UTRECHT UNIVERSITY

MASTER THESIS

A Decision Support System for Blockchain Platform Selection

Author:
J.R.Q. VERKLEIJ
Studentnumber:
3820106

Supervisor:
Dr. S. JANSEN
Second Supervisor:
S. España Cubillo

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Master's Programme of Business Informatics
Department of Information and Computing Sciences

July 2018

UTRECHT UNIVERSITY

Abstract

Beta Faculty
Department of Information and Computing Sciences

Master of Science

A Decision Support System for Blockchain Platform Selection

by J.R.Q. VERKLEIJ

The blockchain technology is a new innovation with the potential to disrupt the world as we currently know it, despite several limitations and challenges to overcome. One of these challenges for software producing organizations is selecting the right technology for their case. In this research, we haveve identified this selection process as a multi-criteria decision making problem. Based on this we have created a Decision Support System which aids developers during the technology selection process of blockchain platforms. Contemporary solutions to this problem were only rather simplistic decision-trees, which struggle with complexity and adaptations. The novelty of this Decision Support System lies in being a feature-based artifact which incorporates ISO Software Quality Aspects and feature prioritization based on the MoSCoW-technique. These contemporary generic blockchain features have been gathered through nine interviews with blockchain experts. Based on prioritized features as input, the Decision Support System gives a score for feasible solutions (e.g. Ethereum or Hyperledger) as result. This Decision Support System was evaluated in three different case-studies for organization creating blockchain-based solutions. In addition to this, the artifact has been validated by a blockchain-domain expert. The main difficulties and obstacles of this whole research were grounded in the immaturity of the blockchain domain as a whole.

Key words: Blockchain, Technology Selection, Multi-criteria Decision Making, Decision Support System

Acknowledgements

After three years of my master Business Informatics, writing this final section of my master thesis feels surreal. I'm grateful i had the opportunity to perform research the past year on a technology which i genuinely think has the potential to re-shape the world as we know it. However, performing research is never trouble-free and along the way you always encounter obstacles which have to be tackled in order to be successful. I think, in the end this thesis is a satisfying combination between both applying theoretical knowledge and tackling these practical obstacles.

I would like to take this opportunity to thank several people. First of all, my thanks go to my first supervisor, dr. Slinger Jansen from Utrecht University. He provided me with clear and straightforward guidance during the whole process. I would also like to thank my second supervisor Sergio Espana Cubillo, for his feedback on my thesis. Special thanks go to Siamak Farshidi for discussing everything related to the creation of the artifact and his ever-enduring patience with me. Next, i would like to express my gratitude to everyone at ShareCompany BIQH but in particular to Marc Bracher and Bernard Schut. In addition to providing me with daily supervision and a case-study for my research i enjoyed the 8 months having a working place in their office. Hereby, i want to show my gratitude to everyone that participated in an interview with me, assisted me with one of the case-studies or aided me in any other way.

Last but certainly not least, i would especially like to thank my family for providing moral support during brighter and darker times while working on my thesis.

Thank you for reading this thesis, i hope you enjoy it.

Jacco Ronaldo Quirinus Verkleij
IJsselstein, August 2018

Contents

Chapter 1

Introduction

"Whereas most technologies tend to automate workers on the periphery doing menial tasks, blockchains automate away the center. Instead of putting the taxi driver out of a job, blockchain puts Uber out of a job and lets the taxi drivers work with the customer directly"
(Vitalik Buterin, co-founder of the Ethereum blockchain)

The blockchain technology has received a massive increase in attention the last few years. Conceptualized by the release of Bitcoin by Nakamoto (2008), the fundamental technology behind it might rise to even higher peaks than Bitcoin itself. Panetta (2017) has placed the blockchain technology just past the peak in the Gartner hype cycle for emerging technologies for 2017, with another 5 to 10 years estimated for mainstream adoption of the technology.

The development of the blockchain traces back to the endless potential the internet itself provided. One of the possibilities the internet now offers is the transacting between individual users, for example, internet banking. One of the latest developments in this domain is so-called cryptocurrencies. Cryptocurrencies can be defined as digital alternatives to traditional government-issued money (Luther, 2013) or web-based, peer-to-peer payment systems that rely on cryptography (Omri, 2013). The underlying technology which enables these cryptocurrencies is called the blockchain.

The blockchain technology can be best described as a distributed ledger technology that solves the double-spending problem (Geranio, 2016; Nakamoto, 2008). The double-spending problem is a potential flaw in digital cash transactions, in which the same digital value can be spent more than once through duplication or falsification. This would lead to inflation (comparable to non-digital counterfeit money), devalues the currency and damages user-trust. Currently, this double-spending problem is dealt with by trusted third-parties, such as banks, notaries, escrow agents or key distribution centers (KDC). The blockchain technology has the disruptive potential to completely replace these trusted third-parties for solving the double-spending problem. Hence the quote showed at the top of the page from Vitalik Buterin (Tapscott and Tapscott, 2016). In that specific example, the obsolete trusted-third party would be the intermediary company Uber, which at present brings together the supply and demand sides of taxi-services.

Blockchain innovations aim to make these trusted-third parties obsolete through reaching consensus about a transaction with the majority of the nodes in a network. The most notable consensus-mechanism is called 'Proof-of-Work'(PoW), which is used in the well known Bitcoin network (Nakamoto, 2008). When transactions are verified they are stored along with other transactions on the same network in a block. These blocks refer to other blocks that have been verified before. Since these blocks are linked to each other they form a chain of blocks which store transactions. Hence the name, blockchain. The longest chain of these blocks will be seen as the general consensus. Due to the fundamental architecture of the blockchain, transactions are

irreversible once verified. In addition to that, a consensus about new transactions will only be achieved through the proof-of-work mechanism if they are correct according to previously verified blocks in the chain. One of the aims of this research is to delve further into technical details and features of the blockchain technology.

Although the initial domain of application of the blockchain was the payment sector, the blockchain technology has the potential to disrupt a tremendous amount of business processes in other industries such as insurance, healthcare, logistics and supply chain management (Milani, Garcia-Banuelos, and Dumas, 2016). The main development that has led to the blockchain being applicable in different industries are so-called smart-contracts. Smart contracts are contracts which are automatically enforced by computer protocols. An online payment will trigger when a preprogrammed condition of a contractual agreement is fulfilled (Crosby et al., 2016).

An example of the blockchain outside of the payment sector is the collaboration between IBM and the Danish logistics firm Maersk for a global cross-border supply chain (Haswell, 2017). This solution manages and tracks the paper trail of tens of millions of shipping containers across the world by digitizing the supply chain process from end-to-end, to enhance transparency and the sharing of information among trading partners. Despite all the potential the blockchain technology offers for implementation, it is at present not fit to replace all processes with the blockchain technology. There are multiple reasons for this. First of all, due to the blockchain technology being a rather recent innovation it still faces technical challenges like scalability and privacy (Deshpande et al., 2017). Another argument for not using a blockchain is that occasionally the blockchain technology has no real value proposition over a centralized database (Lewis, 2017). In addition to this, the blockchain technology faces resistance from people or organizations which are to-be-replaced by blockchain. This resistance of people against new technology is the same as in the past (Noble, 1995). One of the most notable examples being the industrial revolution, in which jobs of humans were replaced by industrial machinery. Comparable to this are physical encyclopedias and maps being made obsolete by Wikipedia and Google Maps.

