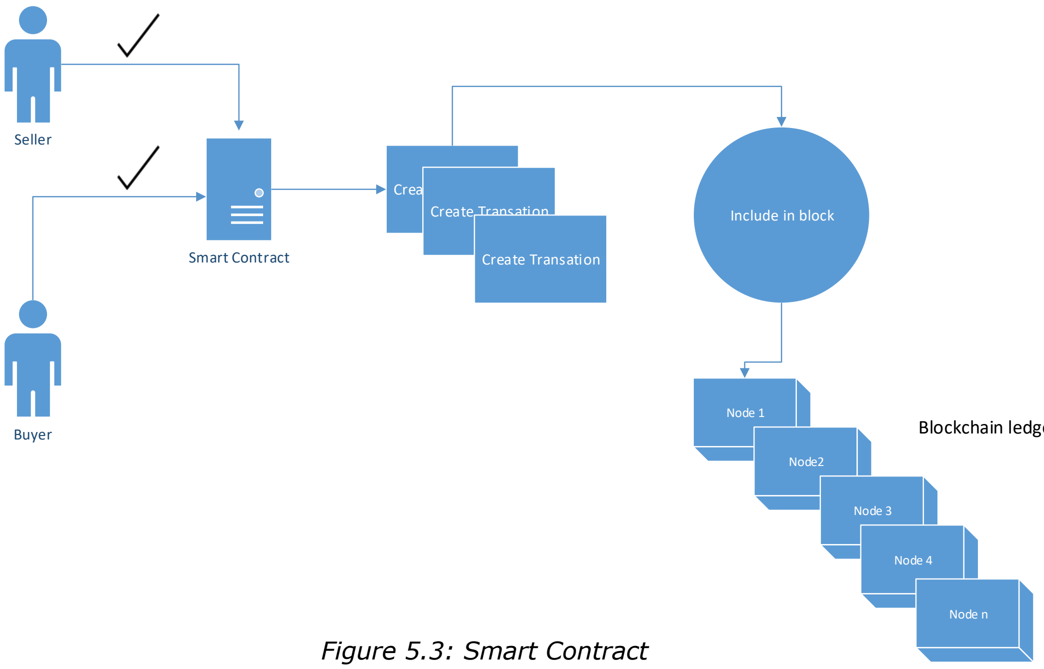

Figure 5.3: Smart Contract

As depicted in figure 5.3, the buyer and seller must agree on the stipulations according to the smart contract. In our case, selling preferences will be treated as data trading conditions that must be validated to conduct successful data trading. The data trade transaction is then recorded on the ledger on all nodes participating over the blockchain network.

5.2.3 Ethereum Blockchain

Initially, blockchain was developed to store the transactions over the network, but later on, new ideas came into play that also served to execute computer programs over the digital ledger. From this innovation, blockchain technology introduced the capability to execute programs and applications in a distributed environment on all the nodes participating in the network. In this context, the Ethereum platform that drives programs over the network in the form of smart contracts was introduced. This platform provides the infrastructure for writing smart contracts. It builds

DAP (decentralised applications) where anyone can implement their functions, rules, terms and conditions to carry out any operation (Vitalik *et al.,* 2013). It can be summarised as follow:

- Ethereum platform allows execution programmes over the blockchain network in the form of smart contracts.
- It allows the creation own cryptocurrency, which means it is not limited to traditional bitcoin currency and provides the functionality to develop its cryptocurrency, such as Tokens.
- Ethereum provides a platform to develop Dapp applications with transparency and security without involving third-party intermediaries.

5.2.4 Solidity

A smart contract executes when a trade occurs between buyer and seller, depending on the terms and conditions. Solidity language is used to code a smart contract within the Ethereum platform, just like Java, C++ or other high-level computer languages. Smart contracts are written using a solidity online platform called Remix and compiled on the same platform for further deployment. Thus blockchain, solidity and DAP work together to form a building block for the whole system, as visualised in figure 5.4 concerning application, software and hardware layers:

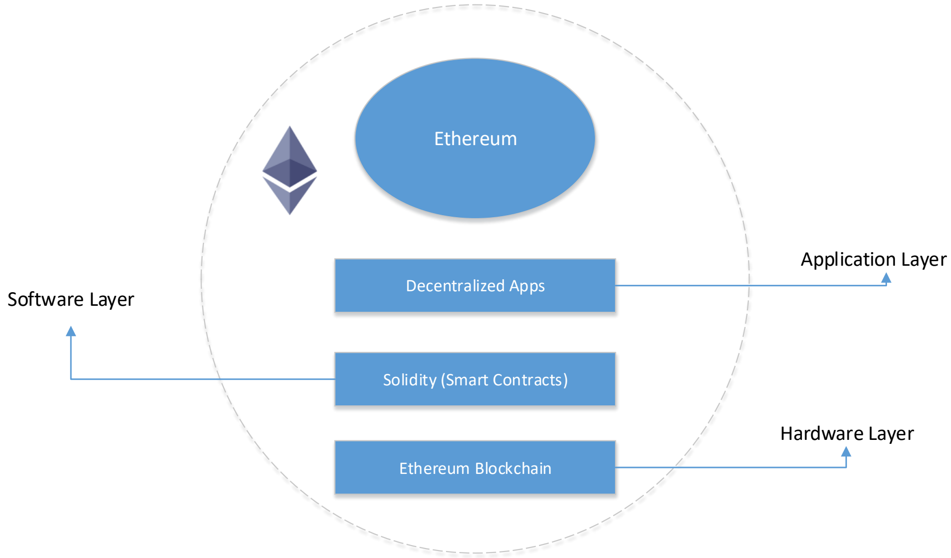

Figure 5.4: Solidity

5.2.5 Ganache

Ganache is already discussed in section 4.4.3, in short, this research will use Ganache personal blockchain for rapid Ethereum and distributed application development.

5.2.6 MongoDB

MongoDB is a distributed database with high availability; horizontal scaling and geographic distribution are built-in and easy to use. MongoDB stores the data in a non-relational manner, making it much faster than traditional RDBMS like MySQL or Oracle. It also supersedes other databases in terms of scalability as data volume grows with time. It can store data in flexible, JSON-like documents, meaning fields can vary from document to document, and data structure can be changed over time. In the DSP trading system, data provided by the seller will be stored in MongoDB to gain all the benefits described above.

5.3 DSP Trading System Modules

The DSP system is divided into different modules based on functionality. These modules are responsible for performing their functions individually and provide an infrastructure for the seller and buyer to conduct the data trading transactions over the blockchain. These modules are controlled by Java microservices based on OOP (Object-oriented programming) to maximise performance and security. Let us discuss these modules individually and evaluate how the system will work. Following is the list of modules.

1- Company Types and Country Creation.

2- Account/Profile & Preferences Creation.

3- Data Selling

4- Data Buying

5.4 Company Types and Country Creation

This module focuses on creating different types of companies. This reference data will be used for buyer and seller to reference during preference creation.

5.4.1 Class Diagram

In this class, the system allows the generation of the company types provided by the system admin, which are used to determine the type of company later on at the time of data trading.

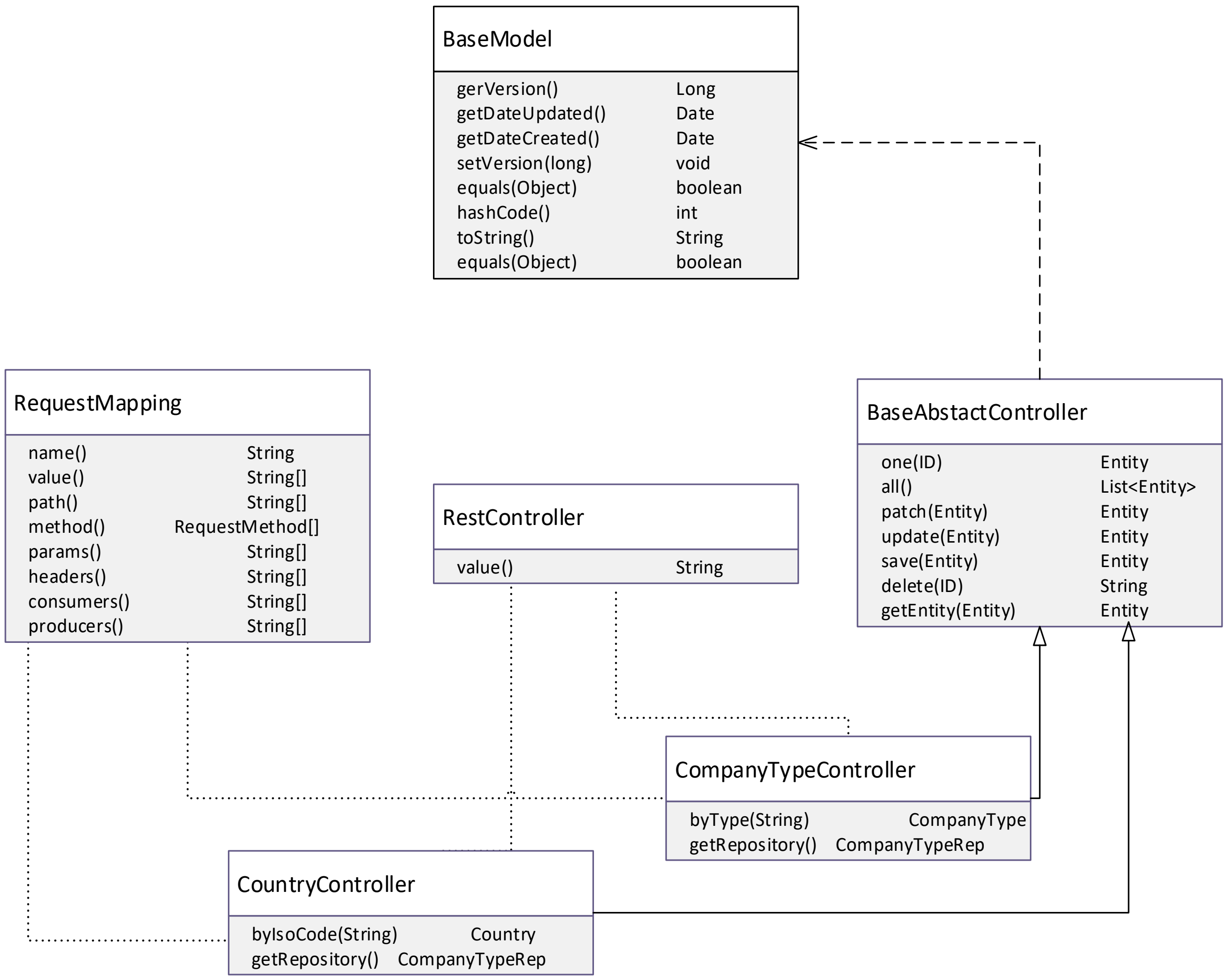

Figure 5.5: Class Diagram of Company type and County creation

Company types can be set up by calling the class method createSaveCompanyTyple() in the InitialDataSetup class built as a helper class for the initial data to the DSP trading system. This method saves the different company types in the system database, which are associated with the different buyer/seller accounts accordingly. This information indicates the type of buyer/seller company. It helps the smart contract to identify the valid buyer as per DSP when the transaction will be made over the system platform and then allow that transaction to be recorded on the blockchain. The Base Model is the parent representative of MongoDB database entities. BaseAbstractController class control the CURD (create, update, read and delete) MongoDB database interfacing operations (with the help of repository objects) and provide abstraction by associating the base model. Essential properties are shown in figure 5.5 above. Request controller and Rest Mapping are the DSP trading system framework classes; they enable our functions to be exposed to http/https. Finally, the CompanyTypeController provides extended functionality of BaseAbstractController.

5.4.2 ERD Diagram

The figure 5.6 shows the Company, CompanyType, Address and Country entities and their relationships. Buyers will add the Company details, whereas sellers will reference the CompanyType and Country in their preferences.

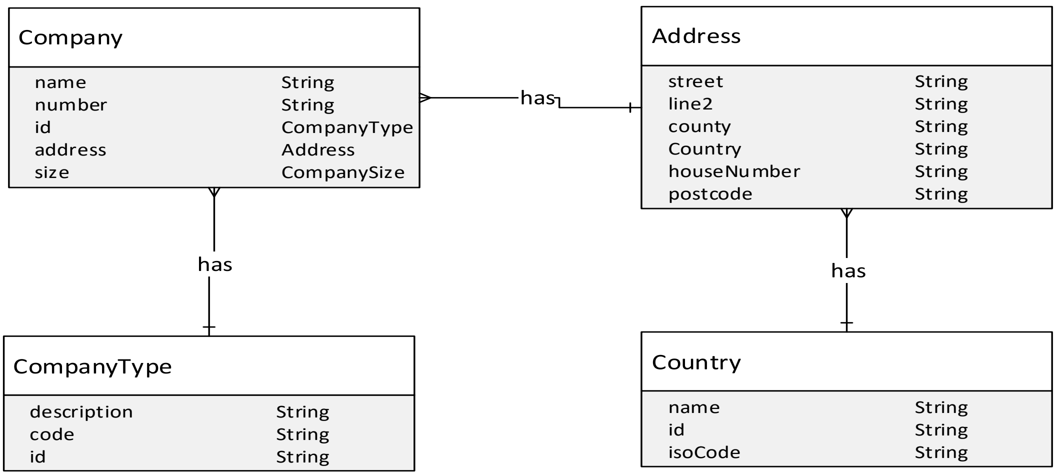

Figure 5.6: ERD diagram – Company Type and Country

5.4.3 Code Snippet

```java
private CompanyType setUpCompanyTypes()
{
    createSaveCompanyType( code: "PLC",  description: "Public limited company");
    createSaveCompanyType( code: "LTD",  description: "Private company limited by shares");
    createSaveCompanyType( code: "Unltd",  description: "Unlimited company");
    createSaveCompanyType( code: "LLP",  description: "Limited liability partnership");
    createSaveCompanyType( code: "IPS",  description: "Industrial and provident society");
    return createSaveCompanyType( code: "RC",  description: "Royal charter");
}

private CompanyType createSaveCompanyType(final String code, final String description)
{
    CompanyType companyType = new CompanyType();
    companyType.setCode(code);
    companyType.setDescription(description);
    return companyTypeRepository.save(companyType);
}
```

Figure 5.7: Code Snippet- Company Type and Country

5.4.4 Postman Screenshot

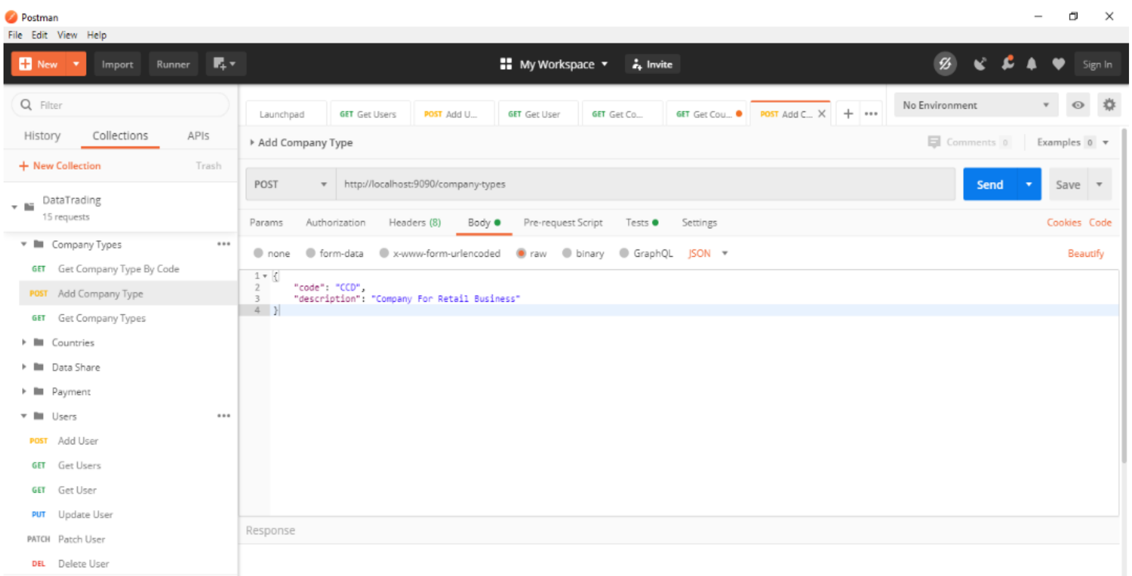

Figure 5.8: Postman Data Posting-Company Type

To create a new company type, data will be posted to store the company type in the database. As shown in the above figure 5.7. "Add Company type” will post the data to the Java microservices that call the class method and store the desired data in MongoDB. The same process will be applied to add the countries by posting the data with the help of “Add Country” in the postman.

