BLOCKCHAIN BASED TRANSACTION PROCESSING SYSTEM

A REFERENCE ARCHITECTURE FOR AN INTEGRATED BLOCKCHAIN BASED TRANSACTION PROCESSING SYSTEM

MASTER THESIS

SPIJKERBOER, E.C. (CHRISTIAN, STUDENT M-BIT)
30 JANUARI 2019

CAPEGROEP
grensverleggend vooruit

UNIVERSITY OF TWENTE.

BLOCKCHAIN BASED TRANSACTION PROCESSING SYSTEM

A reference architecture for an integrated blockchain based transaction processing system

MASTER THESIS

January 2019

Author

Name: E.C. Spijkerboer (Christian)
Programme: MSc Business Information Technology
Institute: University of Twente
PO Box 217
7500 AE Enschede
The Netherlands
E-mail: e.c.spijkerboer@student.utwente.nl

Graduation Committee

Dr. Maria-Eugenia Iacob
Department Industrial Engineering and Business Information Systems
Email m.e.iacob@utwente.nl

Dr. Marten van Sinderen
Department Computer Science
Email m.j.vansinderen@utwente.nl

Tom ten Vregelaar
Company CAPE Groep
Email t.vregelaar@capegroep.nl

UNIVERSITY OF TWENTE.

CAPEGROEP

Acknowledgements

After six and a half years of studying, this master thesis is the result of my graduation for the Master Business Information Technology at the University of Twente. It feels like yesterday when I started the Bachelor Business & IT and now I can call myself a Master of Science. This study was a perfect fit for my. I could combine my love for IT and business management. I would not have achieved all of this without the help of several people.

First I want to thank Maria Iacob and Marten van Sinderen, my supervisors from the University of Twente. Your mixture of critical opinions and enthusiasm, on the research project and my approaches, motivated me to keep going and finish this project. You also supported me on a personal level in some difficult times and kept on pushing, thank you.

I also want to express my gratitude to everyone at CAPE Groep. First of all my supervisor Tom ten Vregelaar. It was a real pleasure working with you. You provided me with great feedback during our meetings, but it was also a lot of fun. I also want to mention Pieter Verkroost, who gave me the opportunity to do this project and helped me structure my work and thoughts. Finally I would like to thank all my other colleagues for assisting me in mastering Mendix and eMagiz for the prototype and all the fun we had after work. Also the other students who have been working alongside me on their own final projects.

Finally I want to thank all my friends I met before and during my study time. You made studying a lot of fun. Heren Jaarclub The Jesters will live on for ever, and Harambee will always be the best volleyball association. And last but certainly not least, I want to thank my parents, sister and brothers. You have been there from the beginning and are always a constant in my live. Your unconditional love for me has helped me through difficult times, but you also were there to pull me back to reality and kept on motivating and pushing me to achieve all of this. I can't thank you enough!

Christian Spijkerboer

Abstract

For small and medium sized enterprises it can be hard to keep their cashflow healthy. They suffer in accessing financial markets, they are considered high-risk borrowers who need to pay higher interest and they have a problem of insufficient liquidity and working capital. There is a Dutch company that tries to help these enterprises, by providing an IT service for reverse factoring. They want to explore the possibilities to improve their service by using blockchain to record all the transactions between different parties. The traditional transaction processing system work with several different components and their design can differ. But they have one important component in common, they are managed by a single party. Using blockchain as the record keeper, an open and transparent transaction processing system could be created, without a trusted third party. During the literature review, no clear and accepted reference architecture for blockchain based systems was found, therefore this became the main objective of this thesis. Using TOGAF and ArchiMate a new reference architecture for blockchain based transactions systems was created. With a case study, a start of validating the architecture was made. The current architecture was migrated to a new possible architecture using the reference architecture and a prototype of the blockchain based transaction system was made. Based on the prototype, it was concluded that the reference architecture has great potential and it could be tool to create blockchain based transaction systems. But due to the limitations in the prototype, more research is needed to improve both the reference architecture and the blockchain techniques in general.

Management Summary

Context

To improve the financial health of supply chains, there is a need to improve the cashflows of small and medium sized enterprises. It can be hard to gain access to cash, due to the extension of trade credit between buyers and suppliers. To reduces this problem, a relatively new research field called Supply Chain Finance emerged. Supply Chain Finance can be described as a set of solutions, such as reverse factoring, to improve the flow of financial resources between interacting organisations. A Dutch company, revered to as The Broker, provides a platform on which these interacting organisations can improve their cash flow using reverse factoring. Together with one of its partners, CAPE Groep, they want to improve their platform and what to investigate the possibility of using blockchain as a record keeping of all the transactions between the organisations using the platform. This research is conducted at CAPE Groep and aims to explore these possibilities and create a standard for blockchain based transaction systems.

Objectives

When creating a system that will record all the transactions between organisations, a transaction processing system will have to be used. Blockchain can provide an openly shared ledger, that is validated by all involved parties, so no trusted third party is needed to keep the ledger valid. One of the objectives of this thesis is to create such a blockchain based transaction system for The Broker. But during the literature study for this research, no clear reference architecture for such a blockchain based systems was found. Therefore the main objective of this research is to create a standard reference architecture for blockchain based transaction systems. This leads to the main research question of this thesis:

What reference architecture can best provide a basis for the design of a blockchain based transaction processing system, and integrations to existing applications?

Methods

The research starts with a literature study to identify the current state of the related research fields, focused on transaction systems and blockchain. The case of The Broker is used to validate to reference architecture that was designed. A case analysis was done, with a process analysis, a value analysis and a design of the current architecture was made. With this information and the knowledge from the literature, the reference architecture was designed, using TOGAF and Archimate. This reference architecture was then validated by using it to migrate the current architecture of The Broker, to a new target architecture. A prototype of the new system was made, using Mendix, eMagiz and Hyperledger Composer. Last an analysis of the impact of the use of the new architecture was done.

Results

According to the literature, transaction processing systems are information systems that record the transactions that have taken place and could be classified as auditing systems. Blockchain could be used to record these transactions between organisations, without the need of a trusted third party. But blockchain itself poses new problems, like the control of the

blockchain, the reliability of the records and the authenticity. Further research is needed to identify the exact problems and find solutions to these problems. The creation of a standard reference architecture of these blockchain based systems could be useful for future research.

The reference architecture was designed with the help of TOGAF's ADM in combination with the Archimate modelling languages. Five quality attributes were defined to validate the quality of the reference architecture. It can be concluded that the reference architecture gives a clear overview, follows the standards and can be useful. But to verify if its correct, the reference architecture was used to migrate the current architecture of the broker to a new target architecture.

The use of the reference architecture for the case study was a success. From the case analysis the processes were used to create a new business layer with the new processes. And the reference architecture provided clear input on how to change the application and technology layer of The Broker, to support a blockchain based transaction processing system. Based on this new target architecture a prototype could be made to verify the architecture. The prototype could be made and it could submit transactions, via a messaging bus, to the blockchain and create a valid audit trail. But there is a big limitation in the testing of the prototype. A blockchain network was setup to validate the incoming transactions, but there was only one peer in the network. To verify if the audit trail using such a system would be really valid, future research is needed. It should be tested with a much bigger network that is active for a longer period of time.

Conclusions

Looking at the reference architecture, the final conclusion of this thesis is that the reference architecture provides a good basis for the design of a blockchain based transaction processing system. It could be used in different business fields and can provide an open ledger, shared between the actors in the business ecosystem, without the use of a trusted third party. Integration with the enterprise systems of the users is simplified by using an enterprise service bus, which is also used to communicate with the blockchain. By adding the communication layer, a reference architecture for a modular use of a blockchain application was developed. But further research is needed to further validate the reference architecture and make improvements.

List Of Figures

List of Tables

Acronyms

BE	Business Ecosystem
CAPE	CAPE Group B.V.
DS	Design Science
DSRM	Design Science Research Methodology
ERP	Enterprise Resource Planning
FSC	Financial Supply Chain
FSCM	Financial Supply Chain Management
KPI	Key Performance Indicator
PoS	Proof of Stake
PoW	Proof of Work
RQ	Research Question
SC	Supply Chain
SCF	Supply Chain Finance
SME	Small and medium sized enterprises
TP	Transaction Processing

Table of Contents

1 Introduction

Small and medium sized enterprises need help to gain access to cash. There is a Dutch company that does this by offering an IT service for reverse factoring. They want to explore the possibilities to improve their platform by using blockchain in their system. This motivation is explained in depth in section 1.1. The research objective is to create a reference architecture for blockchain based transaction systems, the accommodating research questions are given in section 1.2. The research model and research method are given in section 1.3 and 1.4. This research has both a scientific and practical relevance to give more insight in blockchain and provide a reference architecture for blockchain based systems. This can be found in section 1.5. Section 1.6 gives an overview of the rest of this thesis.

1.1 Motivation

For small and medium sized enterprises (SMEs) it can be hard to gain access to cash. They suffer in accessing financial markets, they are considered high-risk borrowers who need to pay higher interest and they have a problem of insufficient liquidity and working capital (Martínez-Sola, García-Teruel, & Martínez-Solano, 2017; Song, Yu, & Lu, 2018; XinXiang, 2012). Since the recent economic downturn, these problems have increased (E. Hofmann, 2011). This is mainly because firms have extended trade credit from their suppliers in order to increase their own liquidity. The SMEs that supply, suffer from this, but are too small to stand up against their big buyers (Coulibaly, Sapriza, & Zlate, 2013; Garcia-Appendini & Montoriol-Garriga, 2013; Ivashina & Scharfstein, 2010).

The environment wherein these trading firms exist, is called a business ecosystem (BE). A BE is a network of organisations: suppliers, distributors, customers, competitors, government agencies. They are all involved in the delivery of a specific product or service through both competition and cooperation (Moore, 1993). BE incorporates all involved parties, which makes it hard to establish a clear view of well-being of the BE (Kim, Lee, & Han, 2008). Within a BE, there can exist multiple supply chains. Parallel to those supply chains, there is a financial supply chains (FSC).

The FSC is the financial flow of a supply chain, next to the physical supply chain(Popa, 2013). It connects the trading partners from order placement to receipt of payment. It carries the flow of the finance in the opposite direction of the flow of goods and services. Managing this is a very important part of supply chain management, which is often underestimated (Wuttke, Blome, Foerstl, & Henke, 2013). Financial supply chain management (FSCM) focusses on tools and processes designed to enhance an organisation's product flow, maximizing profitability and minimizing expenses. It consists of the activities of planning and controlling all financial processes (Popa, 2013; Wuttke, Blome, Foerstl, et al., 2013; Wuttke, Blome, & Henke, 2013).

Supply Chain Finance is a relatively new research field (Huff & Rogers, 2015). Hofmann described SCF as an approach for firms in a supply chain, to jointly create value by controlling, planning and steering the flow of the financial resources (Erik Hofmann, 2005). SCF has grown in popularity since the recent credit crunch, due to the extending payment terms and the decrease of the liquidity of supplying firms (E. Hofmann, 2011). Wuttke et al. described SCF as a solution that enables buying companies to use Reverse Factoring with their entire supplier base (Wuttke, Blome, Foerstl, et al., 2013). It is a way to improve the liquidity for supplying companies (Demica, 2007; Shang, Song, & Zipkin, 2009).

Based on the definitions of these previous studies, we will describe SCF as: *"A set of solutions, such as reverse factoring, to improve the flow of financial resources between interacting organisations across a financial supply chain."* From this definition it can concluded that SCF can help supplying SMEs to increase their liquidity. But the buying companies in the supply chain will have to participate in the solutions. How can these firms be convinced to help the SMEs?

A Dutch company is creating a platform that aims to solve this problem. Buyers can offer early payment of invoices to their suppliers, with dynamic discounts. This improves the liquidity of suppliers, by giving them access to more cash. Buyers who have excess cash hardly receive any interest on that cash. By offering early payments to their suppliers, buyers receive a higher return on their cash, while the liquidity position of the suppliers improves. This is the case that will be used throughout this research. The company wants to remain anonymous, so from now on we will refer to this company and its platform as "the Broker".

The marketing strategy of the Broker is to get big buyers to use the platform, with the promise of a higher return on their cash by providing early payments, with the dynamic discounts, to their suppliers. When a buyer is onboard, suppliers are invited to also join the platform, to connect with their buyers and get access to more cash.

Figure 1 shows the current process that the Broker's platform provides. A buyer places an order and the supplier will deliver the goods or perform a service. In the normal situation, on the due date of the payment the buyer would pay the invoice. But the buyer can also offer to the make an early payment to the supplier, with a certain discount. The discount is calculated by the platform with an algorithm that is based on multiple factors. The supplier can accept the early payment and the buyer will then pay the discounted invoice. The only problem is that the buyers must decide to offer the early payments and start the process.

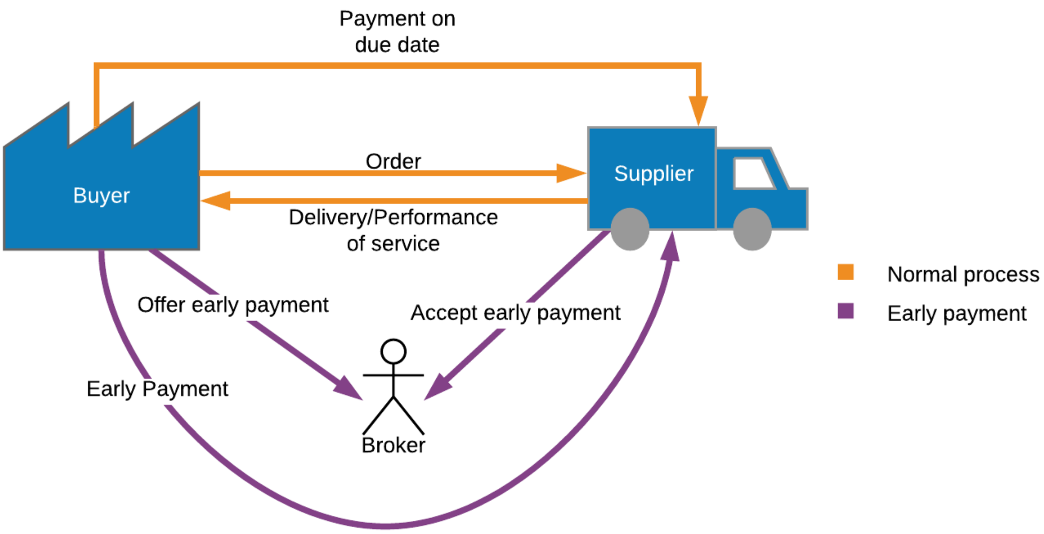

FIGURE 1: CURRENT PROCESS

For now, the Broker focusses on the end of the supply chain, but they want to incorporate the whole supply chain in the system, to improve insight in the financial health of the supply chain. This provides an insight in the financial supply chain, next to the physical supply chain. To make this insight possible, a solution is needed to analyse and present the data and give insight in the financial health of the supply chain. This analysis is done with confidential data of the parties that use the service. That is why the data should be anonymous, so parties cannot extract information from each other. But it is also important that some information is shared.

The analysis of the financial health could also be useful for investors. The Broker wants to create the opportunity for third-party investors to act as factors and invest in invoices. Buyers now post their invoices on the platform, but they do not want to pay earlier. A factor can offer to pay the invoice with the discount to the supplier. When the supplier accepts, he receives the early payment. Then the buyer pays the whole invoice to the factor instead of the supplier. This process is shown in Figure 2.

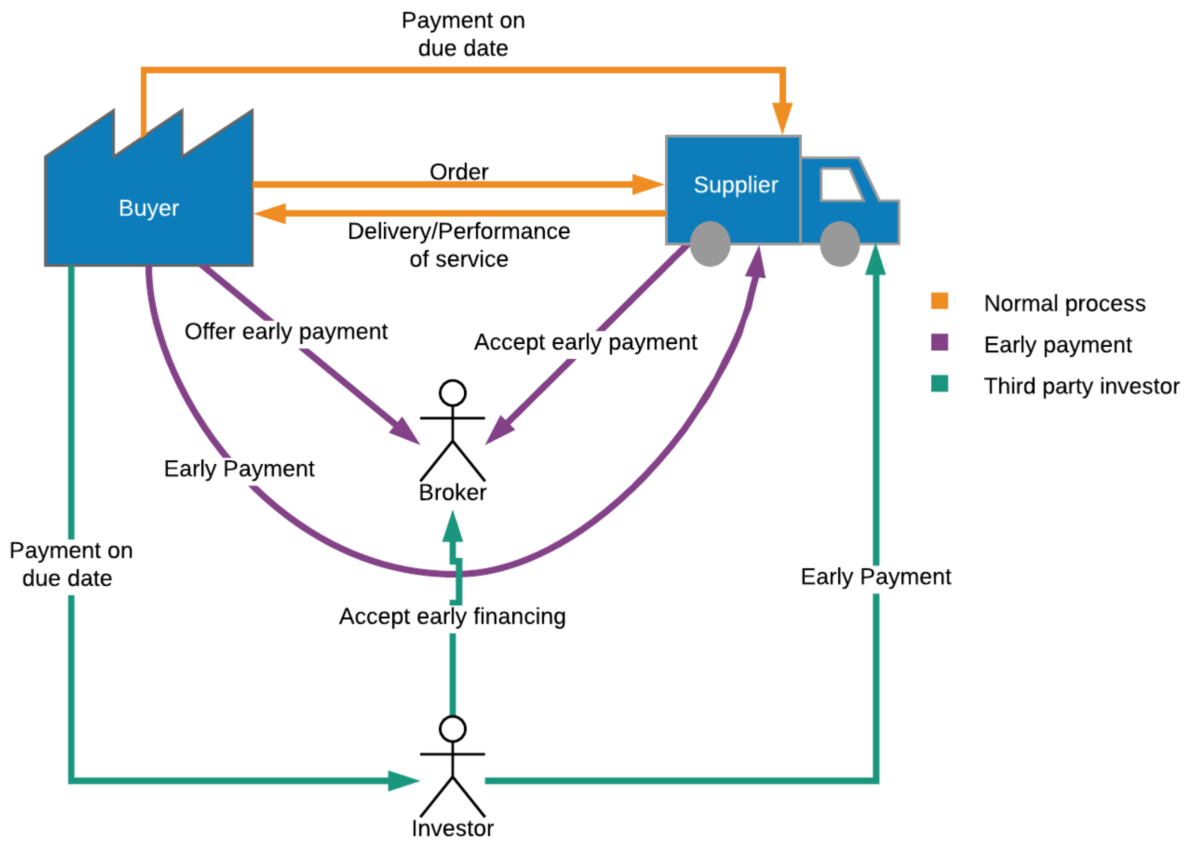

FIGURE 2: PROCESS WITH THIRD PARTY INVESTOR

The Broker's ambition is to manage a fund. Investors store their money in this fund and the Broker will take care of the money trail. This brings some extra responsibilities and a need for a reliable transaction processing system. That is where blockchain could play a big role, by creating a trustworthy audit trail. In a later stage, smart contracts could also play a big role.

The Broker is a partner of CAPE Group B.V. CAPE is a company located in Enschede, Netherlands and specializes in creating integrated IT solutions. CAPE wants to investigate the possibility to use blockchain in TP systems. But the main interest of CAPE is to integrate blockchain based applications with other systems, like ERP systems.

In this thesis, we will explore the possibility of using blockchain in transaction processing systems. The case of the Broker will be used to validate the outcomes of the thesis. CAPE provides resources and guidance in this thesis. The results could give an insight in the possibilities of using blockchain in transaction processing systems and from that a recommendation could be given to the Broker about using blockchain in their system.

1.2 Research Objective

Today, every interaction between enterprises or an individual is recorded, often done by computer systems. These interactions are called business transactions, and they initiate a program. Most of the time, such programs access a shared database and retrieve, create or update an entry. This processing of transactions with computer systems is called transaction processing (TP). TP makes sure that the interactions are recorded correctly and that new processes are started if needed. Combining these transaction processes creates a transaction processing system which has three major functions (Bernstein & Newcomer, 2009).

The first function is obtaining the input for the transaction. This input usually derives from a display, like a webpage were a customer orders a product, or a device, like an IOT sensor. When the input is obtained, a request is constructed. Secondly, the system needs to be able to accept request messages and then call the transaction programs that can process the request. And third, execute the transaction program to complete the work required by the request.

As said, most of the transaction programs execute administrative functions, which involves accessing databases to retrieve, create or update an entry. Database management does play a big part in the TP systems. Transactions must be recorded, and not only that, they must be recorded completely or not at all to ensure the trustworthiness of the records. This is done by using save-points, which are backups of the database on a certain point. When a transaction is aborted, the system can reset to the save-point. There are more processes in place to ensure that the database is correct and consistent, but is there a better or more efficient way? Could blockchain be used as the record keeper?

Blockchain is said to be a relatively new technology, but the first publications of this cryptographic technology were already published in 1991, by Haber & Stornetta (Haber & Stornetta, 1991). Documents were stored in separate blocks and by timestamping and encryptions they were linked and secured. In 1993 they improved the efficiency and reliability of the time-stamping process, by storing multiple files in one block, using a Merkle Tree (Bayer, Haber, & Stornetta, 1993). There was not much written about blockchain after these publications.

The real hype around the blockchain technology was started by Satoshi Nakamoto in 2008 (Nakamoto, 2008), although the term blockchain was not used introduced yet. The words “block” and “chain” were used separately. Nakamoto conceptualized blockchain into the first decentralised cryptocurrency: Bitcoin. It was the first virtual currency that did not have the double spending problem. By creating a peer-to-peer network and blockchain as the public ledger for all transactions, no trusted third party was needed to make transactions (Nakamoto, 2008).

Blockchain was mainly used for cryptocurrencies, but in 2014 the term *blockchain 2.0* emerged. New applications for blockchain were found and made it possible to create more sophisticated blockchain systems with smart contracts. Systems were authorised to pay out dividends or payments for deliveries, based on information in the blockchain. Now, blockchain is trying to become an equal to trusted third parties and has the potential to replace them (Tapscott & Tapscott, 2016).

We hypothesise that payment or trade systems could greatly benefit from a TP system that uses blockchain. Blockchain is an innovative technology, that seems to be able to create a trustworthy audit trail. All transactions will be documented in the blockchain, where a consensus about the transaction is made between all the parties. But there are a lot of unanswered questions about the use of blockchain. Does blockchain bring the same kind of values that traditional systems bring? It is also said that blockchain can ensure trustworthy audit trails, but is this true? And can blockchain create trust between different parties? What kind of architecture should such a system have?

The objective of this thesis is to create a validated reference architecture for blockchain based transaction processing systems. By analysing the case of the Broker, looking at the value exchanges and processes, we want to analyse the influence of the implementation of blockchain.

1.2.1 Research Questions

From the research objective, the following research question is computed:

What reference architecture can best provide a basis for the design of a blockchain based transaction processing system, and integrations to existing applications?

To get the answer to this research question, several sub-questions need to be answered first. The research will start with a literature study. Here the goal is to find reference architectures for both transaction processing systems and blockchain applications. Blockchain is used for cryptocurrencies, but how does it work in other applications? The blockchain for these private applications is smaller, will this affect the security, and does it prevent fraud? Lastly, to test the developed reference architecture, a prototype needs to be built. In the literature study, the options for creating a blockchain application are reviewed. This gives the following sub-questions for the literature review:

SQ1a: What are the common reference architectures for transaction processing systems?

SQ1b: How is blockchain utilised in auditing systems?

SQ1c: What reference architectures are used for blockchain applications?

SQ1d: Which blockchain platforms could be used to create blockchain applications?

To build the reference architecture, an analysis of the current state is needed. The case of The Broker will be used to create this current state. The literature in combination with the analysis should give the answers to the following sub-questions:

SQ2: What are the business processes of the Broker's service and which values does it create and request?

SQ3: What is the current enterprise architecture of the Broker?

With the information of the current state and the knowledge of blockchain implementations, a reference architecture will be made. The information will help to make decisions and support the design of the architecture. This gives the following sub-question:

SQ4: What architecture can be used for the design of a blockchain based transaction processing system?

The architecture that is designed should be validated. Models that are made, will be validated with the help of the Broker and CAPE. A proof of concept of the blockchain based transaction system will be made, to validate the architecture. We will also look at the value a blockchain based system will bring to the Broker, to see if they should switch to such a system.

SQ5: What are the advantages of a blockchain based transaction system, over a traditional transaction system?

SQ6: What are the advantages of the integration possibilities in this reference architecture, over the existing blockchain systems?

SQ7: How could the implementation of blockchain influence the business of the Broker?

With the answers to the sub-questions, the main research question can be answered. The process of this research is given in section 1.3 (Figure 3).

1.3 Research Model

To reach the research goal, several steps need to be taken. A Research Model (Figure 3) is created, to visually represent the process and steps of this research.

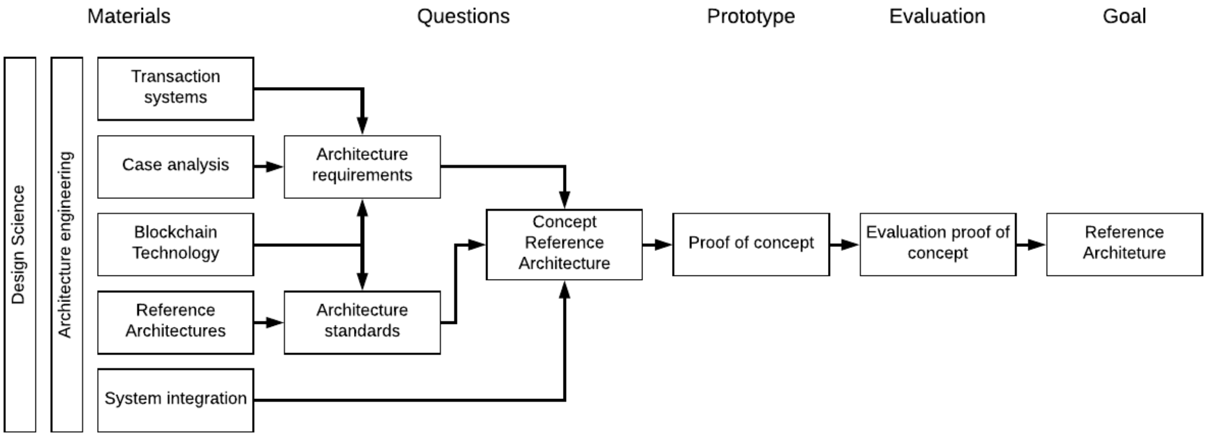

FIGURE 3: RESEARCH MODEL

1.4 Methodology

For this research, the Design Science Research Methodology (DSRM) of Peffers et al. (2008) is used. This research method was designed for design science (DS) in information systems. After the conclusion that there was a lack of a commonly accepted framework for DS, Peffers et al. (2008) created the DSRM. This research method was chosen, because it is a linear model which focusses on identifying problems and solving them with an artifact that helps solving or improving the current state. Figure 4 shows a visual representation of the DSRM.

DSRM consist of 6 activities, which will also be carried out in this research:

1. *Problem identification and motivation.* This was done in sections 1.1 and 1.2. The motivation and problem were identified, which is the possible improvement of transaction processing systems with blockchain, but there is no clear reference architecture and knowledge of the influence of blockchain.
2. *Define the objectives for a solution.* The objectives of this research are defined in the form of research questions in section 1.2.1. Additionally, in chapter 4, the objectives and requirements for the reference architecture are given.
3. *Design and development.* In chapter 4, the reference artifact will be designed. In this research this will be in the form of a reference architecture. This is done based on the current and desired situation.
4. *Demonstration.* To validate the reference architecture, the architecture will be used in the case of the Broker. A proof of concept will be made and a prototype to help validate the architecture. This will be documented in chapter 5.
5. *Evaluation.* After the demonstration an evaluation will be conducted. In chapter 6 the architecture will be evaluated.
6. *Communication.* This thesis will be the communication. Chapter 7 gives the final remarks about all the research questions and the main research question. Also, some recommendations are given.

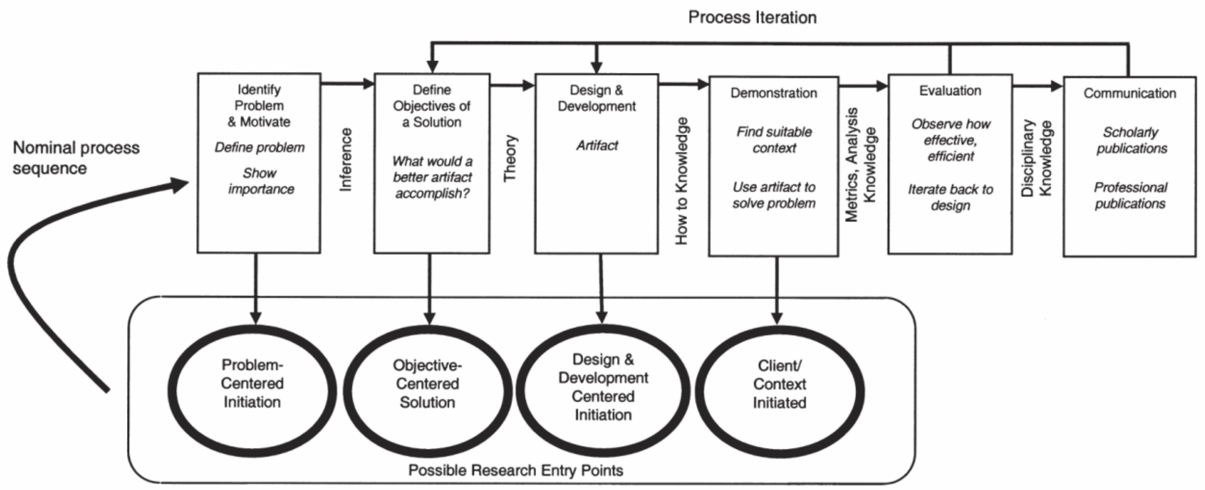

FIGURE 4: DSRM

1.5 Scientific and Practical relevance

This research will have a scientific and practical relevance.