The aim of this research is to expand the body of knowledge related to the blockchain technology. An attempt will be made to create an artifact, in the form of a decision support system (DSS) that will aid in selecting the most desirable blockchain platform for a specific case. This DSS will have its foundation based on the model for multi-criteria decision making (MCDM) presented by Farshidi et al. (2018). The literature research will provide the initial knowledge base for this research. The other data required for this research is collected through expert interviews and documentation. Once the artifact has been created, several case studies will provide input (in the form of requirements and pre-defined expected results) for evaluating the DSS.

Chapter 2

Problem Statement

One of the main issues with respect to the blockchain technology (and the closely related cryptocurrencies) is the lack of literature and scientific research. This is mainly due to the fact that the main body of literature only started to appear around 2013. Around 5 years after the introduction of the Bitcoin blockchain by Nakamoto (Nakamoto, 2008) in 2008. Yli-Huumo et al. (2016) conducted research in 2016 in which they established the situation regarding blockchain development and research at that time. In this research, Yli-Huumo et al. (2016) identified that most of the research done concentrated on security and privacy issues of the blockchain technology. The research on topics (related to challenges and limitations) such as usability and efficiency was described by Swan (2015) to be rather limited. In addition to this, Yli-Huumo et al. (2016) did not find a considerable amount of studies on latency, size and bandwidth, throughput, versioning, hard forks and multiple chains. On top of all of this, almost all the research was performed in the Bitcoin blockchain. This inevitably led to almost no research on smart contracts since the Bitcoin blockchain does not support smart contracts (Nakamoto, 2008).

The research on usability and application of the blockchain technology is rather lacking, not so much from the user perspective but more so from the developer perspective (Swan, 2015). Up until 2017 research from big organizations in the blockchain domain was rather limited, which is quite remarkable due to the possibilities of the blockchain. To indicate this, major organizations such as JPMorgan Chase, Cisco, Accenture and Mitsubishi only started actively seeking blockchain possibilities in 2016 or 2017. A positive exception to this is IBM which started its open-source blockchain-based software project HyperLedger in 2015 (Morabito, 2017). In the following years, the development around the blockchain technology increased, alongside new innovation and adoption. In 2017 it became more apparent to many organizations that the blockchain technology could provide an edge in different industries. In 2017 Gartner placed the blockchain around the peak on the Gartner Hype Cycle for Emerging Technologies for 2017 (Panetta, 2017). In addition to this, media coverage of the blockchain technology received a huge boost when different cryptocurrencies (e.g. Bitcoin, Ethereum, etc.) rose to all-time high valuations. Bitcoin, for example, was valued around 950 US dollar at the beginning of 2017 and around 19600 US dollar in December 2017 (CoinMarketCap.com, 2013).

Next to the limited knowledge in the scientific body and business environment, there are already a considerable amount of blockchain platform alternatives available in the market. Both with (CoinMarketCap.com, 2013) and without a built-in cryptocurrency (Cachin, 2016). This research will merely focus on blockchain platforms rather than all the available alternatives. These blockchain platforms allow for rapid prototyping, development, and deployment of new decentralized blockchain applications (DApps) (Baliga, 2016). These blockchain platforms are mostly open

sourced and available for most to participate and use. Each of this blockchain platforms is designed with specific goals, which dictate its features. Should a company decide they want to develop a blockchain application they have to select the right blockchain platform for their use-case. Not every blockchain platform offers the same features (due to different goals) which are required for a specific case. For example, one blockchain platform might support side-chains for scalability while another platform offers sharding technology to scale. Both implementations would have their own implications for solving a case-specific problem. In addition to this, factors such as suitability, security, scalability, etc. should be considered when picking the most appropriate blockchain alternative (Swan, 2015). Therefore, this technology selection process can be modeled as a multi-criteria decision-making (MCDM) problem that deals with the evaluation of a set of alternatives while taking a set of decision criteria into account (Triantaphyllou et al., 1998).

The combination of the factors hype, limited scientific research, and many potential solutions has led to the odd situation in which a lot of organizations desire to utilize the blockchain technology, but lack the knowledge in selecting the most optimal blockchain platform. Until now there are only a few rather limited solutions for selecting a blockchain platform (Pahl, Ioini, and Helmer, 2017; Rikken, 2018; Wust and Gervais, 2018). Each of these solutions are basic decision trees in which first is determined whether a blockchain is needed in a specific case or not. This is due to that a blockchain implementation only offers significant advantages over a traditional implementation (e.g. scalability of Database Management Systems) when the fundamentals of a potential case match with what blockchain offers (Greenspan, 2016). Should the outcome be (in each of (Pahl, Ioini, and Helmer, 2017; Rikken, 2018; Wust and Gervais, 2018)) that a blockchain is needed, another decision tree is used to determine which blockchain platform to use. However, the output of this decision tree is merely a broad category of available alternatives. Both these decision trees don't take the features of specific blockchain platforms in consideration at all, despite different platforms clearly having different goals (Baliga, 2016). However, this lack of extensive decision tools offers an opportunity for this research to improve on. The problem in this domain can be defined as the following formal problem statement:

Problem Statement: There is a lack of models, frameworks and artifacts for the selection process of a blockchain platform from a developer's perspective

Chapter 3

Research Method

3.1 Research Model

3.1.1 Research Goal

The problem of this domain as identified in Chapter 2, is a lack of artifacts that can assist during the selection process between different blockchain platforms. Thus the most logical main research-question derived from this is:

How can an artifact be developed that assists during the selection process between different blockchain platforms from a developer's perspective?

To reduce the extent of this problem and solve the research question, the main goal of this research is to create an artifact that assists during said process. An additional goal of this research is mapping the current technological advances in the domain of blockchain technology. In practice this means describing the most prominent sub-technologies. However, these two goals should not be seen as two separate projects but rather as intertwined projects. Due to reason that this expanded knowledge-base will be used for the to-be-created artifact.

3.1.2 Decision Making

The artifact proposed for this research will be the 'Blockchain Platform Decision Support System' (BPDSS) with all the fundamental components of a standard Decision Support System as described by Sage (1991). A Decision Support System is a tool that can be used over the full life-cycle and can co-evolve its advice based on evolving requirements. The approach towards creating this BPDSS will be according to the Model-based decision support system for MCDM problems by Farshidi et al. (2018), visualized in Figure 3.1.

This Decision Model applies the six-step decision-making process (as defined by Majumder (2015) that deals with structuring, planning, and solving the problem concerning a set of criteria: (1) Identifying the objective, (2) Selection of the features, (3) Selection of the alternatives, (4) Selection of the weighing method, (5) Applying the method of aggregation and (6) Decision-making based on the aggregation results. In addition, this Decision Model utilizes the MoSCoW prioritization technique (DSDM-Consortium, 2014) to assess criteria weights, uses assessment models to measure the values for non-boolean criteria and utilizes ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge (Farshidi et al., 2018). In this research (Farshidi et al., 2018), the model was applied to construct a DSS that assists during the selection process between different Database Management Systems (DBMS) for Software Producing Organizations (SPOs).