5.4.5 Database Representation

Figure 5.9: Database Representation – Company Type and Country

Company types and countries are stored in MongoDB, representing the unique ID, company type code, and description.

5.5 Account/Profile and Preferences Creation

Buyers or sellers must create an account with the system to achieve the functionality of data trading. The DSP trading system allows to create the accounts for both buyers and sellers and maintains their preferences on the Ganache blockchain associated with the Ethereum account addresses.

5.5.1 Class Diagram

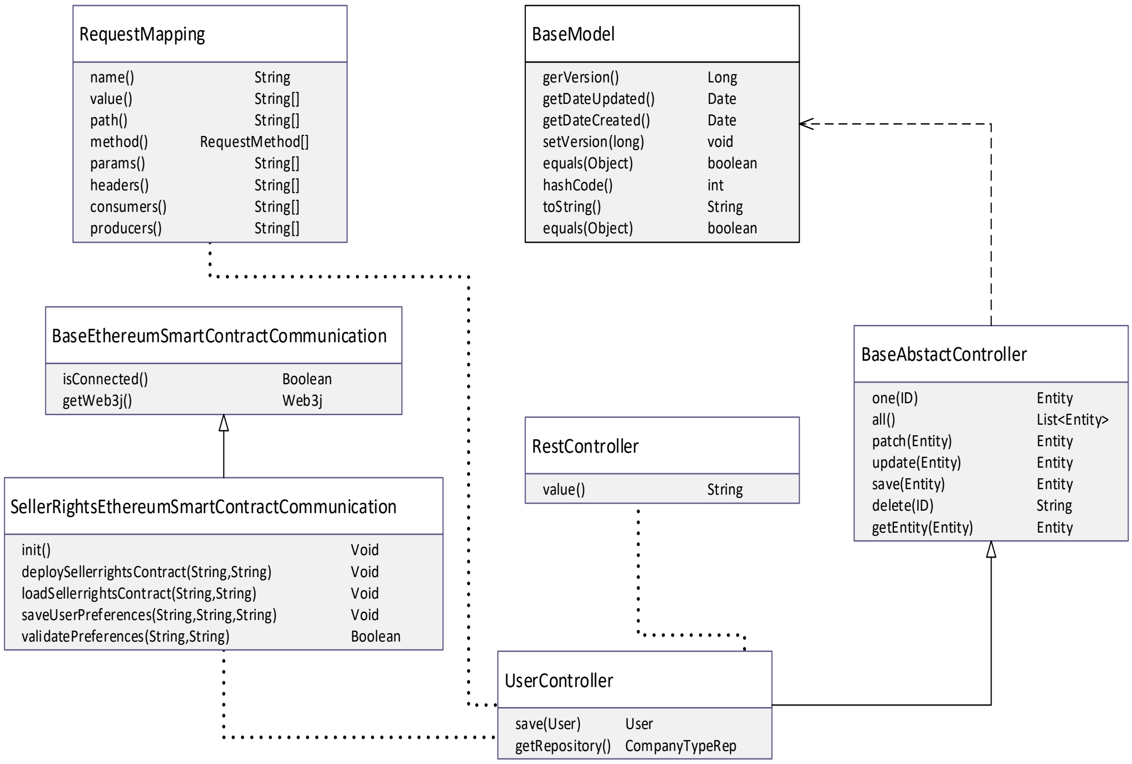

Figure 5.10: Class Diagram- Account/Profile Creation

The whole idea of classes in this module is similar, as described in the previous section 5.4.1. However, UserController class will be responsible for maintaining the seller/buyer profile on the system. The seller or buyer is generalised as an entity and passed to BaseAbstractController class for further processing. At the time of profile creation, DSP is saved to the Ethereum blockchain using the smart contract. Figure 5.8 demonstrate the class diagram for the seller or buyer's account/profile creation module.

5.5.2 ERD Diagram

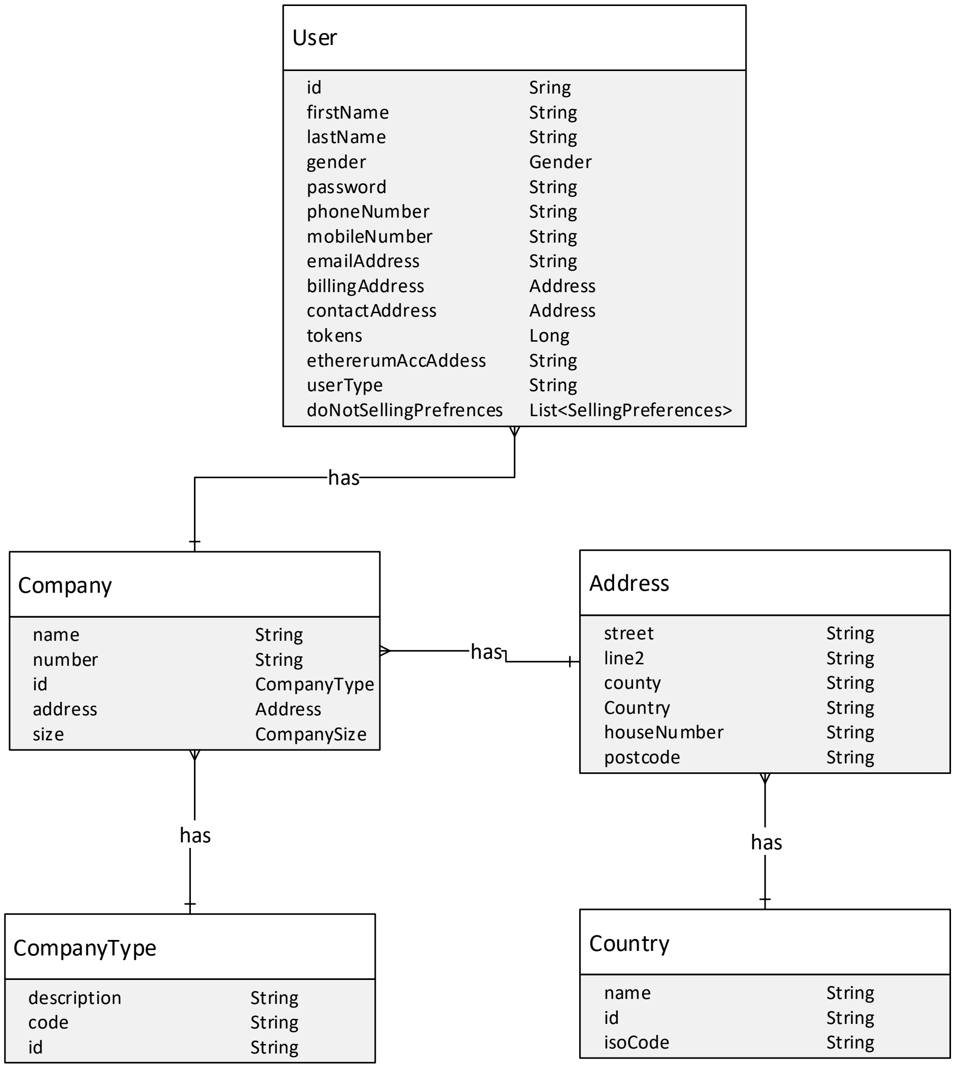

Figure 5.11: ERD Diagram – Account/Profile Creation

User accounts contain the necessary information to validate the system during data trading. In other words, members will be verified on the Ethereum application before doing any transaction or searching the data on the system.

5.5.3 Account Types

Users are categorised into two types, seller and buyer. Both entities are active participants in the data trading life cycle. Let us elaborate on each entity one by one and their behaviour with the smart contract.

i) The seller registers his account by providing the necessary profile information. The system processes the information by interacting with the smart contract to store the DSP on the blockchain. These preferences are essential for data trading. The system communicates with the blockchain by the function in the class, as mentioned above in figure 5.9. saveUserPreferences() method takes the seller's Ethereum account address and saves the preferences on the blockchain by calling the smart contract function. That is an important task of the system because these preferences will be further used for validation. These preferences are composed of reference data such as company types, county or company size. The seller can include none or all preferences to restrict the data trade for buyers.

ii) On the other hand, the buyer registers his account like the seller performing the registration process mentioned in the above section. The only difference is DSP trading system extracts the parameters from the buyer's company object such as country, company type and size of their company and send to the preferences smart contract. This information is used to validate against the seller's preferences, if matched, then the sales will be denied. The same function saveUserPreferences() will be used to save the preferences with the

Ethereum buyer's account address using a smart contract and save this information on the blockchain.

5.5.4 Smart Contract Profile

Seller/buyer preferences parameters will be stored within the Ganache Ethereum blockchain. This is the most important step because DSP will be validated within a smart contract, enabling the seller to restrict the buyer from buying the data on specific conditions already set by the seller. The smart contract will validate those conditions digitally without any third-party involvement in a secure and reliable manner. The seller and buyer preferences-related parameters are stored in the smart contract as key-value pairs identified uniquely by the Ethereum accounts address. The validation procedure allows the buyer to buy the data from the system and pay the amount suggested by the DSP trading system. As the DSP trading system uses Java-based microservices to integrate the web-based Dapp, there must be a mechanism to communicate with the blockchain and achieve the smart contract functionality in the system. Ethereum is one solution that provides interaction with the smart contract over the blockchain. The Web3j Java library is used for communicating with the smart contract and blockchain with the help of Java Spring Boot microservices.

The smart contract is designed in the Solidity language and compiled into byte code. To integrate with the DSP system, compiled smart contract is converted into a wrapper class to access the function within a Java application. To achieve this task, the Web3j command-line tool is used to convert the smart contract byte code resulting in a well-defined wrapper class ready to use in the Java application. DSP is saved on the blockchain as a ready reference, and these preferences will be used to validate the buyers at the time of data trading by the smart contract. As depicted in the class diagram 5.4 saveUserpreferences() method is responsible for saving the preferences on the Ethereum blockchain at the time of seller and buyer profile creation. The account address is stored on the local database, and the system fetches the available account address and links the seller or buyer with that Ethereum address. Finally, a smart contract will store the particular user on the blockchain. Thus, all the participants can register their profiles on the DSP trading system and preferences on the blockchain to perform the data trading.

Smart contract deployment needs the Ethereum account address. Smart contracts were first deployed on the blockchain through Java application using the Web3j Java library. Once deployed on the blockchain, a smart contract can be loaded with the help of the blockchain address. It is necessary to load the smart contract so that smart contract functions can be called from the Java application and process any transaction. In this case, at the time of the seller/buyer account registration, the DSP trading system store the preferences on the blockchain.

5.5.5 Smart Contract Code Snippet

```
pragma solidity ^0.4.21;

import './strings.sol';

contract Sellerrights {
   using strings for *;

    uint32 public u_id = 0;
    uint32 public sell_u_id = 0;
    string foundFlag="Not Found";
    string[] buyerdetails;

    //user structure
struct User {

    string  userGuid;
    string userPrefernces;
 }

    //map the structures
mapping(address => User) public users;

    //create user on blockchain using smart contract
function savePreferences(address _userEthereumAccAddress, string

   //Store the user
   users[_userEthereumAccAddress].userGuid = _userGuid;
   users[_userEthereumAccAddress].userPrefernces= _preferences;
```

Figure 5.12: Smart Contract code snippet

Figure 5.10 shows the smart contract code representing the preference smart contract designed in Solidity language, which is further compiled into byte code using Ethereum remix web utility and wrapper classes are produced with the help of web3j tools so that it can be interacted with the Java application and perform the smart contract operations over the blockchain. This smart contract validates the preferences set by the seller.

5.5.6 Code Snippet Java App

```
@RestController
@RequestMapping("/users")
public class UserController extends BaseAbstractController<User, String>
{
   private UserRepository userRepository;
   private SellerRightsEthereumSmartContractCommunication sellerRightsEthereumSmartContractCommunication;
   public UserController(final UserRepository pUserRepository) { this.userRepository = pUserRepository; }

   @PostMapping
   public User save(@RequestBody User user) {
      User savedUser = getEntity(user);

      sellerRightsEthereumSmartContractCommunication.registerProfile(savedUser);
```

Figure 5.13: Code Snippet Java App – Account/Profile Creation

The above figure 5.13 shows the interaction of a smart contract with the Java application. Using a preferences smart contract, this code represents how the system saves the seller and buyer profiles on the MongoDB database and their preferences on the blockchain.

5.5.7 Postman Screenshot

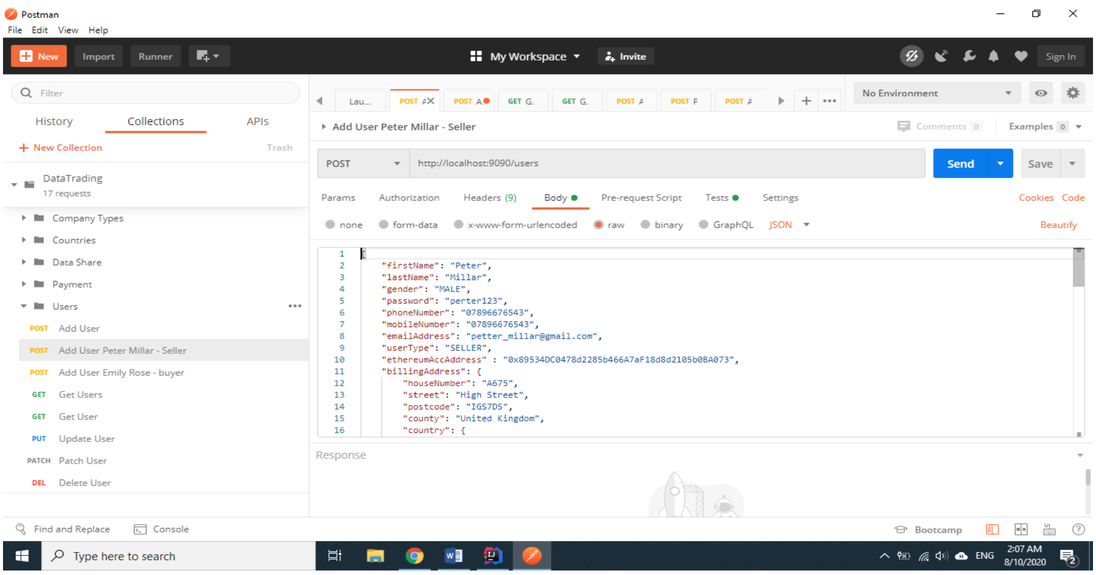

Figure 5.14: Postman data Smart Contract – DSP

Figure 5.14 illustrate the data being posted by the postman to DSP trading system.