1. **Scientific relevance**
   Blockchain is a big hype, but it is a very interesting technology. It is interesting to see in which research fields and innovative ways blockchain can be used. This research will give new insights in the possibilities, but also the limitations of blockchain. It will also provide a new possible reference architecture for blockchain systems.

2. **Practical relevance**
   This research tries to set a standard for blockchain based transaction systems, by creating a new reference architecture. It also takes integration possibilities into account, which could be useful for big companies with lots of integrations. Lastly, this research looks at the value a blockchain based system brings. This could be useful for companies to decide whether they should use blockchain or not.

1.6 Thesis Structure

TABLE 1: DSRM PHASE AND RESEARCH QUESTION PER CHAPTER

Chapter	DSRM phase	Research Questions
1	1, 2	-
2	1	1a, 1b, 1c, 1d
3	1	2, 3
4	2, 3	4
5	4	-
6	5	5, 6, 7
7	6	All

2 Literature Review

To provide a scientific background for this thesis, a literature review has been executed. The literature review will provide the answers to the first four sub questions, also stated in section 2.1. The guidelines for a literature review of Kitchenham and Charters (Kitchenham & Charters, 2007) were followed, together with an exploration of blockchain platforms. The exact methodology is described in section 2.2. The goal was to find reference architectures for both transaction processing systems and blockchain based systems. A reference architecture for transaction processing systems was found and that architecture seems to be widely accepted, see section 2.3. For blockchain there is more discussion and there is no clear reference architecture yet. Blockchain is also promised to be a technology that can create trustworthy audit trails, but there is not an agreed conclusion yet in the literature, see section 2.4. Therefore, in section 2.5, we can conclude that this thesis could be useful in creating a standard reference architecture for blockchain based transaction processing systems.

2.1 Introduction

This literature study is used to provide the answers to the first four sub questions, stated in section 1.2.1. These sub questions are:

1. *What are the common reference architectures for transaction processing systems?*
2. *How is blockchain utilised in auditing systems?*
3. *What reference architectures are used for blockchain applications?*
4. *Which blockchain platforms could be used to create blockchain applications?*

Question one focusses on reference architectures of TP systems. How are they implemented and which parts are crucial? Question two, three and four will focus on blockchain. Because auditing is important in the kind of TP system we want to explore, the question is asked if it is right to assume that blockchain will be useful for that. And reference architectures of blockchain applications, outside of cryptocurrencies, are explored to see how blockchain is used in different applications. These reference architectures could in future research be used to create a reference architecture for a blockchain based TP system. Also some blockchain developing platforms are reviewed, to give a few options on what platform to use when building an actual blockchain system.

2.2 Methodology

The methodology of this literature review is inspired by the guidelines of Kitchenham and Charters (Kitchenham & Charters, 2007). But because of the lack of literature, a more explorative approach is used, that varies on the systematic literature review. Also, because blockchain is a relatively new research topic, especially blockchain applications outside of the cryptocurrencies, an exploration outside the scientific literature is used to get extra information.

2.2.1 Literature Criteria

Per subject, several key-words will be used to find related studies. However, not every study can or will be used, because a lot of it is irrelevant or of low quality. Studies that will be used for the literature review, will have to meet the criteria in Figure 5. These criteria are used to select the first selection of papers. The studies that are selected, are potential studies to be used in the review.

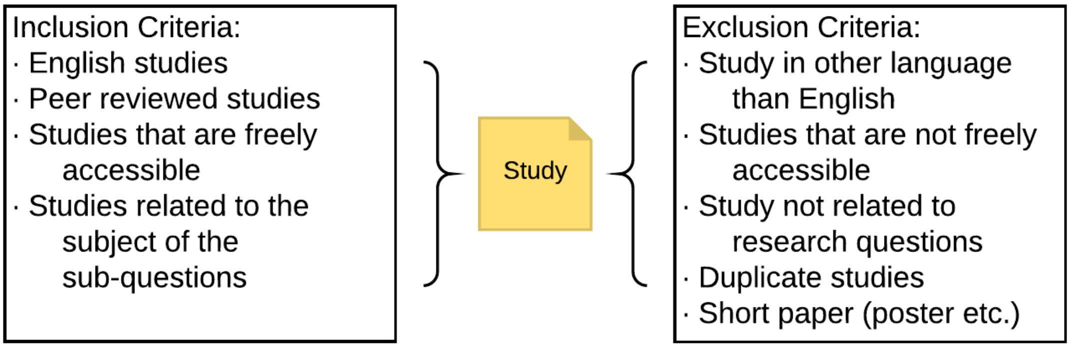

FIGURE 5: LITERATURE CRITERIA

2.2.2 Study Selection Process

The process of the selection of the primary studies is represented in Figure 6. After the database search and the exclusion of papers, based on the literature criteria from Figure 5, the remaining studies will be reviewed based on the abstract and title. This will further reduce the number of potential studies. These potential studies are then reviewed on their full text and then the final selection is made. This gives the set of primary studies, usable per subject.

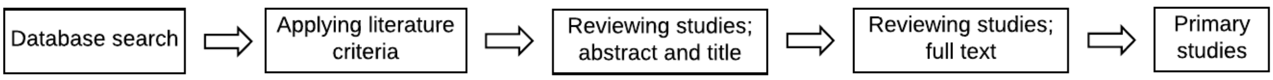

FIGURE 6: STUDY SELECTION PROCESS

2.2.3 Results

Scopus is the literature database that is used to find the information needed. Google Scholar was the second option, because Scopus is not always complete. But after the searches on Scopus, Google Scholar did not add any new studies to the primary studies. In Table 2 the search queries that are executed in Scopus can be found, together with the number of results.

TABLE 2: SEARCH QUERIES

Subject	Query	Results
Transaction Processing systems	TITLE-ABS-KEY (("transaction processing" OR "transaction server") AND reference AND architecture)	19
Blockchain in auditing systems	TITLE-ABS-KEY (blockchain AND audit*)	73
Blockchain reference architectures	TITLE-ABS-KEY (blockchain AND architecture)	180
		272

Combined a total number of 272 potential studies were selected as potential primary studies. The selection process of Figure 6 reduced this to the final set of primary papers. The execution of this process is shown in Figure 7. The set of papers is shown in

Table 3. The papers are discussed in sections 2.3 and 2.4.

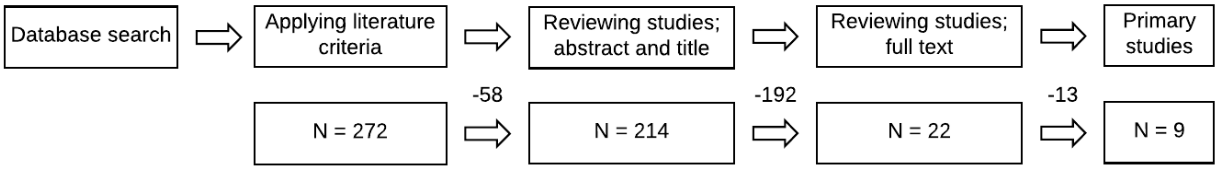

FIGURE 7: PERFORMED STUDY FILTERING

TABLE 3: SELECTED STUDIES

Study	Subject	Paper
S1	Transaction Processing systems	P. Bernstein and E. Newcomer, *Principles of Transaction Processing*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2009.
S2	Blockchain in auditing systems	V. L. Lemieux, “Trusting records: is Blockchain technology the answer?” *Rec. Manag. J.*, vol. 26, no. 2, pp. 110–139, 2016.
S3	Blockchain in auditing systems	S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *Www.Bitcoin.Org*, p. 9, 2008.
S4	Blockchain in auditing systems	G. W. Peters and E. Panayi, “Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money,” *New Econ. Wind.*, pp. 239–278, 2016.
S5	Blockchain in auditing systems	G. Zyskind, O. Nathan, and A. S. Pentland, “Decentralizing privacy: Using blockchain to protect personal data,” *Proc. - 2015 IEEE Secur. Priv. Work. SPW 2015*, pp. 180–184, 2015.
S6	Blockchain reference architectures	K. Leng, Y. Bi, L. Jing, H. C. Fu, and I. Van Nieuwenhuyse, “Research on agricultural supply chain system with double chain architecture based on blockchain technology,” *Futur. Gener. Comput. Syst.*, vol. 86, pp. 641–649, 2018.
S7	Blockchain reference architectures	D. E. O’Leary, “Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems,” *Intell. Syst. Accounting, Financ. Manag.*, vol. 24, no. 4, pp. 138–147, 2017.
S8	Blockchain reference architectures	D. Roman and G. Stefano, “Towards a reference architecture for trusted data marketplaces: The credit scoring perspective,” *Proc. - 2016 2nd Int. Conf. Open Big Data, OBD 2016*, pp. 95–101, 2016.
S9	Blockchain reference architectures	X. Xu *et al.*, “A Taxonomy of Blockchain-Based Systems for Architecture Design,” *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, pp. 243–252, 2017.

2.2.4 Exploration

To find possible blockchain development platforms a few Google searches were executed and from those searches, the gathered information was used for new searches. Eventually we came up with a long list of possible blockchain development platforms, which can be found in Table 4. These platforms were reviewed based on some criteria: popularity and activity of the platform; type of blockchain network; pricing of transactions; consensus algorithm; smart contract functionality. The results of this review can be found in Table 5. From the results, a short list was made including three platforms to be further discussed in section 2.4.4: Ethereum, Quorum, Hyperledger Fabric.

TABLE 4: LONGLIST BLOCKCHAIN PLATFORMS

Platform	Name	Source
P1	Ethereum	"Ethereum Project." [Online]. Available: https://www.ethereum.org/. [Accessed: 19-Jul-2018].
P2	Hyperledger Fabric	"Hyperledger Fabric." [Online]. Available: https://www.hyperledger.org/projects/fabric. [Accessed: 19-Jul-2018].
P3	Hyperledger Sawtooth	"Hyperledger Sawtooth." [Online]. Available: https://www.hyperledger.org/projects/sawtooth. [Accessed: 19-Jul-2018].
P4	MultiChain	"MultiChain - Open source blockchain platform." [Online]. Available: https://www.multichain.com/. [Accessed: 19-Jul-2018].
P5	Openchain	"Openchain - Blockchain technology for the enterprise." [Online]. Available: https://www.openchain.org/. [Accessed: 19-Jul-2018].
P6	Quorom	"Quorum - J.P. Morgan." [Online]. Available: https://www.jpmorgan.com/global/Quorum. [Accessed: 19-Jul-2018].
P7	R3 Corda	"Corda." [Online]. Available: https://www.corda.net/. [Accessed: 19-Jul-2018].
P8	Ripple	"Ripple - One Frictionless Experience To Send Money Globally | Ripple." [Online]. Available: https://www.ripple.com/. [Accessed: 19-Jul-2018].

TABLE 5: BLOCKCHAIN PLATFORM REVIEWS

Criteria	P1	P2	P3	P4	P5	P6	P7	P8
Popularity	High	High	Medium	Medium	Medium	Medium	Medium	Medium
Activity	High	High	Medium	High	Medium	Low	High	Medium
Type of network	Public	Modular	Modular	Private, Permisioned	Private	Permissioned	Permissioned	Permisioned
Crypto-currency	Ether	None	None	None	None	None	None	Ripple
Concensus algorithm	PoW (Casper)	Modular	Modular	Altered PoW	Majority Voting	Majority Voting	Modular	Probabliistic voting
Smart contracts	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Pricing	Free	Free	Free	Free	Free	Free	Free	Paid

2.3 Transaction processing systems

Bernstein and Newcomer wrote the book *"Principles of Transaction Processing"* (Bernstein & Newcomer, 2009). In which, they describe how TP systems work and operate. In chapter 3 they discuss the accepted architecture of the TP systems.

A TP system can be divided into three types of components: functional components, hardware subsystems and operating system processes. The functional components can be decomposed into: front-end programs, request controllers, transaction servers and database systems. In the past, the architecture of these systems was called a three-tier architecture. The front-end program was considered the first tier, the database system with the database the third tier, and everything in between the second tier. Over the years the system became more layered, so it became a multi-tier architecture. A general architecture is shown in

Figure 8. This architecture can be aligned with both service-oriented, by mapping the transaction servers to services, and object-oriented designs, by mapping the transactions servers to business objects.

In this architecture the front-end program receives the input. For example, a form is filled in in a web browser. The front-end program then constructs a request, based on the input from the form. It could respond to some requests itself, but most requests are sent to the next stage of the system. This is done by either storing it in a queue or sending it directly to the request controller. The request controller's responsibility is routing the request to the correct transaction server. This is done by determining the required steps to execute the request and then invoking the correct transaction server. The request controller ensures that no request is lost and aborts the session if an error occurs. The transaction servers run application programs that process the incoming request. They execute program logic to complete the request. The transaction servers usually communicate with one or more database systems, to retrieve, update or insert data.

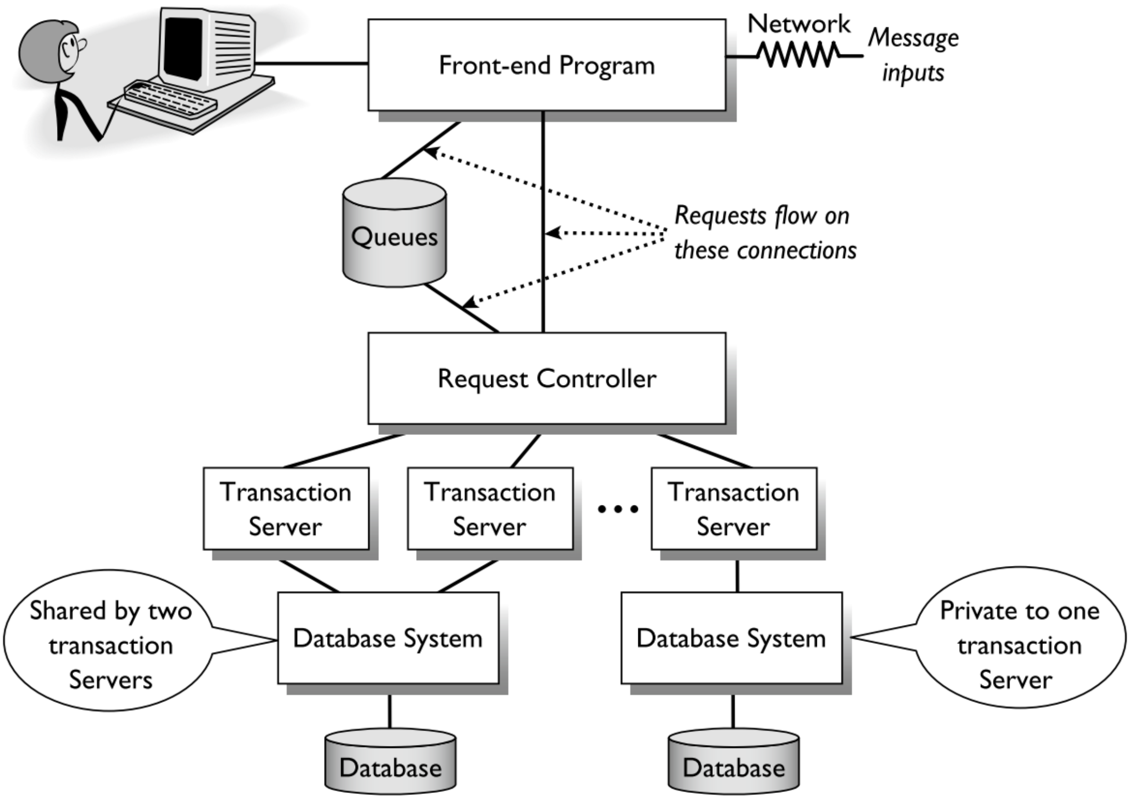

FIGURE 8: TP SYSTEM ARCHITECTURE
SOURCE (Bernstein & Newcomer, 2009)

Usually, these TP systems have two kinds of hardware subsystems, the front-end systems who are close to the input devices, and the back-end systems who are close to the databases. These subsystems can have different configurations, depending on the complexity of the TP system. In complex situations a front-end system will have multiple machines supporting many input devices. The back-end system will then have multiple servers that run different machines, running different applications.

When mapping the functional components of

Figure 8 in processes on the hardware systems, the modern multitier TP systems create a process for each of the functional components. In small systems all the separate processes of the components run on the back-end system. For larger TP systems, the front-end programs run on the front-end system and the other processes run on the back-end system. Both the front and back-end systems could run multiple machines, where the processes run on these separate machines. Therefore, the processes communicate with each other through messaging.

This setup benefits the flexibility of the distribution and configuration of the system. The functions are moved around in the distributed system without modifying the application programs, because the different functions are already separated. It is also easier to scale-up, control and monitor the systems. The main disadvantages of this setup are the performance and the complexity. The components communicate through messaging, instead of local procedure calls, which has a major impact on the performance. The large number of processes and components in the multi-tier architecture greatly increases the complexity.

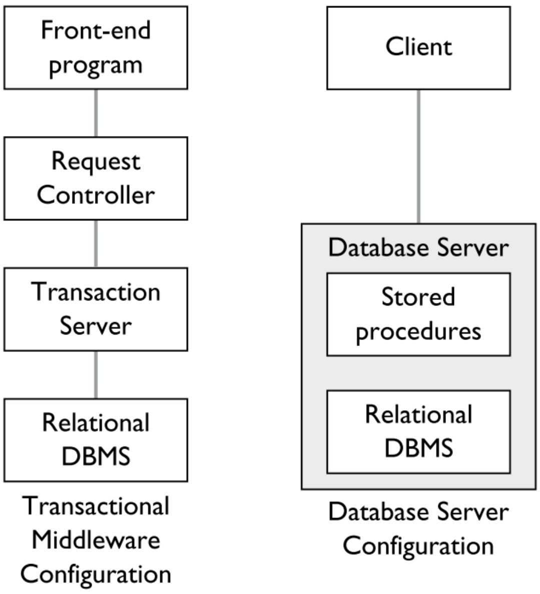

FIGURE 9: TRANSACTIONAL MIDDLEWARE VS DATABASE SERVER
SOURCE: (Bernstein & Newcomer, 2009)

A "new" approach for the TP system architectures is the two-tier architecture. Figure 9 shows the processes next to each other. Instead of using a request controller and a transaction server, stored procedures are used to direct and execute the requests. The two-tier architecture was a popular approach in the 1980s, but because of the lack of scalability and functionality of database servers, the three- and multi-tier architectures became more popular. But due to the increase of functionality to support partitioning and scalable data sharing, the two-tier architecture is considered again.

2.4 Blockchain

In 2008, Bitcoin was created by Nakamoto. In his paper (Nakamoto, 2008), he proposed a peer-to-peer shared ledger, to exchange value between parties without the need of trusted third-party. By timestamping and linking blocks, who are validated by a proof-of-work system, he proposed an auditing valid ledger. The technique was used to create the first cryptocurrency. In later stages, people saw the opportunity to use the blockchain technique in different applications. This evolution let to the term blockchain 2.0. Some of these applications used blockchain as a ledger with a valid audit trail, so there was no need for a trusted party. Peters and Panayi talk about the emerging of blockchain and its potential in transaction processing, by replacing trusted third parties in (Peters & Panayi, 2016).

2.4.1 Basics of blockchain

In blockchain transactions are recorded in a chain of blocks. Figure 10 shows a visual representation of transactions stored in the blockchain. In Figure 11 a visual representation of a blockchain is given. The blocks are chained together by referring to the previous block. When a new block is created, the previous block is hashed. The entire object is put into a hash function, for example SHA-256. This gives a unique outcome. This hash is then put into the new block. When something is changed in a block and the same hash function is performed, it will have a different outcome then it had before. The next block will not reference back to the previous block. Because of this method, blockchain can be used to safely store data, with a low change of tempering. But what if something in a block changes, and then rewrite all the following blocks. The chain would be valid again, and there is no trace of something is changed. This is where consensus algorithms come into play. At this moment, proof-of-work (PoW) and proof-of-stake (PoS) are used the most, but there are a lot more. Especially in the blockchain 2.0 applications, each platform gives its own twist to their consensus algorithm. For now, we will elaborate only on PoW and PoS to give a general understanding of blockchain.

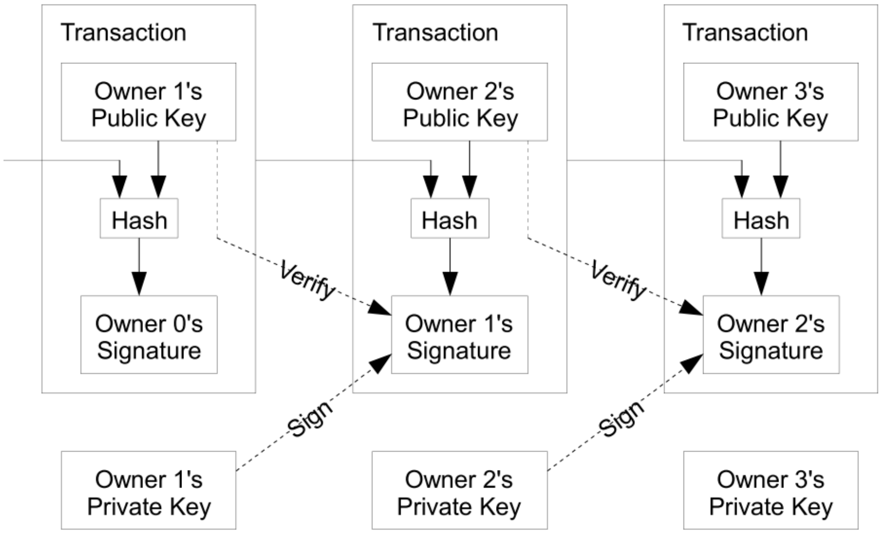

FIGURE 10: BLOCKCHAIN TRANSACTION VISUALISATION
SOURCE: (Nakamoto, 2008)

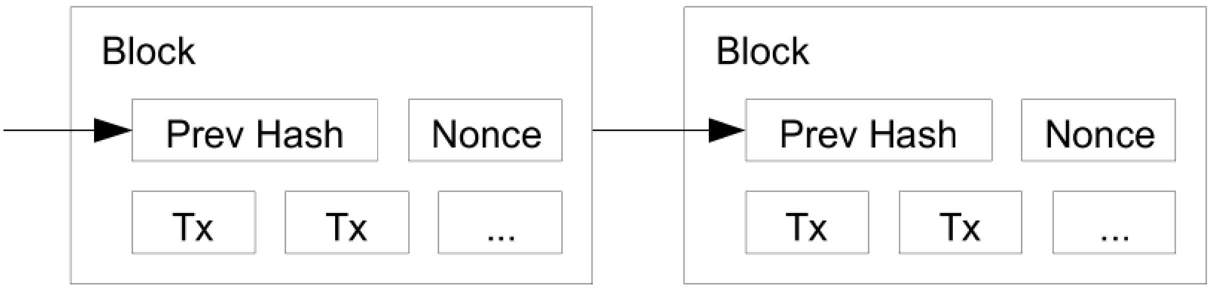

FIGURE 11: BLOCKCHAIN VISUALISATION
SOURCE: (Nakamoto, 2008)

2.4.1.1 Proof of work

When the block is hashed, the outcome is a string of numbers and letters. Every time the same block is hashed, the function will have the same string as the output. Therefore, only certain hashes are accepted as a good hash. At this moment, Bitcoin only accepts a hash that starts with at least 18 zeros. To create such a hash, blocks must be mined. When a new block is ready to be published, the block will be hashed, and the hash outcome will be verified. If it does not have 18 zeros, it is not accepted. Then the nonce comes into play. The nonce is a value in the block that does not contain any data or value. Therefore, the nonce can be changed and the block is hashed again. The outcome will be different and will be verified again. This process is repeated, with all mining nodes in the network, until an acceptable hash is found. This process takes up a lot of computer power, because all the nodes are trying using their power to calculate an acceptable hash. Bitcoin aims to publish one block every ten minutes. When it is going too fast, it increases the number of starting zeros.

Now, when a block is changed, a new acceptable hash needs to be produced. And when this is done, all the following blocks must be recalculated. But if the cheater does not have more than 50% of the CPU power over the entire network, he will not be able to recalculate all the blocks, because new blocks are being added. This PoW algorithm gives blockchain the potential to be the trusted third-party replacer.

This is the main advantage, but there are a few issues with PoW. The biggest problem is, that it is an extremely inefficient process. It requires a sheer amount of power and energy to create blocks, especially because it is a race between the miners. All the miners are using their recourses to find the correct hash, but only one miner will create the block. It is rewarding to use more recourses to mine, so the chance of gaining rewards increases. People and organisations who can invest more resources, have a better chance of mining the blocks than others. As a result, the blockchain is not as decentralized as it wants to be.

2.4.1.2 Proof of stake

PoS makes the entire mining process virtual and replaces miners with validators. The validators will have to lock up some of their coins as stake. They will then start validating the blocks. Meaning that when they discover a block which they think can be added to the chain, they will validate it by placing a bet on it. If the block gets appended, the validators will get a reward proportionate to their bets. This makes a PoS algorithm a lot more resource friendly than PoW, because it is not a race to mine the blocks.

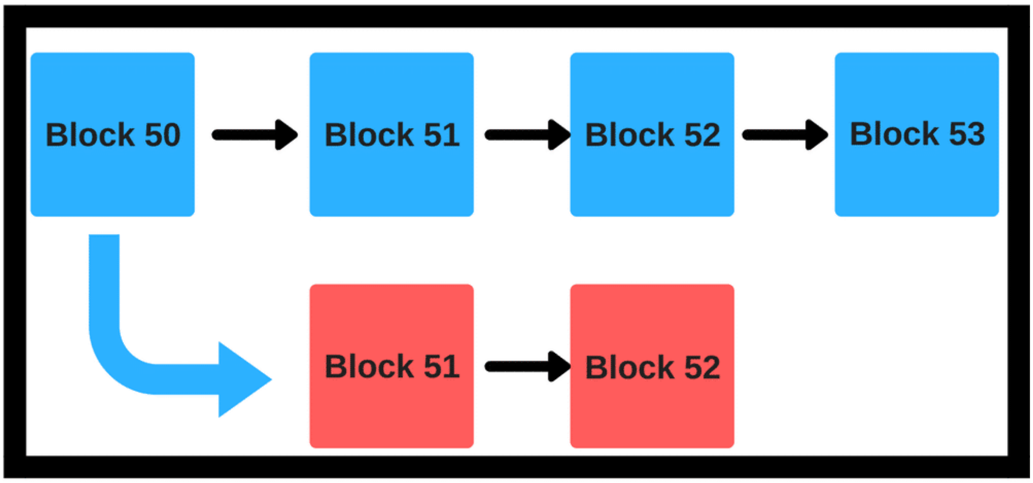

FIGURE 12: BLOCKCHAIN FORK

But the PoS algorithms has a different problem. Figure 12 shows a blockchain. The blue chain is the main line and the red is a fork of the main chain. The red blocks are different and may give false information. The problem is that the other nodes need to keep appending there blocks to the main line. In PoW, someone can try to create a fork, but it needs more people to participate in the new line. Most people will not do that, because it makes no sense for a miner to waste resources on a block that will be rejected by the network anyway. Hence, chain splits are avoided in a PoW system, because of the amount of money that the attacker will have to waste.

However, in PoS it is different. A validator, can put his money in both the red and blue chain, without the fear of repercussion. He will always win and has therefore nothing to lose. This problem is called the “nothing at stake” problem. But due to the big advantages of PoS over PoW, mainly the amount of recourses needed, blockchain applications and platforms are trying to find a way around this problem, but there is not a general solution yet. At the moment of writing, Ethereum is testing with a PoS protocol called Casper, but it is not yet fully implemented.

2.4.2 Blockchain in auditing systems

Blockchain has the potential to be a trusted third party and create valid audit trails, but does the technology live up to those expectations? In (Lemieux, 2016), Lemieux asked that same question. Is Blockchain technology truly suited for the long-term management and preservation of trusted digital records? By using international recordkeeping and digital preservation standards as reference tools, an assessment of the limitations, risks and opportunities of blockchain as a valid audit trail, has been done. Lemieux investigated the use of blockchain in a case of land registration in Honduran, were the Factom solution was used (Paul, Brian, Jack, David, & Peter, 2014), and the Bitcoin blockchain. Lemieux points out several threats/vulnerabilities for the use of blockchain as the trusted record keeper.

The main concerns are: control of the blockchain, the reliability of the records, and the authenticity of the blockchain (Lemieux, 2016). Who controls the blockchain and does the blockchain remain a distributed system? When looking at the actual distribution of the Bitcoin blockchain, it is not as distributed as promoted. Are the recorded records in the blockchain valid and reliable? Although blockchain gives a good insurance that records cannot easily be changed, the initial records need to be valid. Record integrity is the core of the blockchain technology, but the ability to maintain the authenticity of the records is highly dependable upon the system and its security.