3.1.3 Model Selection and Comparison

Farshidi's DSS model Farshidi et al. (2018) offers some advantages over other MCDM methods which were also utilized to address technology selection problems. Examples of other MCDM methods are *Analytic Hierarchy Process (AHP), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Machine Learning (ML) and Fuzzy based decision making* (Abdullah, 2013). However, the problem with many of these methods is that they use pairwise comparison to assess the weight of criteria, which becomes rather time-consuming and complicated as the number of criteria increases (Saaty, 1990). As explained in Chapter 2 both Pahl, Ioini, and Helmer (2017) and Rikken (2018) use a simplistic decision tree as decision method. The main advantage of a decision tree is the ease of use (Petri, 2010). However, decision trees are rather unstable, struggle with complexity and are often relatively inaccurate. To indicate this, a small change in the data can lead to a large change in the structure of the optimal decision tree. This is rather undesirable in an immature domain like blockchain in which still a lot of data can be revised. Thus a decision tree as a method is unsatisfactory for this research.

The novelty of the DSS of Farshidi et al. (2018) lies in utilizing the MoSCoW technique (DSDM-Consortium, 2014) to assess criteria weights and reduce uncertainty, in introducing assessment models to measure the values of non-boolean criteria, in using ISO/IEC quality aspects to indicate the relationship among criteria according to domain experts' knowledge (Farshidi et al., 2018) and being maintainable and evolvable by applying the six-step decision making process by Majumder (2015).

3.2 Blockchain Platform Decision Support System

To create the BPDSS, several data about the respective domain has to be gathered according to Farshidi's model. This data can roughly be divided into several sets: Quality aspects, Domain Features, Domain-Alternatives and the Domain-feature requirements. The Domain-Alternatives are the blockchain platforms available on the market, for example, Hyperledger Fabric blockchain from the Linux Foundation or the Ethereum blockchain by the Ethereum Foundation. These platforms are mainly acquired from documentation and literature or possibly experts on a specific platform. The specific selection criteria for the Domain Alternatives are mentioned in section 3.6.2.

The Domain qualities are metrics to define the quality of software. The ISO/IEC 25010 (ISO, 2011) and Ext. ISO/IEC 9126 (Carvallo and Franch, 2006) are the most general applicable metrics according to Farshidi et al. (2018). These quality aspects are domain independent and thus will all be utilized.

The Domain-Features is a collection of the generic Domain-features which the Domain-Alternatives provide. Novel Domain-features will be excluded. Examples of Domain-Features are smart-contract support or off-chain transactions. Each domain feature has a data-type, which could be boolean or numeric. For example, smart-contracts are supported or not (boolean), while maturity of the platform can be either low, medium or high (numeric). These boolean Domain-Features are gathered through interviews with Domain Experts. The numerical features are: Maturity, Popularity in the market, Innovation and Transaction speed. The numerical value is determined based on different parameters which are supported by literature and Domain expert knowledge.

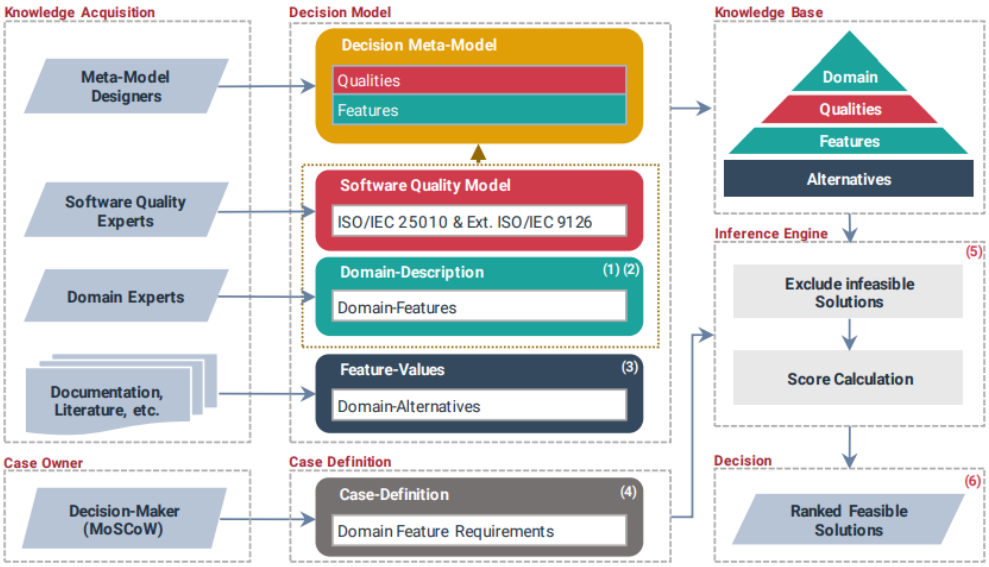


FIGURE 3.1: A model-based decision support system for MCDM problems (Farshidi et al., 2018)

To create the Decision Model the Domain-(sub)Qualities are mapped against which Domain-Features have a positive influence on those qualities. The Domain-Alternatives are mapped against which Domain-Feature they provide in a similar way. The ISO/IEC quality aspects in this model are used to indicate the relationship among the domain features in order to measure the importance of each domain feature based on the domain experts and the decision maker's perspectives. For example, the off-chain transactions feature influences the performance or different consensus-mechanisms influences the fault tolerance quality aspect. The other mapping is between the Domain-Alternatives and the Domain-Features they provide, for example the Ethereum blockchain supports smart-contracts. The lists of Domain-Alternatives, Domain-Features, Domain-Qualities together with the mapping based on Domain Experts form the Decision Model. In addition to this there the domain feature requirements. The Domain feature requirements provide the decision-makers with the ability to prioritize each of the Domain-Features based on the MoSCoW-prioritization technique (DSDM-Consortium, 2014). This technique categorizes the feature-requirements into either must-have, should have, could have or won't have.

The last step is entering the Knowledge Base (Domain Qualities, Domain Features, Domain Alternatives, and mappings) into the Inference Engine. This Inference Engine calculates the score for each feasible alternative based on the mapping and the MoSCoW prioritization. Infeasible solutions are left out of this calculation. Eventually, the output of the DSS is a list of Feasible Solutions that are ranked according to the score calculated by the Inference Engine. A working instance of the DBMS and Cloud Service Provider (CSP) selection DSS has been made available online.

3.3 Research Questions

The process of creating the Blockchain Platform DSS (according to the model presented by Farshidi et al. (2018)) will be done by splitting up the different steps of the

Research Questions	
MQ	How can an artifact be developed that assists during the selection process between different blockchain platforms from a developer's perspective??
RQ 1	What are the technologies related to the blockchain technology that are relevant for the creation of the artifact?
RQ 2	What are the contemporary features and platforms in the blockchain domain?
RQ 3	Which identified blockchain features have a positive influence on different software quality aspects?
RQ 4	Which features are offered by each of the available blockchain platforms?
RQ 5	Is the created decision support system applicable in the business environment?