5.5.8 Database Representation


```
  _id: "User-2d6c760c-2008-4f18-ac70-d5c52c5d5c96"
  firstName: "Peter"
  lastName: "Millar"
  gender: "MALE"
  password: "perter123"
  phoneNumber: "07896676543"
  mobileNumber: "07896676543"
  emailAddress: "petter_millar@gmail.com"
> billingAddress: Object
> contactAddress: Object
> company: Object
  tokens: 400
  ethereumAccAddress: "0xf9B3a97B3476492fc83fBC96f1b528e6419eF8D8"
  userType: "SELLER"
> allowSellingPreferences: Array
v doNotAllowSellingPreferences: Array
  v 0: Object
      country: DBRef(country, Country-4c3c5d00-f10e-4270-b40f-b8dd96ea5cb7, undefined)
      companyType: DBRef(companytype, CompanyType-228b5ac0-338e-484b-9b42-1df1d46d051b, undefined)
      companySize: "LARGE"
  > 1: Object
  version: 1
  dateUpdated: 2020-08-11T16:59:34.735+00:00
  dateCreated: 2020-08-11T16:59:34.727+00:00
  _class: "uk.ac.brunel.research.datatrading.model.User"
```


Figure 5.15: Database Representation – Account/Profile Creation

The above figure 5.15 DB object shows a user object of type SELLER. Apart from the profile information, the doNotAllowSellingPreferences attribute references the preferences smart contract on the Ganache blockchain.

5.6 Data Selling

This module emphasises data selling activity from the seller. After successful registration on the system, the seller can sell the data readily available for the buyer to purchase. At this stage, the seller preferences are stored in a smart contract. The data seller awarded the DAT tokens for selling the data on the DSP trading system. This process involved the ownership changing from seller to system admin, including the transfer of DAT tokens from Sysadmin to the seller. The workflow of the process starts with searching data on the system. After viewing data, the buyer can purchase that data on successful preference validation with the buyer's profile on the Ganache blockchain.

5.6.1 Class Diagram

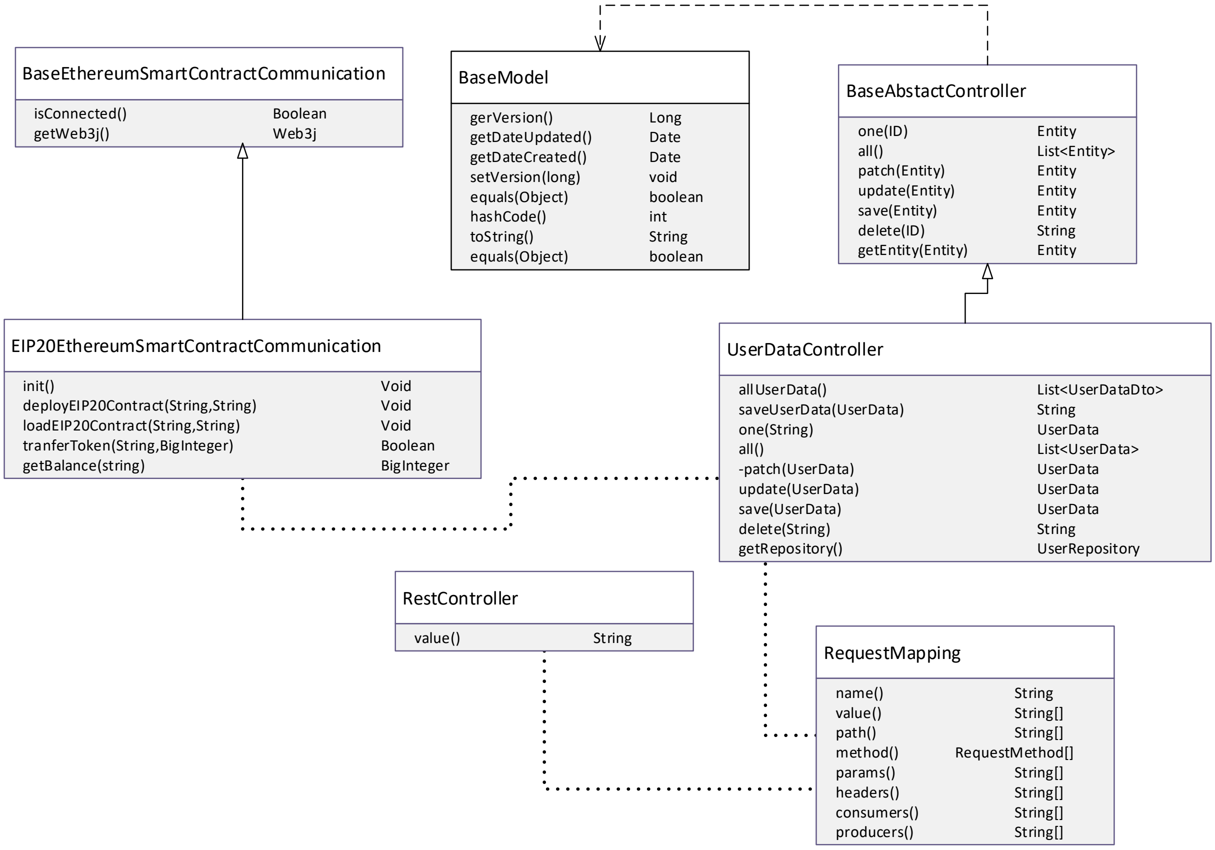

Figure 5.16: Class diagram – Data Selling

Data selling is handled by the methods defined in the UserDataController class. When the seller provides the data, saveUserData() method invoked with the userdata as a parameter and store the data to MongoDB database. Seller data can be fetched from the database b by calling one() method or allUserData() for all seller data according to buyer requirements. Rest of the class methods facilitate other functions of data selling as shown in figure 5.16.

5.6.2 ERD Diagram

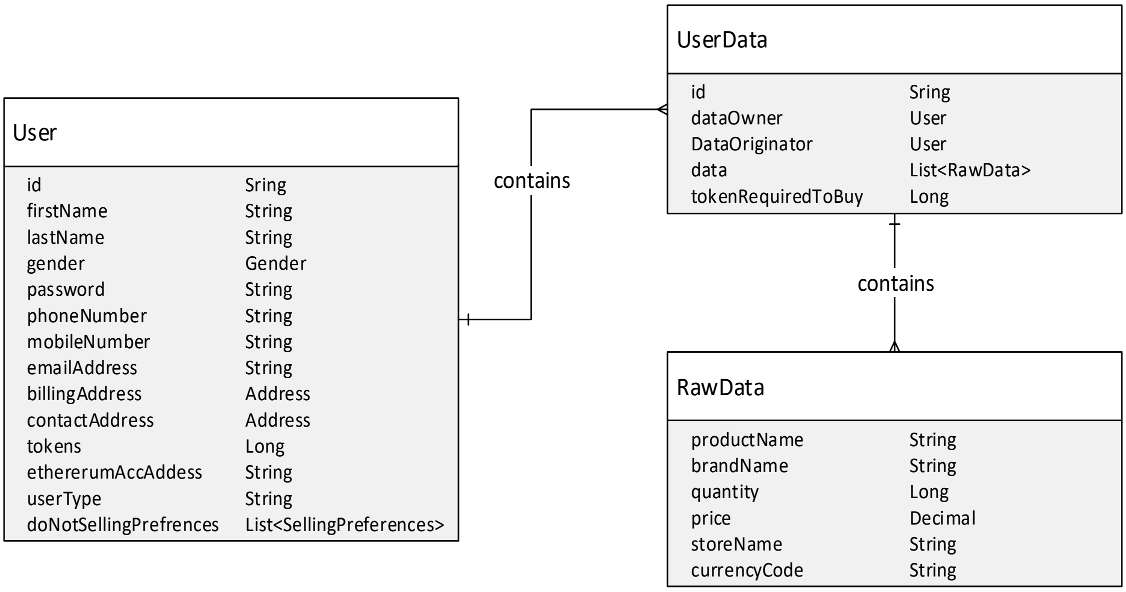

Figure 5.17: ERD diagram – Data Selling

Sellers can sell the data having different fields like product name, brand name, quantity and price, as shown in figure 5.17. Tokens are awarded to the seller depending upon the number of data rows provided and saved in the field 'tokenRequiredToBuy'.

5.6.3 Smart Contract Token Standard

Initially, there was no smart contract programming standard to transfer cryptocurrency from one account to another. Every developer writes his code to achieve the transferring process, which is the tedious practice of doing that task. Developers experienced many difficulties in designing the smart contract in terms of debugging, exception handling and security. Considering such difficulties, the token payment mechanism is standardised to overcome the problem faced by smart contract designers. For this reason, the EIP20 standard was introduced in 2015 to design the token payment smart contacts within the Ethereum echo system. That standardisation eases the smart contract designers to adopt the predefined methods that process the cryptocurrency tokens between parties involved in trading.

The DSP trading system uses EIP20 token standard in the smart contract and takes advantage of hassle-free transactions over the Ethereum blockchain. A seller is awarded tokens as a count of data provided, which are transferred from the system admin to that seller using the EIP20 token standard. The payment controller smart contract is deployed on the blockchain using the Web3j library in the Java application and then loaded into that contract within the same application when the seller sells the data. Tokens are transferred from the system admin to the seller by calling 'transferTokens()' function. This function transfers the token to seller's Ethereum account address from the system admin address.

5.6.4 Smart Contract Code Snippet

```
pragma solidity ^0.4.21;

import "./EIP20Interface.sol";


contract EIP20 is EIP20Interface {


    uint256 constant private MAX_UINT256 = 2**256 - 1;
    mapping (address => uint256) public balances;
    mapping (address => mapping (address => uint256)) public allowed;


    string public name;                   //fancy name: eg Simon Bucks
    uint8 public decimals;                //How many decimals to show.
    string public symbol;                 //An identifier: eg SBX


    function EIP20(
        uint256 _initialAmount,
        string _tokenName,
        uint8 _decimalUnits,
        string _tokenSymbol
```

Figure 5.18: Payment Smart Contract code snippet

Figure 5.18 snippet shows the EIP20 token smart contract designed in Solidity language, inspired by JavaScript that provides the functionality of implementing smart contracts run on Ethereum. The primary purpose of the EIP20 smart contract is to deal with token transfers and get balances of tokens associated with the user on the blockchain. These accounts are the blockchain account addresses linked with the users to save the information and use those addresses for the financial transaction of tokens later on. The above EIP20 token standards are responsible for custom token movement from the system admin account to the seller account concerning blockchain addresses linked with the seller and system admin to perform the token transactions.

5.6.5 Code Snippet Java App

```java
@PostMapping("/share")
public String saveUserData(@RequestBody UserData userData) {

    boolean success = false;
    String add = "";

    // Save the Data and User.
    final UserData savedUserData = userDataRepository.save(userData);

    try {
        final User dataOriginator = userRepository.findById(savedUserData.getDataOwner().getId()).get();
        add = dataOriginator.getEthereumAccAddress();

    } catch (Exception e) {
        return "Please add seller account first......";
    }
```

Figure 5.19: Code Snippet java App – Data Selling

Java application communicates with the local database repository MongoDB to save seller data and EIP20 smart contract used to award tokens to the seller.

5.6.6 Database Representation

```
_id: "UserData-a7ef9ac0-f803-428d-ae21-03316911e80c"
dataOwner: DBRef(user, User-6e5142f2-5e64-4697-b720-428d5f2be173, undefined)
data: Array
  0: Object
    productName: "Cold Drink"
    brandName: "Pepsi"
    quantity: 1
    price: "10"
    currencyCode: "GBP"
    storeName: "ASDA"
  1: Object
version: 0
dateCreated: 2020-08-05T18:05:13.593+00:00
_class: "uk.ac.brunel.research.datatrading.model.UserData"
```

Figure 5.20: Database Representation – Data Selling

Data is stored in the form of an array of objects, and dataOwner attribute references back to the original seller, as shown in the above data object.

5.7 Data Buying

In this section, the data buying module will be defined. As mentioned earlier, the proposed system consists of two smart contracts. In this module, the preferences contract played an essential role in validating the seller and buyer preferences and controlling the data trading. Buyers can register their profile, and DSP collects preferences-related parameters and stores them in smart contracts. The buyer can then list the data provided by the seller in section 5.6 and attempt to purchase. However, the smart preference contract will either allow or deny the transaction based on the preferences set by the seller. For example, if the seller sets the country:Russia, then no buyers from Russia can buy the data block.

5.7.1 Class Diagram

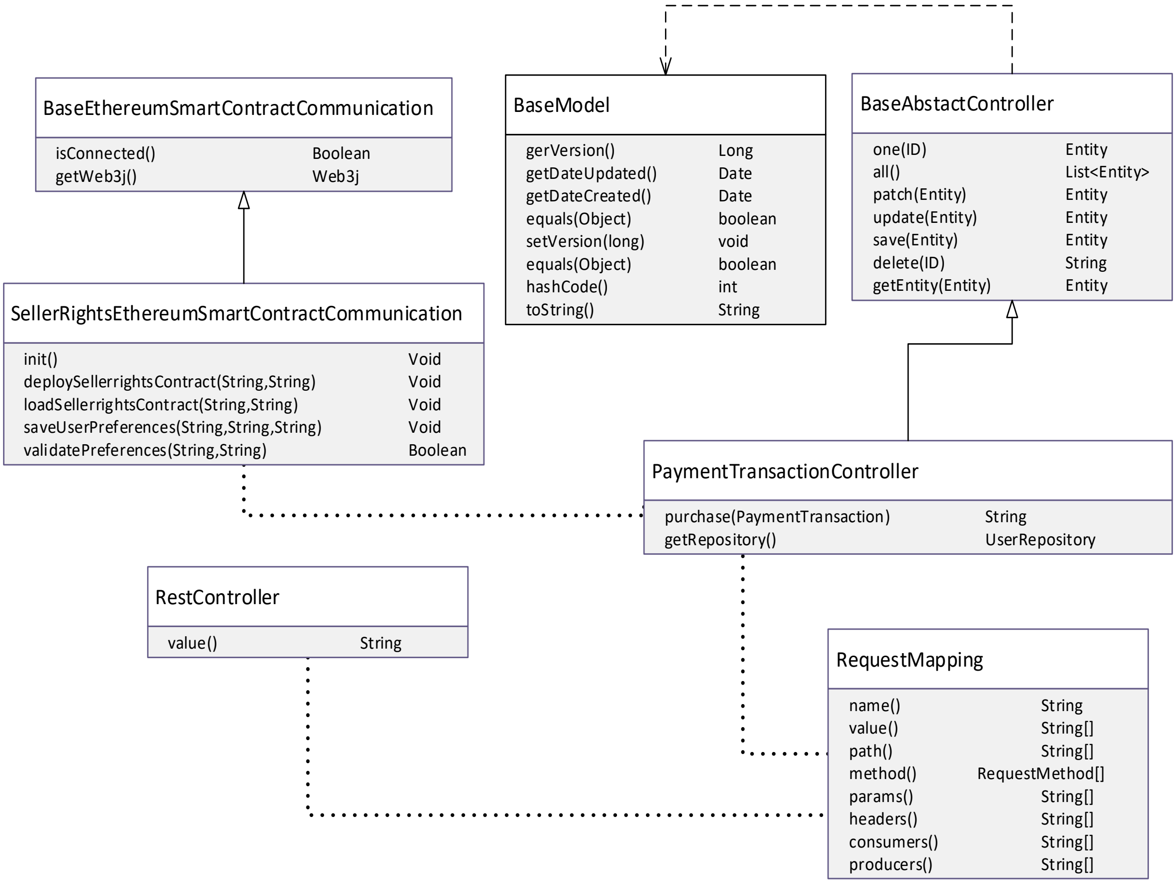

Figure 5.21: Class diagram – Payment

The buyer lists data. Then attempts to purchase the data the system calls PaymentTransactionController, the purchase method as shown in the figure 5.21. The purchase() method handles the payment mechanism between the buyer and the system admin. Considering the seller and buyer Ethereum account addresses as the input of a function, the smart contract retrieves the DSP stored at the time of seller registration. The evaluation will occur using a smart contract's function validatePreferences(). Once the buyer satisfies the preferences pre-defined by the seller, then the DSP trading system allows the buyer to access the data. The transaction will be completed after this step.