These concerns are based on the implementation of blockchain in the Factom solution but could be generalised to similar solutions. The paper concludes that these concerns are mainly an advice for future implementations of the blockchain technology and should be investigated. The conclusion of the paper is: “*Overall, however, the message is one of caution about the role of Blockchain technology as a comprehensive public recordkeeping and digital preservation solution, even while acknowledging its apparent advantages as a low-cost transaction validation mechanism.*” (Lemieux, 2016, p. 134)

Zyskind, Nathan and Pentland have a more positive view on the blockchain technology as a trusted record keeper. In a world where Facebook has 300 petabytes of personal data, they have created a solution to use blockchain to protect personal data (Zyskind, Nathan, & Pentland, 2015). They elected blockchain, because they state that blockchain 2.0 projects have demonstrated how blockchain can serve other functions than cryptocurrencies, requiring trusted computing and auditability. The problem with those projects were the data was shared publicly. They created a new platform that enables users to control their data, without compromising security or limiting companies' and authorities' ability to provide personalized services. This was done by combining a blockchain, re-purposed as an access-control moderator, with an off-blockchain storage solution. The writers claim that the ledger can act as legal evidence for accessing data, since it is computationally tamper-proof.

2.4.3 Blockchain reference architectures

Recently, researchers and organisations are trying to use the blockchain technology in numerous applications. "Blockchain" was almost at the peak of Gartner's hype curve in 2016 and 2017 (O'Leary, 2017). Therefore, O'Leary tried to summarize the newest developments of blockchain in Accounting and Supply Chain Systems. In section four, he shows four possible blockchain architectures (Figure 13) for different needs for the blockchain based system, in different organisations. Option 1 is a public blockchain, like Bitcoin, where everyone can participate. Option 2 is a private blockchain, which is used internally within an organisation. A pairwise corporate use of blockchains is visualized in option 3, where a private blockchain is shared between two companies. This could be useful for firms that outsource a substantial amount of production, so a shared ledger is useful. And last option 4 illustrates multiple consortium companies using the same private blockchain. For example, this can be used for transactions between the companies. These are only the architectures of the blockchain, not the actual blockchain based systems.

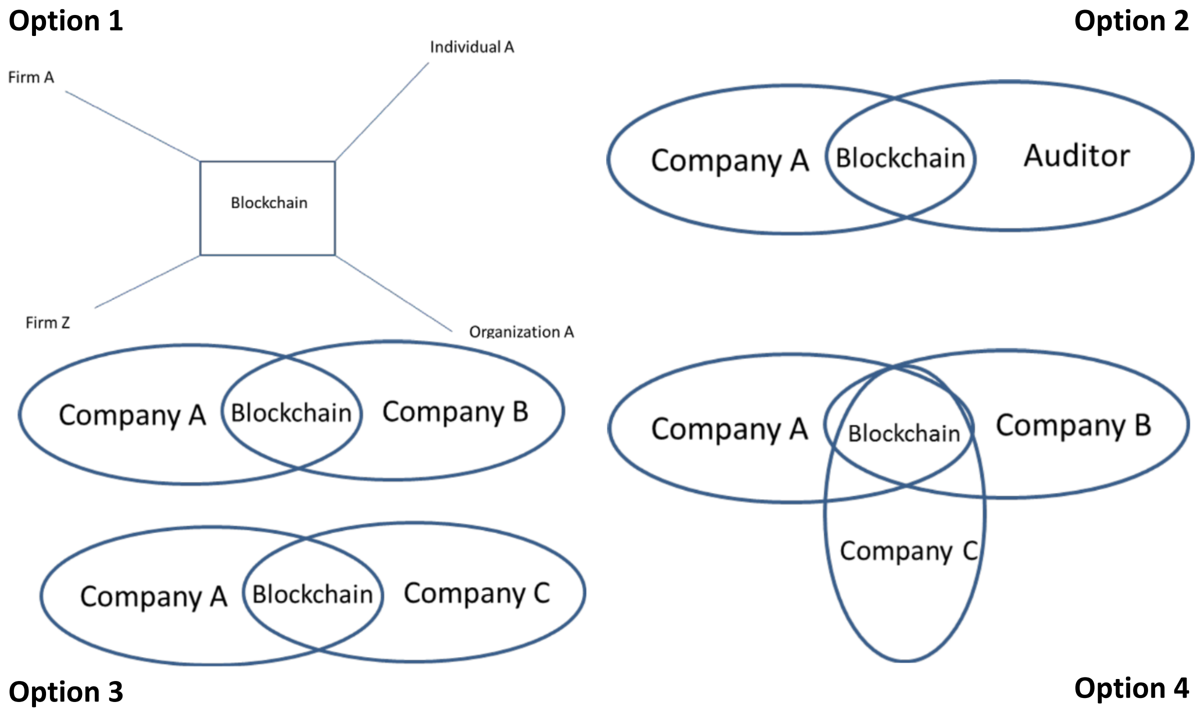

FIGURE 13: FOUR BLOCKCHAIN ARCHITECTURES
SOURCE: (O'Leary, 2017)

Roman and Stefano created a reference architecture for trusted data marketplaces in (Roman & Stefano, 2016). The paper discusses challenges and opportunities related to sharing data. This data is needed for the process of credit scoring. Credit scoring regulates the way companies and individuals lend money from bank for example, and it establishes the level of trust between companies in their transactions. From their analysis, they identified the main data-related challenges for data sharing. To mitigate those issues, they introduce the concept of a “trusted data marketplace”. They outlined an architecture that can be used for future implementations of credit scoring ecosystems. They claim that the architecture could also be used as a reference for data sharing systems, where data privacy is necessary. The proposed architecture can be found as Figure 14. The underlying principle of the design is the ability for different parties to jointly store and process data, while keeping the data completely private. They propose to use blockchain to improve the trust between the parties, but also the possible implementation of smart contracts. Further they use homomorphic encryption, that enables the ability to perform computations of data without decrypting it first. Although the design of the architecture is there, they have not tested it, but it could be used as a reference. (Roman & Stefano, 2016)

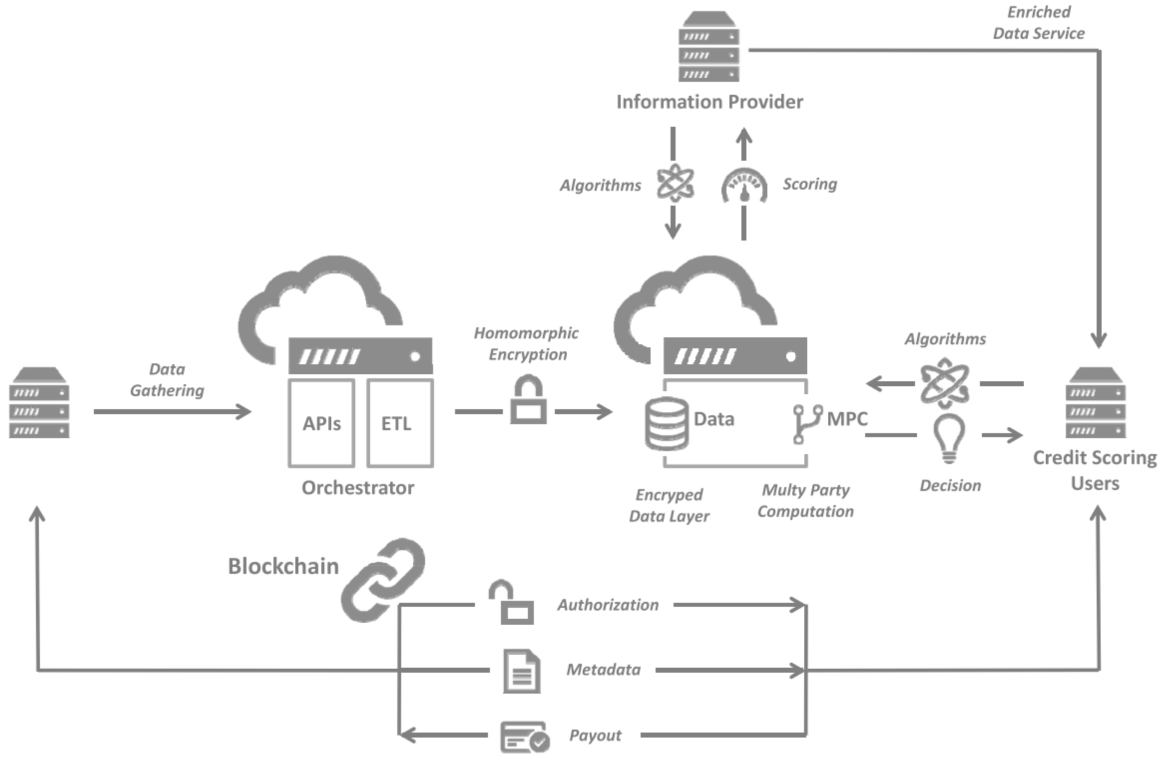

FIGURE 14: REFERENCE ARCHITECTURE FOR TRUSTED DATA MARKETPLACE
SOURCE: (Roman & Stefano, 2016)

Kaijun et al. created a double chain architecture based on blockchain technology in (Leng, Bi, Jing, Fu, & Van Nieuwenhuyse, 2018). Different than most blockchain based system is that they created two separate chains. One for the user information and the other for the transaction data. This new structure is shown in Figure 15. They believe this results in three major benefits. The first being that any node in the system can view the transaction data, without knowing the private information of the enterprises. Secondly, by dividing the data, the amount of information that is recorded by the nodes will be reduced, which is beneficial for the speed of the system. And last, it is easier to expand the system. They used a proof of stake algorithm as the consensus algorithm. This requires less computing time and power to ensure the normal operation of the blockchain. The research results implicate that a double-chain structure can guarantee the transparency and security of transition information and privacy of enterprise information. The double chain also significantly improves the credibility and overall efficiency of the system.

As shown above, there is a sheer variety of possible architectures for blockchain systems. But it is difficult to find reference architectures for specific implementations of blockchain based systems. Blockchain is still very young, so everything is still in development. To bring some clarity, Xu et al. propose how to classify and compare blockchains and blockchain-based systems to assist with the design and assessment of their impact on software architectures (Xu et al., 2017). The taxonomy captures the major architectural characteristics of blockchains. It is intended to help with the architectural considerations, like availability, security and performance, for a blockchain based system. This taxonomy could be very useful to pick the right architecture.

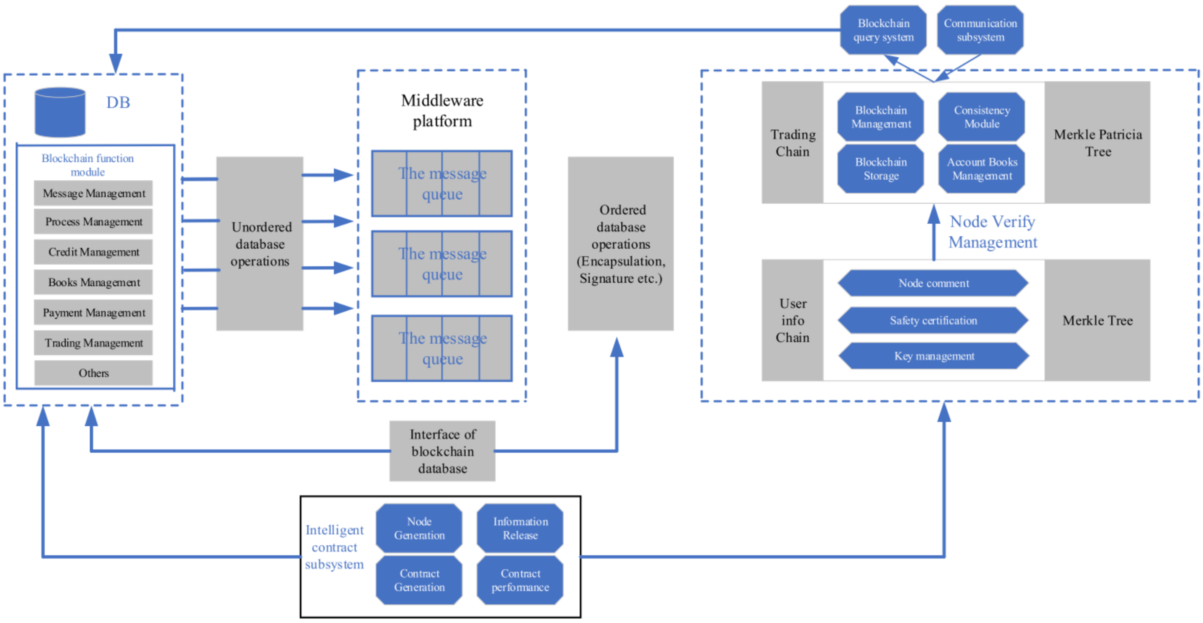

FIGURE 15: DOUBLE CHAIN STRUCTURE BLOCKCHAIN
SOURCE: (Leng et al., 2018)

2.4.4 Blockchain platforms

The architectures discussed in section 2.4.3, can be implemented by hand. But there are also platforms which provide the ability to set up a blockchain system. These platforms provide the ability to create blockchain applications. In this section, three blockchain platforms will be discussed: Ethereum, Quorum and Hyperledger (fabric).

Ethereum is one of the biggest blockchain providers in the world. Created in 2014 by Vitalik Buterin, Ethereum is a decentralized platform, that provides an infrastructure to build a personal blockchain application or cryptocurrency. The applications run on a custom build blockchain by Ethereum, which enables users to build their own applications, save the data in a distributed network and trade value using the Ether currency. With the use of smart contracts, a wide variety of features can be built into the applications. As of this moment, Ethereum is testing a new consensus protocol called Casper. It is a PoS based consensus protocol, instead of the PoW based protocol Ethash, which was used since the beginning of Ethereum. As soon as the scaling issues of Casper have been resolved, Ethereum will continue to use Casper as its new consensus protocol and totally transfer from a PoW to a PoS consensus protocol. (Dienelt, 2016; 'The First Version of Ethereum's Casper Upgrade Has Been Published - CoinDesk', 2018)

Where Ethereum can be used in many applications and as a cryptocurrency, Quorum is an enterprise-focused version of Ethereum. Developed by J. P. Morgan in 2015, Quorum is designed to handle high speed and high-throughput processing of private transactions. It uses a vote-based consensus algorithm instead of PoW, which greatly improves the performance of the applications. This protocol is called QuorumChain. Although the performance of Quorum is good, it does affect the trustworthiness of the chain, because of the voting system. But because of the performance, it is a massive step forward towards implementing blockchain in the financial consortium. (JP Morgan Chase, 2016; 'Quorum - J.P. Morgan', n.d.)

Hyperledger is a bundle of open-source projects, hosted by the Linux Foundation. As of now, it contains 10 projects, both frameworks and tools. As of now, Hyperledger Sawtooth and Fabric are the frameworks that are active and some more frameworks are still in development. Hyperledger Fabric was launched in 2016, with the goal to create enterprise-grade distributed ledger frameworks and codebases. With its modular architecture it allows plug-and-play components around consensus and membership services. Fabric leverages container technology to host smart contracts, who comprise the application logic of the systems. The difference with Ethereum is that Fabric is much more modular, and it has no currency like Ether in Ethereum. ('Hyperledger Fabric', n.d.; 'Hyperledger Fabric — Key Concepts', n.d.)

There are more platforms, as shown in Table 4, that provide blockchain enterprise applications, like R3 Corda and Ripple. But they are not applicable in this case. R3 Corda is not really a blockchain, although it provides implementation of a distributed ledger (Hearn, 2016). Ripple is a blockchain solution for global payment, but it focusses on payments. It is not sufficient to create fully functional applications, but it does use an interesting consensus protocol, which is much faster than Bitcoin (Schwartz, Youngs, & Britto, 2014).

2.5 Conclusion

The goal of this literature review was to gather reference architectures for both TP systems and blockchain applications. Also, the utilisation of blockchain in auditing systems has been discussed and some blockchain platforms were reviewed. Four research questions had been formulated to guide the literature search. For each of these topics a literature search was done. From this search came that recent reference architectures for TP systems are scarce. The most recent paper that was found was from 2012, but it did not contain useful information. The book from Bernstein and Newcomer, did contain a lot of information, but is a bit outdated.

SQ1a: What are the common reference architectures for transaction processing systems?

In the literature we found that TP systems are often not classified as actual TP systems. TP systems are information systems that record the steps that have been taken and execute the according business logic. This means that the architectures of those systems could also be used to find references for new systems. But the architecture of Bernstein and Newcomer,

Figure 8, is commonly used in all sort of system that are recording transactions. So, this mechanism should be integrated into the systems that processes transactions.

SQ1b: How is blockchain utilised in auditing systems?

Where recent literature for TP systems was limited, there is an overflow of literature about the blockchain technology. The big problem here is that a lot the studies are about the Bitcoin blockchain or other cryptocurrencies. The actual research field of blockchain 2.0 is still in development. When looking into the use of blockchain technology in auditing system, there was not a clear answer. There is a stream of studies that take the advantages of blockchain as the truth and base their new applications for the blockchain technology on it. It was refreshing to see a thorough analysis by Lemieux (Lemieux, 2016), who compared standards for auditing of data with the possibilities of blockchain. She identified several threats to blockchain systems, which could bring new problems. But instead of discouraging one to use blockchain for auditing, she encourages to keep researching the blockchain technology for these kinds of systems. The potential is clear, but the potential treats must be considered. In this the research field could make a lot of progress.

SQ1c: What reference architectures are used for blockchain applications?

The same can be said for reference architectures of blockchain applications. Because the blockchain technology is only being used outside of cryptocurrencies for a few years now, there is no clear architecture yet. What can be seen is that blockchain can be a great technology for data sharing between organisations/individuals, but data privacy can still be a problem. In (Leng et al., 2018) they created a new and innovative architecture, were they use two separate blockchains. But it is still hard to choose a good architecture for the setup of a blockchain based system. Xu et al. created a framework that could assist in choosing the right architecture (Xu et al., 2017). But there is still a lot of room for improvement.

SQ1d: Which blockchain platforms could be used to create blockchain applications?

But the tools to build such applications, are developing rapidly. Although a lot of the development platforms and tools are still in development themselves, there is a big pool to choose from. Ethereum is a good choice, but to use it to build private blockchains, to interact and share data between organisations, it is not perfect. Mainly because all the transactions cost Ether and are not free. Quorum is the Ethereum based version for enterprise-ready blockchain applications. But the problem with Quorum is, that the development has been slow as of late. Therefore in our opinion, Hyperledger Fabric is the best choice if you want to implement a new blockchain technology, at this moment. The technology is developing rapidly, so in a few months, the landscape could be totally different.

As said, especially on the blockchain technology, the research fields could use a lot of improvements. An interesting step could be to create a reference architecture that combines TP systems with blockchain technology. Some blockchain systems are already classified as TP systems, but then it is only one system. An improvement could be to create an architecture that creates a TP system which uses blockchain to save transaction data, which can be shared among systems, as a distributed database. The potential is to create a distributed database which can be used by multiple systems, that is not owned by a trusted third party. The potential is there, but because of the young age of blockchain, there is no clear architecture to implement such systems. Therefore, we think that by achieving the research goal of this thesis, we could set a first step towards a commonly accepted reference architecture for blockchain based systems.

3 Case Analysis

This chapter describes the case analysis of the Broker. The goal is to answer the second and third research question. This is done by first interviewing employees of the Broker to get information about the processes. This information was then used to do a process and value analysis. The current and target process of the early payment process are described in this chapter. Also, two e3-value models are created to show the values the reverse factoring service of the Broker brings. Last, an architecture of the current situation is modelled in Archimate. This shows the current business, application and technology layer of the application of the Broker. All this information can be used in the following chapters.

3.1 Introduction

As said in the introduction, the Broker has created a platform that aims to improve the cashflow between buyers and their suppliers. Buyers can offer early payment of invoices to their suppliers, with dynamic discounts. This improves the liquidity of suppliers, by giving them access to more cash. Buyers who have excess cash hardly receive any interest on that cash. By offering early payments to their suppliers, buyers receive a higher return on their cash, while the liquidity position of the suppliers improves. The marketing strategy of The Broker is to get big buyers to use their platform, with the promise of a higher return on their cash by providing early payments, with the dynamic discounts, to their suppliers. When a buyer is onboard, suppliers are invited to also join the platform, to connect with their buyers and get access to more cash.

To extract the information of this case, several analyses will be completed. CAPE had already looked at the main process of the early payments. This information gives a background of the process and will be used in the process analysis. To verify this information and gather more information about the business processes and the value exchanges, several interviews have been conducted.

For all the analysis, the main actors must be identified. In this case, there are a lot of actors, but they can be categorized in 4 groups.

1. *The Broker:* The Broker is the creator of the service and does the development and maintenance of its platform. They have the responsibility to act as the trusted third-party. They guarantee, that within the system, the information about the users is save, the system works without issues and there will be no double spending.
2. *Suppliers:* The suppliers use the system to gain the opportunity of receiving an early payment. They can accept the offers of early payments.
3. *Buyers:* The buyers use the system to offer early payments and by doing this they can receive a discount on their invoices. They must upload the invoices, pay, possibly the early payments, the suppliers or they will have to pay the investors.
4. *Investors:* The goal of the Broker is to expand its service, so investors can act as factors. The investors can use the system to offer early payments to the suppliers. They will then receive the full invoice amount from the buyers, to gain return on their investment.

3.2 Analysis Methods

This section explains the analysis methods that were used in this research. To gather the information about the case, several interviews were done with employees of the Broker. This information is then used for the process and value analysis. Last, the current enterprise architecture is modelled in Archimate, with the business, application, and technology layer.

3.2.1 Interviews

Interviewing can be used as a method to gather information and ask experts their opinion. Three types of interviews can be distinguished: structured, semi-structured and unstructured interviews (Bernard, 2006). For both the information extraction and architecture validation, semi-structured interviews were conducted. This type of interviews provides the structure to interview experts about topics that were predetermined, while also giving the possibility to go deeper into relating topics (Cohen & Crabtree, 2006).

In total, three interviews were conducted in this research. The parties present during these interviews were one of the senior programmers of the Broker, the head marketing and acquisition of the Broker and several people from CAPE groep. The first two interviews were focussed on analysing the business ecosystem the Broker is has positioned itself, what business services it provides, and more in dept on the technical level. The first interview was used to gather information about its current processes and architecture. The second interview was used to evaluate and validate the process and value analyses. The last interview was used to evaluate the new architecture and the influence of the implementation of blockchain on the business of the Broker and its value streams. This information is discussed in chapter 6.

3.2.2 Process Analysis

The information that was gathered through the interviews will be used as input for the process analysis, together with an earlier analysis of CAPE groep, that was done prior to this research. To give insight in the processes the Unified Modelling Language (UML) will be used to create activity diagrams (Rumbaugh, Jacobson, & Booch, 2004).

3.2.3 Value analysis

To get an insight in the value creation the broker provides, the e3-value modelling technique of Gordijn & Akkermans is used (2003). It shows the creation of value in a network, by trading economic values between firms. It is emphasised that these models should not be confused with process or activity models, the models show what is exchanged and by whom, but not how. This model was created to give insight in the feasibility of e-commerses. E3-value modelling uses "*value viewpoints"* to show the value creation logic of a firm, which is at the highest level abstraction (Akkermans & Gordijn, 2003).

The concepts that are involved in an e3 value model are described below (Akkermans & Gordijn, 2003). Their relations are visualized in Figure 16.

1. *Actor:* an actor is an independent economic and/or legal entity. Each actor should be capable of making a profit or to do utility increase.
2. *Value object:* value objects are services, goods, money, or even consumer experiences, which are exchanged between actors.

3. *Value port:* value ports are used by actors to show to its environment that they want to provide or receive value objects.
4. *Value offering:* a value offering models what an actor offers or requests from its environment.
5. *Value interface:* a value interface is composed of one or more value offerings. The value interface models that an actor is willing to offer something of value but wants to receive something of value in return.
6. *Value exchange:* a value exchange connects two value ports with each other
7. *Market segment:* a market segment is a concept that breaks a market, consisting of actors, into segments that share common properties. It shows a set of actors, that share one or more of their value interfaces, with their value objects, from an economic perspective.
8. *Composite actor:* a composite actor clusters the value interfaces of other actors and it has its own value interfaces. This grouping of actors is used to show a value constellation and could also represent partnerships.
9. *Value activity:* a value activity is a collection of operational activities which are performed by an actor.

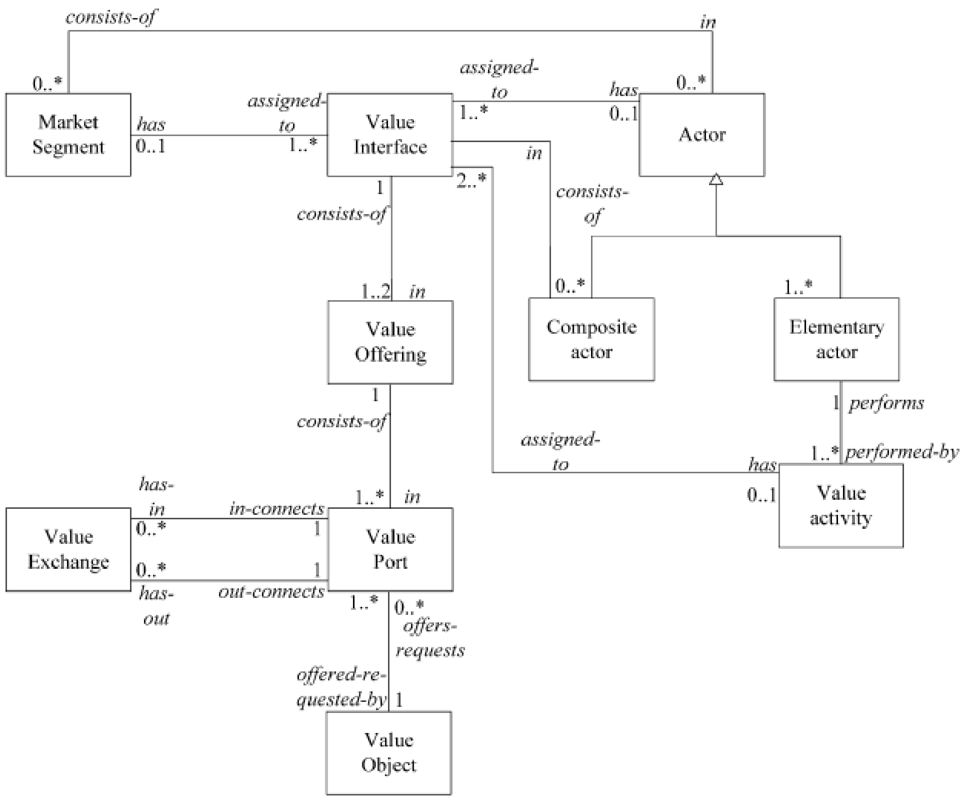

FIGURE 16: CONCEPTS OF E3-VALUE AND THEIR RELATIONSHIPS
SOURCE: (Akkermans & Gordijn, 2003)

3.2.4 Modelling the architecture

To get an overview of the impact of the implementation of blockchain in the current system of the Broker, the whole enterprise architecture of the broker must be analysed. An enterprise architecture includes everything needed in managing and developing an organisation. It takes a holistic view of its business processes, information systems and technology layer (Niemi, 2006). Niemi states that an enterprise architecture provides a tool for aligning and integrating strategy, people, business and technology. It enables an agile enterprise that is continually evolving within dynamic environments. A language used to model these enterprise architectures is Archimate (2017).

Archimate is an open and independent language that can be used for the development, maintenance and operationalization of business processes, organisational structures, information flows and IT systems. The full framework is presented in 6 layers: Strategy, Business, Application, Technology, Physical, and Implementation and Migration (Group, 2017). The Archimate core framework consists of the Business, Application, and Technology layer. In this research, only the core will be used. Figure 17 shows an overview of the concepts used in Archimate. It is a service-oriented model, the higher layers use services provided by the lower layers.