TABLE 3.1: The main research question and the relevant sub-research questions

model into sub-research questions. This section will elaborate further on the different sub-questions related to the Main-research question. Table 3.1 mentions all the sub-research questions alongside the main research question, which was explained in section 3.1.

RQ 1: What are the technologies related to the blockchain technology that are relevant for the creation of the artifact? Before all the steps of Farshidi's model can be initiated, a base of knowledge with respect to blockchain has to be established which will serve as the foundation for the other research questions. This section will (but not limit itself to) explaining the technical details of the blockchain technology, related technologies, possibilities and challenges and implementation in practice. Answering this research question will be the main part of the literature study. The main deliverable of this phase is an initial list of Domain-Features which serves as input for the interview protocol in the data collection phase.

RQ 2: What are the contemporary features and platforms in the blockchain domain? This question deals with identifying the Domain Features and Domain Alternatives required in Farshidi's model. In addition to this, it increases the knowledge base with respect to blockchain technology. The available blockchain platform alternatives will be based on the literature research and as auxiliary sources several alternative experts (for example solution vendors). The Domain Features are based on the knowledge of blockchain domain experts and the literature research and documentation serve an auxiliary role. This is further elaborated on in section 3.6.1.

RQ 3: Which identified blockchain features have a positive influence on different software quality aspects? This question deals with establishing the relationship between the identified Domain Features of the blockchain domain and the positive influence they potentially have on each of the Domain Qualities from ISO 25010 or Ext. 9216. The mapping of these relationships will be done based on the knowledge of Domain Experts. The mapping will also be referred to as SF-mapping deliverable. The SF-mapping will serve as input for the creation of the DSS.

RQ 4: Which features are offered by each of the available blockchain platforms? This question deals with establishing the relationship between the different blockchain platforms (Domain Alternatives) and which Domain Features they offer. The mapping of this will be based on documentation and potentially experts on specific platforms. This mapping will be referred to as the FA-mapping deliverable. The FA-mapping will serve as input for the creation of the DSS.

RQ 5: Is the created decision support system applicable in the business environment? Before this question can be answered obviously first the artifact has to be built based on the mapping done in the other research questions. The DSS will be created

in a similar fashion as the DSS on this website. Once this the DSS had been built it has to be validated whether it solves the business problem. Since the blockchain technology is so new, the created DSS can't be compared against similar projects. Therefore this research will focus on the validation of the artifact on whether the DSS output for certain case studies is valid according to domain expert feedback.

3.4 Design Science Research

The research approach for this problem is the Design Science Research method. This due to "Design science addresses research through the building and evaluation of artifacts designed to meet the identified business need" (Hevner, 2007). In the problem statement the business need for such an artifact has been elaborated on, alongside the scientific gap and what kind of artifact will be created. Figure 3.2 shows the Design Science Research Framework applied to this research. Design Science Research is a creative, and often iterative, problem-solving process (Hevner et al., 2004).

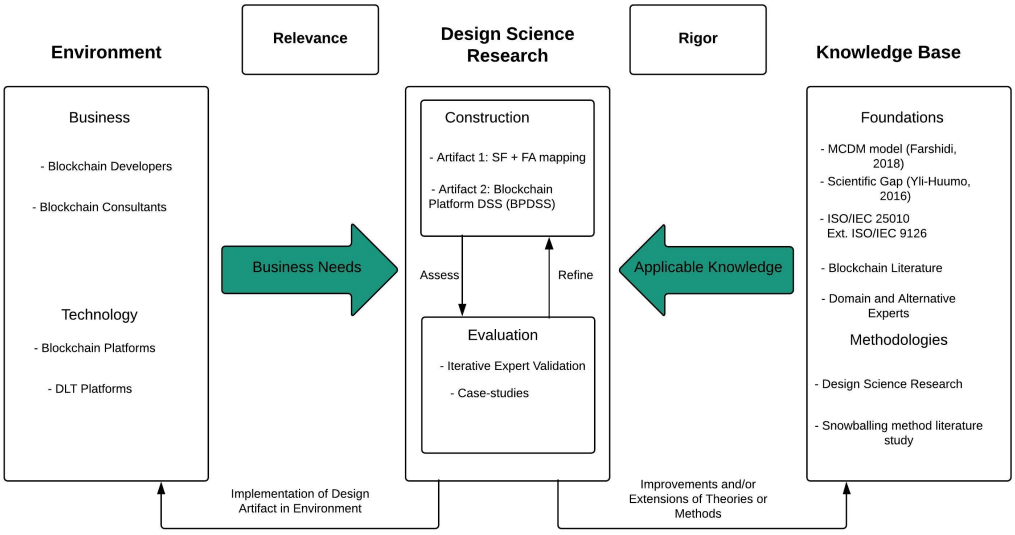


FIGURE 3.2: Information Systems Research Framework applied to this research

Since Figure 3.2 is rather theoretical more concrete phases were generated. These three more tangible phases are shown in Figure 3.3. Table 3.2 shows the sub-research questions and the relevant research methods for each question. The following sections in this chapter will explain the literature study, Experts Interviews, Documentation Analysis, Case Study Evaluation and Expert Validation in more detail.

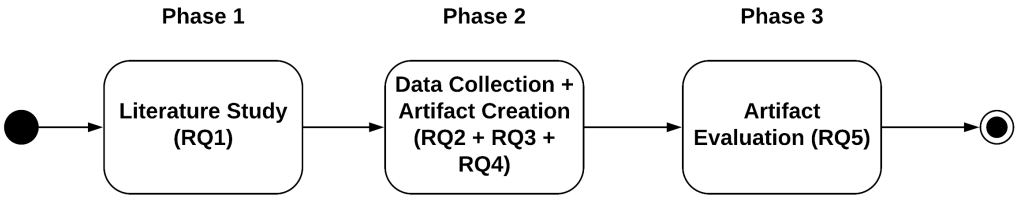


FIGURE 3.3: Project-phases

Phase	Research Question	Main Research Method	Auxiliary Research Method
Phase 1	Sub-Research Question 1	Literature Study	Document Analysis
Phase 2	Sub-Research Question 2	Expert Interviews	Document Analysis
	Sub-Research Question 3	Expert Interviews	Document Analysis
	Sub-Research Question 4	Document Analysis	Expert Interviews
Phase 3	Sub-Research Question 5	Case Study Evaluation	Expert Opinion

TABLE 3.2: Research-phases with respective Research Questions and Research Methods

3.5 Literature Study Protocol

This section will elaborate on the literature research protocol. The literature research protocol that will be used applies mainly to answering RQ1. This literature research will be approached with the snowballing method, also known as chain-referral sampling as explained by Wohlin (2014). The snowballing method is a non-probability sampling method used when characteristics to be possessed by samples are rare and difficult to find. Because the blockchain technology is such a new innovation, research on it is rather specific and rare. This research will both utilize forward snowballing and backward snowballing. A literature study based on backward snowballing starts with an initial source and additional sources are based on the reference list of this initial source (Jalali and Wohlin, 2012). Identifying new sources based on forward snowballing is the reverse process of backward snowballing. Thus, new sources are identified by searching for sources that cite the initial source. Both these methods are used to make the literature research as thorough as possible.