5.7.2 ERD Diagram

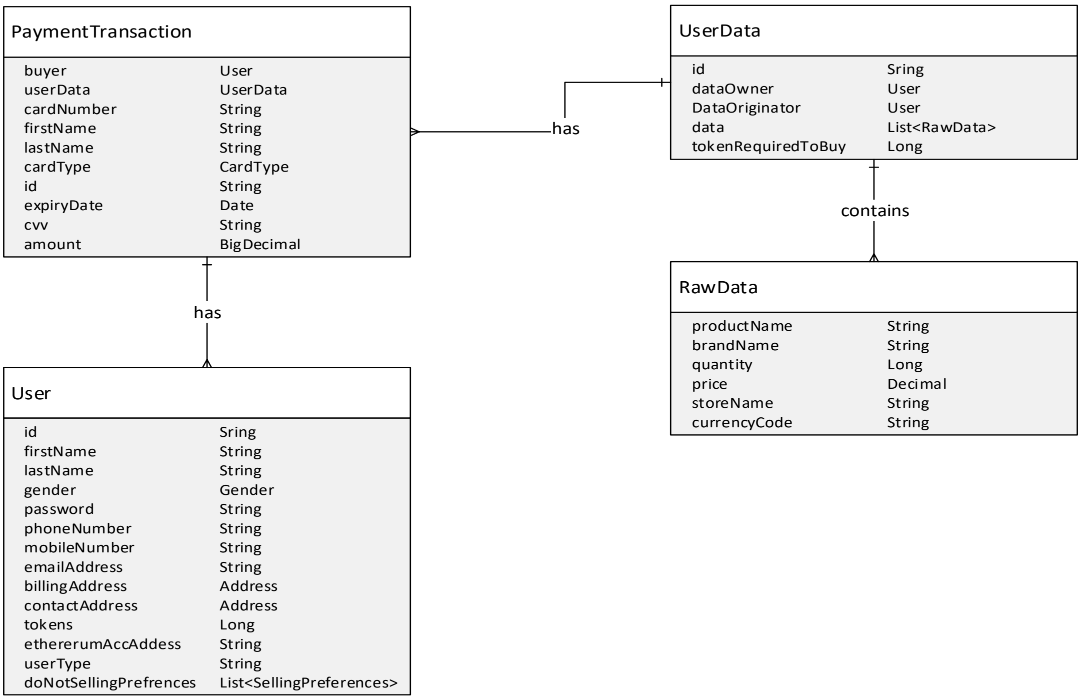

Figure 5.22: ERD diagram – Data Buying

Payment details contain the payment method, card number, expiry date, name on the card and CVV. For this research, we keep it as a mock payment. Whenever a buyer purchases the data from the system, the record will be maintained its association with the seller-id and the data-id linked with it. This data will no longer be available for trading, and buyer can have unlimited access to the data block.

5.7.3 Preferences Smart Contract Validation

The preferences smart contract will validate these seller preferences against the buyer ones digitally without any human or third-party effort. In this module, validation will take place to control data trading. The system will allow the buyer to buy the data if they satisfy the seller's preferences.

The buyers list the data, which also contains the price to pay. This price is calculated on the number of data rows calculated by the system. Buyer and seller preferences are stored in preferences smart contract as key-value pairs identified by the Ganache blockchain. Once validate the preferences smart contract parses the key-value pairs, compares key by key with values, and evaluates if the buyer matches any of the key-value defined by the seller. If matched, then the purchase will be denied; otherwise, will be allowed.

5.7.4 Preferences Smart Contract Code Snippet

```
function validateSellerRights(address _sellerEthereumAccAddress,address _buyerEthereumAccAddress)
{
        //buyer details parsing
        parseBuyerPref(_buyerEthereumAccAddress);

        //prasing seller prefrences
       var s = users[_sellerEthereumAccAddress].userPrefernces.toSlice();
       var delim = "|".toSlice();
       var parts = new string[](s.count(delim));
       for(uint i = 0; i < parts.length; i++)
        {
         parts[i] = s.split(delim).toString();
        }
```

Figure 5.23: Preferences Smart Contract validation

When a buyer lists the data and is ready to purchase the selected data, the system behind the scene validates the buyer using a smart contract whether the buyer qualifies for the purchase. This process will take place with the help of implemented business logic in the smart contract. DSP trading system calls validataSellerRights() function by providing the Ganache blockchain address of the seller and buyer. These addresses are used to fetch the DSP and the buyer's preferences. After this stage, both DSP and buyer's profile respective parameters will be matched in the function mentioned above. If the match is found, the smart contract system returns the true or false, indicating whether the party is a valid buyer. The system allows the buyer to purchase the data or deny the transaction, depending on the validation.

5.7.5 Code Snippet Java App

```
@PostMapping("/purchase")
public String purchase(@RequestBody PaymentTransaction paymentTransaction)
{
    final Optional<User> buyer = userRepository.findById(paymentTransaction.getBuyer().getId());
    final Optional<UserData> userData = userDataRepository.findById(paymentTransaction.getUserData().getId());

    boolean found = false;
    String buyerInfo = "";

      // smart contract validation on the basis of seller preferences and buyer info
       String sellerEthererumAddress = userData.get().getDataOriginator().getEthereumAccAddress();
       found = sellerRightsEthereumSmartContractCommunication.validatePreferences(
               sellerEthererumAddress,buyer.get().getEthereumAccAddress());
```

Figure 5.24: Code Snippet – Data Buying

As described earlier, Java microservices will communicate with the blockchain smart contract for performing the validation task and the validation code is embedded in the smart contract. The preferences smart contract is already deployed and loaded in the memory so that Java application can communicate with the Ganache blockchain. Thus, the smart contract validation function is called to progress the data trading activity between the seller and buyer, whether successful or unsuccessful, depending on the validation.

To perform the validation process, purchase() method will be called at the time of purchasing data and interact with the smart contract for the validation process. The trade will take place on successful indication from the smart contract, and data will be referenced from the system admin to the buyer. The buyer will pay the amount according to the data volume and get unlimited data access that will no longer be available on the data repository for further trade.

5.7.6 Postman Screenshot

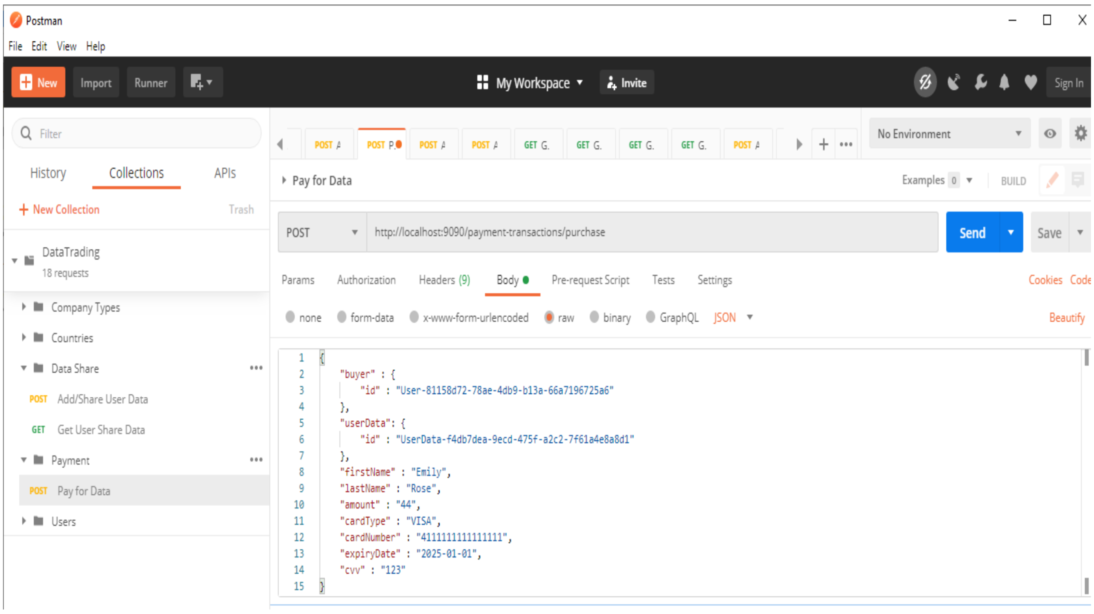

Figure 5.25: Postman DSP validation

Data can be viewed and purchased by posting the request to a Java application. The buyer attempts to pay by providing the necessary details. These inputs will be used to validate the buyer with the DSP. Upon successful validation, data ownership will change to the buyer.

5.7.7 Database Representation


```
_id: "PaymentTransaction-c575c2e2-7a70-4463-a78f-99d85464c07c"
buyer: DBRef(user, User-81158d72-78ae-4db9-b13a-66a7196725a6, undefined)
userData: DBRef(userdata, UserData-f4db7dea-9ecd-475f-a2c2-7f61a4e8a8d1, undefined)
firstName: "Emily"
lastName: "Rose"
amount: "44"
cardNumber: "4111111111111111"
cardType: "VISA"
expiryDate: 2025-01-01T00:00:00.000+00:00
cvv: "123"
version: 0
dateCreated: 2020-08-11T17:03:28.614+00:00
_class: "uk.ac.brunel.research.datatrading.model.PaymentTransaction"
```


Figure 5.26: Database Representation – Data Buying

Figure 5.26 shows the DB object of the payment transaction with the payment details along with the buyer details.

5.8 Thesis-specific Requirements Implemented

Table 5.1 lists the thesis-specific requirements from table 3.4 designed and implemented in chapters 4 and 5. These build artefacts will be validated in the next chapter 6.

Requirement	R. No.
As a seller, I want to define my selling preferences so that my selling rights will be protected.	3.3
As a system admin, I need to add company types and countries so that sellers and buyers can reference them for their account helping to define the seller preferences.	3.10

As a process, I want to store the seller preferences on the smart contract over the blockchain, so that data selling is controlled.	3.12
As a process, I want to store the buyer's preference related attributes on the blockchain, so that these details will be used to evaluate with the help of smart contract before data purchase to protect their selling rights.	3.14
As a process, I need to execute the DSP smart contract before purchase to control the data purchase.	3.16

Table 5.1: Thesis-specific requirements implement

5.9 Summary

This chapter covered the low-level design and the implementations of requirements list out in table 3.4 in light of the high-level design provided in chapter 4. As a novelty, DSP trading system data trading is shaped into a new paradigm in which data selling is based on seller preferences within smart contracts. This concept gives the seller a level of trust and control as they can now define their preferences to whom they are interested in selling their data. A modular approach has been adopted from sections 5.3 to 5.7 and detailed each module so that the scenarios would be visible at a glance. The application is elegantly designed and integrated by incorporating the latest and state-of-the-art computer technologies to take advantage of the rapid blockchain application development. All modules are described with the help of class diagrams, ERD diagrams and database representation.

The process of data trading focused on DSP and smart contract guarantees to perform the validation on its own to control the data selling using smart contract over the blockchain, which answers the research question of this thesis mentioned in section 2.6. Chapter six will focus on validating the thesis-specific requirements mentioned in table 5.1.

Chapter 6: DSP VALIDATION WITHIN SMART CONTRACT

6.1 Introduction

This chapter aims to validate the demonstrable DSP trading system where the DSP smart contract controls trading. Designing and implementing the DSP trading system as per requirements from table 3.4 have been achieved in the previous chapters 4 and 5, respectively. Validation integration test cases are required to validate the thesis-specific requirements (Table 5.1) along with some domain-specific ones. First, test scenarios will be analysed based on Tables 3.4 and 5.1, along with the BPMN diagram (Figure 6.1). Then each scenario will be validated by test steps; hence test results will be produced.

Section 6.2 discusses the DSP validation test scenarios, and from sections 6.3 to 6.5, each scenario found in table 6.1 has been tested with detailed steps, and results have been provided. Finally, section 6.5 focuses on generalizing the preferences smart contract program to an algorithm so that the effectiveness of allowed and denied transactions will be compared while selecting random buyers and sellers from the pre-defined list during simulation.

6.2 DSP Validation Test Scenarios

We have a demonstrable data trading system controlled by DSP at this stage. However, we need to validate the DSP trading system against the novel requirements mentioned in table 5.1 and some of the domain-specific requirements from table 3.4. We need BPMN (business process model notation) figure 6.1 to visualise the system. Figure 6.1 is designed after analysing figures 4.1 and 5.1. Figure 6.1 helps to extract the test scenarios listed in table 6.1.

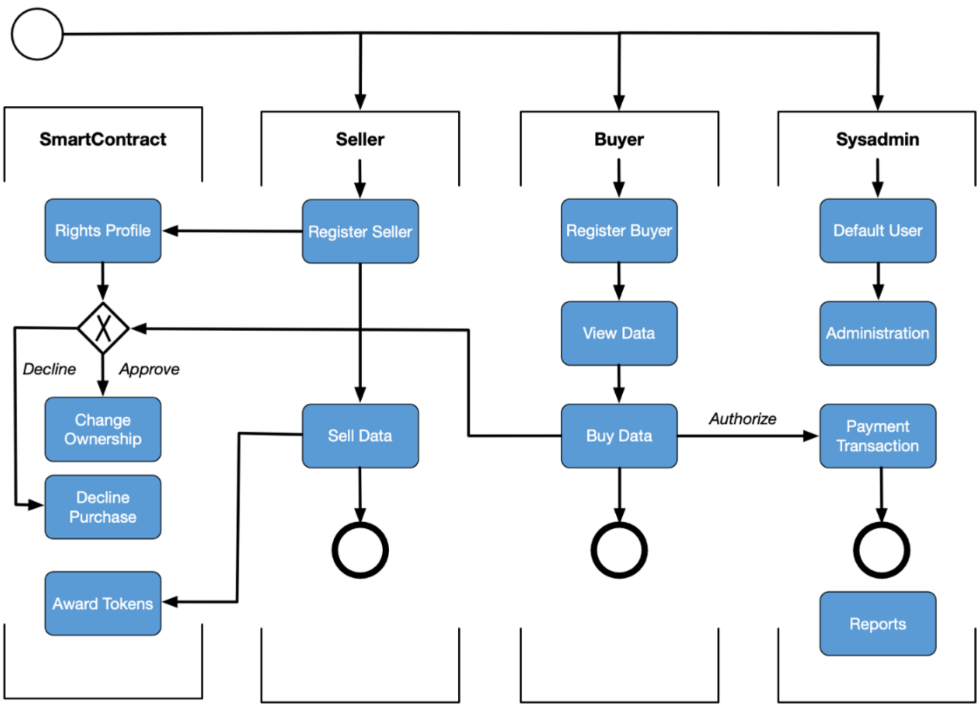

Figure 6.1: Data Trading (DSP within the smart contract) Business Process Model Notation

Figure 6.1 shows a seller registering and selling their data and awarded by tokens. The buyer also registers first and then views the data to buy. Furthermore, buying business process is controlled by the preferences smart contract, and the transaction will be either Approved or Declined based on the seller's preferences. Not all, but some domain-specific system admin test scenarios will be covered.

Test scenarios	**Integration test scenarios description**	**Requirement references, table 3.4**
Test scenario validation 1	The seller "Peter Millar" will log in to the system and register themself to provide the profile and preferences details. The system will store the preferences in the smart contract. The seller will sell their data and be awarded tokens. Then getting seller details will return the same preferences previously set by the seller.	3.1, 3.2, 3.3, 3.4, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13
Test scenario validation 2	Buyer "Dmitri Ivanovich" list the data after registering their details and attempts to buy. However, Dmitri Ivanovich does not comply with the seller's preferences and the preferences smart contract will deny the transaction.	3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.14, 3.16
Test scenario validation 3	Buyer "James Dean" list the data after registering their details and attempts to buy. The seller preferences smart	3.5, 3.6, 3.7, 3.8, 3.9,

	contract will allow the transaction as this James Dean will comply with the seller's preferences.	3.10, 3.11, 3.14, 3.16
Test scenario validation 4	The preferences smart contract program will be generalised to an algorithm so that the effectiveness of allowed and denied transactions will be compared while selecting random buyers and sellers from the pre-defined list during simulation.	3.16

Table 6.1: Test scenario validations

Table 6.1 shows the test scenarios. To validate these scenarios, we will go through each of them by defining the BDD style test steps and finally show the Postman test results of these steps.