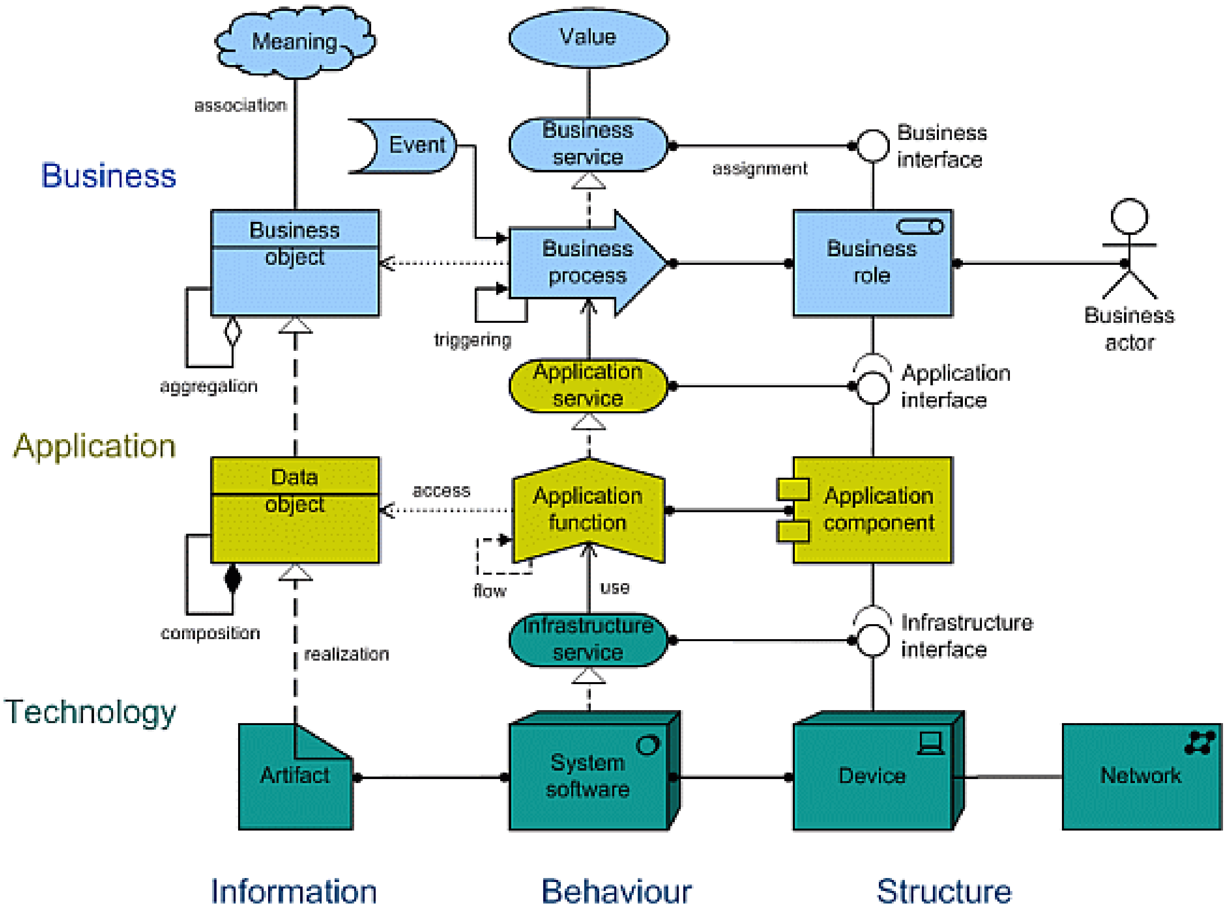

FIGURE 17: ARCHIMATE CONCEPTS

3.3 Process Analysis

In this section the important finding relating to the early payment process are presented. In the introduction, an overview of the process is given and explained. Here the processes are explained in more detail in activity diagrams in UML. First the normal process of paying an invoice is visualised. This is followed by the processes of early payment when parties use the service provided by the Broker. These processes can later be used for the designing the enterprise architectures.

3.3.1 Normal process

Figure 18 shows the process that is followed between two companies when the supplier invoices the buyer. This is done after the products or services are delivered to the buyer. A payment term is discussed beforehand. A supplier then sends an invoice to the buyer. He must manually enter the invoice in his enterprise system (ES) of choice. The invoice will then be approved by the buyer if he agrees. Then payment phase will initiate and eventually the supplier will receive its money. This process is not instantaneous, because the payment term could go up to 60 days.

3.3.2 Early payment process

The early payment process is shown in Figure 19. It starts the same as the normal process, but now the buyer has the chose to offer an early payment. When the buyer decides to offer an early payment, he must upload this invoice to the Broker's platform. The system will then daily calculate the possible discount the supplier should give to buyer, depending on the day of payment. Each day, the supplier can choose to accept the early payment. If he doesn't accept, the early payment process will end by taking the normal payment path. If he does accept, he must manually update the invoice in his ES. The Broker will notify the buyer, that the early payment is accepted, and he receives the new invoice. The buyer too must manually change his information about the invoice. Then the payment phase is initiated.

3.3.3 Early payment process with investors

As said in the introduction, the Broker wants to give investors the opportunity to acts as factors. By doing this, the supplier is no longer completely dependent on the buyer to improve the cashflow. The Brokers ultimate goal is to set up a fund for investors to invest in. The Broker will then take responsibility to invest the money, by acting as the factor for suppliers. But to achieve this, they first want to give investors the opportunity to act as factors themselves. If the number of investors does not grow, because they have to put in too much work, the Broker wants to involve to the fund keeper. But first they want to expand the system to a market place for investors.

This implementation of investors has an impact on the current early payment process. Figure 20 shows the extra complexity it will bring to the process. Both the previous processes are kept intact, so if the buyer elects to offer an early payment, the process stays the same. But if the buyer elects not to offer an early payment, he is confronted with another choice. He can give permission to the Broker to offer the invoice to investors, so they can act as factors. This permission is important, because the buyer will be held accountable if he does not pay the investor.

If the buyer gives his permission to the investors, investors can choose to offer their investment to suppliers and facilitate the early payment. If a supplier accepts, the investor will pay the discounted invoice to the supplier. The buyers will then have to change their invoice as well, because they will now be invoiced by the investor, instead of the supplier. On the due date of the invoice, the buyer will then pay the full amount to the investor.

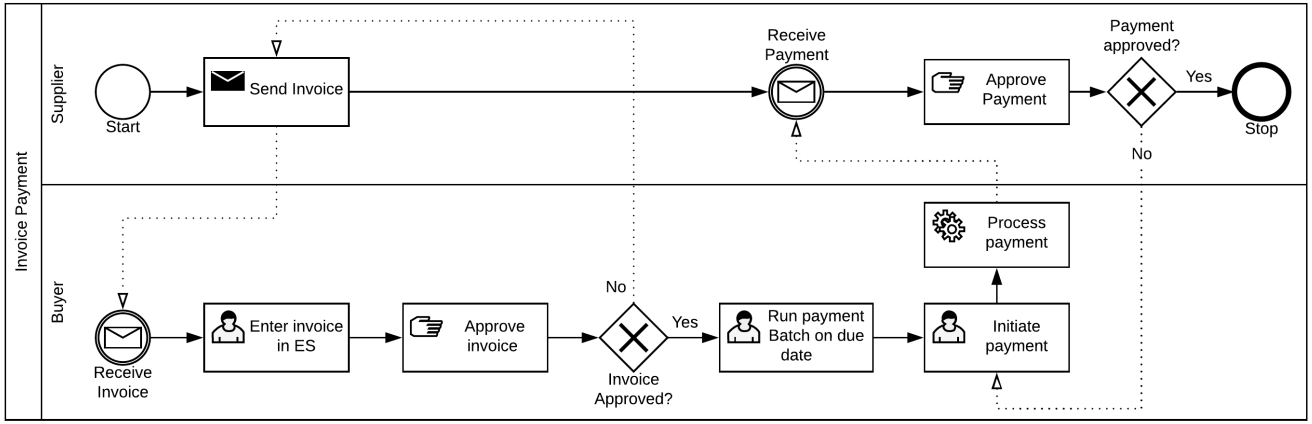

FIGURE 18: NORMAL PAYMENT PROCESS

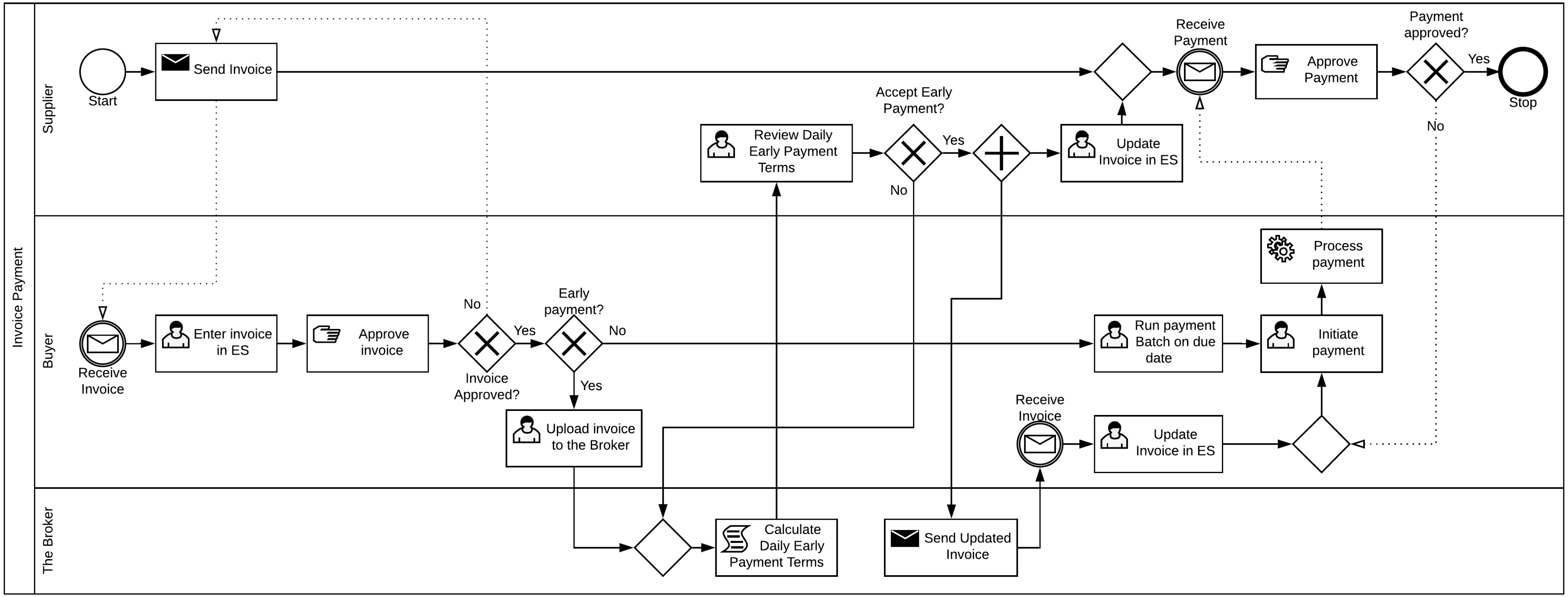

FIGURE 19: EARLY PAYMENT PROCESS

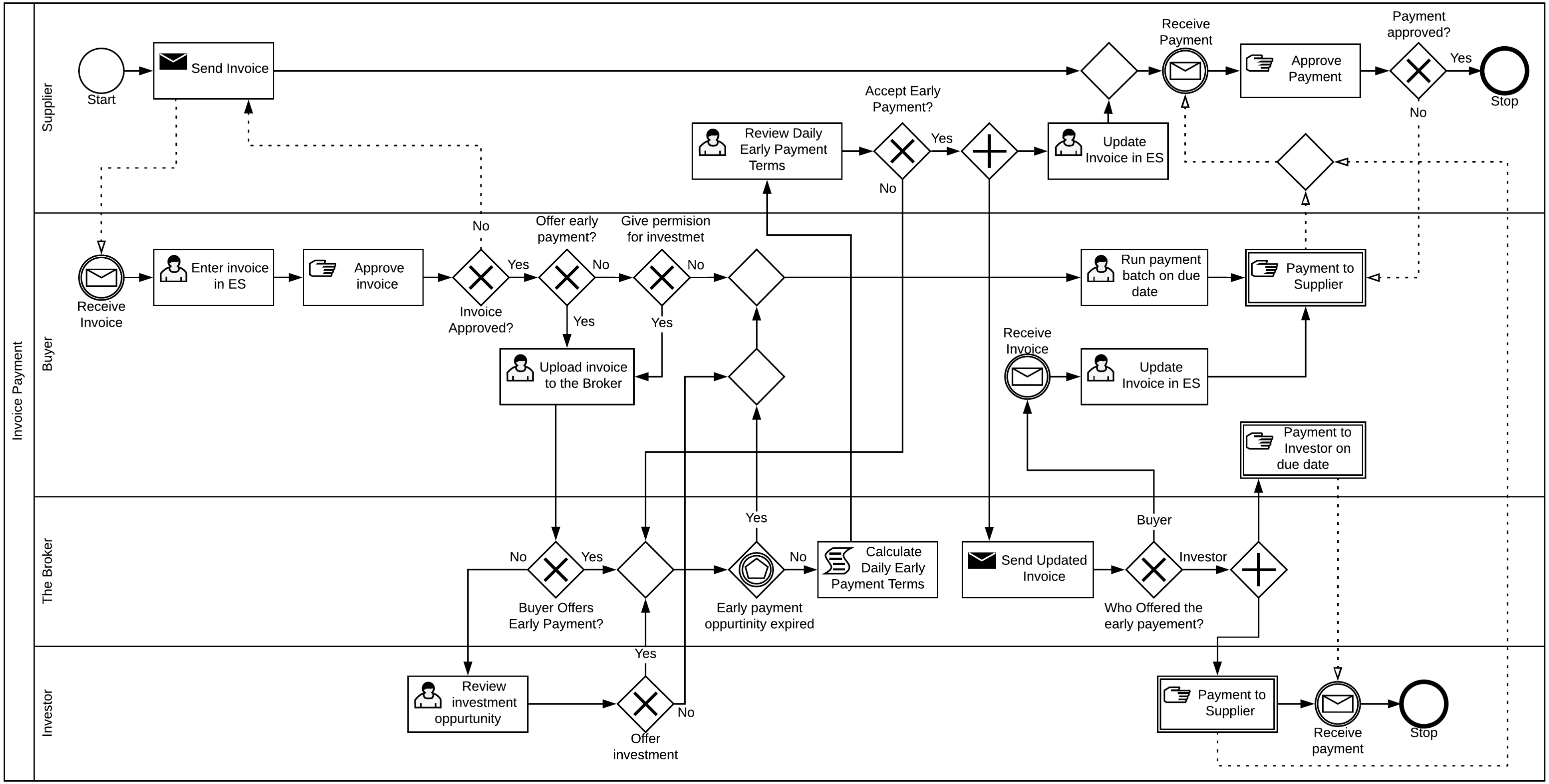

FIGURE 20: EARLY PAYMENT PROCESS WITH INVESTOR

3.4 Value Analysis

As said in the introduction of this chapter, there are four actors involved in the service that the Broker provides: the Broker, the suppliers, the buyers and eventually the investors. They all benefit from providing/using the service. In this section, these benefits are analysed by using e3-value modelling. In the following sections both the current and target situation are analysed and modelled.

3.4.1 Current value model

Figure 21 is the e3-value model of the current situation. In the current situation are 3 actors: the Broker, suppliers and buyers. The buyer and supplier are modelled as market segments. All the buyers share the same value interfaces and objects. The same goes for the suppliers.

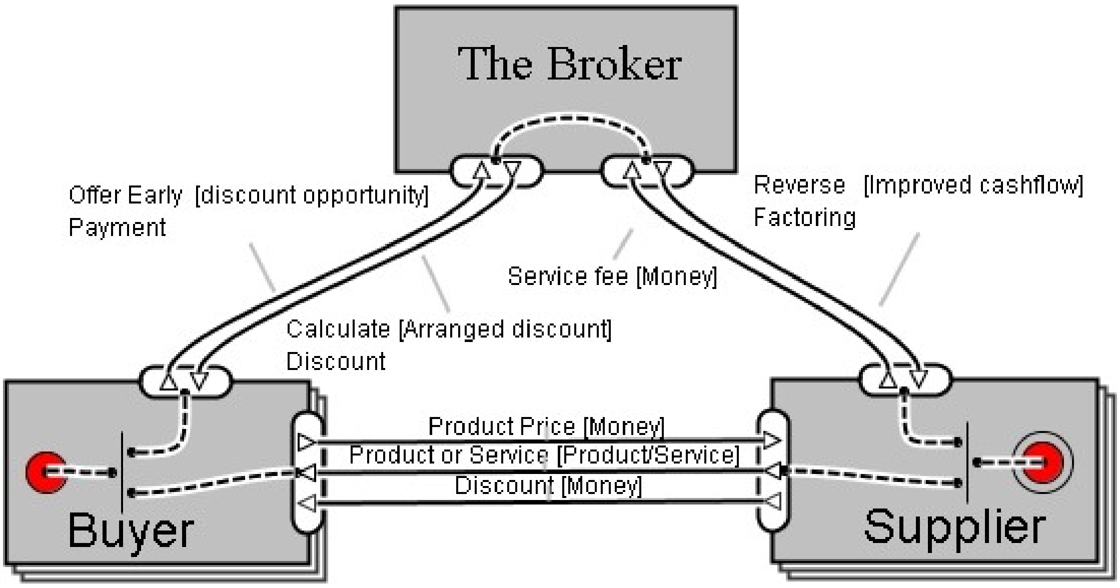

FIGURE 21: E3-VALUE MODEL CURRENT SITUATION

The buyers and suppliers exchange multiple values. The supplier delivers products or services to the buyer. In return the supplier receives the price for these deliverables in the form of money. But this can come at a cost of giving a discount, if the supplier accepts an early payment. This is related to the value exchanges between the buyer/supplier and the Broker.

The Broker provides a service to both parties that facilitates the reverse factoring process. The Broker needs both buyers and suppliers to receive value. The value that the Broker wants to receive is money. The suppliers need reverse factoring to improve their cashflow. Buyers can benefit from reverse factoring, because they receive a discount, but they are not in need of reverse factoring. Therefore the Broker decided that the suppliers will need to pay a service fee to the Broker. This service fee is a percentage of the discounted. This is where the Broker receives his desired value. In return he provides an improved cashflow to the supplier.

But for the Broker to receive this service fee, buyers need to offer early payments to the suppliers, through the Broker. This gives the Broker the opportunity to create value for the supplier. In return the Broker arranges the discount, through their dynamic discount algorithm. Then the buyer can receive his discount value, in the form of money, from the supplier. In this way all parties receive their desirable values, but they need each other.

3.4.2 Target value model

When the Broker reaches his goal of integrating the last actor, the investors, the value model will change. Figure 22 shows the e3-value model of the target situation. The value streams between the supplier and buyer, and supplier and the Broker stay the same. But the Buyer will have to provide a new value to its environment. This will be explained after the introduction of the investors.

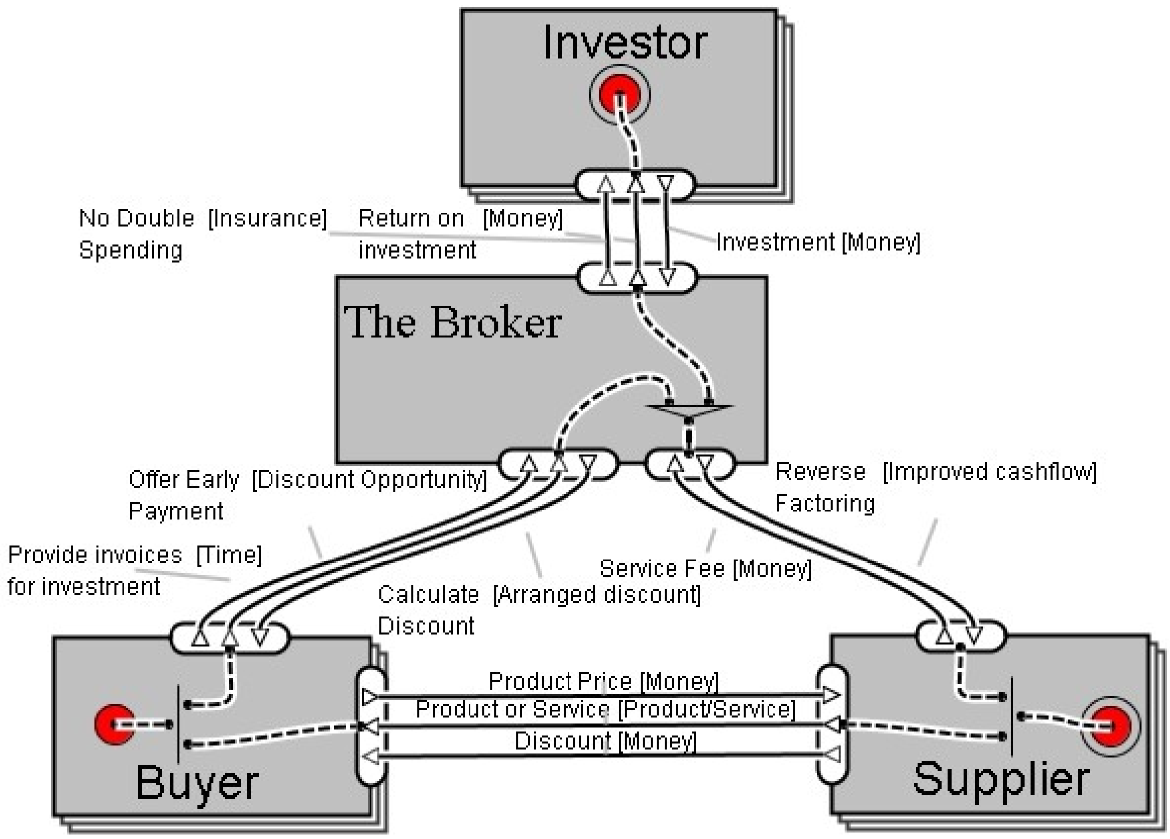

FIGURE 22: E3-VALUE MODEL TARGET SITUATION

The investors are added through a new market segment. The same as for the buyers and suppliers, all the investors share their value interfaces and objects. By acting as a factor for the suppliers, facilitated by the Broker, the investors also give the suppliers an improved cashflow. This is done by facilitating investments in the form of money. In return they receive a return on their investment, also in the form of money. Also, the Broker provides an insurance to the investors, that the suppliers are not selling their invoices to multiple investors within the platform of the Broker.

With the input of the investors, the Brokers value towards the supplier increases. He can provide the supplier with more early payments, through the factoring process. If the suppliers will accept more early payments, the Broker will receive more money by the service fees. So far, all the actors benefit from the implementation of the investors, but for the buyers it creates an extra contribution.

Buyers will have to give their permission to the Broker, to make an invoice available for investors. By doing this, they enable the other actors to proceed with the early payment process. When this succeeds, the buyers will have to pay the investor instead of the supplier and therefore will have to change the information of their invoice in their systems. So, the other actors will not create value, without the buyers offering their invoices for investment, which takes time and therefore costs money. But they won't receive extra value from doing this. This could be a problem. What is the incentive for buyers to participate, when they do not receive any new value?

3.5 Current Architecture

In this section, the current architecture of the application of the Broker is explained. The architecture will show how the business processes are related to the applications and technologies. This will be useful for creating the new architecture, where blockchain will be implemented. The business layer will contain the contain a simplified version of the early payment process described in section 3.3.2. Also, the application and technology layer will mainly show the functions and services that are related to the business layer. This simplification is done to make the architecture clearer. Services like user management are neglected, because we think that these parts will not influence the goal of this research.

The current architecture shows how an involved party can take the role of both buyer and supplier. A buyer uses the application to upload invoices. In this research the focus lies on the early payment process and not on the uploading of invoices into the system, so we will not go into dept in the upload process. Suppliers can use the application to select invoices for early payment. These functions are supported by the application and technology layer. To create a clear distinction between environments within the architecture, each grey area indicates and environment, given an appropriate name. In the current architecture, there is a small part for the party's technology layer, next to the Broker's technology layer. This layer will defer per party, so it is generalized to Enterprise Systems. This is where the buyer and supplier store their invoices and must make the changes.

The current architecture is shown in Figure 23 as well as in Appendix A: Architectures. In the following sections, each of the layers will be clarified.

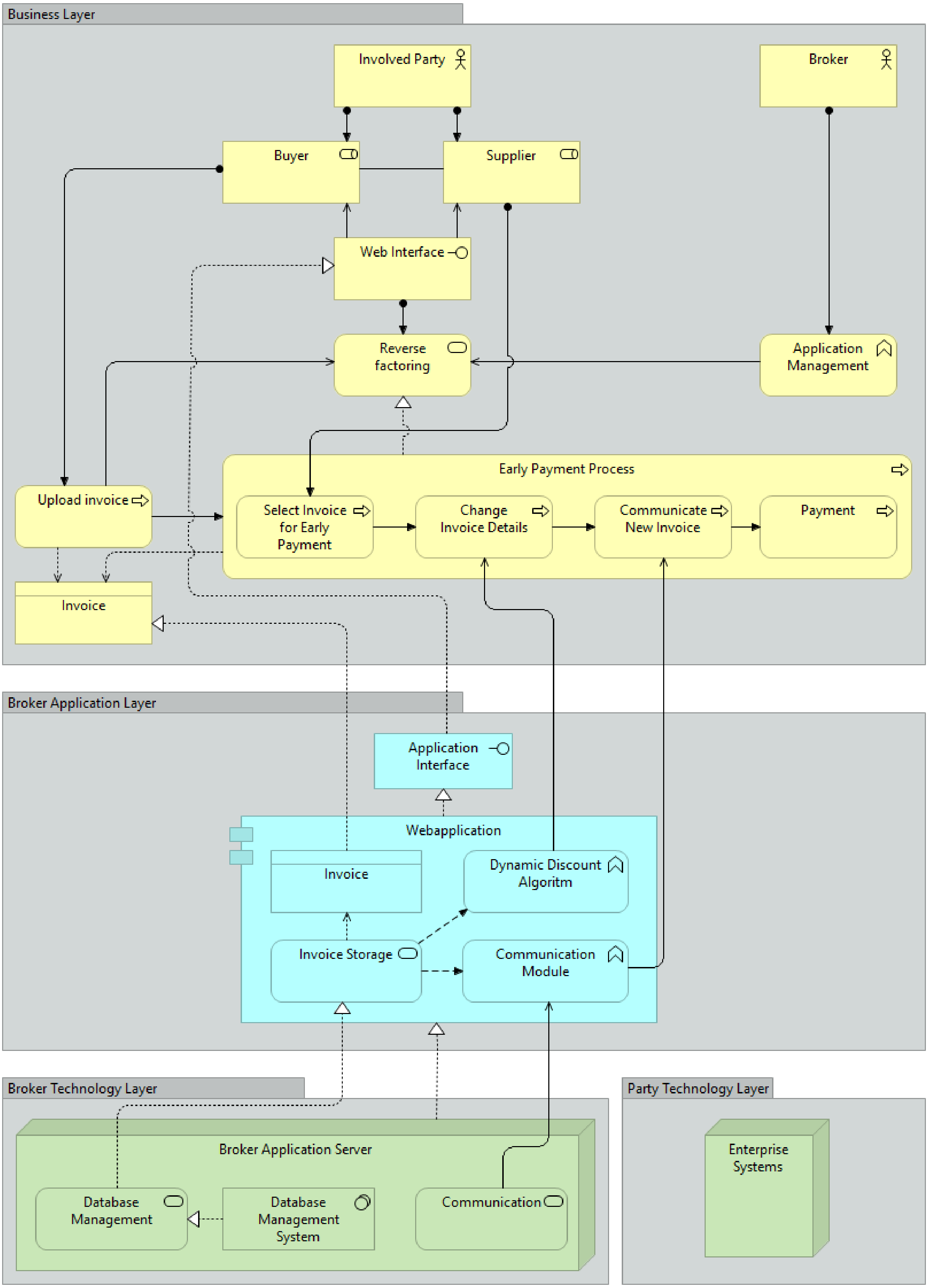

FIGURE 23: CURRENT ARCHITECTURE OF THE BROKER

3.5.1 Business Layer
The business layer is the top layer of the architecture, indicated by the yellow blocks. It is also shown in Figure 24.

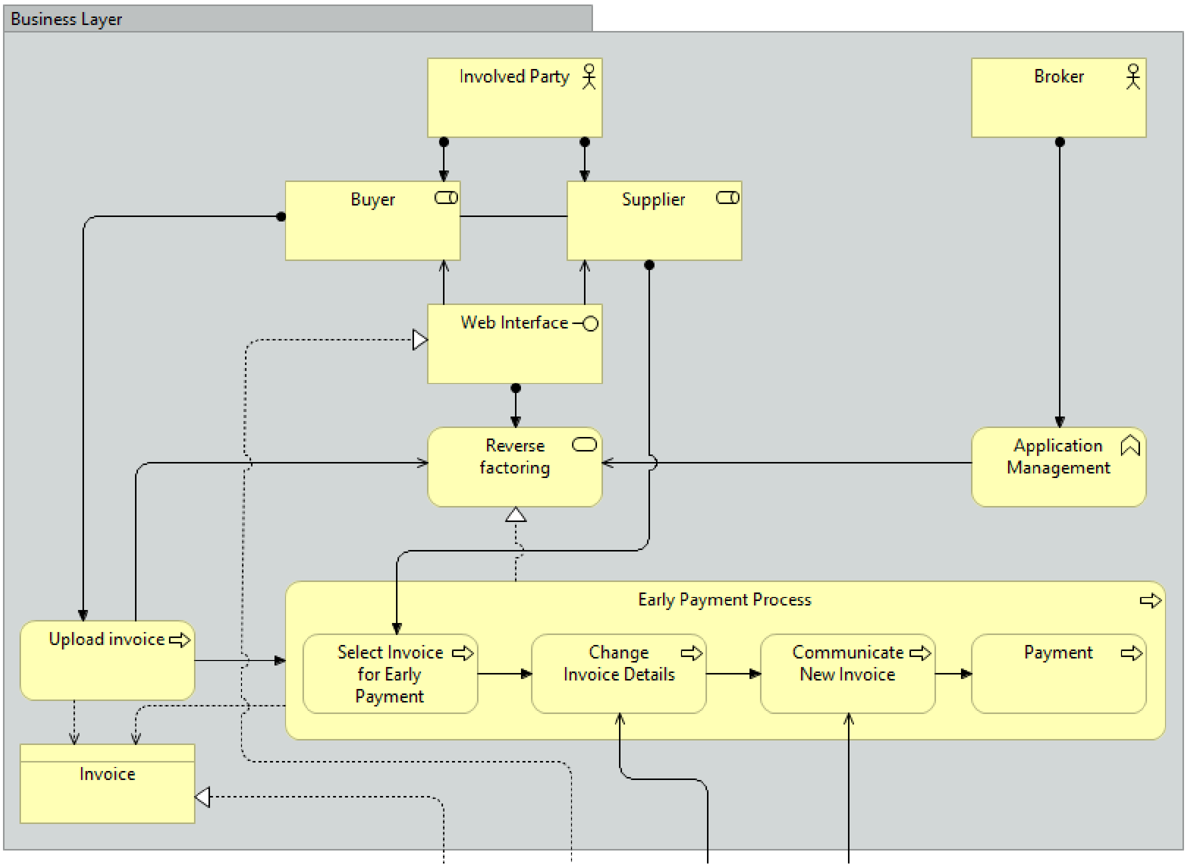

FIGURE 24: BUSINESS LAYER - CURRENT ARCHITECTURE

In the top, two actors are described: *the Broker* and *Involved Parties*, although in the activity diagrams and e3-value models three actors were described. For the application of the Broker, both the buyer and supplier are users of the system, so for the application they are the same actor. An involved party can choose to take a business role within the architecture, a *buyer* or a *supplier.*

The main service that the Broker provides with this application is reverse factoring. This is the business service that services the business roles, through an interface. The parties use a web interface to communicate with the application and by doing that receiving the reverse factoring serves from the Broker, who develops and manages the application. The reverse factoring service is implemented by the early payment process. This is the core process of the application.