The starting point for the forward snowballing method is the initial research which presents the blockchain technology and Bitcoin cryptocurrency by Nakamoto (2008). This source is the most appropriate for forward snowballing since the different versions of this study have been cited in great quantity in more recent blockchain research.

The starting point for the backward snowballing method is the study of Yli-Huumo et al. (2016). This source is appropriate for backward snowballing since it cites various other sources that conducted research on the blockchain technology. In addition to this, the paper by Yli-Huumo et al. (2016) has been published fairly recently, so it is less likely more recent sources are not cited. However, forward snowballing will also be applied to the study of Yli-Huumo et al. to prevent the exclusion of more recent studies.

3.6 Data Collection + Artifact Creation

3.6.1 Expert Interviews

This section will further elaborate on the role the experts used in this research, as well as how they will be selected and how their knowledge will be captured.

Expert Selection

The snowballing method of identifying sources will be used for finding blockchain experts as well. The reason for this comes down to the rather limited number of knowledgeable blockchain experts because the blockchain domain is rather immature. Identification of the Domain Features and Domain Alternatives not only requires a domain expert to have a thorough understanding of the technical side of

the blockchain technology but also on the business side of blockchain implementations. This is even more so true for mapping the Domain Features, Alternatives, and Qualities and finally evaluating the created artifact in the form of the DSS. Thus identifying the initial blockchain experts is the most important, since these experts can refer to other experts which might provide knowledge from different perspectives on the blockchain technology. The starting point will be finding preferably an academic blockchain expert (e.g. experts working at universities/research institutes) due to affiliation with scientific research and credible domain knowledge. In addition to this also potential blockchain experts listed on LinkedIn or other domain professional websites will be contacted to greatly increase the potential pool of candidates. The goal is to have roughly half of the experts being academics and the other half more business focussed (e.g. public speakers, architects, developers, etc.). For the non-academic blockchain experts to be selected for this research they have to conform to specific requirements. Several years of relevant experience with the blockchain technology and no commercial incentive with respect to this research (excludes cryptocurrencies speculators as well) being the most important criteria. Potential interviewee candidates were officially invited to take part in this research by the means of a letter, which can be found in Appendix K (Dutch) and Appendix L (English).

Interview Protocol

The blockchain expert interviews will be the main data collection element of this research. The interviews that will be conducted are broadly categorized into a sequential order. The first set of interviews will be with blockchain domain experts to identify all the Domain-Features. The next set of interviews is optional depending on whether all the Domain-Alternatives can be identified based on documentation and literature. Should this not be the case then additional interviews with alternative experts will be conducted. Before the last set of interviews can be conducted all the Domain-Alternatives and Features have to be identified since these will be used as input for the interviews. Together with blockchain domain experts, the Software Quality aspects will be mapped against the Domain-Features (SF). Should documentation and benchmarking not be sufficient for mapping the Domain-Features against the Domain-Alternatives (FA) then additional interviews will be conducted. These additional interviews will be with alternative experts and/or blockchain platform vendors. The interviews will be semi-structured (Institute, 2009) due to the nature of the required interaction between the interviewer and interviewee. The interview protocol shall be created based on the interview protocol utilized by Farshidi et al. (2018). The input for this interview protocol will be an initial set of Domain-Features based on the Literature Study of this research (with the numerical features + parameters from Farshidi et al. (2018)) and an initial set of Domain-Alternatives (as explained in Section 3.6.2. It is expected that the initial set of Domain-Features will initially diverge with new additions to the list of Domain-Features. After this divergence, it is expected that some Domain-Features will be removed as they are not considered generic or are not mentioned by the blockchain experts. This process of additions/removals is documented in Appendix B. Before the interviews are conducted according to the created protocol, the protocol shall first be validated by one of the academics of Utrecht University. The interviews will be recorded with an audio-recording device. After the completion of this research, these recordings will be deleted. In addition to this, all the participants and citations (if applicable) will remain anonymous. The main inference and processing of the data shall be

done during the interactive sessions with the experts. However, the recorded data will also be analyzed as well with a goal to identify the generic blockchain Domain Features. The interview protocol can be found in Appendix B.

3.6.2 Alternative Selection

This section will briefly explain the scope of this research. Due to the many existing blockchain platforms, obviously a selection has to be made. From a practical point of view, it is impossible to include all blockchains due to the time constraint. The scope of this research excludes all blockchains that are no blockchain platforms. In addition to this only platforms with their own main-net currently running will be selected for now. It should be noted however, that some alternatives will be included that are strictly speaking not conform the definition of blockchains. But rather these platforms are distributed ledger technologies (DLT), which is an umbrella term for blockchains among other technologies (Mainelli and Smith, 2015). Should other prominent alternatives release their own blockchain main-net these will be considered to be included in the BPDSS as well. Fortunately, due to the nature of Farshidi et al. (2018)'s model the BPDSS can be extended relatively effortless with additional blockchain alternatives should new alternatives meet the selection criteria. The initial five domain alternatives that will be used as input for the Interview Protocol (Section 3.6.1) are the 'most prominent' contemporary blockchain platforms. Three of these five domain alternatives will be the blockchain platforms that have the highest market capitalization according to CoinMarketCap. The other two initial domain alternatives (which are not linked to any native currency) are selected based on a quick analysis of (grey)literature, articles and documentation to determine the most noteworthy alternatives.

3.6.3 Document Analysis

As mentioned in Table 3.2, the main method to gather information related to RQ4 will be the analysis of related documents. Documentation analysis will also be the auxiliary approach to collect additional information besides the main methods (literature study and expert interviews). Document analysis is a form of qualitative research in which documents are interpreted by the researcher to give voice and meaning around an assessment topic (Bowen, 2009). Examples of documents that can and will be analyzed are whitepapers, business plans, benchmarking studies, company updates, training guides or customer recommendations.

3.6.4 Artifact Creation

The input for the creation of the BPDSS will be based on the model by Farshidi et al. (2018). In practice, this means the Domain-Features, Domain-Alternatives, Domain-Qualities and the mapping between these. An important aspect, however, is that this is an incremental process, and adaptations to the DSS will be made constantly on the expanding base of knowledge. Appendix P shows screenshots of the BPDSS, which is also available online.