6.3 Test Scenario Validation 1

Following is test scenario 1 as per table 6.1.

The seller "Peter Millar" will log in to the system and register themself to provide the profile and preferences details. The system will store the preferences in the smart contract. The seller will sell their data and be awarded tokens. Then getting seller details will return the same preferences previously set by the seller.

6.3.1 Integration Test Steps

```
Feature: Seller preferences
  As a new seller
  I want to register with my preferences
  So that I can log in and sell my data to the system with
  mine preferences loaded.
```

Step #1.1	Get al.l Countries
Given	The user brings up Postman to get al.l countries
When	A user call Get al.l Countries web services
Then	Successfully get the response for all countries for the user to load the one he is interested in

Step #1.2	Get Country (United Kingdom)
Given	The user brings up Postman to get country details
When	A user call Get Country web services with ISO code GBR
Then	Successfully get the response and Postman {{countryID}} variable will be assigned with the system ID of "United Kingdom"

Step #1.3	Get Country (Russia)
Given	The user brings up Postman to get country details
When	A user call Get Country web services with ISO code RUS
Then	Successfully get the response and Postman {{notAllowedCountryID}} variable will be assigned with the system ID of "Russia"

Step #1.4	Get al.l Company Types
Given	The user brings up Postman to get al.l company type
When	A user call Get al.l Company Types web services
Then	Successfully get the response for all company types for the user to load the one he is interested in

Step #1.5	Get Company Type (Industrial and provident society)
Given	The user brings up Postman to get company type details
When	A user call Get Company Type web services with Type code IPS
Then	Successfully get the response and Postman {{ notAllowedCompanyTypeID}} variable will be assigned with the system ID of " Industrial and provident society"

Step #1.6	Add New Seller Peter Millar with preferences
Given	Peter Millar a new seller brings up Postman for registration
When	A Peter Millar call for a register web services

And	He enters the following valid registration details including the email petter_millar@gmail.com and password perter123.
	"firstName": "Peter",
	"lastName": "Millar",
	"gender": "MALE",
	"password": "perter123",
	"phoneNumber": "07896676543",
	"mobileNumber": "07896676543",
	"emailAddress": "petter_millar@gmail.com",
	"userType": "SELLER",
	"billingAddress": {
	"houseNumber": "A675",
	"street": "High Street",
	"postcode": "UB31LH",
	"county": "Middlesex",
	"country": {
	"id": "{{countryID}}"
	}
	},
	"contactAddress": {
	"houseNumber": "A675",
	"street": "High Street",
	"postcode": "UB31LH",
	"county": "Middlesex",
	"country": {
	"id": "{{countryID}}"
	}
	}
	
	Note: {{countryID}} is Postman variable already loaded from Step *#1.2*.

AND	He enters the following preferences related Details "doNotAllowSellingPreferences": [

	{
	"country": {
	"id": "{{notAllowedCountryID}}"
	},
	"companyType": {
	"id": "{{notAllowedCompanyTypeID}}"
	},
	"companySize": "SMALL"
	}
	]
	
	Note: {{notAllowedCountryID}} and {{notAllowedCompanyTypeID}} are Postman variables already loaded from Step *#1.3 & #1.5*.

Then	Successfully get the response and Postman {{ sellerUserID}} variable will be assigned with the system ID of "Peter Millar"

Step #1.7	Login & Get Token
Given	The user brings up Postman to call to login
When	A user call login & get token web services with their username and password.
Then	Successfully get the response, and Postman {{access_token}} variable will be assigned the OAuth2 token. This token will be used to access the secure APIs

Step #1.8	Seller Sold his data awarded tokens
Given	The user brings up Postman to sell his data
When	A user call to sell data web services with a payload of data as JSON.
Then	Successfully get the response after data sold and message of awarded token and ownership changed to system admin

Step #1.9	Get Sold Data
Given	The user brings up Postman to get the sold data
When	A user call to get the sold data web services
Then	Successfully get the response with data sent back as JSON.

Table 6.2: Seller preferences test steps

6.3.2 Test Results

The above test steps, 1.1 to 1.9, have been set up in the Postman, and run the test steps to get the following test results. Each test step asserted with the successful response code and Postman variables have been set up while running the test so that tests can be automated.

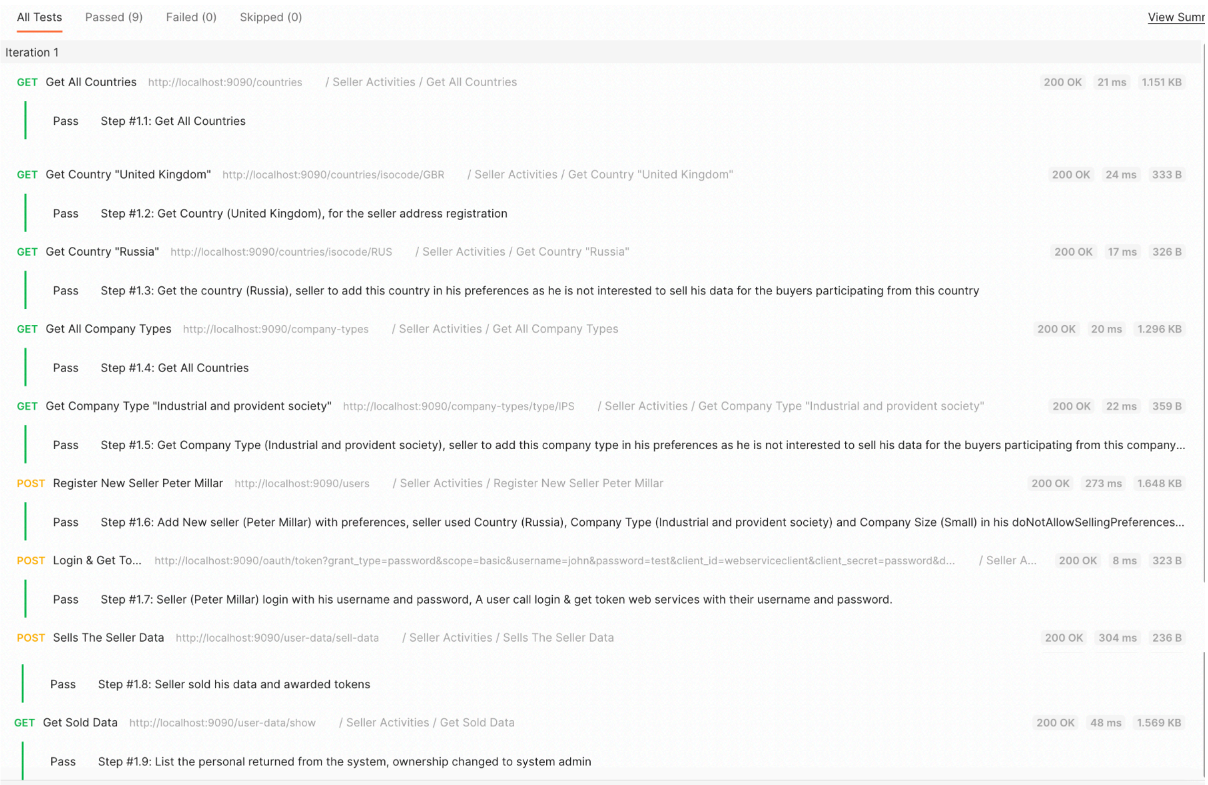

Figure 6.2: Test results of feature seller preferences

All the test cases have passed from steps 1.1 to 1.9, concluding that the seller registered for his profile and preferences on the smart contract and sold his data successfully.

6.4 Test Scenario Validation 2

Following is test scenario 2 as per table 6.1.

Buyer "Dmitri Ivanovich" list the data after registering their details and attempts to buy. However, Dmitri Ivanovich does not comply with the seller's preferences and the preferences smart contract will deny the transaction.

6.4.1 Integration Test Steps

```
Feature: Buyer not allowed to purchase
  As a new buyer
  I want to register with my company details
  So that I can log in and buy data
```

Step #2.1	Get al.l Countries
Given	The user brings up Postman to get al.l countries
When	A user call Get al.l Countries web services
Then	Successfully get the response for all countries for the user to load the one he is interested in

Step #2.2	Get Country (Russia)
Given	The user brings up Postman to get country details
When	A user call Get Country web services with ISO code RUS
Then	Successfully get the response and Postman {{countryID}} variable will be assigned with the system ID of "Russia"

Step #2.3	Get al.l Company Types
Given	The user brings up Postman to get al.l company type
When	A user call Get al.l Company Types web services
Then	Successfully get the response for all company types for the user to load the one he is interested in

Step #2.4	Get Company Type (Industrial and provident society)
Given	The user brings up Postman to get company type details
When	A user call Get Company Type web services with Type code IPS

Then	Successfully get the response and Postman {{companyTypeID}} variable will be assigned with the system ID of " Industrial and provident society"

Step #2.5	Add New Buyer Dmitri Ivanovich
Given	Dmitri Ivanovich a new buyer brings up Postman for registration
When	A Dmitri Ivanovich call for register web services
And	He enters the following valid registration details including the email dmitri_ivanovich@gmail.com and password mypassword123.

```
{
    "firstName": "Dmitri",
    "lastName": "Ivanovich",
    "gender": "MALE",
    "password": "mypassword123",
    "phoneNumber": "+7123-4567-901",
    "mobileNumber": "+7123-4567-901",
    "emailAddress": "dmitri_ivanovich@gmail.com",
    "userType": "BUYER",
    "billingAddress": {
        "houseNumber": "176",
        "street": "ul. Lesnaya d. 5, kv.",
        "postcode": "123456",
        "county": "MOSKVA",
        "country": {
            "id": "{{countryID}}"
        }
    },
    "contactAddress": {
        "houseNumber": "176",
        "street": "ul. Lesnaya d. 5, kv.",
        "postcode": "123456",
        "county": "MOSKVA",
        "country": {
            "id": "{{countryID}}"
        }
    },
    "company": {
        "name": "XYZ IPS",
        "number": "ABC-53442",
        "size": "LARGE",
        "type": {
            "id": "{{companyTypeID}}"
        },
        "address": {
            "houseNumber": "176",
            "street": "ul. Lesnaya d. 5, kv.",
            "postcode": "123456",
            "county": "MOSKVA",
            "country": {
                "id": "{{countryID}}"
            }
        }
    }
}
```

	Note: {{countryID}} & {{companyTypeID}} are Postman variables already loaded from *Step #2.2 & #2.4*
Then	Successfully get the response and Postman {{ buyerUserID}} variable will be assigned with the system ID of "Dmitri Ivanovich"

Step #2.6	Login & Get Token
Given	The user brings up Postman to call to login
When	A user call login & get token web services with their username and password.
Then	Successfully get the response, and Postman {{access_token}} variable will be assigned the OAuth2 token. This token will be used to access the secure APIs

Step #2.7	Get Data To Buy
Given	The user brings up Postman to get data to buy
When	A user call to get data to buy web services
Then	Successfully get the response with data sent back as JSON and Postman {{soldDataID}} variable will be assigned with the system ID of data block

Step #2.8	Pay for Data - Not Allowed
Given	The user brings up Postman to buy the data
When	A user call to purchase data web services

AND	{
	"buyer" : {
	"id" : "{{buyerUserID}}"
	},
	"userData": {
	"id" : "{{soldDataID}}"
	},
	"firstName" : "Dmitri",
	"lastName" : "Ivanovich",
	"amount" : "44",
	"cardType" : "VISA",
	"cardNumber" : "4111111111111111",
	"expiryDate" : "2025-01-01",
	"cvv" : "123"
	}
	
	Note: {{buyerUserID}} & {{soldDataID}} are Postman variable and already loaded from *Step #3.5 & #3.7*

Then	Successfully get the response with error message from preference smart contract that you are not allowed for this purchase.

Table 6.3: Buyer not allowed to purchase test steps

6.4.2 Test Results

The above test steps, 2.1 to 2.8, have been set up in the Postman, and run the test steps to get the following test results. Each test step asserted with the successful response code and Postman variables have been set up while running the test so that tests can be automated.

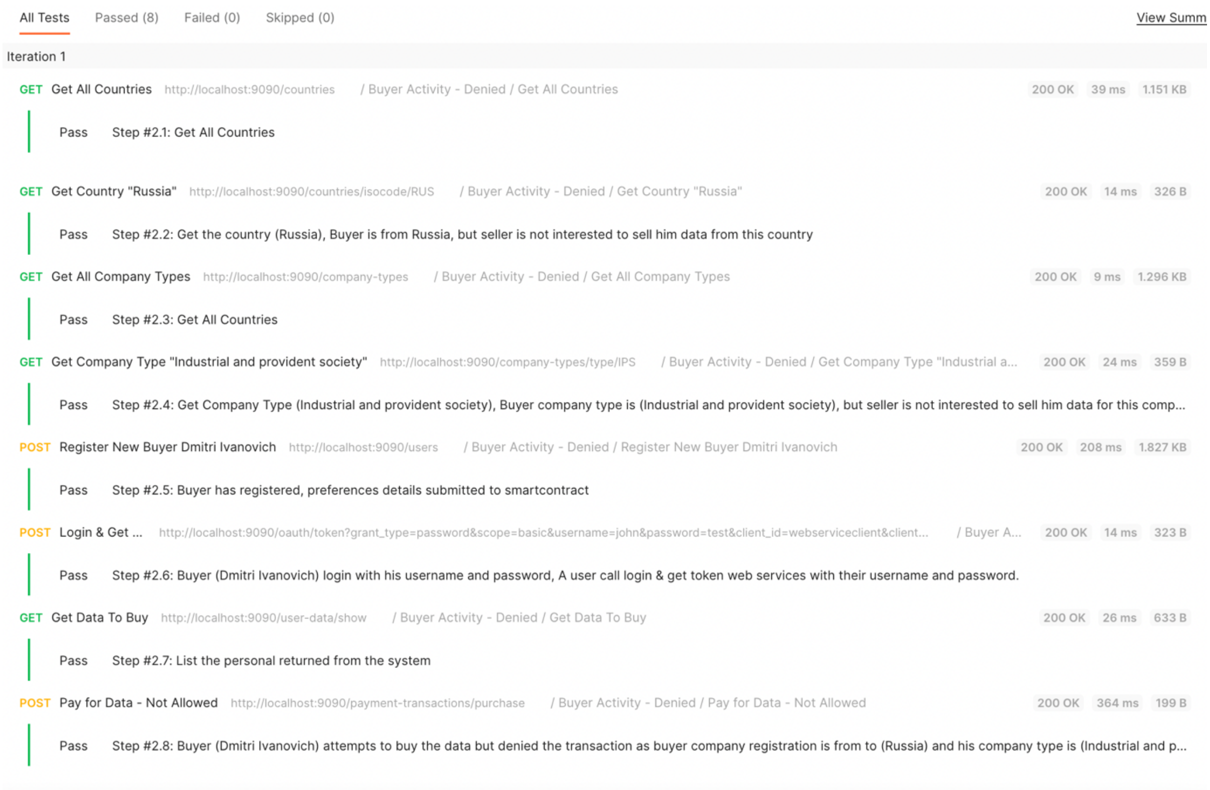

Figure 6.3: Test results of feature buyer not allowed to purchase

All the test cases have passed from steps 2.1 to 2.8, concluding that the preference smart contract denied the transaction initiated by the buyer due to his preferences parameters not complying with the seller's preferences.