The buyer's main responsibility in the application is to upload invoice to the platform. As said, we are not going into detail of the uploading of invoice, that is not relevant for this research. But without this process, the early payment process cannot start. This is because the responsibility of the supplier is, selecting invoices for early payment. Without the business objects invoice, the supplier is not able to start the early payment process. This process is described in Figure 19 of section 3.3.2. When an invoice is selected for early payment, the invoice details change, mainly the price and due date. This is then communicated to the parties, this is done via the application or email. Last the payment process starts, from where the supplier will receive its money.

3.5.2 Application Layer

The application layer is the middle layer of the architecture, indicated by the blue blocks. It is also shown in Figure 25.

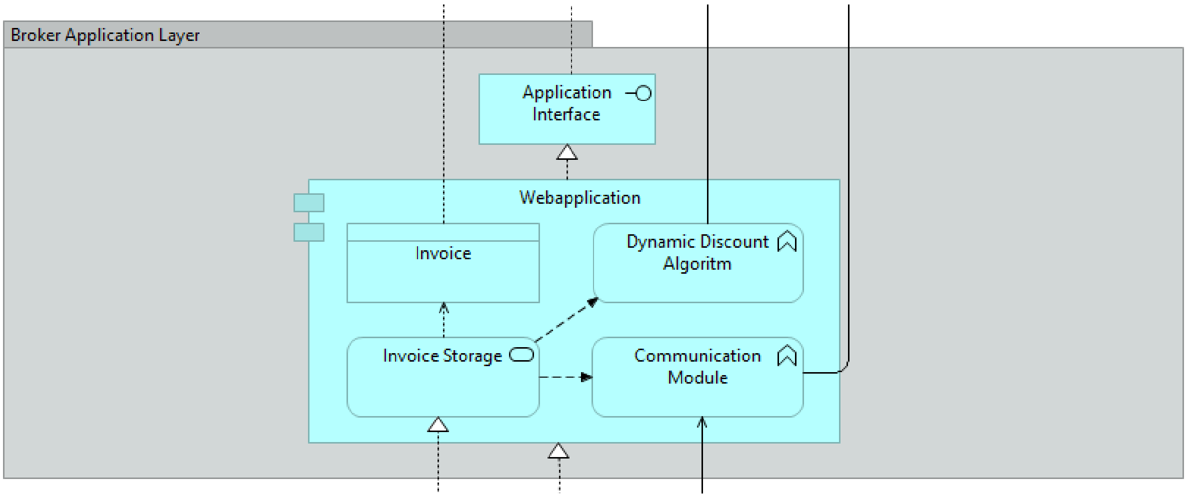

FIGURE 25: APPLICATION LAYER - CURRENT ARCHITECTURE

For the business roles to use the system, they use a web interface. This is an application interface, implemented by the web application of the Broker, the application component. The web application is composed of multiple functions and services. The dynamic discount algorithm function serves the change invoice details from the business layer. It calculates the discount for the early payment of the selected invoice. And the communication module function service the communication of the new invoice details. As said, this is done via the application and by email. Last the web application has an invoice storage service. This service is responsible for managing all the invoices in the application and making them available for the other functions and services that need the invoices.

3.5.3 Technology Layer

The technology layers are in the bottom layer of the architecture, indicated by the green blocks. It is also shown in Figure 26.

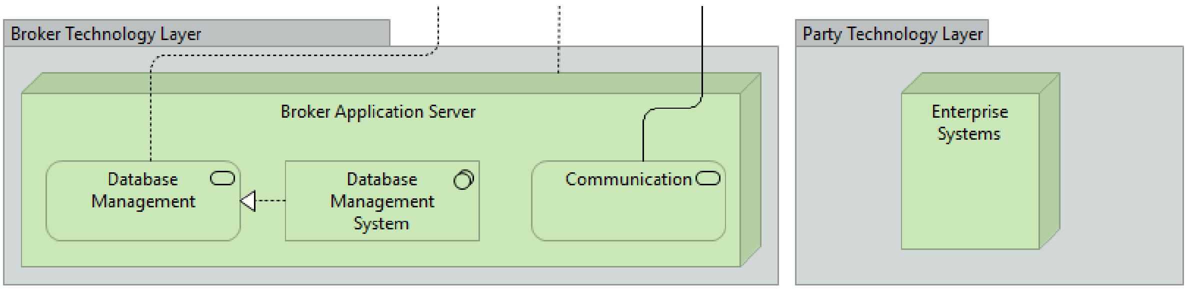

FIGURE 26: TECHNOLOGY LAYER - CURRENT ARCHITECTURE

The Broker has his own server to host the web application. To create a clear overview of the architecture, we modelled the server as one node. In reality, the application hosting is done by three distinct servers. They share the load and serve as backup for each other. For now, we will assume that it is one server. On this server run all the technologies to make the application possible. The most important services are the communication and database management. They are responsible for the serving and implementation of the corresponding application service and function.

The party technology layer was created to show that in the current situation there is no direct communication between the application of the Broker and the Enterprise Systems of the parties who use the system.

3.6 Conclusion

The core process of the Brokers service is the early payment process, described in Figure 19. This is needed to let the buyers and suppliers use the reverse factoring service the Broker provides. By doing this, the suppliers cashflow will improve, the buyer will receive a discount on his invoices and the Broker will receive a service fee from the suppliers, described in Figure 21. This answers the second research question:

> *SQ2: What are the business processes of the Broker's service and which values does it create and request?*

But this will change when the Broker will try to implement investors in the service. The early payment process gets more complicated, described in Figure 20. The value streams for the Broker and supplier will improve and the investors will receive value from joining the service. But the buyers will have to put in an extra effort for the new process to work, described in Figure 22.

By using this information, an architecture of the current situation was created, Figure 23. In section 3.5 this architecture is explained. This gives the answer to research question three:

> *SQ3: What is the current enterprise architecture of the Broker?*

The conclusion is that it will be possible to implement the investors into the system. By knowing the current architecture, we can create a target architecture, to clearly see what must be changed. The value models show that the addition of investors to the system will create more value for the Broker and suppliers, but not for the buyers. To get the buyers on board, the process of uploading and changing the invoices, should be as simple as possible. This should be considered when creating the target architecture.

4 The Reference Architecture

The main goal of this research is to create a reference architecture for blockchain based transaction systems that can be integrated with other systems. In this chapter a new reference architecture is developed. This is done by using a structured method. A quality assessment is done based on enterprise architecture quality attributes. The outcome is a reference architecture that can be used to design blockchain bases transaction systems.

4.1 Methodology

This section will describe the methodology and validation used to create the reference architecture. But to create a reference architecture, it is necessary to first get a clear understanding of what a reference architecture is.

4.1.1 Definition

The concept reference architecture belongs to the field of enterprise architecture. A reference architecture has been defined as the fundamental organisation of a system, embodied in its components, the relationship of these components to each other and the environment, and the guiding for its evolution and design (IEEE, 2000).

The reference architecture is an abstraction and can be used as a blueprint to create enterprise architectures within a specific domain. It provides the principles, guidelines and best practices to create a concrete architecture. It describes the components that should be used, but these need to be selected for specific systems. (Angelov, Grefen, & Greefhorst, 2009)

An architecture can effectively and efficiently document and communicate the core of a business, the information systems that are used, and the IT infrastructure for each of the relevant stakeholders (Iacob, Jonkers, Quartel, Franken, & Van den Berg, 2012).

So, in the case of this research: the reference architecture should give a blue print on how to structure a specific enterprise architecture that uses, or wants to use, a blockchain based transaction processing system. The reference architecture should provide a basic structure of the essential components, but can be used by different enterprises, with different core businesses.

4.1.2 Development method

To develop this reference architecture, the method of Iacob, Jonkers, Quartel, Franken & Van den Berg (2012) will be used. The approach is based on TOGAF's Architecture Development Method (ADM). With the use of a multi-phased cycle, TOGAF is used to relate the Business, Information Systems and IT structure. Iacob et al. (2012) use TOGAF in combination with the modelling language Archimate. Archimate is an open and independent language that can be used for the development, maintenance and operationalization of business processes, organisational structures, information flows and IT systems. We already used Archimate to create the current architecture of the Broker in chapter 3. In section 0 Archimate is explained.

Figure 27 shows the phases of TOGAF, together with the corresponding layers of Archimate. As seen the ADM is an iterative process. This will not be done in this research. In this research The preliminary phase and the phases A through E are executed.

The preliminary phase has already been done in the previous chapters. The vision for the architecture was introduced in the introduction. The literature review and the case analysis of the Broker were used to gain information about the domain. Also, a vision for the architecture is created, but in section 4.2 several objectives will be defined for the architecture. In the development phase, phase B, C and D will be completed, by creating an architecture consisting of a business, application and technology layer. The reference architecture will then be used in the broker case, to create a new enterprise architecture for the target situation of implementing both blockchain and the investors. This will be one of the opportunities and solutions of phase E. The new enterprise architecture will then be used to create a prototype of the opportunity, to examine if it feasible. This solution will then be the topic of the evaluation of the reference architecture. In this evaluation, an impact analysis will be done of the new enterprise architecture to see if a migration plan is needed for the broker, to actually implement the prototype and migrate to the new architecture. The end will be a recommendation to the Broker if they need to go on to phase F in ADM

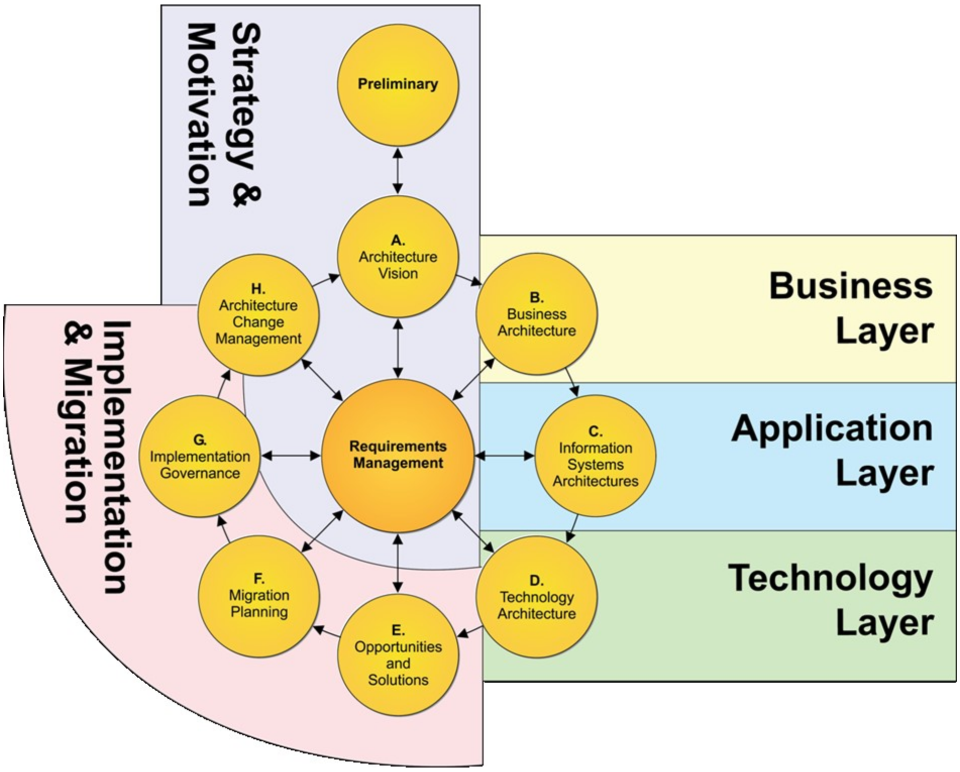

FIGURE 27: TOGAF ADM AND ARCHIMATE
SOURCE: (Group, 2017)

4.1.3 Evaluation method

Also, the quality of the reference architecture will be evaluated. Niemi and Pekkola (2013) identified quality attributes for enterprise architecture products and services. The reference architecture will be used to create new enterprise architectures, so it should have a high quality to contribute to the quality of the enterprise architectures. Five of the quality attributes for the EA products will be used:

1. **Clarity and conciseness:** The architecture should provide an overall and clear view. It should only use the necessary model elements to avoid confusion and use a logical order of elements from top to bottom.

2. **Granularity:** The architecture conveys basic information but gives a sufficient level of detail at the same time.
3. **Uniformity and cohesion:** The architecture satisfies the EA standards, does not replicate already existing architectures and can be used to transform a current architecture to a to-be architecture
4. **Correctness:** This means the architecture should be completely, so no missing elements, and up to date.
5. **Usefulness:** The architecture has a clear purpose in the appropriate context, which should also be clear.

4.2 Objectives

As Figure 27 shows, a central part of TOGAF's ADM is requirement management. Also phase two of the DSRM states the importance of defining objectives for the solution. Therefore several objectives for the reference architecture will be defined. These objectives could later be used to validate the reference architecture. The objectives were made in cooperation with both the Broker and CAPE groep. They were generalized to serve the reference architecture.

1. **The applications based on this reference architecture will be able to save their transactions is a blockchain:** The main goal of the research is to create a reference architecture for a blockchain based transaction system. Therefore this objective is the most important one to be achieved. Users of the system will have to build a consensus together to be able to submit transactions. This should be clear in the reference architecture.
2. **Creating this reference architecture will start the creation of a standard for modelling blockchain in enterprise architectures:** In the literature, no reverence architecture could be found that modelled blockchain in an enterprise architecture. From that we derive that this has not been done before. With this reference architecture we want to propose a standard for modelling blockchain in enterprise architectures.
3. **The reference architecture has a high architectural quality:** The architectural quality of the architecture will be evaluated using the five quality attributes, mentioned in the previous section, of Niemi and Pekkola (2013).
4. **The reference architecture should support different business objectives:** By achieving this objective, the reference architecture could be used by a diverse spectrum of applications. As seen in the literature review, there is no clear reference architecture for the application of blockchain, outside of cryptocurrencies. The creation of such a reference architecture could facilitate new initiatives and speed up the development of blockchain technologies and applications.
5. **The reference architecture should support integration possibilities:** Although an objective is to support a diverse spectrum of enterprises, blockchain applications will always be used when dealing with multiple stakeholders. These stakeholders share an application but will most likely have their own enterprise systems as well. Therefore it will be useful to provide integrations with these systems of the stakeholders.

4.3 Reference Architecture

In this section, the reference architecture will be presented. The information gathered in chapters two and three is used to come to this architecture. Each layer will be explained, to give a clear view of the architecture. The reference architecture is shown on the next page, Figure 29, and can be found in Appendix A: Architectures.

4.3.1 Business layer

The business layer is the top layer of the architecture, indicated by the yellow blocks. It is also shown in Figure 28.

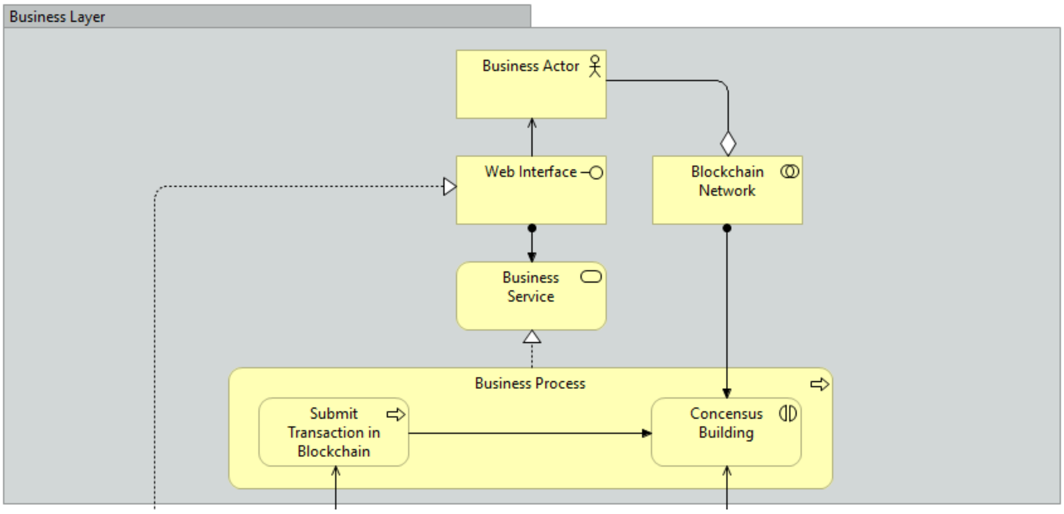

FIGURE 28: BUSINESS LAYER - REFERENCE ARCHITECTURE

One off the objectives was to create a reference architecture for a diverse group of applications. With this reference architecture, a diverse spectrum of business processes and services can be supported, if they contain a form of the indicated elements.

The first element are the business actors within the enterprise environment of the architecture. These actors could take multiple roles are could be modelled separately. One important note is there should be multiple participant using the system. They can be modelled in one actor element, but it is required to have multiple participant. This is because these participants need to be part of the blockchain network collaboration. Together they are responsible for the consensus building interaction within the business process.

The actors are served by a web interface, that exposes the business services that are provided. This business service is implemented by a business process. Again, this could be any business process, but it has to mandatory elements: A business process “submit transaction in blockchain” and a business interaction “consensus building”. The business process is responsible for submitting the transaction to the blockchain. This will trigger the consensus building interaction. This is modelled as an interaction, because the business actors will come to a consensus together whether the transaction is valid. Although this happens automatically, the actors are responsible, through the blockchain network, for this interaction. After the consensus building succeeds, the business process could go on.

By using this business layer for the refence architecture, the reference architecture can support multiple business processes and services.

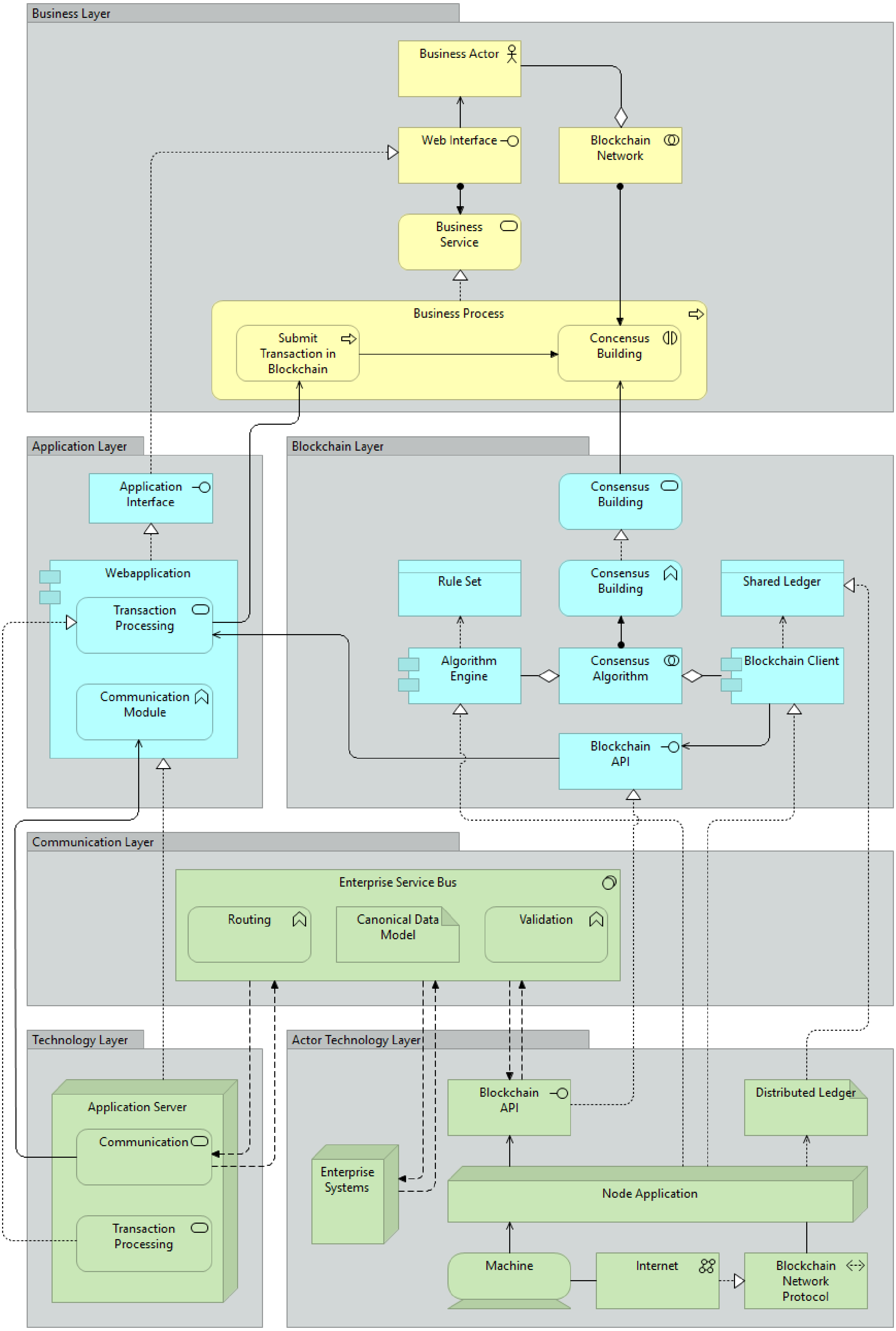

FIGURE 29: REFERENCE ARCHITECTURE

4.3.2 Application layer

The application layer is the middle layer of the architecture, indicated by the blue blocks. It is also shown in Figure 30. The application layer is divided into two separate groups. These groups are used to indicate the environment the elements are being used. First the Application Layer group will be discussed.

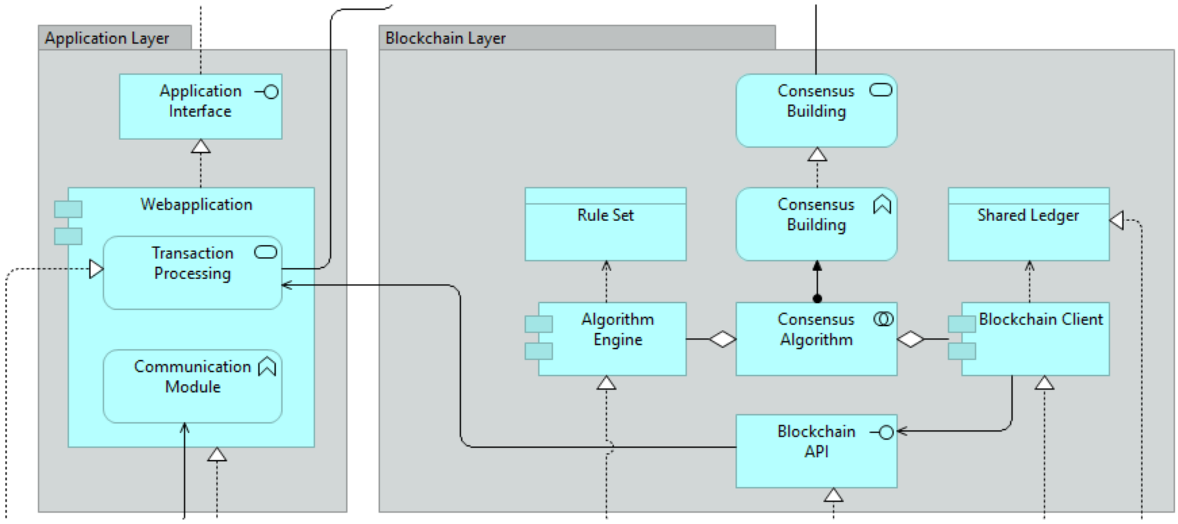

FIGURE 30: APPLICATION LAYER - REFERENCE ARCHITECTURE

The application layer group is where the actual application from the service is being implemented. The web application is a software component that implements an application interface. This interface exposes the software component to the environment. It implements the business interface from the business layer. The software component must have at least two elements: the application service “Transaction processing” and the application function “Communication module”. The transaction processing is necessary to submit the transaction to the blockchain. It serves the business process “Submit transaction to blockchain” to make this process possible. The communication module is necessary to send the transaction to the blockchain and to provide the possible integrations with other systems.

The Blockchain Layer group are the application elements of the blockchain. They run on the node application of the actor’s technology layer. This already indicates that each actor is hosting a node in the blockchain network. Each of these nodes has all the blockchain application elements. The blockchain client is a software component that is responsible for the communication with the outside world, it serves a blockchain API to expose the shared ledger to the transaction processing of the web application. This is the application data object that represents all the transactions that are saved in the blockchain. The blockchain client can access this ledger to retrieve data, or to submit a new transaction when the consensus algorithm succeeds. The blockchain client is modelled as a different software component then the algorithm engine. Although they both run in the same environment, they have very different functions. The algorithm engine is responsible for the work that must be done for building a consensus. This is mainly the calculations of the hashes for the blockchain. How the consensus is built between all the nodes is registered in the data object “rule set”. This also indicated that this reference architecture can be used by all consensus algorithms. The algorithm engine and the blockchain client, do have to work together to do their part in the consensus building, the algorithm engine will do the calculations and the blockchain client will provide the data and will take care of the communication. This consensus building collaboration is responsible for the consensus building function, which implements the consensus building service. This consensus building service then service the business interaction consensus building. This indicates that all the blockchain layers will have to collaborate to reach a consensus.

Although it is complicated to model a blockchain system, we think that this best gives an overview of the application elements that are involved in one node of a blockchain network.

4.3.3 Technology layer

The technology layer is the bottom layer of the architecture, indicated by the green blocks. It is also shown in Figure 31. The technology is divided into three different groups. A Communication layer, a Technology layer, and an Actor technology layer.

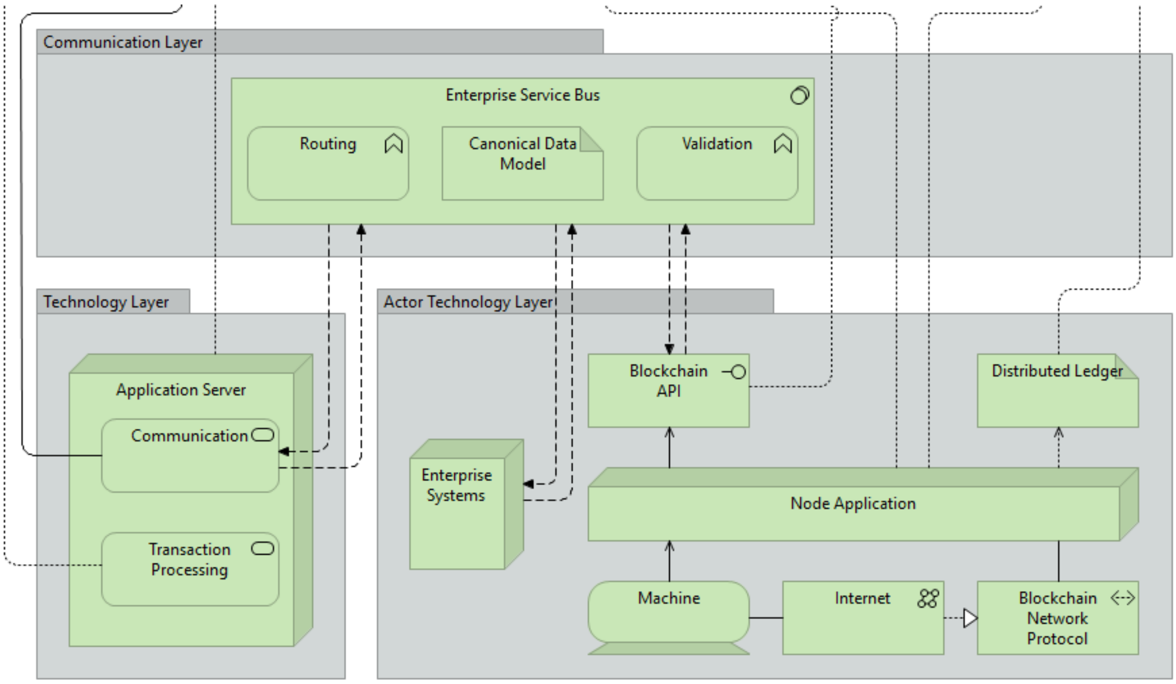

FIGURE 31: TECHNOLOGY LAYER - REFERENCE ARCHITECTURE

First, the Technology layer group. This is the application server that hosts the web application. The server is modelled as one node, to keep it clear, but this could be multiple servers. The application server implements the software component web application. The application server should at least two technology services: “Communication” and “Transaction Processing”. The transaction processing service is the implementation of the application transaction processing service. The communication service is responsible for the communication with the enterprise service bus, which brings us to the communication layer.