3.7 Artifact Evaluation

The evaluation of the created artifact will be largely based on a study of Prat, Comyn-Wattiau, and Akoka (2014) in which they describe evaluation methods for artifacts

created during design-science research. Since this research creates an artifact based on design science research the evaluation method from the study of Prat, Comyn-Wattiau, and Akoka (2014) should be appropriate. The evaluation method that will be used is: 'Demonstration of the use of the artifact with several real examples' from the study of Prat, Comyn-Wattiau, and Akoka (2014). This evaluation is based on analysis and logical reasoning to measure to which degree the goal dimension of the artifact is met. The goal is evaluated based on the efficacy, validity and generality criteria. The efficacy is the degree to which the artifact produces its desired effect (Venable, Pries-Heje, and Baskerville, 2012). The validity criteria is defined as the degree to which the artifact works correctly (Straub, Boudreau, and Gefen, 2004). Reliability is encompassed by validity as well. Artifact generality is goal generality (Aier and Fischer, 2011): the broader the goal addressed by the artifact, the more general the artifact. The relativeness of the evaluation is absolute as the results will not be compared to other artifacts, since there are currently no similar implementations which use the DSS method in the blockchain domain. Secondly, the efficiency and usefulness of the BPDSS is evaluated through three exploratory theory-testing case studies. The unit of analysis is a unique technology selection decision in a software product. These case studies involved: Defining the Domain Feature requirements and prioritizing them and the second phase comparing the feasible DSS solution with their own chosen solutions. Due to the immaturity of the field, a blockchain expert will evaluate the results from the case studies as well on validity.

3.8 Process-Deliverable Diagram

This section elaborates on the Process-Deliverable Diagram (PDD) relevant for this research, as explained by Weerd and Brinkkemper (2008). The main purpose of a PDD is to provide a clear overview of the steps or activities that are needed to create the deliverables of each step in a specific technique or method. In general, the PDD is divided into two sides, left and right. The left side contains the process steps and the right side contains the deliverables for each step, connected with broken lines. The PDD for this research is shown in Appendix J. This PDD visualizes the explanation of this chapter. On the process side of the PDD are the different phases along with the sub-research questions they relate to. The right side shows the specific deliverables as explained in section 3.2.

Chapter 4

Literature Study

This chapter presents the contemporary literature regarding the blockchain technology, a technical explanation of the technology, various consensus-mechanisms, smart contracts, permission-models and various other information. This chapter is advantageous to understand the context of the blockchain domain.

4.1 Technical Foundation

The introduction of the blockchain technology was in 2008, in the paper 'Bitcoin; A peer-to-peer electronic Cash-System' by Satoshi Nakamoto (pseudonym of a still unknown author) (Nakamoto, 2008). In this paper, Nakamoto presents an electronic cash system, nowadays more widely known as the Bitcoin cryptocurrency. Cryptocurrency can be simplified as digital money, an alternative payment currency to traditional money like the US Dollar or Euro valutas. With 'traditional' transactions on the internet, financial institutions serve as trusted third parties to process these electronic payments and avoid the double-spending of currency. However, the role of these trusted third parties has its weaknesses. For example, completely non-reversible transactions are not possible with this traditional system. This increases potential mediation costs, transaction costs and decreases privacy and the potential amount of transactions (Nakamoto, 2008). Physical currency avoids the need for a trusted third party since a physical coin or note can only be given out once, this is not affected by the double-spending problem. However, no such a mechanism existed yet in the digital world to make trusted third parties obsolete with respect to the double-spending problem. The blockchain technology was introduced as a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of chronological order of transactions (Nakamoto, 2008).

4.1.1 Transactions

Nakamoto defines an electronic coin (or cryptocoin/cryptocurrency) as a chain of digital signatures. During the transfer of such a cryptocoin, (e.g. several Bitcoins from person A to person B) the owner of the coin signs a hash of the previous transactions of that particular coin and the public key of the receiver and adds this to the end of this chain of digital signatures. This is visualized in Figure 4.1.

The problem is however, the receiver has no means of knowing whether the sender of the transaction didn't double spend the transaction. Prior to the blockchain technology, a frequent way of solving this problem would be a trusted third party validating these transactions for double spending. However, this would mean the introduction of a centralized authority accompanied by its drawbacks. Yet, by communicating each transaction publicly the need for a centralized authority dwindles.

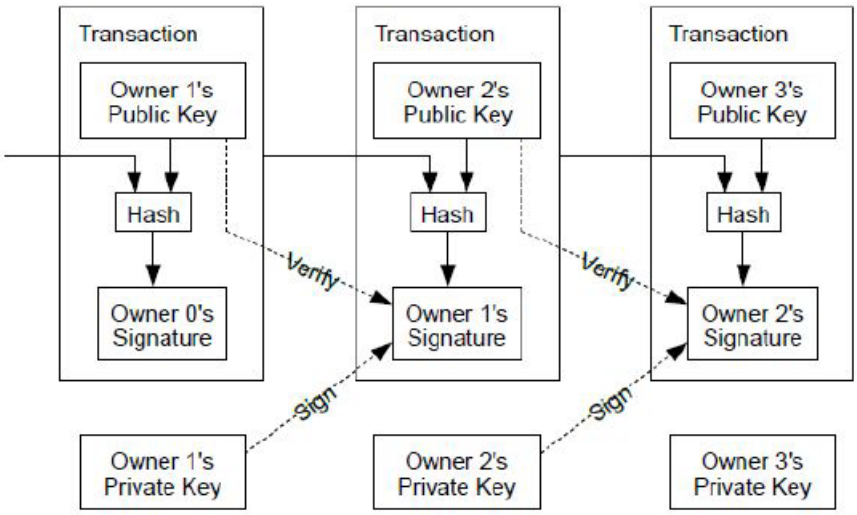


FIGURE 4.1: Signing of a chain of digital signatures (Nakamoto, 2008)

All participants agree on a single correct history of transactions and is expanded with the first received transaction on the condition that the history of the other transactions is agreed upon as well. To determine which transaction was received first, a timestamp server was proposed as a solution. The timestamp server works by taking a hash of a block of items to be time-stamped and widely publishing the hash (Nakamoto, 2008). This proves that the data must have existed at that time in order to get into the hash. All the timestamps together form a chain by referring to an earlier timestamp in its hash, as visualized in Figure 4.2. The blocks are related to a certain hash which forms a chain, hence the name blockchain.

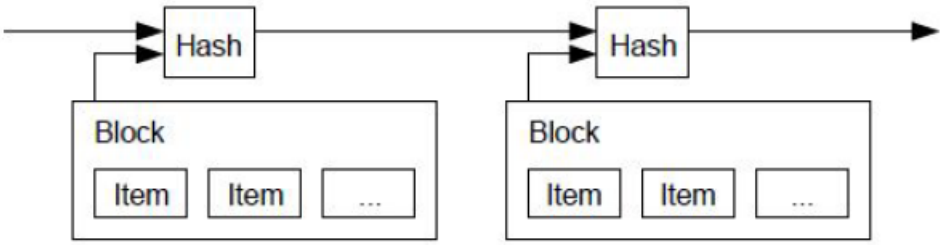


FIGURE 4.2: The visualization of a blockchain through hashing and timestamping (Nakamoto, 2008)

In order for the peer-to-peer network to reach a consensus on the current valid state of the blockchain, there are several mechanisms. The first introduced consensus mechanism (for the Bitcoin) utilizes the proof-of-work (PoW) mechanism. A proof-of-work is a piece of data which is difficult to produce (costly or time-consuming) but easy for others to verify. In the case of the Bitcoin, this is the Hashcash PoW system. In this Hashcash PoW mechanism, all new transactions are collected in a new block and made public to all other nodes in the network. For a new block to be accepted by network participants, miners must complete proof-of-work which covers all of the data in the block. This becomes increasingly difficult to limit the rate at which new block can be generated by the network to one every 10 minutes. For a block to be

valid it must hash to a value less than the current target; this means that each block indicates that work has been done generating it. Each block contains the hash of the preceding block, thus each block has a chain of blocks that together contain a large amount of work. Changing a block (which can only be done by making a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain. This protects the blockchain from tampering.