6.5 Test Scenario Validation 3

Following is test scenario 3 as per table 6.1.

Buyer "James Dean" list the data after registering their details and attempts to buy. The seller preferences smart contract will allow the transaction as this James Dean will comply with the seller's preferences.

6.5.1 Integration Test Steps

```
Feature: Buyer allowed to purchase
  As a new buyer
  I want to register with my company details
  So that I can log in and buy data
```

Step #3.1	Get al.l Countries
Given	The user brings up Postman to get al.l countries
When	A user call Get al.l Countries web services
Then	Successfully get the response for all countries for the user to load the one he is interested in

Step #1.2	Get Country (United Kingdom)
Given	The user brings up Postman to get country details
When	A user call Get Country web services with ISO code GBR
Then	Successfully get the response and Postman {{countryID}} variable will be assigned with the system ID of "United Kingdom"

Step #3.3	Get al.l Company Types
Given	The user brings up Postman to get al.l company type
When	A user call Get al.l Company Types web services
Then	Successfully get the response for all company types for the user to load the one he is interested in

Step #3.4	Get Company Type (Private company limited by shares)
Given	The user brings up Postman to get company type details
When	A user call Get Company Type web services with Type code LTD
Then	Successfully get the response and Postman {{companyTypeID}} variable will be assigned with the system ID of "Private company limited by shares"

Step #3.5	Add New Buyer James Dean
Given	James Dean a new buyer brings up Postman for registration
When	A James Dean call for register web services

And	He enters the following valid registration details including the email jamesDean@gmail.com and password difficultpass123.
	{
	"firstName": "James",
	"lastName": "Dean",
	"gender": "MALE",
	"password": "difficultpass123",
	"phoneNumber": "07896676543",
	"mobileNumber": "07896676543",
	"emailAddress": "jamesDean@gmail.com",
	"userType": "BUYER",
	"billingAddress": {
	"houseNumber": "D-87",
	"street": "Mission Street",
	"postcode": "90876",
	"county": "",
	"country": {
	"id": "{{countryID}}"
	}
	},
	"contactAddress": {
	"houseNumber": "D900",
	"street": "19 Hill Street",
	"postcode": "IK27S",
	"county": "",
	"country": {
	"id": "{{countryID}}"
	}
	},
	"company": {
	"name": "Muffler Limited",
	"number": "TAC-0098",
	"size": "LARGE",
	"type": {
	"id": "{{companyTypeID}}"
	},
	"address": {
	"houseNumber": "11",
	"street": "West Road",
	"postcode": "GT56D",
	"county": "Hilton",
	"country": {
	"id": "{{countryID}}"
	}
	}
	}
	}
	
	Note: {{countryID}} & {{companyTypeID}} are Postman variables already loaded from *Step #3.2 & #3.4*

Then	Successfully get the response and Postman {{ buyerUserID}} variable will be assigned with the system ID of "James Dean"

Step #3.6	Login & Get Token
Given	The user brings up Postman to call to login
When	A user call login & get token web services with their username and password.
Then	Successfully get the response, and Postman {{access_token}} variable will be assigned the OAuth2 token. This token will be used to access the secure APIs

Step #3.7	Get Data To Buy
Given	The user brings up Postman to get data to buy
When	A user call to get data to buy web services
Then	Successfully get the response with data sent back as JSON and Postman {{soldDataID}} variable will be assigned with the system ID of data block

Step #3.8	Pay for Data - Not Allowed
Given	The user brings up Postman to buy the data
When	A user call to purchase data web services

AND	{
	"buyer" : {
	"id" : "{{buyerUserID}}"
	},
	"userData": {
	"id" : "{{soldDataID}}"
	},
	"firstName" : "James",
	"lastName" : "Dean",
	"amount" : "44",
	"cardType" : "VISA",
	"cardNumber" : "4111111111111111",
	"expiryDate" : "2025-01-01",
	"cvv" : "123"
	}
	
	Note: {{buyerUserID}} & {{soldDataID}} are Postman variable and already loaded from *Step #3.5 & #3.7*

Then	Successfully get the response with congratulations from the preference smart contract that you are allowed for this purchase and ownership transfer to "James Dean"

Table 6.4: Buyer allowed to purchase test steps

6.5.2 Test Results

The above test steps, 3.1 to 3.8, have been set up in the Postman, and run the test steps to get the following test results. Each test step asserted with the successful response code and Postman variables have been set up while running the test so that tests can be automated.

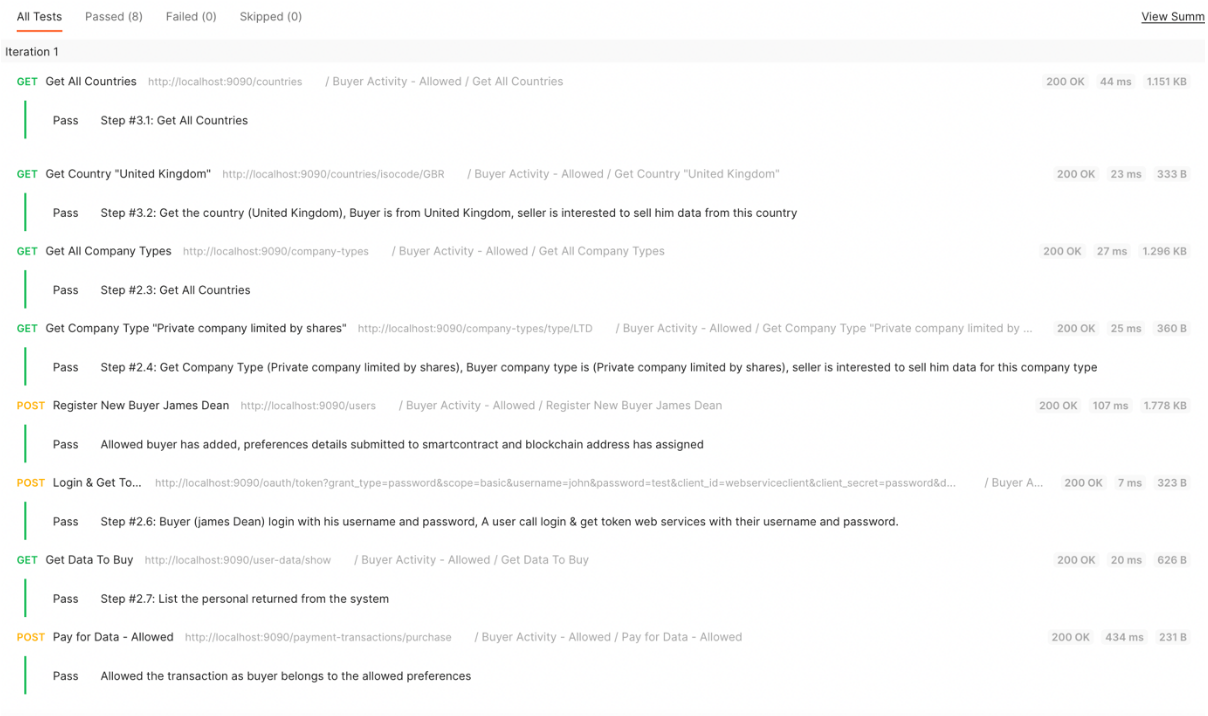

Figure 6.4: Test results of feature buyer allowed to purchase

All the test cases have passed from steps 3.1 to 3.8, concluding that the preference smart contract allowed the transaction initiated by the buyer due to his preferences parameters complying with the seller's preferences.

6.6 Test Scenario Validation 4

Following is the test scenario 4 as per table 6.1

The preferences smart contract program will be generalised to an algorithm so that the effectiveness of allowed and denied transactions will be compared while selecting random buyers and sellers from the pre-defined list during simulation.

To simulate the preference algorithm, the following model formulation is required.

6.6.1 Model Formulation

The preferences smart contract is generalised to find the model with a particular set of information as an input and perform the validation process of the smart contract and produce the output. The data set is a DSP collection containing the information in the equations below.

P = {'COMPANY SIZE', 'COMPANY COUNTRY', 'TYPE OF THE COMPANY'}

B = {'LARGE', 'RUSSIA', 'TOYS'} – Buyer from Russia Company

S = {'SMALL', 'ISRAEL', 'BETTING'} – Tony as Seller

Where P contains a set of company size, company country, and type of company, as the model drives the data trading transaction between buyer and seller; therefore, B holds the parameter values such as Large, Russia and company type as Toys. Similarly, S is comprised of DSP as Small, Russia and Betting, referring to the set P.

$P \neq \emptyset$ (non-empty) set of Preferences.

$\mathbb{P} = \{a_1, a_2, a_3 .... a_n\}$

Before simulating the model, it is worth mentioning here that P must not be empty and have to initialise with non-empty values. P can have a range of values in the form of $a_1, a_2, a_3 .... a_n$, which denotes a set of information related to each entity.

DSP are chosen randomly as well as buyer's profile attributes by the simulation model, which contains non-empty values and must belong to P set as mentioned in the following equations.

$B \neq \emptyset$ (non-empty)

The buyer's parameters system automatically chosen from his profile

$B = \{X : X \in \mathbb{P}\}$

Seller Preferences can be $S = \emptyset$ (Empty) subset of $\mathbb{P}$

$S = \{Y : Y \in \mathbb{P}\}$

The smart contract takes these parameters in the simulation model and performs the validation process by matching the buyer's profile parameters and DSP that was selected automatically previously. The Intersection process between two sets occurs among the set of B and S, which means that trade is valid if there is no match between these two sets. Otherwise, the trade becomes invalid and denies the buyer from purchasing the data.

$B \cap S = \emptyset$ (True of this condition will let the Smart Contract issue a valid purchase)

6.6.2 Model Calibration

Anylogic simulates the model. Parameters are applied to the simulation tool for testing the smart contract functionality. For this purpose, the seller and buyer entities are created, and each entity's parameter list is updated, as mentioned in Table 6.1.

S#	Description	Fields	Type
1	The components of DSP	Country, Company Type and Company Size	Collection as Array
2	Buyer profile preferences as parameters	Company Country, Company Type and Company Size	Single set

Table 6.5: Simulation parameters of seller and buyer

Collection entities such as buyers and sellers participate in this simulation. Exchange act as a simulation agent. Each entity element holds the parameter values treated as inputs for the model calibration. It means that the exchange carries the collection of sellers and buyers participating in data trading. The next step is randomly choosing to route a buyer and a seller from the collection and then route those participants to the smart contract simulation to validate DSP.

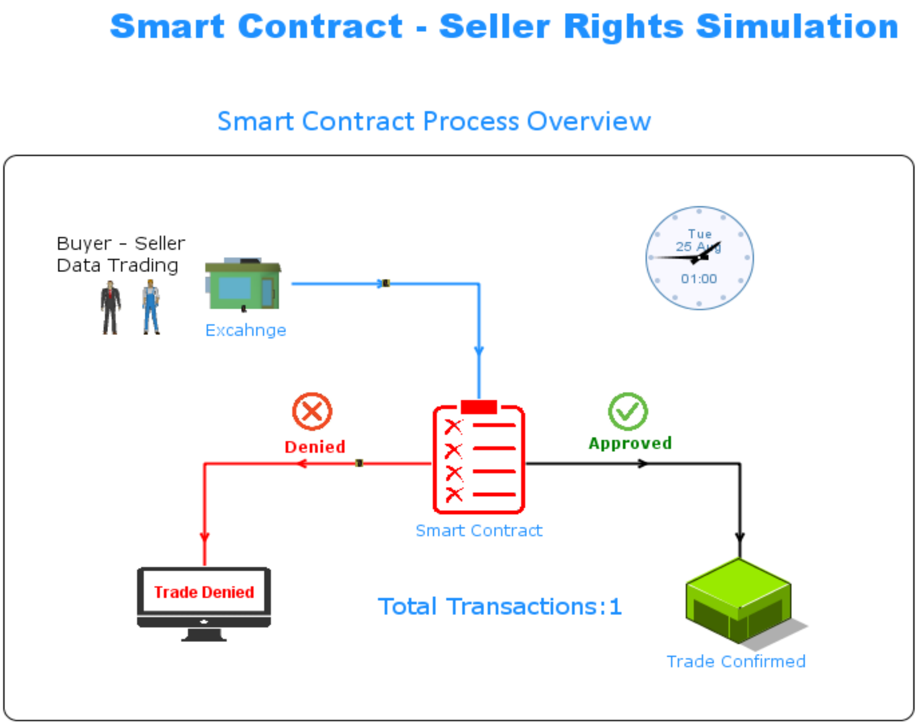

Figure 6.5: Smart Contract process overview

The output derived from the smart contract simulation validation shows the type of data trading transaction having status as approved and denied.

6.6.3 Test Results

Each transaction has been simulated and recorded individually. The preferences of each entity involved in the transaction are compared with participating buyers' and sellers' parameters.

Simulation results are shown in the different types of graphs and charts. This evaluation depends upon the different simulation cases reflected as successful and unsuccessful and the total number of transactions that occur during the simulation.

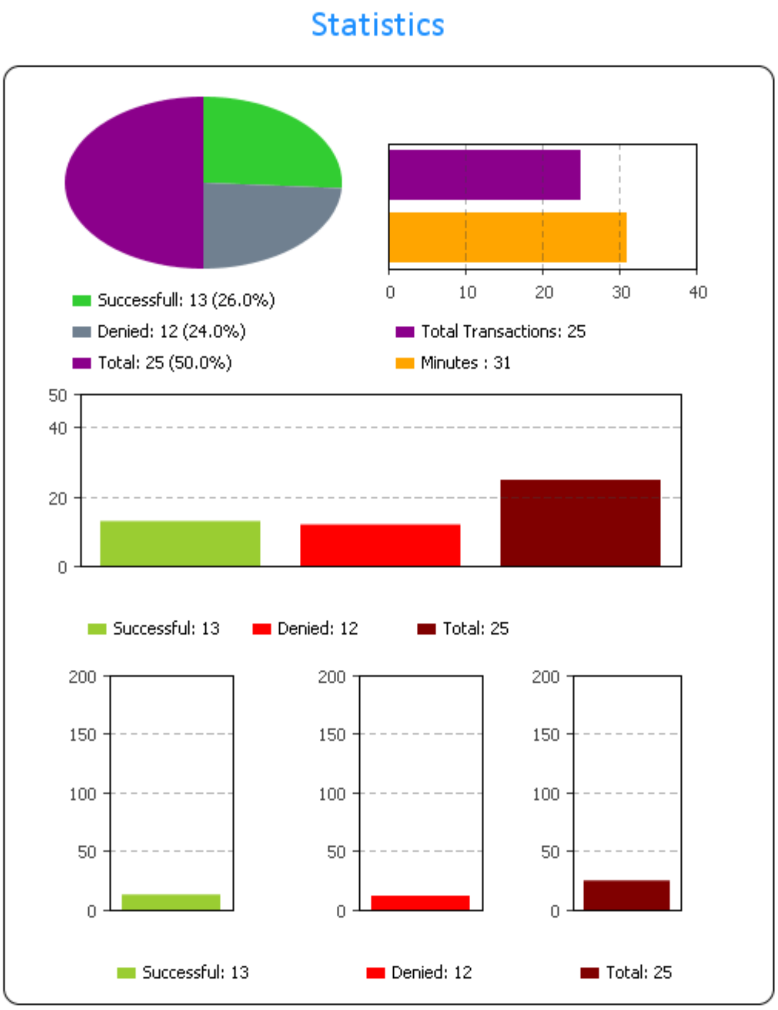

Figure 6.6: Output – Statistics

Figure 6.3 shows 13 successful and 12 denied transactions for a sample simulation run. The model has been simulated with a series of iterations, and the results are recorded in table 6.2.