To create fast integrations with other system, we choose to use an enterprise service bus to handle the communication between all the different technology nodes. This enterprise service bus is modelled as a system software component. The enterprise service bus can be hosted anywhere. It has some general elements like a routing and validation functions and it uses a canonical data model to map incoming messages to the desired output for each system. The enterprise service bus is not only responsible for the integration with enterprise systems of the users, but we want to introduce it as a medium of communication between the application and a blockchain. We will elaborate on that choice later, but first take a look at the user's technology layer.

The user's technology layer represents the technology layer of each of the actors/participants within the system. It has a node called enterprise systems. We cannot know the exact technological environment of each actor, but we want to indicate that with this setup, it is easy to integrate enterprise systems, via the enterprise service bus. What each actor should have when using the system is the blockchain setup.

Each user must be active in order to participate in the blockchain network. Therefore each user should run a node application on a machine. This could be a virtual machine, or hosted on a local server, but it should be active. The nodes communicate with each other when a new transaction is submitted and will build a consensus. This is done directly between the nodes, over the internet using a specific blockchain network protocol. This protocol will depend on the blockchain technology that is used. The node application implements both the blockchain client and the algorithm engine. As said, they were separated to indicate the difference, but they run on the same node. The nodes will communicate with each other through their own protocol, but the communication with the application goes via the blockchain interface and the enterprise service bus.

As said, we choose to let the communication between the application and the blockchain go through the use the enterprise service bus. We choose to do that to create a modular architecture. Now the application and blockchain technology can be easily replaced. In the current systems, when you want to switch the blockchain technology, you will have to rebuild the entire application and vice versa. Now one can keep existing and the other part is easily changeable. Another advantage is that in theory the system should be faster. Because the enterprise service bus receives all the messages and responses, it is the first node to be notified when a consensus is reached, and a transaction is successfully submitted. The enterprise service bus can then notify all the integrated systems. A last advantage is that a blockchain can now easily be used by multiple servers or even applications in general. Each system can easily integrate with the blockchain, because it is already up and running. This could be useful to create bigger blockchains, to better guarantee a valid audit trail. But this will also bring extra problems. These will be discussed in the evaluation in chapter 6.

By using these technology layers, the reference architecture can be used to create easy integrations between the applications and enterprise systems of the users. Also, the blockchain is modelled in a simple and clear way.

4.4 Conclusion

This chapter gives the answer to the fourth research question:

SQ4: What architecture can be used for the design of a blockchain based transaction processing system?

The answer to this question is the newly designed reference architecture, that can be used to for the design of a blockchain based transaction system. The reference architecture can be found in section 4.3, Figure 29, or in Appendix A: Architectures. This was done by using the method presented by Iacob et al. (2012), using TOGAF's ADM in combination with the Archimate modelling language. In section 4.1.3 five quality attributes were defined to validate the quality of the architecture:

1. **Clarity and conciseness:** The reference architecture has given an overall and clear view of the environment, and it uses a logical order of elements. It could have simplified the application layer of the blockchain a bit more, but this would left out important details.
2. **Granularity:** Following the previous point, the blockchain layer could have been simplified, but than it would lack a sufficient level of detail.
3. **Uniformity and cohesion:** The architecture is build using the Archimate standards, which are followed. Also, a reference architecture like this has not been done before and this architecture can be used to transform current architectures to a to-be architecture.
4. **Correctness:** This point is hard to judge, because this has never been done before. The demonstration and evaluation of the architecture will show if the architecture is complete and therefore correct.
5. **Useful:** This has been met, the architecture can be used for the design of blockchain based transaction systems.

From this assessment we concluded that the quality of the reference architecture is fairly high. This because the correctness of the architecture is difficult to assess at this moment. Next to the quality attributes, also five objectives for the reference architecture in section 4.2:

1. **The applications based on this reference architecture will be able to save their transactions is a blockchain:** The only mandatory business process is the submit transaction to blockchain process. This shows that the reference architecture should be used for applications that want to do that. By using a business collaboration and interaction, the users of the system will have to create consensus to validate the transaction. When this is completed the transaction is submitted into the blockchain.
2. **Creating this reference architecture will start the creation of a standard for modelling blockchain in enterprise architectures:** In the blockchain layer and user technology layer, we propose a standard for modelling blockchain in enterprise architectures. We think that is a correct way to model blockchain, but we don't know if this is the best way possible.
3. **The reference architecture has a high architectural quality:** We concluded that the architectural quality of the reference architecture is fairly high. This because it is difficult to assess the correctness of the architecture before it being used and tested.

4. **The reference architecture should support different business objectives:** This has been achieved. By using this business layer for the refence architecture, the reference architecture can support multiple business processes and services.
5. **The reference architecture should support integration possibilities:** This has also been achieved by the implementation of an enterprise service bus in the communication layer. Through this ESB new system can easily be integrated.

To answer the main research question and see if this reference architecture is actually the good reference architecture for the design of a blockchain based transaction system, the use of the reference architecture should be demonstrated. This will be done in the next chapter.

5 Demonstration of Reference Architecture

To validate the reference architecture, the case of the Broker is being used to test the reference architecture. This is done by first migration the current architecture of the broker to a target architecture, by using the reference architecture. This migration was successful. Also, to demonstrate the correctness of the reference architecture, a prototype of the target architecture was built. Although it was succeeded to build a small blockchain based transaction system using Emagiz as messaging platform, the result could not help a lot in the validation. This because the prototype could not be tested in a large blockchain network with multiple users simultaneously using the system.

5.1 Migrate Architecture

In this chapter the reference architecture created in chapter 4, will be tested with the case of the Broker. In chapter 3 a case analysis was done, with a process analysis of the current situation and the target situation. Also, an enterprise architecture of the current situation was made. To demonstrate the reference architecture, the enterprise architecture of the current situation will be migrated to a new architecture which uses blockchain to record the transactions. This will be done using the target process, were the investors are also implemented.

5.1.1 Business layer

First the business layer from the current situation, Figure 24, needs to be expanded, because of the implementation of the investors. As shown in the target process, Figure 20, this implementation does complicate the process. But looking at the simplification in the architecture, it will not complicate the business layer. The business process “Select invoice for early payment” covers this whole process. This process becomes like described in the target process, but for clarity in the architecture, this will not be modelled.

Just like buyers and supplier, the investors are regarded as users of the system, so they are also modelled under “Involved party”. A new business role is added for the investors. They act the same as suppliers. They can use the business interface to acts with the system. Within the system they can offer early payments to the suppliers, when buyers approve this. So, they are also responsible for the process “Select invoice for early payment”.

The second change in the business layer is the introduction of blockchain. The mandatory process “Submit transaction in blockchain” must be added. Also, the “blockchain network” collaboration and the business interaction “Consensus building” needs to be added in the process. But this is done very easily. All the actors in the system will be participating the blockchain network collaboration, including the Broker and the investors.

These changes present the new business layer, shown in Figure 32.

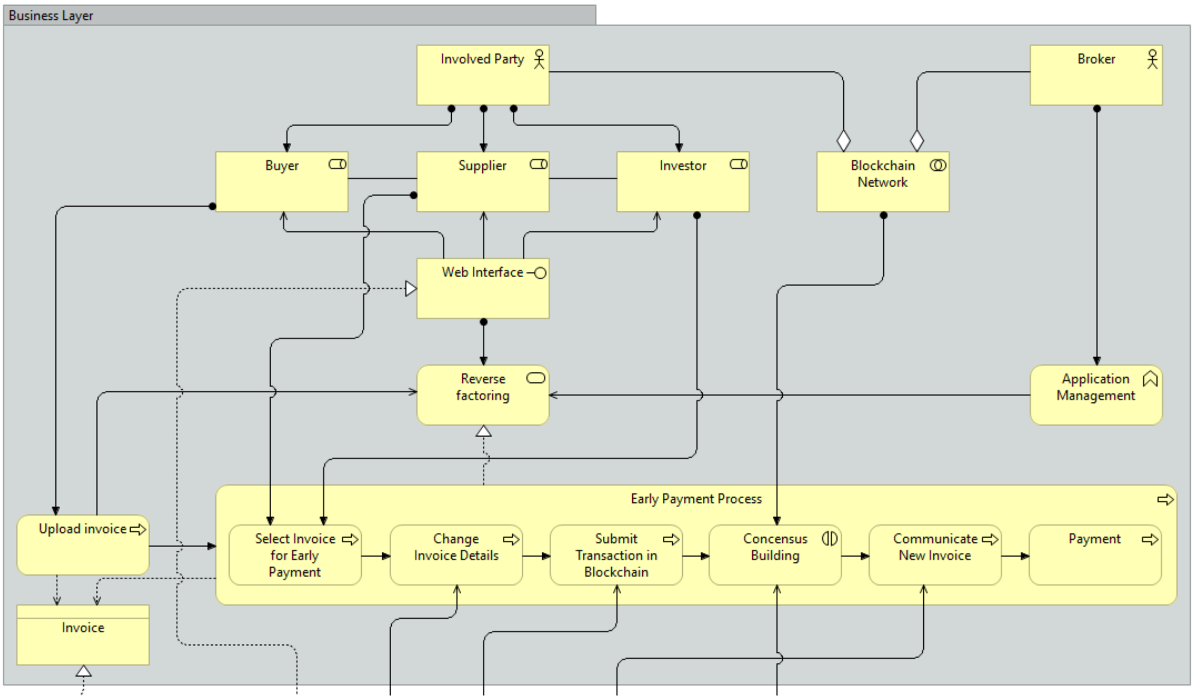

FIGURE 32: BUSINESS LAYER - TARGET ARCHITECTURE

5.1.2 Application layer

The addition of investor in the system will not affect the current application layer, Figure 25, but the addition of blockchain does. A new group is added, the Blockchain Layer. This layer is the same as in the reference architecture. To use the blockchain and to serve the "Submit transaction in blockchain" service, a new application service is added to the Web application: "Transaction processing". This results in the new application layer, shown below in Figure 33.

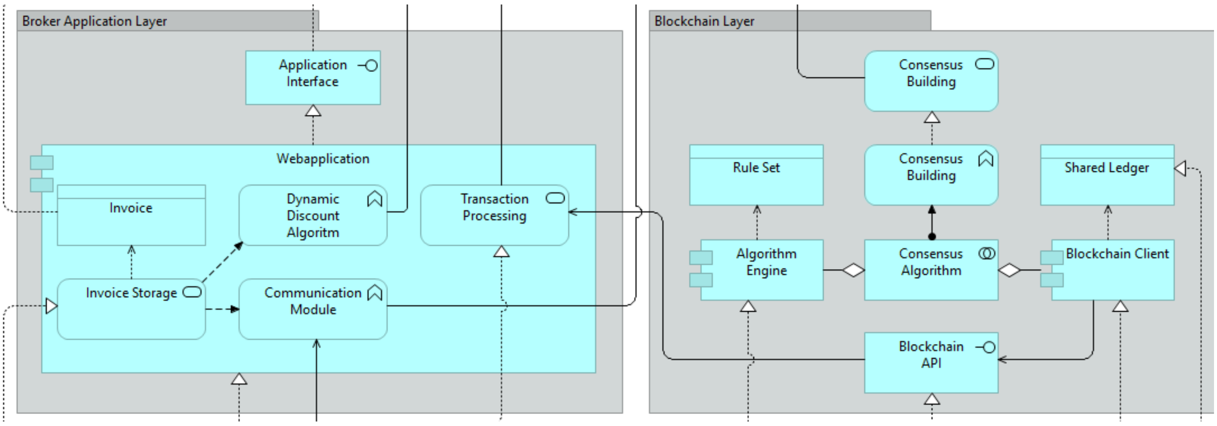

FIGURE 33: APPLICATION LAYER - TARGET ARCHITECTURE

5.1.3 Technology layer

The last migration that needs to be done is the current technology layer, Figure 26. Two new groups are added from the reference architecture, the "Communication layer" and the "Party Technology Layer". Both layers are the same as in the reference architecture, only now the "User Technology Layer" is called "Party Technology Layer".

For the "Broker Technology Layer", the technology service "Transaction Processing" is added to implement the application service "Transaction Processing". Also, the technology service

“Communication” is now responsible for the communication between the application and the Enterprise Service Bus. The result is shown below, Figure 34.

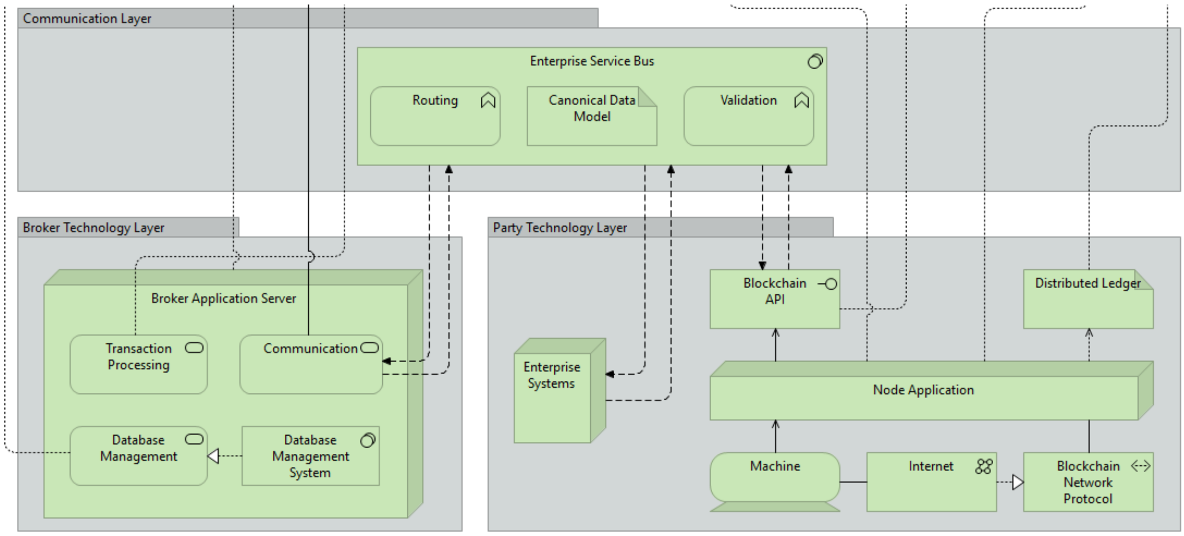

FIGURE 34: TECHNOLOGY LAYER - TARGET ARCHITECTURE

5.1.4 The complete target architecture

The migration of all the enterprise layers, using the reference architecture, result in a new enterprise architecture: The target architecture, Figure 35, which can also be found in Appendix A: Architectures.

By doing this migration, the first step of phase E of the ADM and evaluation of the reference architecture are completed. In the case of the Broker the reference architecture can be used to migrate the current enterprise architecture to a new enterprise architecture. The next step of phase E and the evaluation is creating a prototype, to see if a system build on this architecture will work.

5.2 Methodology Prototype

This section describes a prototype made for the Broker case. The aim is to create a simplified version of the platform of the broker, but were the transactions are stored in a blockchain and the communication is done through an enterprise service bus. First the scope of the prototype is defined. Then the approach of building the prototype is presented, together with the technologies that were used. Last a description of the prototype.

5.2.1 Scope

The goal of the prototype is to prove that in theory an implementation of an enterprise architecture, based on the reference architecture, can work. Because there is not yet such a system or a similar system, the prototype should be built from scratch. This is also the interest of CAPE Groep. CAPE Groep is interested in the use of blockchain, in combination with Mendix and Emagiz. These are the tools that will be used to build the web application and the enterprise service bus.

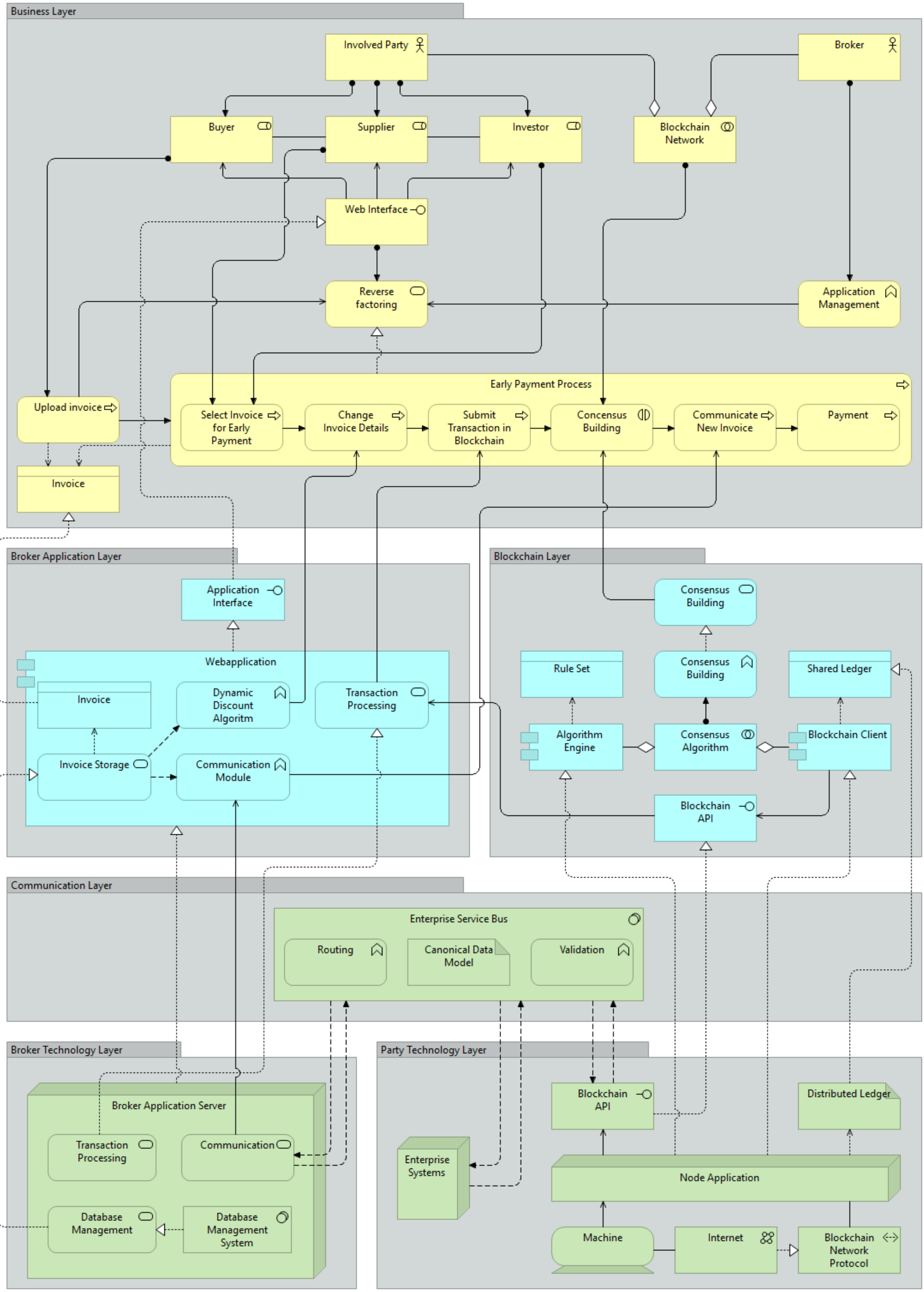

FIGURE 35: TARGET ARCHITECTURE

To validate the reference architecture, the prototype should be able to submit transactions in a blockchain through the enterprise service bus. Ideally, we would also like to create an integration with an enterprise system, but it has been proven that an ESB is capable of these integrations, so this will be out of scope for this prototype. Also, the implementation of the investors is left out of the prototype. This would create extra complexity, but won't support the goal of investigating the use of an ESB in combination with the blockchain and web application.

Therefore the prototype has the following objectives:

1. **Recreate a simplified version of the web application of the broker:** We are going to develop a similar web application as the Broker to simulate the Business, Application and Technology layer of the new architecture for the Broker's platform.
2. **Set up a blockchain network that can store transactions:** We are going to setup a blockchain network that can store the transaction information of the early payment agreements.
3. **Develop an enterprise service bus that facilitates the communication between the web application and the blockchain:** The ESB will act as the communication layer from the architecture between the application and the blockchain.

5.2.2 Techniques

The objectives are separated in three parts, which correlated with three different systems that need to be made: the blockchain, the web application and the ESB. Three different techniques will be used to build these systems. The blockchain network will be created using Hyperledger Composer, the web application with Mendix and the ESB using Emagiz. In the next sections these techniques will be explained.

5.2.2.1 Hyperledger Composer

To build the blockchain network for this prototype, Hyperledger Composer will be used. Hyperledger Composer is one of the projects of the Linux Foundation. It is a development platform to quickly create a Hyperledger Fabric based blockchain network. You can setup your blockchain network, using a web application, or download the developers tools on a Linux machine. When you have setup the network, a Fabric network will be compiled and can be deployed. (Hyperledger Composer)

Hyperledger Fabric, or simply Fabric, introduces a blockchain architecture that aims at flexibility, scalability and confidentiality. Due to its flexible nature and the use of Composer, it is a widely used platform to create prototypes and proof of concepts in many different use cases. Fabric follows a novel execute-order-validate paradigm for distributed execution of untrusted code in different environments.

In the first phase, the execution, a client node in the network signs and sends a transaction to endorser nodes for execution. The endorsers execute the transaction, which contains the identity of the submitting client, the transaction data and a transaction identifier. The transaction then enters the ordering phase, which uses a pluggable consensus protocol, to create an ordered sequence of endorsed transaction grouped in blocks. This consensus protocol is based on the Byzantine-fault tolerant (BFT) protocol, with an introduction of a novel hybrid replication paradigm in the Byzantine model. The state of the blocks are then broadcasted to all the peers in the network and the final phases starts, the validation phase. Each of the peers validates the state changes of the blockchain. When this is completed and the new endorsed transactions in the block are accepted, the changed state becomes the new blockchain state. An high level overview of theses phases is visualised in Figure 44. (Androulaki et al., 2018)

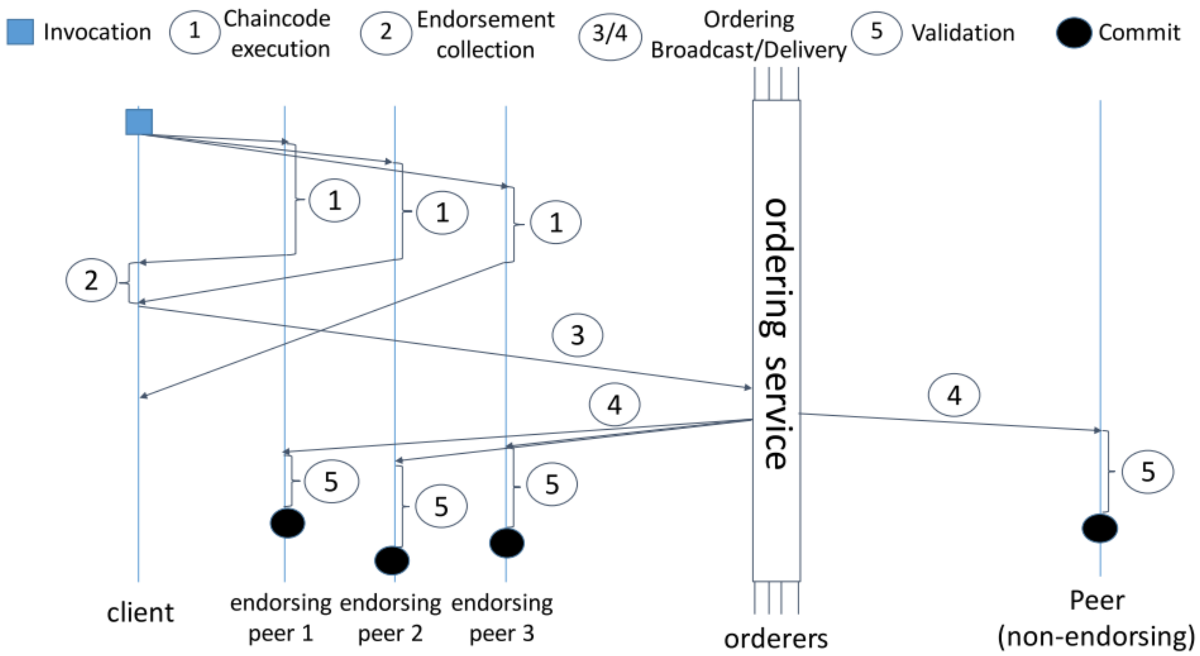

FIGURE 36: FABRIC HIGH LEVEL TRANSACTION FLOW

We will use Hyperledger Composer/Fabric, because it is a good platform for prototyping and quickly setting up a potential blockchain network. It could also potentially be used as the final blockchain network, but the performance and scalability could be an issue, which was also stated by Androulaki et al. (2008).

5.2.2.2 Mendix

Mendix is a low code application Platform as a Service (aPaaS). It can be applied to develop applications quickly, without the use and knowledge of coding languages. This platform will be used to build the web application, because the use of Mendix is common within CAPE Groep and they want to investigated the use of Mendix in combination with blockchain. Mendix offers a model and process flow driven way to develop applications, and the applications can be deployed easily and quickly for testing purposes.

5.2.2.3 eMagiz

eMagiz is a web-based integration Platform as a Service (iPaaS). The eMagiz tool is developed by CAPE in the Mendix platform. With eMagiz, integration solutions are built to connect different systems with each other via a bus. It can validate input and output messages and can map and transform the messages to different outputs for other systems. Because CAPE often builds solutions for customers with both Mendix and eMagiz, standard integration solutions have been made. eMagiz offers a Mendix like model and flow based driven development experience. eMagiz could be a perfect tool to connect Mendix application and the Fabric network, so eMagiz will be used for the communication bus.

5.3 Prototype description

In this section, the prototype itself will be descripted. This will be done by describing each of the systems in the flow of the use of the systems. First the blockchain network, with it's with the user interface and the messages to submit the transactions in the blockchain. Then the Mendix application, with the user interface and the messages to submit the transactions in the blockchain. Then the last part, the eMagiz bus. The Mendix application will send messages to the bus, which then has to send it to the fabric endorser, to submit a transaction. Last the general overview of the prototype will be given and the flow of the messages.

5.3.1.1 Blockchain

For the blockchain network, Hyperledger Composer is used. It is started by creating a domain model in which you define the participant is the network, the assets, the transactions that can be done on these assets and events that take place during these transactions. The goal of the prototype is to validate the architecture, so it was kept simple. Figure 37 shows the domain model.

```
namespace org.thesis.network

asset Invoice identified by invoiceID {
    o String invoiceID
    o DateTime creationDate
    o DateTime paymentDate
    o Double totalAmount
    --> Party buyer
    --> Party supplier
}

participant Party identified by partyID {
    o String partyID
    o String name
}

transaction EarlyPaymentTransaction {
  --> Invoice invoice
  o Double discountPercentage
  o Double newAmount
  o DateTime newPaymentDate
}

event EarlyPaymentEvent {
    --> Invoice invoice
    o Double newAmount
    o Double oldAmount
    o DateTime newPaymentDate
    o DateTime oldPaymentDate
}
```

FIGURE 37: DOMAIN MODEL - BLOCKCHAIN NETWORK

Invoices shared by parties as buyer and supplier. An early payment transaction can be executed to change the discount percentage, the total amount that needs to be payed and the payment date. When such a transaction is executed, the early payment event stores extra information, like the old total amount to be payed and the old payment date.

To execute the transaction, the chain code behind the transaction needs to be defined. This is a small function that states what should be done. Figure 38 shows the chain codes logic. It defines what changes have to be made to the invoice and it commits the transaction, with the event.

```
/**
 * Track the trade of a commodity from one trader to another
 * @param {org.thesis.network.EarlyPaymentTransaction} tx - The early payent transaction instance
 * @transaction
 */
async function earlyPaymentTransaction(tx) {
    //Save old value and date
    let oldAmount = tx.invoice.totalAmount;
    let oldPaymentDate = tx.invoice.paymentDate;

    //Update the invoice values
    tx.invoice.totalAmount = tx.newAmount;
    tx.invoice.paymentDate = tx.newPaymentDate;

    //Get the invoice registry
    let invoiceRegistry = await getAssetRegistry('org.thesis.network.Invoice');

    //Update the invoice
    await invoiceRegistry.update(tx.invoice);

    //Emit an event for the modified invoice
    let event = getFactory().newEvent('org.thesis.network', 'EarlyPaymentEvent');
    event.invoice = tx.invoice;
    event.oldAmount = oldAmount;
    event.newAmount = tx.newAmount;
    event.oldPaymentDate = oldPaymentDate;
    event.newPaymentDate = tx.newPaymentDate;
    emit(event);
}
```

FIGURE 38: TRANSACTION LOGIC - BLOCKCHAIN NETWORK

With the domain model and chain code defined, Composer can create a Fabric network. An administrator has to be defined and other participants can be given access to the network to submit transactions. This is done via an automatically generated API. With this API, new invoice and participants can be entered and then transactions can be done on these invoices. But to not bother the users of the network with difficult post and get calls to the API, an application is made that can talk with the API.