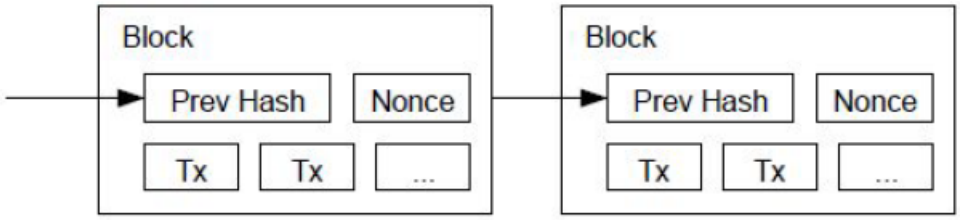


FIGURE 4.3: Proof-of-Work consensus mechanism (Nakamoto, 2008)

4.2 Consensus Mechanisms

Despite PoW currently being the most adopted way of reaching consensus on the blockchain, there are other consensus mechanisms as well been implemented. This is due to scalability issues of PoW, leading to latencies on the order of an hour for a single transaction and high energy costs Vukolic (2016). PoW is also considered rather slow with at the time of writing only processing approximately 7 transactions/second on the Bitcoin blockchain. In comparison Visa processes around 2.000 transactions per second (Narayanan et al., 2016). The 4 other most important alternatives to PoW are currently Proof-of-Stake and its derivatives, (Delegated and Federated) Byzantine fault tolerance, Federated Byzantine Agreement and Proof-of-elapsed time (Baliga, 2017).

Currently, the most important alternative to PoW is Proof-of-Stake (PoS). The main advantage PoS has over PoW are the high energy costs of PoW and scalability. PoS completely replaces the mining operation with a voting mechanism based on staking cryptocurrency. The PoS algorithm pseudo-randomly selects validators for the creation of a new block. Should the node vote for an invalid transaction the stake of the node is burned. This tackles the Nothing-at-Stake problem in which nodes can vote for multiple forks of the blockchain with nothing at stake. In addition to losing their stake as an incentive to act honest, nodes receive a dividend based on the size of their stake as a reward for rightful voting. The NXT blockchain currently utilizes this concept and the Ethereum blockchain will shift from PoW to PoS in 2018 (Prisco, 2017).

To understand the other consensus models, it is important to understand that Blockchain platforms can roughly be categorized into three main variants, being: Public, Consortium and Private (Buterin, 2015). In a public blockchain everyone can participate as a node in the network to take part in the consensus process in addition to reading and writing transactions. These blockchains are generally considered to be fully decentralized. Bitcoin and Ethereum are prime examples of this. 'Permissionless' is another term commonly used to refer to public blockchains (Perretta, 2017). In a private blockchain, permissions are restricted to one organization. Likely examples of the application of private blockchains include database management,

internal auditing, etc. These blockchains are considered centralized and are often referred to as 'Permissioned' blockchains.

Consortium blockchains are blockchains where the consensus process is controlled by a preselected set of nodes which are semi-trusted and verified members. An example of a consortium blockchain is a group of financial institutions, each of which operates a node. To reach a consensus on the validity of a block for example 10 out of the 12 nodes have to approve it. Consortium blockchains are a hybrid between public and private blockchains but in practice often require permission before a party can participate in the network.

Practical Byzantine fault tolerance (PBFT) is a consensus mechanism that solves the Byzantine Generals problem (Lamport, Shostak, and Pease, 1982). The Byzantine Generals problem boils down to how to reach consensus when faced with untrustworthy and malfunctioning actors that threaten to destabilize the network. PBFT is used in the Hyperledger Fabric blockchain, which is a consortium blockchain developed by the Linux Foundation (Cachin, 2016).

Delegated Byzantine Fault Tolerance (dBFT) solves the Byzantine Generals problem by querying a random node in the network about the state of the network until >66 percent of the network agrees with that random node. This implies that dBFT assumes at least two-thirds of the network operates not maliciously. Currently, the public NEO blockchain utilizes this consensus mechanism (NEO-Foundation, 2017).

Federated Byzantine Agreement (FBA) is utilized by the public Ripple and Stellar blockchains to tackle the Byzantine Generals problem (Mazieres and Stellar-Development-Foundation, 2016). Both are real-time gross settlement systems aimed at supporting financial institutions completing a high amount of transactions/sec. The instantiation of the FBA in Ripple works in an iterative way. A batch of transactions first has to be approved by at least 50 percent of the nodes to become a candidate set. After this, it is pushed further for higher approval ratings until 80 percent (super-majority) of the nodes approve of the specific candidate set. The Stellar blockchain started as a hard-fork of the Ripple blockchain, however, is more decentralized through changes to the FBA. In a study performed by Baliga (2017) the discussed consensus mechanisms are compared against each other, see Table 4.1.

	PoW	**PoS**	**PoET**	**BFT and variants**	**Federated BFT**
Blockchain type	Permissionless	Both	Both	Permissioned	Permissionless
Transaction finality	Probalistic	Probalistic	Probalistic	Immediate	Immediate
Transaction rate	Low	High	Medium	High	High
Token needed?	Yes	Yes	No	No	No
Cost of participation	Yes	Yes	No	No	No
Scalability of peer network	High	High	High	Low	High
Trust model	Untrusted	Untrusted	Untrusted	Semi-trusted	Semi-trusted
Adversary Tolerance	<=25 percent	Depends on specific algorithm used	Unknown	<=33 percent	<=33 percent

TABLE 4.1: A comparison of blockchain consensus mechanisms by Baliga (2017)

Some lesser applied and more novel consensus mechanisms are the following: Proof-of-Activity, Proof-of-Burn, Proof-of-Capacity and Proof-of-Importance (Castor, 2017). In addition to the discussed consensus mechanisms, new consensus models are still emerging to address other limitations of contemporary ones.

4.3 Smart-Contracts

In addition to the blockchain technology, and related to online transactions, a new technology was developed by Szabo (1994) called Smart contracts. Smart contracts are contracts which are automatically enforced by computer protocols. An online payment will trigger when a preprogrammed condition of a contractual agreement is fulfilled (Crosby et al., 2016). However, no use for smart contracts was found until the emerge of the blockchain technology. Now the blockchain technology enables the registration, verification, and execution of the smart contracts by checking if the conditions are met. In practice, this would mean the introduction of a smart contract could make e.g. a real-estate agent obsolete when buying or selling a house under the assumption the conditions in the smart-contract are fulfilled.