Iteration	Successful Trades	Denied Trades	Total Trades
1	10	30	40
2	35	10	45
3	15	20	35
4	9	14	15

Table 6.6: Simulation transactions

Table 6.2 compares the successful and denied trade simulation iterated four times to show the effectiveness of the generalised model.

6.7 Summary

This chapter validated the trading system controlled by DSP smart contract. Table 6.1 list the test scenarios for validations purpose. Section 6.3 presented the test steps related to Seller preferences - Test scenario validation 1. Then Postman test automation is used to evaluate the test scenario and provide the test results. Section 6.4 evaluated the buyer not allowed to purchase - Test scenario validation 2. Section 6.5 provided the test results of the buyer allowed to purchase - Test scenario validation 3. Finally, section 6.6 simulated the DSP formulation to find the effectiveness of the generalised preferences smart contract algorithm by comparing the successful and denied trade simulation iterations from table 6.2. This chapter provides sufficient evidence that DSP smart contract controls data trading, which answers our research questions from section 2.6. Next, Chapter 7 will utilise the validations against the requirement from tables 3.4 and 5.1 to demonstrate that the requirements have been met and validated.

Chapter 7: CONTRIBUTION AND FUTURE RESEARCH

7.1 Introduction

Chapter 7 concludes the research and presents key findings. Previous chapter 6 has proved that the DSP smart contract can control data trading between seller and buyer over the blockchain. This research gives a new paradigm of controlling data trading by incorporating DSP within the smart contract. DSP feature gives the seller a sense of control and trust during data trading. A seller defines these preferences to ensure that data selling will only be allowed to those buyers who comply with their preferences. Novelty gives an idea of incorporating DSP within the smart contract to control data selling over blockchain and opens new doors for the researchers to extend the features in this domain further.

Overview of the research discussed in section 7.2. Section 7.3 focused on research findings, section 7.4 described the research limitations, section 7.5 provided future recommendations, and finally, this chapter concluded.

7.2 Research Overview

The literature review in section 2.4 highlighted the possibilities and applicability of personal data trading within the blockchain and its associated features. After a gap analysis of the related research mentioned in table 2.1, this research raised a question in section 2.6 as "Can a Data Seller Preferences (DSP), using Smart Contracts (SC) over blockchain, be developed to control data selling?". Furthermore, the analysis, design and implementation of DSP smart contract artefacts have been realised to control the data trading using the seller preferences. Finally, the novel requirements have been validated, including some domain-specific ones, to prove the DSP can control the data trading within the blockchain, which directly answers our research question in section 2.6.

7.3 Research Findings

This research conducted a detailed analysis of existing literature (section 2.4) and compared the results to produce the gap analyses in table 2.1 within the domain of data trading over the blockchain. Then asked the research question.

"Can a Data Seller Preferences (DSP), using Smart Contracts (SC) over blockchain, be developed to control data selling?"

In the light of gap analysis, the concept of controlling data selling using a DSP smart contract was found novel and set up following objectives to answer this research question.

Objective 1:

Analyse and investigate the major artefacts from the literature related to data trading over the blockchain and capture the domain-specific and novel requirements.

Objective 2:

Design and develop the requirements investigated in Objective 1. Appropriate design tools and technologies are required to build a demonstratable trading system to control the data selling based on DSP.

Objective 3:

Scenario-based integration tests are required to validate the novel or thesis-specific requirements.

Objective 1 has been achieved by listing domain-specific and thesis-specific requirements in table 3.4. Table 7.1 shows the thesis-specific requirements.

Requirement	**R. No.**
As a seller, I want to define my selling preferences so that my selling rights will be protected.	3.3
As a system admin, I need to add company types and countries so that sellers and buyers can reference them for their account helping to define the seller preferences.	3.10
As a process, I want to store the seller preferences on the smart contract over the blockchain, so that data selling is controlled.	3.12
As a process, I want to store the buyer's preference related attributes on the blockchain, so that these details will be used to	3.14

evaluate with the help of smart contract before data purchase to protect their selling rights.	
As a process, I need to execute the DSP smart contract before purchase to control the data purchase.	3.16

Table 7.1: Thesis-specific requirements implemented

Objective 2 has been achieved by designing and implementing the requirements in chapters 4 and 5. By the end of chapter 5, we had a demonstrable DSP trading system, which helped to achieve this research's third objective.

Objective 3 has been achieved in chapter 6 by analysing scenario-based integration tests validating the novel requirements from table 7.1. Table 7.2 provides the test results to answer our research question: Yes, DSP smart contract can control data trading over the blockchain. This is the key contribution of our research apart from design and implementation artefacts.

Test scenarios	**Features**	**Result**
Test scenario validation 1	Seller preference	✓
Test scenario validation 2	Buyer not allowed to purchase	✓
Test scenario validation 3	Buyer allowed to purchase	✓
Test scenario validation 4	Simulation of DSP algorithm	✓

Table 7.2: Test scenarios results

7.4 Research Limitations

This research introduced a novel approach in data trading over blockchain to control the data trading using preferences smart contract, however, the research applied basic or limited preferences parameters. Data selling is controlled by DSP within a smart contract over the blockchain. The DSP includes country, type of buyer's business (e.g., beverages, medicine) and business size (e.g., small, medium or large). Smart contract validates the buyer based on these preferences. These DSPs are limited to attributes including a buyer's country, company type and size of the company. Additionally, more complex preferences parameters can be incorporated into the data trading process.

7.5 Future Recommendations

As discussed in section 7.4, this research is limited to the basic set of preferences to prove that data trading can be controlled via DSP using the smart contract in data trading. However, future research can extend the DSP to more complex ones. Moreover, this research kept the distribution of DAT tokens equal once the seller sold their data. However, this mechanism can be improved to reward the quality data seller more and reduce tokens with less data quality. Similarly, buyers can be offered more price for the quality data and less otherwise. More importantly, DSP can be extended and play an essential role in social media data selling on the trading platform. Future research will define new types of preferences focusing on social media data, which have more than one category, such as emotion, politics and entertainment. These new preferences will help to control selling to different types of buyers. Finally, as the trading platform discourages the download of the data, it can create more problems, so the trading platform can be visualised as a reporting engine with some pre-built reports and SQL-like interface or more advanced ones to generate customer reports.

This research is not limited to data trading. It can be incorporated with other sectors such as public, financial, cross-industry, supply chain, energy, health and technology, where one party agrees with the terms and conditions or rules set by the other party to conduct business transactions. The core idea behind the research gives it a broader sense and covers most sectors.

7.6 Conclusion

This research added a novel feature of controlling data trading over blockchain using the DSP smart contract. The data sellers will enjoy more confidence and control while selling their data. To summarise, this research analysed the research gaps concerning related work covered in chapter 2. Requirements were determined in chapter 3 with the help of literature, and artefacts have been produced focusing on the research question. DSR methodology helped to produce artefacts. Chapters 4 & 5 covered the design and implementation of artefacts accordingly. Finally, DSP within smart contact has been evaluated and simulated in chapter 6.

DSP will be an excellent feature for trading platform owners to provide more facilities to their customers. Although the DSP introduced a basic level of controls, they are open to being extended to more complex ones for future research.

References

Trade in Data, Accessed online on 10 June 2020, https://www.law.ox.ac.uk/business-law-blog/blog/2018/05/trade-data-constructive-limits-personal-data-ownership.

The story of data: How did we get here? Accessed online on 10 June 2020, https://martechtoday.com/the-story-of-data-how-did-we-get-here-215146.

Using Big Data to Predict Consumer Behaviour and Improve ROI, Accessed online on 10 June 2020, https://www.brandpoint.com/blog/using-big-data-to-predict-consumer-behavior-and-improve-roi/.

Blockchain and big Data: A great marriage, Accessed online on 20 June 2020, https://www.finextra.com/blogposting/16596/blockchain-and-big-data-a-great-mariage.

J. F. Rayport and B. J. Jaworski, E-commerce, Irwin/McGraw-Hill, 2000.

Charles Arthur "Tech giants may be huge, but nothing matches big data" Available online at: https://www.bernardmarr.com/default.asp?contentID=1177 Accessed 1/02/2020.

Jake Frankenfield "What is Data Anaalutics" Available online at: https://www.investopedia.com/terms/s/social-data.asp, Accessed 6/02/2020.

Sherman "Nonfinancial Vs. Financial Information" Available online at: https://bizfluent.com/info-7758431-nonfinancial-vs-financial-information.html [Accessed 10/02/2020].

Elea Feit "Your Data Is Shared and Sold…What's Being Done About It?" Available online at: https://knowledge.wharton.upenn.edu/article/data-shared-sold-whats-done/, Accessed 1/04/2020.

Tucker and Neumman "Buying Consumer Data? Tread Carefully." Available online at: https://hbr.org/2020/05/buying-consumer-data-tread-carefully, Accessed 5/04/2020.

"List of big data companies" Available online at: https://en.wikipedia.org/wiki/List_of_big_data_companies , Accessed 11/02/2020.

HENRY H. ECKERSON "GDPR Reference Guide: All 99 Articles in 25 Minutes" Available online at: https://www.eckerson.com/articles/gdpr-reference-guide-all-99-articles-in-25-minutes [Accessed 1/02/2020].

World online, Accessed online on 10 Nov 2020, https://www.duo.uio.no/bitstream/handle/10852/55350/Thesis---Final---Update.pdf?sequence=1&isAllowed=y

Bing, Jon. 2009. "Building cyberspace: a brief history of Internet". In Lee A. Bygrave and Jon Bing's Internet Governance: Infrastructure and Institutions (ed), 8-47. Oxford: Oxford University Press.

Max Roser, Hannah Ritchie and Esteban Ortiz-Ospina (2016) - "Internet". Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/Internet' [Online Resource]

T. Jung, X. Li, W. Huang, J. Qian, L. Chen, J. Han, J. Hou, and C. Su, "Accounttrade: Accountable protocols for big data trading against dishonest consumers," in IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, May 2017, pp. 1–9.

G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trust Management in Decentralized IoT Access Control System," in IEEE ICBC'2020. IEEE, 2020, p. 9.

G.-H. Kim, S. Trimi, and J.-H. Chung, ''Big-data applications in the government sector,'' Commun. ACM, vol. 57, no. 3, pp. 78–85, 2014.

M. Mussa and S. Rosen, ''Monopoly and product quality,'' J. Econ. Theory, vol. 18, no. 2, pp. 301–317, 1978.

Z. Brakerski and V. Vaikuntanathan, ''Efficient fully homomorphic encryption from (standard) LWE,'' SIAM J. Comput., vol. 43, no. 2, pp. 831–871, 2014.

C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, "Fine-grained two-factor protection mechanism for data sharing in cloud storage," IEEE Trans. Inf. Forensics Security, vol. 13, no. 1, pp. 186–196, Jan. 2018.

F. McKeen et al., "Innovative instructions and software model for isolated execution," in Proc. 2nd Int. Workshop Hardw. Architectural Support Secur. Privacy, May 2013, p. 10.

M. J. Yang, ''A design of data trading platform based on cryptology and blockchain technology,'' Inf. Commun. Technol., vol. 10, pp. 24–31, 2016.

X. Cao, Y. Chen, and K. J. R. Liu, ‘‘Data trading with multiple owners, collectors, and users: An iterative auction mechanism,’’ IEEE Trans. Signal Inf. Process. Over Netw., vol. 3, no. 2, pp. 268–281, Jun. 2017.

D. Markovikj, S. Gievska, M. Kosinski, and D. J. Stillwell, “Mining facebook data for predictive personality modeling,” in Seventh Interna- tional AAAI Conference on Weblogs and Social Media, 2013.

S. Zheng, L. Pan, D. Hu, M. Li and Y. Fan, "A Blockchain-Based Trading Platform for Big Data," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 991-996, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162759.

G. F. Camilo, G. A. F. Rebello, L. A. C. de Souza and O. C. M. B. Duarte, "A Secure Personal-Data Trading System Based on Blockchain, Trust, and Reputation," 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 379-384, doi: 10.1109/Blockchain50366.2020.00055.

F. Liang, W. Yu, D. An, Q. Yang, X. Fu and W. Zhao, "A Survey on Big Data Market: Pricing, Trading and Protection," in IEEE Access, vol. 6, pp. 15132-15154, 2018, doi: 10.1109/ACCESS.2018.2806881.

W. Dai, C. Dai, K. -K. R. Choo, C. Cui, D. Zou and H. Jin, "SDTE: A Secure Blockchain-Based Data Trading Ecosystem," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 725-737, 2020, doi: 10.1109/TIFS.2019.2928256.

W. Xiong and L. Xiong, "Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning," in IEEE Access, vol. 7, pp. 102331-102344, 2019, doi: 10.1109/ACCESS.2019.2928325.

T. Weber and W. Prinz, "Trading User Data: A Blockchain Based Approach," 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019, pp. 547-554, doi: 10.1109/IOTSMS48152.2019.8939246.

Enjolras, Bernard. 2014. "Big Data og samfunnsforskning: Nye muligheter og etiske utfordringer". Tidsskrift for samfunnsforskning, 1(1):80-88. Idunn.

Robin Kurzer "The story of data: How did we get here?" Available online at: https://martechtoday.com/the-story-of-data-how-did-we-get-here-215146 6, Accessed 1/02/2020.

"How do companies use my loyalty card data? Available online at: "https://www.bbc.com/news/technology-43483426, Accessed 1/02/2020.

Wieringa, R.: Design science as nested problem solving. In: Proceedings of the 4th Inter- national Conference on Design Science Research in Information Systems and Technology. ACM, Philadelphia (2009)

Bernard Marr "Where Can You Buy Big Data? Here Are The Biggest Consumer Data Brokers" Available online at: https://www.bernardmarr.com/default.asp?contentID=1177 Accessed 1/02/2020.

David Lazarus "Shadowy data brokers make the most of their invisibility cloak" Available online at: https://www.latimes.com/business/story/2019-11-05/column-data-brokers, Accessed 25/02/2020.

Su, et al. "A Blockchain-Based Data Trading Framework with Trusted Execution Environment" https://arxiv.org/pdf/2007.06813.pdf

Lyon, David. 2014. "Surveillance, Snowden, and Big Data: Capacities, Consequences, Critique". Big Data & Society, 1(2):1-13. Sage Journals.

Corey Mitchell, 2019, Access online on 15 10 June 2019, https://www.investopedia.com/terms/t/trading-partner-agreement.asp.

Venable, John R.; Pries-Heje, Jan; and Baskerville, Richard L., "Choosing a Design Science Research Methodology" (2017). ACIS 2017 Proceedings. 112.

Algorithmic trading, 2015, Accessed online on 12 June 2019, https://en.wikipedia.org/wiki/Algorithmic_trading.

Alharby, M. and van Moorsel, A., 2017. Blockchain-based smart contracts: A systematic mapping study. arXiv preprint arXiv:1710.06372.

Atzei, N., Bartoletti, M. and Cimoli, T., 2017, April. A survey of attacks on Ethereum smart contracts (sok). In International Conference on Principles of Security and Trust (pp. 164-186). Springer, Berlin, Heidelberg.

Chiu, J. and Koeppl, T.V., 2019. Blockchain-based settlement for asset trading. The Review of Financial Studies, 32(5), pp.1716-1753.

Cohen, L.R., Samuelson, L. and Katz, H., 2017. How securitization can benefit from blockchain technology. The Journal of Structured Finance, 23(2), pp.51-54.

Cong, L.W. and He, Z., 2019. Blockchain disruption and smart contracts. The Review of Financial Studies, 32(5), pp.1754-1797.

Crosby, M. et al., 2016. Blockchain technology: Beyond bitcoin. Applied Innovation, 2(6-10), p.71.

Dai, W. et al., 2019. SDTE: A secure blockchain-based data trading ecosystem. IEEE Transactions on Information Forensics and Security, 15, pp.725-737.

Destefanis, G. et al., 2018, March. Smart contracts vulnerabilities: a call for blockchain software engineering? In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 19-25). IEEE.