5.3.1.2 Mendix application

The Mendix application should give the users an interface to view see their own invoices, both the invoices that they have to pay to their suppliers and the invoices that have to be payed to them by their buyers. First a domain model was defined to support these functionalities, which can be seen in Figure 39.

Here parties can be created, which is linked to a user account to log in to the application. Parties can be linked to invoices, which have an ID, creation date, payment date and total amount. When an early payment transaction is made, it will be submitted to the blockchain and discounted invoice is created. The other entities are used to submit transactions to the blockchain and to gather data from the blockchain.

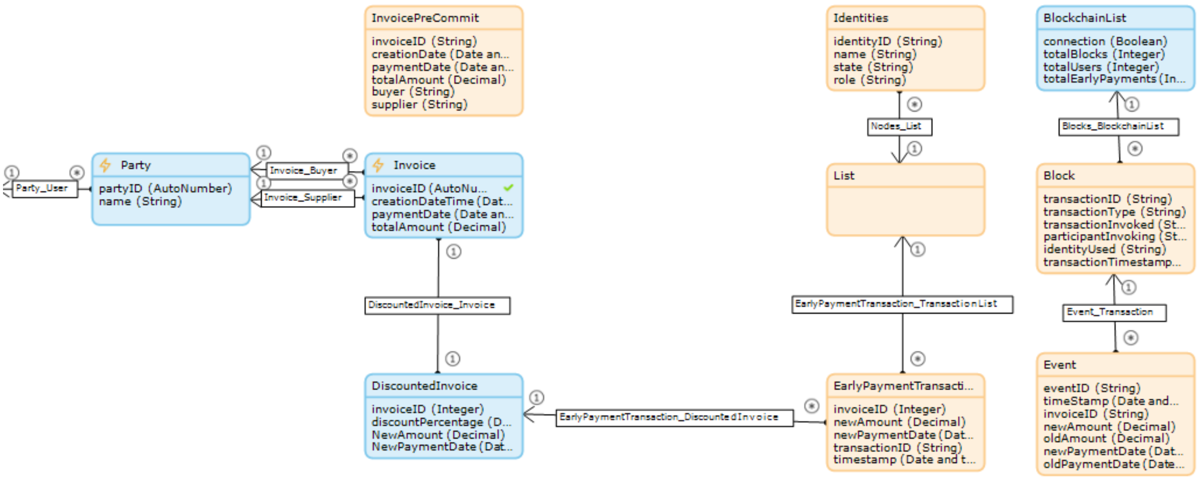

FIGURE 39: DOMAIN MODEL - MENDIX APPLICATION

The process of submitting a transaction to the blockchain goes through multiple microflows, but the main submit to the blockchain is the microflow in Figure 40. The pre-created discounted invoice is send to a web service, that is consumed by an eMagiz connector. eMagiz has a standard connector for Mendix, which makes it easy to send requests to eMagiz. The data in de message that is send is show below. When a new party or invoice is created in the Mendix application, the party or invoice is also submitted to the blockchain with the same process.

<DiscountedInvoice>
<invoice>resource:org.thesis.network.Invoice#2</invoice>
<discountPercentage>2</discountPercentage>
<newAmount>14067.9</newAmount>
<newPaymentDate>2019-01-30T23:00:00.000Z</newPaymentDate>
</DiscountedInvoice>

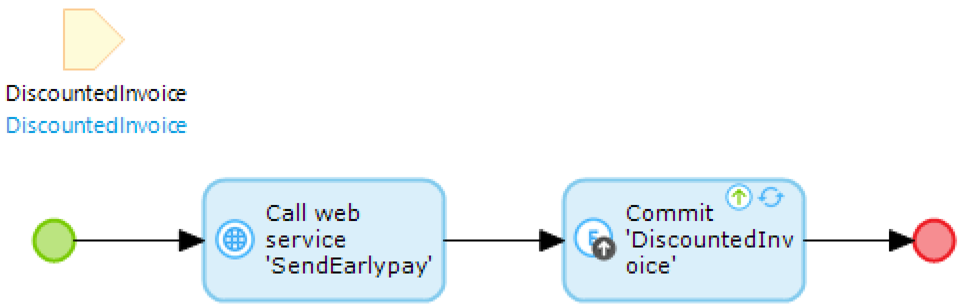

FIGURE 40: SUBMIT TRANSACTION - MENDIX APPLICATION

When an early payment transaction is submitted, the users can see the details of the transaction. The information of the transaction is then retrieved from the blockchain via the API and shown in a screen. An example is shown in Figure 41. Administrators have a bigger dashboard with more information to manage the blockchain. All this information can be gathered from the API. And as said all these request are send to eMagiz and then eMagiz retrieves the information from the blockchain and sends it back to the Mendix application.

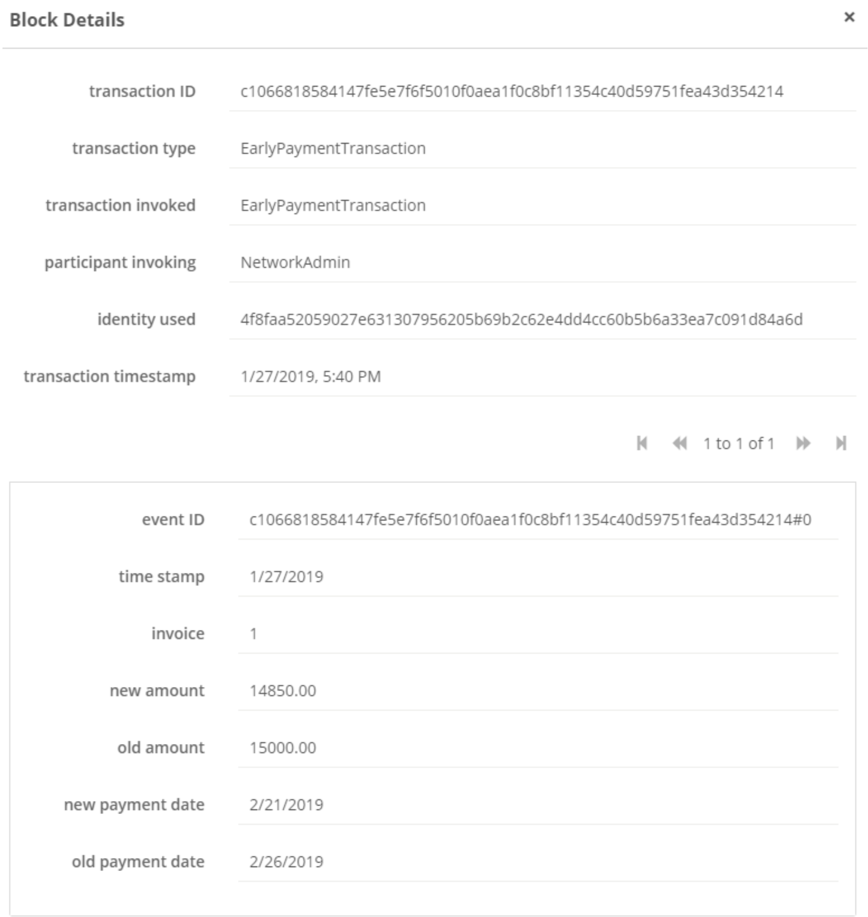

FIGURE 41: TRANSACTION DETAILS - MENDIX APPLICATION

5.3.1.3 eMagiz bus

The eMagiz bus is responsible for the communication between the blockchain and the web application. For now the bus connects two systems, the web application and the blockchain API. All the requests are initiated by the web application. Six different request can be made. The Blocks request is a synchronous call that requests all the blocks that are in the blockchain. It will receive an answer from the blockchain. The same goes for EarlyPayments, that retrieves all the early payment transactions, and Identities, that retrieves all the nodes in the network and their information. EarlyPayment, Invoice and Party are asynchronous request, request that don't receive a response unless something goes wrong, and they submit a new early payment transaction, invoice or party to the blockchain. Figure 42 shows an overview of the bus.

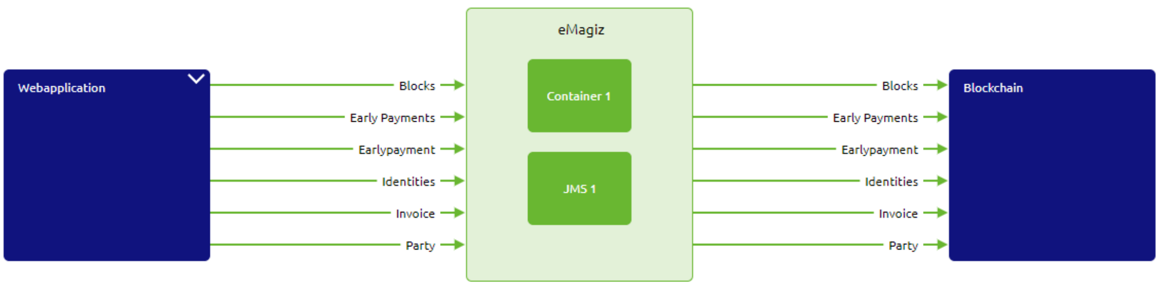

FIGURE 42: OVERVIEW - EMAGIZ BUS

To send the message correctly from one system to another, eMagiz uses a canonical data model, similar to the domain model for Mendix. In our bus, the domain models are the same, because that is also the data the blockchain API wants to receive. Figure 43 shows the domain model.

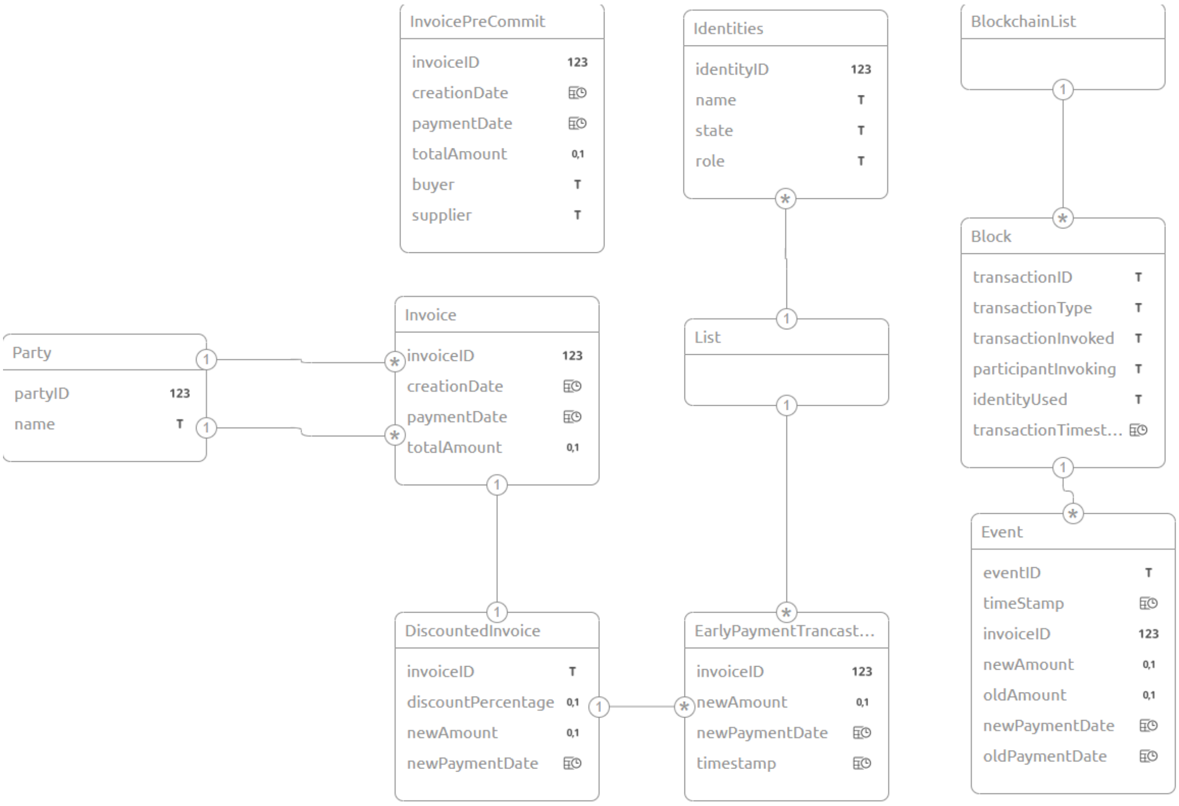

FIGURE 43: DOMAIN MODEL - EMAGIZ BUS

The messages that the bus receives from the web application are received in XML (see section 5.3.1.2) are mapped to the canonical data model. The message is then mapped to system message requested by the blockchain API. The data is the same, but it has to be in JSON instead of XML. eMagiz has transformation tools to format the data to the requested format. Below a message is shown to submit a new early payment transaction to the blockchain. eMagiz also adds the headers required by the blockchain API.


```
{
    "invoice":"resource:org.thesis.network.Invoice#3",
    "discountPercentage":"3",
    "newAmount":"14692.59",
    "newPaymentDate":"2019-02-04T23:00:00.000Z"
}
```


5.4 Conclusion

In the first section of this chapter, we successfully migrated the current enterprise architecture of the Broker to the desired new target enterprise architecture, using the reference architecture. From that we can conclude that it is possible to use the reference architecture to create an enterprise architecture for a blockchain based transaction processing system.

The next topic in this chapter was the development of the prototype. A Mendix application that represented the application of the Broker was developed with success, although the scope was small. Also, a messaging bus was created using Emagiz, also with success. Last a blockchain network was setup that could record the transaction information of early payments of invoices. This was also a success, but in the validation it showed a big limitation. This will be discussed in the next chapter.

6 Evaluation

In this chapter the last evaluation of the refence architecture is done. First the prototype is evaluated. Then the final research questions will be answered. First the advantages of blockchain based transaction systems above the traditional transaction system. Then the advantages of the integration options provided in the reference architecture. Lastly, a new value analysis is conducted to look at the impact of blockchain on the value streams.

6.1 Prototype Evaluation

When looking at the prototype, a small proof of concept of the possible new system of the Broker was created. the objectives that were described in section 5.2.1 were achieved, but not in the form that was desired for the evaluation.

The Mendix web application was a success. A simplified version of the platform of the Broker was recreated. Buyers can “upload” their invoices. When this had happened, suppliers can select an invoice for early payment. They can request a new payment date and therefore the price of the invoice would lower. The dynamic discount algorithm of the Broker was not used, because they want to keep this secret, but that does not influence the outcomes.

When a supplier requests the early payment, the web application sends a message to Emagiz. Emagiz then converts the message to the Canonical Data Model and sends a message to the rest API of the Hyperledger Fabric module. When the data format is correct, the transaction is submitted to the blockchain and the transaction with corresponding event is a new block in the blockchain.

It can be concluded that in theory an implementation of the reference architecture works. But there are some limitations to this prototype, causing us to emphasize that it in theory the implementation can work, but there is need for better testing, with a bigger prototype setup.

The big limitation to this prototype is the fact that a blockchain network deployed, but the network only had one node. The Byzantine-fault tolerant consensus protocol was used, which has been proven to work, but due to limitation of having one node in the network the validation is not reliable. Setting up multiple nodes in the network was not possible, due to lack of resources. Therefore no realistic case testing could be done.

Also performance wise, no realistic case could be tested. All the applications were run on a single machine, the blockchain on a virtual machine using virtual box. Therefore the communication between the Mendix application and the eMagiz connector were done locally. The communication with the API did go over the internet, but on a local network. No real testing could be done, which does impact the conclusions we can derive from this prototype.

It was proven that, in theory, a blockchain based transaction system using this enterprise architecture, based on the reference architecture, can be build. But nothing can be said about the performance or scalability of the system, which are two of the major issues existing blockchain systems. It would also have been interesting to see if the enterprise service bus would influence the performance, but this could not be tested.

6.2 Advantage blockchain implementation

In this section research question five will be answered:

> *SQ5: What are the advantages of a blockchain based transaction system, over a traditional transaction system?*

To answer this question, first we refer back to the literature study. Traditional transaction processing systems are well established and are widely used. They record every transaction between parties, review the transaction and only submit the transaction when it is correct. These standards can be implemented by the developers. They also work fast; the systems can process large amounts of transactions in a short time. All the transactions are stored in a database, to be recalled and reviewed at any given time. But this is done by one or multiple parties separately.

This is where blockchain has a major advantage over the traditional systems. Blockchain is a ledger, that records all the transactions and each participant has a copy of this ledger. By using a consensus algorithm, the transactions are verified and can be approved or denied from the ledger. With good consensus protocols, participants can be withhold to change transactions. If a transaction is changed, the chain is broken, and other participants will notice the fraud. Blockchain can create trust between parties, without the need of a trusted third party, that uses a traditional transaction system to record all the transactions.

But by sharing the blockchain ledger with each other, every participant has access to all the data. There are ways to avoid this, which are outlined in the research of Leng et al. (2018). But Lemieux (2016) shows that there are still uncertainties about the blockchain technology. The blockchain can guarantee a valid audit trail, but it needs a good consensus protocol to achieve this. Proof of Work is the most used algorithm, but it vulnerable to a 51% attack, which is dangerous for small blockchains. This also degrades the scalability of the blockchain system.

6.3 Advantages of communication layer

In the reference architecture the advise of using an enterprise service bus, or messaging platform is given, which brings us to research question six:

> *SQ6: What are the advantages of the integration possibilities in this reference architecture, over the existing blockchain systems?*

By implementing an enterprise service bus, or messaging platform, between the blockchain and web application, a more modular system is created. When the web application needs to be replaced by a totally new system, only the integration with the ESB has to be replaced. The same goes for the blockchain models. When something changes, there is no need to rebuild the web application, only the date transformations in the ESB has to be changed. These are the direct advantages when only using one application with a blockchain. But when this is the case, we don't recommend using an ESB.

An ESB becomes interesting when you are integrating with more than three systems. In the case of the Broker, it is very useful to build integrations with the enterprise systems of their users. This is especially useful to ease the effort of the buyers to change their invoice information, by doing this automatically when a new transaction is approved by the blockchain network. Because the confirmation comes directly from the blockchain, the enterprise service bus could send messages to all the enterprise systems of the users involved.

6.4 Impact of the target enterprise architecture

This section will discuss the impact that the new target enterprise architecture of the Broker could have on the broker. Chapter 5 shows how the current architecture was migrated to the target architecture and what the impact of the reference architecture had on the current architecture. In this section the focus will lie on the impact of the target architecture on the value analysis from section 3.4.

6.4.1 Current situation with blockchain

Figure 44 shows the impact that the new architecture could have on the value streams. This could be surprising, because the Broker is not an actor anymore. The philosophy and vision of blockchain is to cut out the trusted third party, which in this case is the Broker. That is how Bitcoin became big as there was no need for bank to transfer money.

The Broker acts as an intermediary between the buyer and the supplier. He stores the data and records all the transactions, to avoid fraud. Blockchain could provide the same service as a technology.

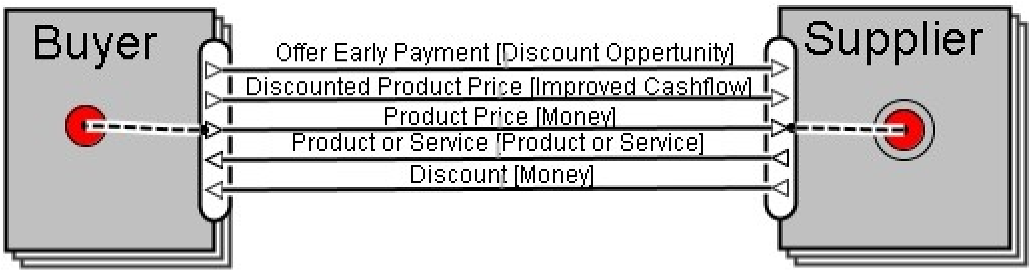

FIGURE 44: E3-VALUE MODEL CURRENT SITUATION WITH BLOCKCHAIN

There is a big chance that the impact will not be this drastic. A value that is not mentioned is the development and maintenance of the application. This is done by the Broker. They created the platform in which they create value for the other actors. But in theory, if one or multiple actors want to take on this effort of developing the same platform using blockchain, they do not need a third party like the Broker, which is beneficial for the suppliers, because they will not have to pay the service fee to the Broker.

6.4.2 Target situation with blockchain

In the case of the investors being implemented, it turns out to be the same, which is shown in Figure 45. Also, in this case, blockchain could replace the broker entirely. And in this case, it is more likely to happen. The investors need the Broker in the current situation without blockchain. The Broker guarantees that there is no double spending, suppliers that sell their invoices to multiple investors. With blockchain this is not possible, within the system. Of course suppliers could go outside the system to sell their invoices, but that problem also exists with the Broker as trusted third party. But why is it more likely that the Broker will be cut out in the case of investors joining the environment?

This is because there are more actors who could be willing to take the responsibility, or share the responsibility with multiple actors, to develop and maintain the platform. And there are not only more actors involved, but with the introduction of the investors, you invite a rich group of actors. They could be willing to invest money into the system, because they could receive a higher return. Currently the Broker asks a service fee from the suppliers. Investors could offer the suppliers to maintain and develop the system in return for a bigger discount, which would still be lower than the Broker's service fee. However, even though cutting out the broker is possible in theory, it is still not probable.

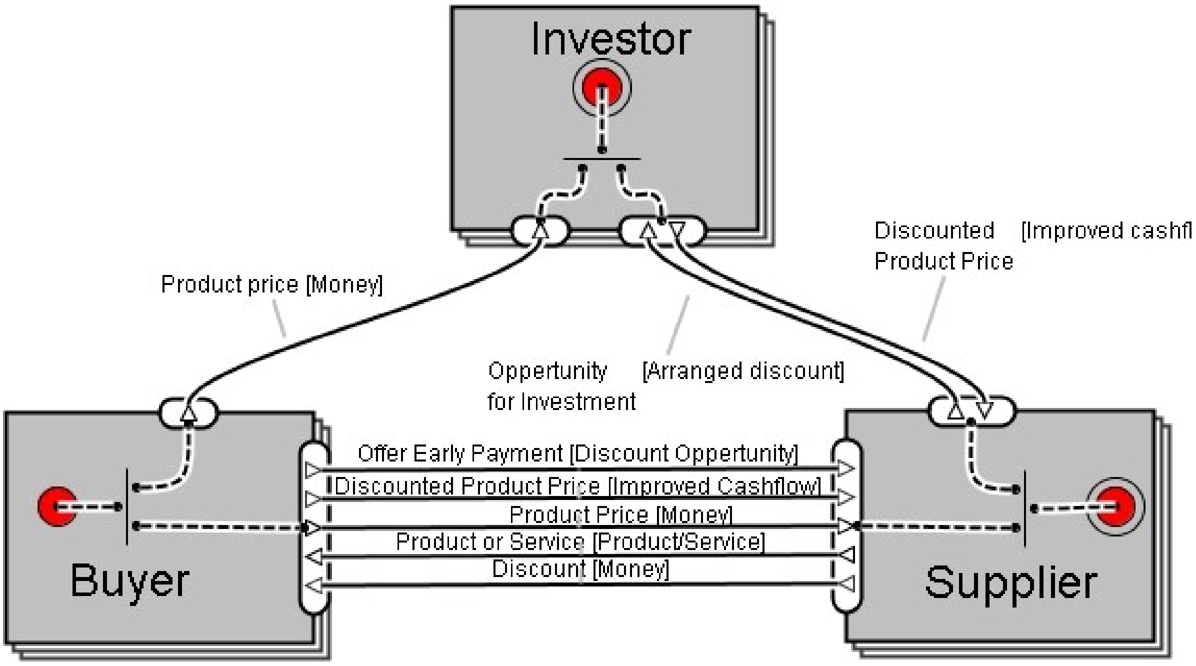

FIGURE 45: E3-VALUE MODEL TARGET SITUATION WITH BLOCKCHAIN

7 Final remarks

In this chapter, the final remarks of this research will be given. First the answers to the research questions will be stated. These answers will then be used to answer the main research question of this thesis and give the final conclusion. Also, the contribution of this research will be discussed and recommendations for future directions in research and practice will be given.

7.1 Conclusions

The main research question of this thesis is:

What reference architecture can best provide a basis for the design of a blockchain based transaction processing system, and integrations to existing applications?

To answer this question, ten research questions were stated. We divided these questions in four groups, which were answered in the previous chapters. First, we will answer these supporting research questions, to come to our final conclusion of this research.

7.1.1 Reference architectures and blockchain

The first set of research questions were the questions, that were answered with the literature study. To create a reference architecture for a blockchain based transaction processing system, we searched the literature for reference architectures for TP systems. This gave us the answer to the first research questions

SQ1a: What are the common reference architectures for transaction processing systems?

Although literature about TP systems seems to be outdated, we found the commonly accepted book of Bernstein and Newcomer (2009). This book gives an overview of the transaction processing systems. But our main conclusion is that TP systems are so common and diverse, that they are difficult to categorize.

Blockchain is mainly known for its utilisation in cryptocurrencies. By using blockchain, people have created currencies that do not require banks to be transferred. Through the technology, people do not need a trusted third party. The exploration of the application of blockchain in other systems is a hot topic. We want to explore the possibility of using blockchain in a TP system, to process and record transactions to create a valid audit trail. Therefore we looked at the already existing applications.

SQ1b: How is blockchain utilised in auditing systems?

When looking into the use of blockchain technology in auditing systems, there was no clear answer. There is a stream of studies that take the advantages of blockchain as truth and base their new applications for the blockchain technology on it. It was refreshing to see a thorough analysis by Lemieux (2016), who compared standards for auditing of data with the possibilities of blockchain. She identified several threats to blockchain systems, which could bring new problems, like privacy. But she encourages future researchers to keep exploring, because the technology has great potential.

To create a reference architecture for a blockchain based TP system, we wanted to combine the reference architecture for TP systems with the architectures of blockchain applications. Therefore we looked at the existing literature about architectures for blockchain applications.

SQ1c: What reference architectures are used for blockchain applications?

Because the blockchain technology is only being used outside of cryptocurrencies for a few years now, we could not find a clear architecture. What can be seen is that blockchain can be a great technology for data sharing between organisations/individuals, but data privacy can still be a problem. Leng et al. (2018) created a new and innovative architecture, where they use two separate blockchains. But it is still hard to choose a good architecture for the setup of a blockchain based system. Xu et al. created a framework that could assist in choosing the right architecture (Xu et al., 2017). But there is still a lot of room for improvement. Therefore we want to propose a new reference architecture in this research that can be used for multiple blockchain applications.

Lastly, for the development of the prototype, a review of the existing blockchain development platforms was conducted.

SQ1d: Which blockchain platforms could be used to create blockchain applications?

Although a lot of the development platforms and tools are still in development themselves, there is a big pool to choose from. We put them next to each other and evaluated them. Ethereum is a good choice, but it is not perfect for building private blockchains, interact and share data between organisations. Mainly because all the transactions cost Ether. Quorum is the Ethereum based version for enterprise-ready blockchain applications. But the problem with Quorum is, that the development has slowed down, and it is questionable if it will succeed. Therefore in our opinion, Hyperledger Fabric is the best choice if you want to implement a new blockchain technology, at this moment. The technology is developing rapidly, so in a few months, the landscape could be totally different.

7.1.2 Case analysis

An analysis of the current situation of the Broker was done to use the case for the validation of the reference architecture. The reference architecture should be able to assist in changes the Broker has to make to get from the current situation to the target situation. Therefore we looked at the current business processes of the Broker and we did a value analysis. This is captured in research question two.

SQ2: What are the business processes of the Broker's service and which values does it create and request?

Section 3.3 described the processes in which the Broker is involved. Figure 19 and Figure 20 show these processes. Inviting investors into the application will complicate the process, but mainly for the Broker himself, although the main downside is for the buyer. He can give permission to the investor, to offer an early payment to the suppliers. But if this succeeds the buyer will have changed his payment information on the invoice, because he will have to pay the investors instead of the suppliers. This also reflects in the value analysis.

Section 3.4 describes this value analysis. Two e3-value models were created, to gain insight in the value gains of the actors. Figure 21 shows that the Broker creates value for all actors by offering the reverse factoring service. The cashflow of the suppliers will improve, the buyers will receive a discount on their invoices and the Broker will receive a service fee from the suppliers. When the implementation of investors succeeds, these values will improve, except for the buyers. Figure 22 shows the impact investors will have on the environment. They will gain value with the return on their investment, the cashflow of the suppliers will improve even more and therefore the Broker will receive more money. But the buyers do not gain from this implementation, instead they will have to put in more effort for the other actors to gain value. It should therefore be important to make this process as easy as possible for the buyers, so the Broker won't have to request too much effort from the buyers.

As said, for the reference architecture to be able to assist in the transformation from the current to the target architecture, we had to model the current architecture of the Broker.

> *SQ3: What is the current enterprise architecture of the Broker?*

The process analysis and the information gathered from the interviews was used to create the current architecture of the Broker. This can be found in section 3.5, Figure 46, or in Appendix A: Architectures. The architecture shows the business, application and technology layer. In the business layer, a simplified version of the processes is modelled and how they serve the involved parties, buyers and suppliers. The application layer serves the business layer, by providing the services and functions to assist the business processes. The technology layer serves the application layer. In the current situation, there is no integration between the application of the Broker and the enterprise systems of the buyers and suppliers.