Feng, S. et al., 2018, December. Competitive data trading in Wireless-Powered Internet of Things (IoT) crowdsensing systems with blockchain. In 2018 IEEE International Conference on Communication Systems (ICCS) (pp. 289-394). IEEE.

Gatteschi, V. et al., 2018. Blockchain and smart contracts for insurance: Is the technology mature enough? Future Internet, 10(2), p.20.

Gichigi, T. (2018). A brief history of blockchain. [online] Medium. Available at: https://medium.com/coinmonks/a-brief-history-of-blockchain-70c519d3053 [Accessed 19 Dec. 2019].

Governatori, G. et al., 2018. On legal contracts, imperative and declarative smart contracts, and blockchain systems. Artificial Intelligence and Law, 26(4), pp.377-409.

Jin, R. et al., 2019. Blockchain-Enabled Charging Right Trading Among EV Charging Stations. Energies, 12(20), p.3922.

Kiyomoto, S., Rahman, M.S. and Basu, A., 2017, June. On blockchain-based anonymized dataset distribution platform. In 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA) (pp. 85-92). IEEE.

Liang, F. et al., 2018. A survey on big data market: Pricing, trading and protection. IEEE Access, 6, pp.15132-15154.

Liang, G. et al., 2018. Distributed blockchain-based data protection framework for modern power systems against cyber-attacks. IEEE Transactions on Smart Grid, 10(3), pp.3162-3173.

Lu, Q. and Xu, X., 2017. Adaptable blockchain-based systems: A case study for product traceability. IEEE Software, 34(6), pp.21-27.

Lypchenko, S. (2019). How Can We Benefit from Using Blockchain Technology? - DZone Security. [online] dzone.com. Available at: https://dzone.com/articles/how-can-we-benefit-from-using-blockchain-technolog [Accessed 19 Dec. 2019].

Marr, B. (2018). A Very Brief History Of Blockchain Technology Everyone Should Read. [online] Forbes.com. Available at: https://www.forbes.com/sites/bernardmarr/2018/02/16/a-very-brief-history-of-blockchain-technology-everyone-should-read/#cb57ace7bc47 [Accessed 19 Dec. 2019].

Mattila, J., 2016. The blockchain phenomenon–the disruptive potential of distributed consensus architectures (No. 38). ETLA working papers.

Mingxiao, D. et al., 2017, October. A review on consensus algorithm of blockchain. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2567-2572). IEEE.

MLSDev (2019). Blockchain Architecture Basics: Components, Structure, Benefits & Creation. [online] Medium. Available at: https://medium.com/@MLSDevCom/blockchain-architecture-basics-components-structure-benefits-creation-beace17c8e77 [Accessed 20 Dec. 2019].

Morkunas, V.J., Paschen, J. and Boon, E., 2019. How blockchain technologies impact your business model. Business Horizons, 62(3), pp.295-306.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf.

Nguyen, Q.K., 2016, November. Blockchain-a financial technology for future sustainable development. In 2016 3rd International Conference on Green Technology and Sustainable Development (GTSD) (pp. 51-54). IEEE.

Notheisen, B., Cholewa, J.B. and Shanmugam, A.P., 2017. Trading real-world assets on blockchain. Business & Information Systems Engineering, 59(6), pp.425-440.

Pilkington, M., 2016. 11 Blockchain technology: principles and applications. Research handbook on digital transformations, 225.

Puthal, D. et al., 2018. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. IEEE Consumer Electronics Magazine, 7(4), pp.6-14.

Somdip, D., 2018. A proof of work: Securing majority-attack in blockchain using machine learning and algorithmic game theory (Doctoral dissertation, Modern Education and Computer Science Press).

Sun, Y. et al., 2016. Internet of things and big data analytics for smart and connected communities. IEEE access, 4, pp.766-773.

Vujičić, D., Jagodić, D. and Ranđić, S., 2018, March. Blockchain technology, bitcoin, and Ethereum: A brief overview. In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-6). IEEE.

Yli-Huumo, J. et al., 2016. Where is current research on blockchain technology? —a systematic review. PloS one, 11(10), p.e0163477.

Zheng, Z. et al., 2017, June. An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE International Congress on Big Data (BigData Congress) (pp. 557-564). IEEE.

Zheng, Z. et al., 2018. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4), pp.352-375.

Zhou, J. et al., 2018, July. Distributed Data Vending on Blockchain. In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 1100-1107). IEEE.

Zuo, C. et al., 2017. Fine-grained two-factor protection mechanism for data sharing in cloud storage. IEEE Transactions on Information Forensics and Security, 13(1), pp.186-196.

Zyskind, G. and Nathan, O., 2015, May. Decentralizing privacy: Using blockchain to protect personal data. In 2015 IEEE Security and Privacy Workshops (pp. 180-184). IEEE.

A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, "MedRec: Using blockchain for medical data access and permission management", Proc. Int. Conf. Open Big Data (OBD), pp. 25-30, Aug. 2016.

F. Tian, "A supply chain traceability system for food safety based on HACCP blockchain & Internet of Things", Proc. Int. Conf. Service Syst. Service Manage. (ICSSSM), pp. 1-6, Jun. 2017.

R. AlTawy, M. ElSheikh, A. M. Youssef, G. Gong, "Lelantos: A blockchain-based anonymous physical delivery system", 2017, [online] Available: http://eprint.iacr.org/2017/465.

Marr, B. (2018). A Very Brief History Of Blockchain Technology Everyone Should Read. [online] Forbes.com. Available at: https://www.forbes.com/sites/bernardmarr/2018/02/16/a-very-brief-history-of-blockchain-technology-everyone-should-read/#cb57ace7bc47 [Accessed 19 Dec. 2019].

Randall Stephens. (2019). A peer-to-peer bitcoin cash system. [online] medium.com Available at: https://medium.com/swlh/the-creator-of-bitcoin-satoshi-nakamoto-is-most-likely-this-guy-8723eddb517c [Access 8 April 2020]

Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

V. Buterin et al. A next-generation smart contract and decentralized application platform. white paper, 2014.

R. Kumaresan and I. Bentov. How to use bitcoin to incentivize correct computations? In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014.

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

Object Management Group (2011) Unified modelling language superstructure specification v2.4. http:// www.omg.org/spec/UML/2.4/Superstructure

Todd Grimm: The Human Condition: A Justification for Rapid Prototyping. Time Compression Technologies, vol. 3 no. 3. Accelerated Technologies, Inc. May 1998. Page 1.

Guarino, N., "Formal ontology and information systems", Proceedings of FOIS, (1998), 81–97.

Arsov, D., 2017. Periodic Table of Cryptocurrencies: Blockchain Categorization. Available at SSRN 3095169.

Simon, H. (1996). The sciences of the artificial. 3rd edition. The MIT Press, Cambridge, MA

Dennis, R., Owenson, G. and Aziz, B., 2016, October. A temporal blockchain: a formal analysis. In 2016 International Conference on Collaboration Technologies and Systems (CTS) (pp. 430-437). IEEE.

Dinh, T.T.A., Liu, R., Zhang, M., Chen, G., Ooi, B.C. and Wang, J., 2018. Untangling blockchain: A data processing view of blockchain systems. IEEE Transactions on Knowledge and Data Engineering, 30(7), pp.1366-1385.

Generator, m. 2020. What is mathwashing?. [online] Mathwashing.com. Available at: https://www.mathwashing.com/ [Accessed 9 Jan. 2020].

Glesne, C., 2016. Becoming qualitative researchers: An introduction. Pearson. One Lake Street, Upper Saddle River, New Jersey 07458.

Goldkuhl, G., 2012. Pragmatism vs interpretivism in qualitative information systems research. European journal of information systems, 21(2), pp.135-146.

Medium. 2020. The Limits of Prototyping. [online] Available at: https://medium.theuxblog.com/the-limits-of-prototyping-8661c33fe4cf [Accessed 9 Jan. 2020].

Nurgaliev, I., Muzammal, M. and Qu, Q., 2018, November. Enabling blockchain for efficient spatio-temporal query processing. In International Conference on Web Information Systems Engineering (pp. 36-51). Springer, Cham.

Samuel, O., Javaid, N., Awais, M., Ahmed, Z., Imran, M. and Guizani, M., 2019, July. A blockchain model for fair data sharing in deregulated smart grids. In IEEE Global Communications Conference (GLOBCOM 2019).

Shah, P. (2019). Part II: Designing Smart Contracts: Considerations Involved. [online] Entrepreneur. Available at: https://www.entrepreneur.com/article/332657 [Accessed 6 Feb. 2020].

Venable, J., Pries-Heje, J. and Baskerville, R., 2016. FEDS: a framework for evaluation in design science research. European Journal of Information Systems, 25(1), pp.77-89.

Nick Szabo, Smart Contracts: Building Blocks for Digital Markets (1996), http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html.

Duncan, C. Ho, A. Saxena, and J. F. Chin, “Idc futurescape: Worldwide datacenter 2017 predictions − APEJ implications,” International Data Corporation, Tech. Rep., 2017

Feng, S., Wang, W., Niyato, D., Kim, D. I., & Wang, P. (2018). Competitive Data Trading in Wireless-Powered Internet of Things (IoT) Crowdsensing Systems with Blockchain. 2018 IEEE International Conference on Communication Systems (ICCS).

Gao, W., Yu, W., Liang, F., Hatcher, W. G., & Lu, C. (2018). Privacy-Preserving Auction for Big Data Trading Using Homomorphic Encryption. IEEE Transactions on Network Science and Engineering.

Hess, T., Matt, C., Benlian, A. and Wiesböck, F., 2016. Options for formulating a digital transformation strategy. MIS Quarterly Executive, 15(2).

Li, Z., Kang, J., Yu, R., Ye, D., Deng, Q. and Zhang, Y., 2017. Consortium blockchain for secure energy trading in industrial Internet of things. IEEE transactions on industrial informatics, 14(8), pp.3690-3700.

Oh, H., Park, S., Lee, G. M., Heo, H., & Choi, J. K. (2019). Personal Data Trading Scheme for Data Brokers in IoT Data Marketplaces. IEEE Access, 1–1.

Ølnes, S., Ubacht, J. and Janssen, M., 2017. Blockchain in government: Benefits and implications of distributed ledger technology for information sharing.

Darren Duxbury, Tommy Gärling, Amelie Gamble & Vian Klass (2020) How emotions influence behavior in financial markets: a conceptual analysis and emotion-based account of buy-sell preferences, The European Journal of Finance, 26:14, 1417-1438, DOI: 10.1080/1351847X.2020.1742758

Cravens, David W., Thomas N. Ingram, Raymond W. LaForge, and Clifford E. Young (1993), "Behavior-Based and Outcome-Based Salesforce Control Systems," Journal of Marketing, 57 (October), 47–59.

Karen E. Flaherty, Todd J. Arnold & C. Shane Hunt (2007) The Influence of the Selling Situation on the Effectiveness of Control: Toward a Holistic Perspective, Journal of Personal Selling & Sales Management, 27:3, 221-233, DOI: 10.2753/PSS0885-3134270302

M.G. Cox, P.M. Harris, Design and use of reference data sets for testing scientific software, Analytica Chimica Acta, Volume 380, Issues 2–3, 1999, Pages 339-351, ISSN 0003-2670,

https://doi.org/10.1016/S0003-2670(98)00481-4.

Tian, L., Li, J., Li, W., Ramesh, B., & Cai, Z., (2019). Optimal Contract-based Mechanisms for Online Data Trading Markets. IEEE Internet of Things Journal, 1–1.

West, D.M., 2016. How 5G technology enables the health Internet of things. Brookings Center for Technology Innovation, 3, pp.1-20.

Zhao, Y., Yu, Y., Li, Y., Han, G., & Du, X. (2019). Machine learning based privacy-preserving fair data trading in big data market. Information Sciences, 478, 449–460.

Antonopoulos, A.M. (2014), Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly Media, Sebastopol, CA

“Smart contract,” 2016, [Online; accessed 25-Feb-2017]. [Online]. Available: https://en.wikipedia.org/wiki/Smart_contract

Wood, Gavin. " Ethereum: A secure decentralised generalised transaction ledger." Ethereum Project Yellow Paper (2014).

Preetisha Dayal, 2018, “What are Blockchain Wallets?” [Online]. Available: https://www.newgenapps.com/blog/what-is-blockchain-wallet-and-how-does-it-work

B. Yu, J. Wright, S. Nepal, L. Zhu, J. Liu and R. Ranjan, "IoTChain: Establishing Trust in the Internet of Things Ecosystem Using Blockchain," in IEEE Cloud Computing, vol.5, no. 4, pp. 12-23, Jul./Aug. 2018, doi: 10.1109/MCC.2018.043221010.

Ethereum Wiki. A Next-Generation Smart Contract and Decentralized Application Platform. Available online: https://github.com/ethereum/wiki/wiki/White-Paper (accessed on 15 March 2020).

Fabian Vogelsteller , Vitalik Buterin, 2015, “Eip20 - erc20 token standard,” Available online: https://tinyurl.com/ycd8mzb3 (accessed on 20 March 2020).

Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. Computer Networks. Available online:https://www.cs.mun.ca/courses/cs6910/IoT-Survey-Atzori-2010.pdf (accessed on 23 March 2020).

Nick Szabo, Smart Contracts: Building Blocks for Digital Markets (1996), http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html.

Vitalik Buterin. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. 2013a. URL https://github.com/ethereum/wiki/wik i/White-Paper.

P. Bhattacharjee, P.K.Ray, Patient flow modelling and performance analysis of healthcare delivery processes in hospitals. A review of reflections, Computers & Electrical Engineering 78 (2014).

The AnyLogic Company, 2019 The AnyLogic Company (Ed.) (2019). Available online at https://www.anylogic.com/, updated on 4/29/2019. Google Scholar

Simon HA. Rationality in psychology and economics. J Bus. 1986;59(4 Pt 2):S209–24. [Google Scholar]

Sterman JD. Learning from evidence in a complex world. Am J Public Health. 2006; 96:505–14. [Google Scholar]

Railsback SF, Grimm V. Agent-based and individual-based modeling: a practical introduction. Princeton: Princeton University Press; 2012. [Google Scholar]

V. Buterin, 2014 "A next-generation smart contract and decentralized application platform.," Available online at: https://github.com/ethereum/wiki/wiki/White-Paper/, Accessed 01/08/2020.

N. Szabo. (1994). Formalizing and Securing Relationships on Public Networks. [Online]. Available: http://www.firstmonday. org/ojs/index.php/fm/article/view/ 548/469

Buterin and Vitalik. (2017). ETHEREUM: A Next Generation Smart Contract and Decentralized Application Platform. [Online]. Available: http://ethereum.org/ethereum.html

B. A. Tama, B. J. Kweka, Y. Park, and K.-H. Rhee, ''A critical review of blockchain and its current applications,'' in Proc. Int. Conf. Elect. Eng. Comput. Sci. (ICECOS), Aug. 2017, pp. 109–113

J. Stark, 2016 "Making sense of blockchain smart contracts," Available online at: http://www.coindesk.com/making-sense-smart-contracts/ [Ac-cessed 02/08/2020].

G. Karame and S. Capkun, ''Blockchain security and privacy,'' IEEE Security Privacy, vol. 16, no. 4, pp. 11–12, Jul. 2018.

Ream, J., Chu, Y., Schatsky, D. (2016). Upgrading blockchains: Smart contract use cases in industry. [Blog] Deloitte Insights. Available at: https://www2.deloitte.com/us/en/insights/focus/signals-for-strategists/using-blockchain-for-smart-contracts.html [Accessed 10 Aug. 2020].

Ellis Solaiman , Todd Wike and Ioannis Sfyrakis "Implementation and evaluation of smart contracts using a hybrid on- and off-blockchain architecture" Available online at: https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.5811 [Accessed 10/08/2020].