7.1.3 Reference architecture

The main research question is to find the best reference architecture for a blockchain based transaction system. From the literature study, we found that there was no satisfying reference architecture, so we decided to create one. We propose a new reference architecture for blockchain applications, focussed on transaction processing. We therefore asked ourselves the fourth research questions.

> *SQ4: What architecture can be used for the design of a blockchain based transaction processing system?*

To create this architecture, we used TOGAF's ADM method in combination with the modelling language Archimate. We took the requirements of the Broker, that could be generalized, into account. By looking at the blockchain technology and other reference architectures, we came up with the reference architecture that can be found in section 4.1, Figure 29, and Appendix A: Architectures.

In our opinion, the reference architecture could be used for every business layer if they need a blockchain system. One requirement is that there should be a group of users, who can participate in the blockchain network, otherwise the consensus building will not work (properly).

We separated the application layer for use of the system, with the blockchain layer. The blockchain layer will be present at every actor. The same goes for the actor technology layer. The blockchain layer consists of an algorithm engine and a blockchain client. These two parts should be separated, because their functions differ. The blockchain client is used to access the blockchain. This can be hosted at every user, but that is not necessary. The algorithm engine will be present at every user. It uses a rule set to get is consensus algorithm. This can defer from proof of work, to proof of stake, which makes this architecture very flexible.

The communication with the blockchain and the application will go through an enterprise service bus or messaging platform. By doing this, both the application and blockchain become modular and easily replaceable. Also, one blockchain network can be used by multiple applications and the implementation can be quick. The implementation of the communication layer also brings the advantage of easy integrations with other systems, like enterprise systems of users.

By providing this new reference architecture for blockchain applications, which has multiple advantages, we believe we bring more clarity in the world of blockchain development.

7.1.4 Validation

The last group of research questions support the validation of the architecture. They review certain parts of the architecture and the impact it will have in the case of the Broker. With the prototype we proved that in theory you can build a blockchain based transaction system using an enterprise architecture, based on the reference architecture. But we cannot say anything about the performance or scalability of the system, which are two of the major issues existing blockchain systems. Due to the limitations of the systems and lack of resources no real testing, apart from unit testing, has been done. Therefore the reference architecture will have to undergo more validation for it to be used by others. But the last research questions can be answered, starting with research question five:

> *SQ5: What are the advantages of a blockchain based transaction system, over a traditional transaction system?*

One of the major advantages of blockchain is the distributed data. Blockchain is a ledger, that records all the transactions and each participant has a copy of this ledger. If a transaction is changed, the blockchain is broken and other participants will notice the fraud. With blockchain you can replace the trusted third party, that uses a traditional transaction system to record all the transactions.

But by sharing all the data you give up privacy. Leng et al. (2018) found a way around this, but this added a lot of complexity. Lemieux (2016) showed that there are still uncertainties about the blockchain technology. It can be concluded that blockchain has advantages over the traditional processing systems, but also some disadvantages. The scalability of blockchain remains an issue, together with small concern for the valid audit trail. Blockchain is mainly useful for small networks, were the participants use an equal consensus protocol, who don't want to involve a trusted third party to keep the records for them. In that area, blockchain has major advantages of the traditional systems.

In the reference architecture we also advise users to use an enterprise service bus, or messaging platform. We looked at the advantages of doing this.

> *SQ6: What are the advantages of the integration possibilities in this reference architecture, over the existing blockchain systems?*

The implementation of enterprise service bus, or messaging platform, in the reference architecture, gives a new kind of modular architecture. When the web application needs to be replaced by a totally new system, you only need to replace the integration with the ESB. The same goes for the blockchain models. An ESB becomes even more interesting when you are integrating with more than three systems. In the case of the Broker, it is very useful to build integrations with the enterprise systems of their users. This is especially useful to ease the effort of the buyers to change their invoice information, by doing this automatically when a new transaction is approved by the blockchain network. Because the confirmation comes directly from the blockchain, the enterprise service bus could send messages to all the enterprise systems of the users involved.

Therefore we advise developers of new blockchain applications, to look at your application and long-term plans using this architecture. When you see the possibility of multiple integrations with other systems, for example enterprise systems, but also more applications using the same blockchain, use an enterprise service bus at the start. This will save a lot of time on the long term.

Now that we looked at the validation of the reference architecture, we also wanted to look at the influence of this reference on the current situation.

> *SQ7: How could the implementation of the new reference architecture, influence the business of the Broker?*

By applying the reference architecture to the current architecture, it shows that a lot of complexity is added. The transaction processing of the current system will have to change completely. Also, an enterprise service bus will have to be implemented, but this will be very useful. By using an ESB, users will not have to manually change their invoice information, but this could be handled automatically. This will improve the user experience of using the Broker's service.

But should the Broker implement the blockchain based transaction processing? Therefore we created a new version of the e3-value models. We looked at what values blockchain could bring to the system and it turns out that blockchain could in theory replace the Broker's position. In the current situation, the Broker acts as the trusted third party, who automatically handles the negotiations and stores the transaction information. It guarantees that suppliers cannot request early payments for the same invoice multiple times, within the system. Blockchain could also fulfil these roles. Blockchain was designed to cut out third parties and create a network of participants, who can create transactions between each other. When the Broker implements blockchain, the Broker will lose its value creations to the blockchain. The actors will now receive the values from each other, because of the blockchain system, so why should they still create value for the Broker?

Although it takes a lot of effort and it is unlikely to happen, in theory the actors in the environment could set up their own blockchain based system, without the Broker, to receive their desired values, without offering values to the Broker. Therefore we advise the Broker not to implement blockchain in their system. They will add a lot of complexity, without gaining extra benefits from it. They could achieve goals and values, by using their traditional system. We do recommend the Broker to implement an ESB for the user experience of the users, especially for the buyers.

7.1.5 Final conclusion

With the answers to all the research questions, it is now possible to answer the main research question:

What reference architecture can best provide a basis for the design of a blockchain based transaction processing system, and integrations to existing applications?

Looking at the reference architecture, it is our final conclusion that our reference architecture provides a good basis for the design of a blockchain based transaction processing system. It could be used in different business fields and can provide an open ledger, shared between the actors in the business ecosystem, without the use of a trusted third party. Integration with the enterprise systems of the users is simplified by using an enterprise service bus, which is also used to communicate with the blockchain. By adding the communication layer, we developed a reference architecture for a modular use of a blockchain application. But we can not conclude that this is the best reference architecture, due to the lack of testing. This is our main recommendation for future research.

7.2 Contributions

This research contributes to the science by providing the new reference architecture for blockchain based transaction systems, with easy integration options. This reference architecture could support future research into the application of blockchain in different sectors. We also investigated the influence blockchain could have on the value streams within a business ecosystem. This could also be used in future research.

The contribution to practice lies in the advice we can give cases such as the Broker case. Blockchain is technology with lots of potential, but only in specific cases. We would advise cases like the Broker, not to use blockchain. Blockchain should sometimes be regarded as a threat instead of an opportunity. With our prototype we contributed to the knowledge of CAPE Groep, by demonstrating how Mendix and Emagiz could communicate with a blockchain. Emagiz could benefit from building integrations with blockchain platforms, to make quick integrations between applications that want to use blockchain and enterprise systems.

7.3 Recommendations

The result of this research is a reference architecture supplemented with a prototype. The focus on blockchain and integrations, and the limitations to the prototype provide some recommendations for future research.

Firstly, the prototype has not been tested with a large blockchain network. We could not identify the scalability of the prototype or the transaction speed. Therefore it is also difficult to recommend the best consensus protocol for specific situations. To further validate this reference architecture, a large blockchain network needs to be set up and all those participants should participate in various consensus algorithms. This will give a better insight in the speed and scalability of this architecture.

Secondly, coherent with the previous recommendation, the research into blockchain should continue. It is shown that proof of work has its flaws, but it is still the most used consensus algorithm. There are new upcoming developments, but these are yet to be completed and tested.

The last recommendation for future research is for Archimate. Archimate was used to create the architectures. We presented a way to model blockchain in Archimate, but it could be an interesting topic to see if Archimate could be improved. Now the blockchain network was modelled using business and application interactions, but we think that with some new elements, blockchain could be made even more clear in a reference architecture for blockchain based transaction systems.

References

Akkermans, J. M., & Gordijn, J. (2003). Value-based requirements engineering: exploring innovative e-commerce ideas. *Requirements Engineering*. https://doi.org/10.1007/s00766-003-0169-x

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., … Yellick, J. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceeding EuroSys '18 Proceedings of the Thirteenth Eurosys Conference*. https://doi.org/10.1145/3190508.3190538

Angelov, S., Grefen, P., & Greefhorst, D. (2009). A classification of software reference architectures: Analyzing their success and effectiveness. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA 2009*. https://doi.org/10.1109/WICSA.2009.5290800

Bayer, D., Haber, S., & Stornetta, W. S. (1993). Improving the Efficiency and Reliability of Digital Time-Stamping. In *Sequences II* (pp. 329–334). https://doi.org/10.1007/978-1-4613-9323-8_24

Bernard, H. R. (2006). Interviewing: Unstructured and semistructured. In *Research Methods in Anthropology*. https://doi.org/10.1525/aa.2000.102.1.183

Bernstein, P., & Newcomer, E. (2009). *Principles of Transaction Processing*. *Principles of Transaction Processing* (2nd ed.). San Francisco, CA: Morgan Kaufmann. https://doi.org/10.1016/B978-1-55860-623-4.X0001-7

Cohen, D., & Crabtree, B. (2006). Semi-structured Interviews Recording Semi-Structured interviews. *Qualitative Research Guidelines Project*. https://doi.org/10.1021/la503432x

Coulibaly, B., Sapriza, H., & Zlate, A. (2013). Financial frictions, trade credit, and the 2008-09 global financial crisis. *International Review of Economics and Finance*, *26*, 25–38. https://doi.org/10.1016/j.iref.2012.08.006

Demica. (2007). *Steady Supply - The Growing Role of Supply Chain Finance in a Changing World*. *Demica*. https://doi.org/10.1016/B978-0-12-397037-4.00003-X

Dienelt, J. (2016). *Understanding Ethereum*.

Garcia-Appendini, E., & Montoriol-Garriga, J. (2013). Firms as liquidity providers: Evidence from the 2007-2008 financial crisis. *Journal of Financial Economics*, *109*(1), 272–291. https://doi.org/10.1016/j.jfineco.2013.02.010

Group, T. O. (2017). ArchiMate® 3.0.1 Specification. In *Open Group Standard*. Retrieved from http://pubs.opengroup.org/architecture/archimate3-doc/toc.html

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, *3*(2), 99–111. https://doi.org/10.1007/BF00196791

Hearn, M. (2016). Corda: A distributed ledger. *Whitepaper*, 1–56.

Hofmann, E. (2005). Supply Chain Finance: some conceptual insights. *Logistik Management - Innovative Logistikkonzepte*, *80*, 203–214. https://doi.org/10.1016/j.resuscitation.2008.09.013

Hofmann, E. (2011). Natural hedging as a risk prophylaxis and supplier financing instrument in automotive supply chains. *Supply Chain Management*, *16*(2), 128–141. https://doi.org/10.1108/13598541111115374

Huff, J., & Rogers, D. S. (2015). Funding the Organisation through Supply Chain Finance: A Longitudinal Investigation. *Supply Chain Forum*, *16*(3), 4–17. https://doi.org/10.1080/16258312.2015.11716348

Hyperledger Composer - Create business networks and blockchain applications quickly for Hyperledger. (n.d.). Retrieved 19 July 2018, from https://hyperledger.github.io/composer

Hyperledger Fabric. (n.d.). Retrieved 19 July 2018, from https://www.hyperledger.org/projects/fabric

Hyperledger Fabric — Key Concepts. (n.d.). Retrieved 18 July 2018, from http://hyperledger-fabric.readthedocs.io/en/release-1.2/key_concepts.html

Iacob, M., Jonkers, H., Quartel, D., Franken, H., & Van den Berg, H. (2012). *Delivering Enterprise Architecture with TOGAF® and ArchiMate®*.

IEEE. (2000). Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE Std 1471-2000*. https://doi.org/10.1109/IEEESTD.2000.91944

Ivashina, V., & Scharfstein, D. (2010). Loan syndication and credit cycles. In *American Economic Review* (Vol. 100, pp. 57–61). https://doi.org/10.1257/aer.100.2.57

JP Morgan Chase. (2016). Quorum Whitepaper, 1–8. Retrieved from https://www.blocksg.com/single-post/2017/12/27/Quorum-Whitepaper

Kim, H., Lee, J.-N., & Han, J. (2008). The role of IT in the relationship between business ecosystem's healthiness and flagship firm's performance: A conceptual foundation and empirical validation. In *PACIS 2008 - 12th Pacific Asia Conference on Information Systems: Leveraging ICT for Resilient Organisations and Sustainable Growth in the Asia Pacific Region*.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. *Engineering*, *45*(4ve), 1051. https://doi.org/10.1145/1134285.1134500

Lemieux, V. L. (2016). Trusting records: is Blockchain technology the answer? *Records Management Journal*, *26*(2), 110–139. https://doi.org/10.1108/RMJ-12-2015-0042

Leng, K., Bi, Y., Jing, L., Fu, H. C., & Van Nieuwenhuyse, I. (2018). Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, *86*, 641–649. https://doi.org/10.1016/j.future.2018.04.061

Martínez-Sola, C., García-Teruel, P. J., & Martínez-Solano, P. (2017). SMEs access to finance and the value of supplier financing. *Revista Espanola de Financiacion y Contabilidad*, *46*(4), 455–483. https://doi.org/10.1080/02102412.2017.1345196

Moore, J. F. (1993). Predators and prey: a new ecology of competition. *Harvard Business Review*, *71*(3), 75–86.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, 9. https://doi.org/10.1007/s10838-008-9062-0

Niemi, E. (2006). Enterprise architecture benefits: Perceptions from literature and practice. In *International Business Information Management (IBIMA)* (pp. 161–168). https://doi.org/10.1016/j.ncl.2010.12.002

Niemi, E., & Pekkola, S. (2013). Enterprise Architecture quality attributes: A case study. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.1109/HICSS.2013.201

O'Leary, D. E. (2017). Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems. *Intelligent Systems in Accounting, Finance and Management*, *24*(4), 138–147. https://doi.org/10.1002/isaf.1417

Paul, S., Brian, D., Jack, L., David, J., & Peter, K. (2014). Factom White Paper. *Factom*. Retrieved from https://www.factom.com/devs/docs/guide/factom-white-paper-1-0

Peffers, K. E. N., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Peters, G. W., & Panayi, E. (2016). Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. *New Economic Windows*, 239–278. https://doi.org/10.1007/978-3-319-42448-4_13

Popa, V. (2013). The financial supply chain management: A new solution for supply chain resilience. *Amfiteatru Economic*, *15*(33), 140–153.

Quorum - J.P. Morgan. (n.d.). Retrieved 19 July 2018, from https://www.jpmorgan.com/global/Quorum

Roman, D., & Stefano, G. (2016). Towards a reference architecture for trusted data marketplaces: The credit scoring perspective. *Proceedings - 2016 2nd International Conference on Open and Big Data, OBD 2016*, 95–101. https://doi.org/10.1109/OBD.2016.21

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modeling Language Reference Manual Second Edition*. *Zentralblatt für Bakteriologie, Parasitenkunde, Infektionskrankheiten und Hygiene. Erste Abteilung Originale. Reihe A: Medizinische Mikrobiologie und Parasitologie*. https://doi.org/10.1017/CBO9781107415324.004

Schwartz, D., Youngs, N., & Britto, A. (2014). Consensus Whitepaper: A Novel Consensus Algorithm | Ripple. Retrieved from https://ripple.com/consensus-whitepaper/

Shang, K. H., Song, J.-S., & Zipkin, P. H. (2009). Coordination Mechanisms in Decentralized Serial Inventory Systems with Batch Ordering. *Management Science*, *55*(4), 685–695. https://doi.org/10.1287/mnsc.1080.0981

Song, H., Yu, K., & Lu, Q. (2018). Financial service providers and banks' role in helping SMEs to access finance. *International Journal of Physical Distribution and Logistics*

Management, *48*(1), 69–92. https://doi.org/10.1108/IJPDLM-11-2016-0315

Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money ... *Sage Publications, Inc.*, 384. https://doi.org/10.1515/ngs-2017-0002

The First Version of Ethereum's Casper Upgrade Has Been Published - CoinDesk. (2018). Retrieved 18 July 2018, from https://www.coindesk.com/first-version-ethereums-casper-upgrade-published/

Wuttke, D. A., Blome, C. C., Foerstl, K., & Henke, M. (2013). Managing the innovation adoption of supply chain finance - Empirical evidence from six european case studies. *Journal of Business Logistics*, *34*(2), 148–166. https://doi.org/10.1111/jbl.12016

Wuttke, D. A., Blome, C., & Henke, M. (2013). Focusing the financial flow of supply chains: An empirical investigation of financial supply chain management. *International Journal of Production Economics*, *145*(2), 773–789. https://doi.org/10.1016/j.ijpe.2013.05.031

XinXiang, H. (2012). Research of the risks and strategies of SME's supply-chain financing. In *Proceedings of the 2012 2nd International Conference on Business Computing and Global Informatization, BCGIN 2012* (pp. 144–146). https://doi.org/10.1109/BCGIN.2012.44

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., … Rimba, P. (2017). A Taxonomy of Blockchain-Based Systems for Architecture Design. *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, 243–252. https://doi.org/10.1109/ICSA.2017.33

Zyskind, G., Nathan, O., & Pentland, A. S. (2015). Decentralizing privacy: Using blockchain to protect personal data. *Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015*, 180–184. https://doi.org/10.1109/SPW.2015.27

Appendix A: Architectures

A1: Current Architecture

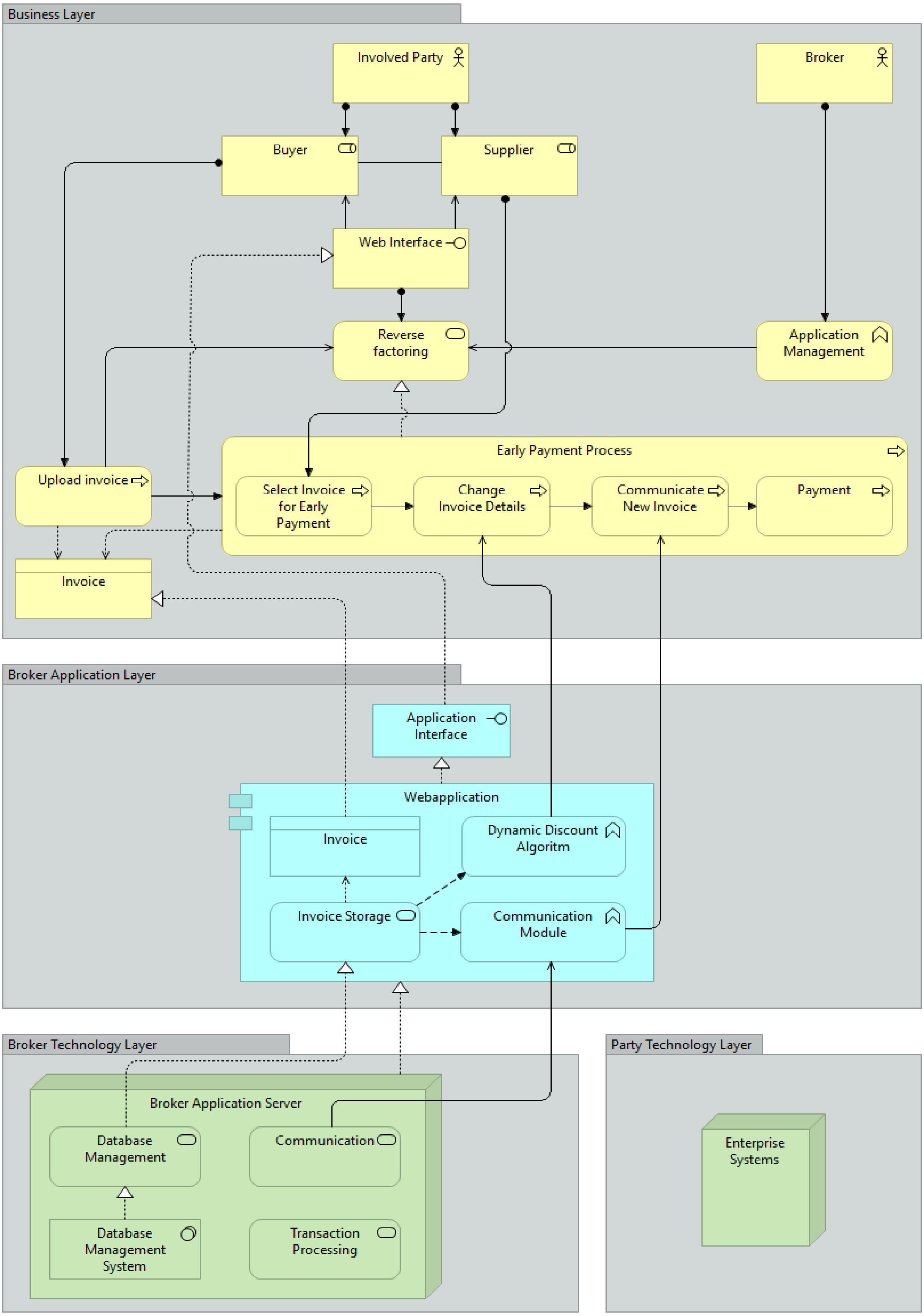

FIGURE 46: CURRENT ARCHITECTURE OF THE BROKER

A2: Target Architecture

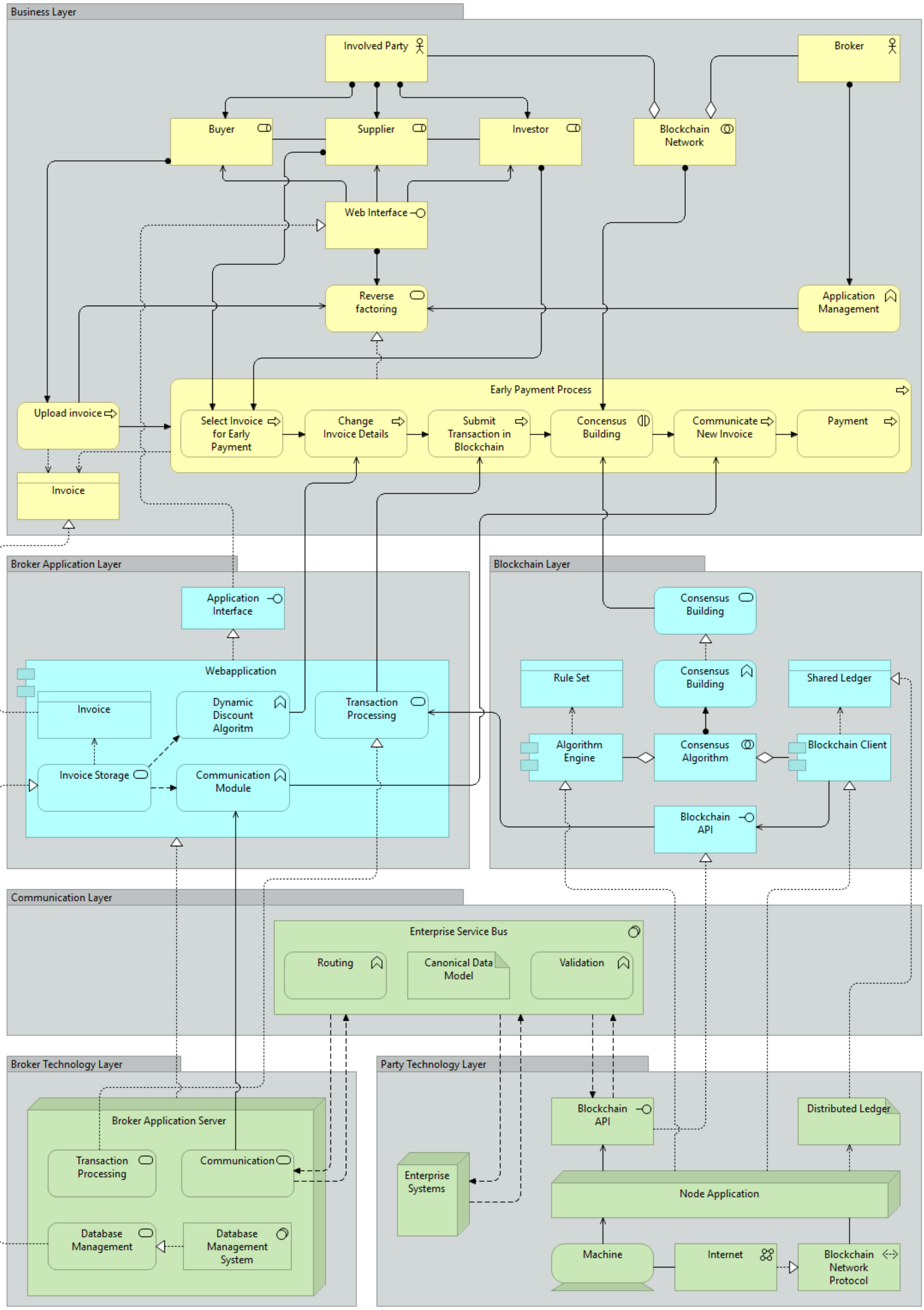

FIGURE 47: TARGET ARCHITECTURE

A3: Reference Architecture

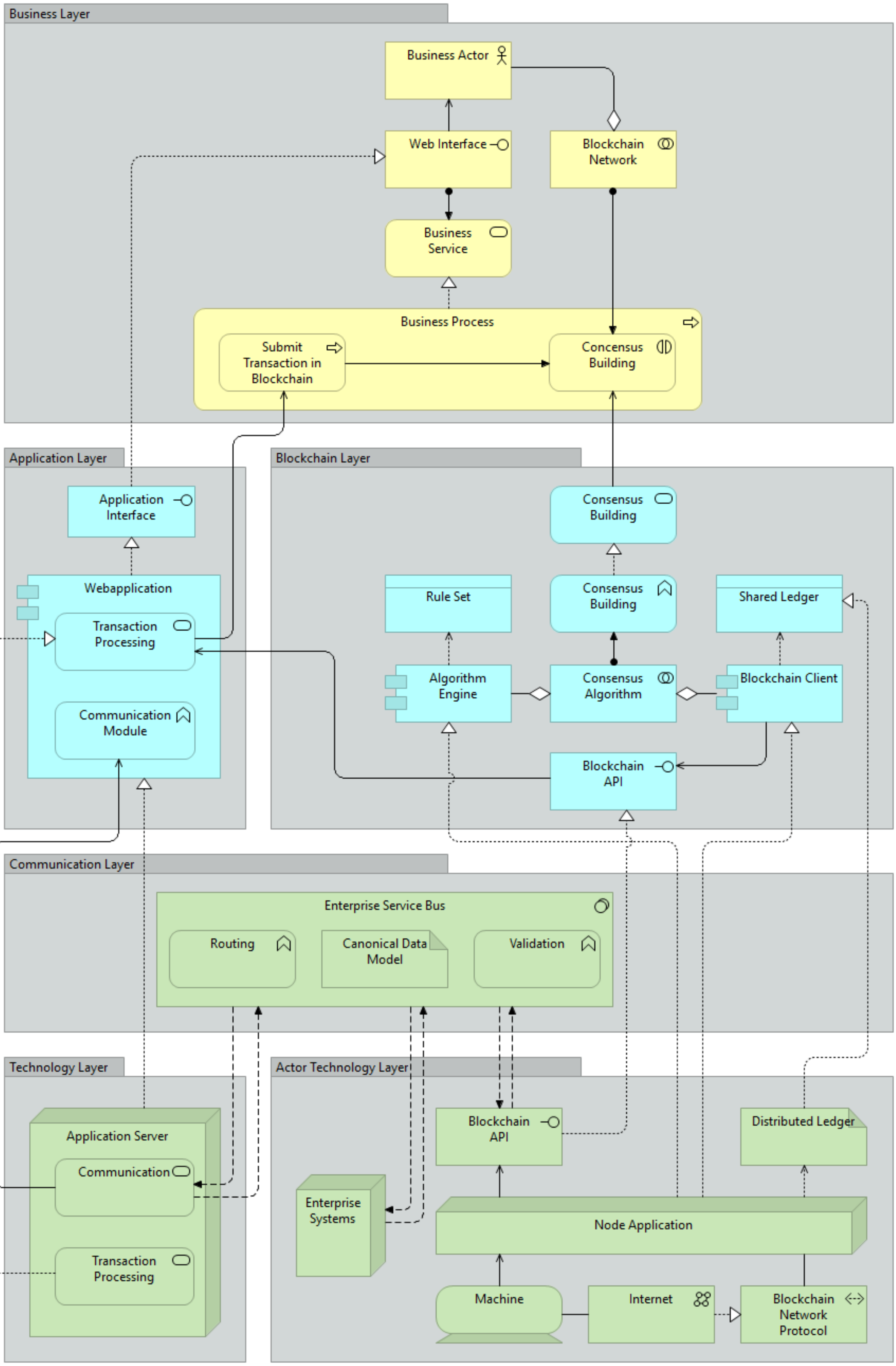

FIGURE 48: REFERENCE ARCHITECTURE