An example of the source code of a smart contract written in the Solidity language is shown in Figure 4.4. In addition to Solidity, smart contracts can also be programmed in other programming languages such as Java, Golang, Javascript, C++, Python and .NET (NEO-Foundation, 2017; Rosic, 2017).

```
1  contract MetaCoin {
2    mapping (address => uint) balances;
3
4    function MetaCoin() {
5      balances[tx.origin] = 10000;
6    }
7
8    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
9      if (balances[msg.sender] < amount) return false;
10     balances[msg.sender] -= amount;
11     balances[receiver] += amount;
12     return true;
13   }
14
15   function getBalance(address addr) returns(uint) {
16     return balances[addr];
17   }
18 }
19
```

FIGURE 4.4: Example of the source-code of a smart-contract written in Solidity

4.4 Scalability

One of the main contemporary challenges of public permissionless blockchains is the ability to scale with respect to transaction-speed and keeping transactions-costs low when the size of the peer-network increases (Deshpande et al., 2017; Vukolic, 2016; Herrera-Joancomartí and Pérez-Solà, 2016; Koteska, Karafiloski, and Mishev, 2017). This is indicated by the Bitcoin blockchain only being capable of performing roughly 7 transactions per second, as mentioned in section 4.2. Another prominent example was the clogging of the Ethereum-blockchain at the end of 2017 due to the popular dApp 'Cryptokitties' (BBC, 2017). This congestion roughly tripled transactions costs and greatly increased transaction-time on the Ethereum blockchain (Etherscan.io, 2018).

4.4.1 Scalability Trilemma

According to Ethereum co-founder Vitalik Buterin the scalability issue in the blockchain domain comes down to the scalability trilemma (Buterin, 2018). The scalability trilemma claims that blockchain systems can only at most have two of the following three properties:

- Decentralization (defined as the system being able to run in a scenario where each participant only has access to $\mathbf{O}(\mathbf{c})$ resources)
- Scalability (defined as being able to process $\mathbf{O}(\mathbf{n}) > \mathbf{O}(\mathbf{c})$ transactions)
- Security (defined as being secure against attackers with up to $\mathbf{O}(\mathbf{n})$ resources).

With **c** referring to the amount of computational resources (e.g. bandwidth, storage, computation) and **n** referring to the size of the ecosystem. Currently, the choice between consensus-mechanisms (4.2) highly correlates with which two properties a blockchain platform is skewed towards (Baliga, 2017). In the case of the Ethereum blockchain (PoW consensus-mechanism), it is decentralization and secure, but lacks in scalability with respect to transaction-rate and transaction-costs (Ethereum-Foundation, 2014). In general permissioned blockchain platforms sacrifice decentralization in favor of scalability and trust (at least until this trilemma is solved), in for example the case of Ripple Network (Schwartz, Youngs, and Britto, 2014) which operates with a FBA consensus-mechanism.

4.4.2 Flawed Solutions

Buterin (2018) mentions three easy, but flawed solution categories for solving the scalability trilemma, being: Using many different altcoins, increasing the block-size limit and merge mining. The problem with using many different altcoins (many different blockchains) is that it greatly reduces security.
Increasing the block size limit can be in certain circumstances be a viable solution when block-size is the main limiting factor reducing scalability. However, ever-increasing the block size limit inevitably forces nodes using consumer hardware to drop out due to the increased required computational power. In this situation, only a relatively small amount of supercomputers would be running the blockchain, which can lead to great centralization risk.
The third is merge mining, a technique where there are many chains, but all chains share the same mining power (or, in proof of stake systems, stake). In theory this could greatly increase the throughput, in practice however, it greatly increases computational power and storage load required by each node. In fact, it is just a stealthy form of an increase in the block size limit and thus also decreases decentralization.

4.4.3 Potential Solutions

Fortunately, there are several potential solutions to the scalability trilemma in development. Sharding technology, off-chain state channels, side-chains, and plasma technology a are examples of these potential solutions (Poon and Buterin, 2017).

One of the main issues that the Ethereum blockchain has difficulties with scalability is that every node has to process all transactions and has to store the entire state of every account balance, contract code and storage. Although this provides a large amount of security it greatly limits scalability to the point that a blockchain cannot process more transactions than what a single node is capable of processing.

A possible solution to this is sharding. With sharding, only a small subset of nodes has to verify a subset of transactions (Buterin, 2018). So with the amount of nodes processing transactions increasing with N, this would, in theory, mean increased throughput scaling with N as well. The network remains secure as long as a sufficient amount of nodes verify each transaction.

Off-chain state channels are another potential solutions currently in development with the most prominent example being the Lightning Network (LN) on the Bitcoin Blockchain (Poon and Dryja, 2016). A state channel is a two-way communication channel between participants in a network which enable them to conduct interactions, which would normally occur on the blockchain, off the blockchain. What this will do is that it will decrease transaction time exponentially since there is no longer dependence on a third party like a miner to validate every single transaction. Once certain conditions are satisfied (e.g. certain time lapsed or the total amount of transactions done worth more than x US dollar) the state channel is closed. Once the state channel is closed only the final result of the entire set of transactions has to be validated by the main blockchain.

A sidechain is an independent cryptographic ledger that is linked directly to the main blockchain without jeopardizing its speed and performance (Back et al., 2014). Those side-chains can be seen as customizable instances secured by the main blockchain. Possible unwanted actions on a side-chain don't negatively affect the main blockchain. In addition to this side-chains can improve the interoperability between different blockchains as well by utilizing different aspects from several blockchains in their own instance. Lisk is the first blockchain platform which utilizes the concept of sidechains to extend the scalability of the system without undermining the overall speed and performance. Each sidechain is customizable and they are protected by a group of 101 master nodes. These master nodes use the same proof-of-stake (PoS) mechanism as used by the parent Lisk network.

Another promising technology that could scale blockchains is plasma technology (Poon and Buterin, 2017). Like state channels, Plasma is a technique for conducting off-chain transactions while relying on the underlying Ethereum blockchain to ground its security. However, plasma takes this one step further by allowing for the creation of child-chains (rather similar to side-chains) attached to the main Ethereum blockchain. These child-chains can have lower child-chains of itself as well, and so forth. Plasma is basically a many branching blockchain linked to one root blockchain. First, a set of smart-contracts is created on the Ethereum main-chain that serves as the root for so-called plasma child-chains. The Plasma root contains the basic state-transition rules of these child-chains, records hashes of the child-chains state, and serves as a bridge that lets users move assets between the Ethereum main-chain. These child-chains can have its own consensus algorithm (for example PoS) independent from the main Ethereum-chain when created.

These different scaling possibilities should not be seen as exclusives to each other, but rather as complementary. With certain weaknesses of a potential solution being mitigated by the strengths of another technology.

Besides the different consensus-algorithms, Smart-Contracts and different scalability solutions there are currently more technologies related to blockchains. Several of these technologies will be discussed in this section related to privacy, resilience, security and incentive in the form of tokens.

4.5 Privacy

At the moment privacy is a well-discussed topic, especially with the European GDPR regulation coming into effect in 2018 (European-Union, 2016). Privacy and blockchain make up for an unusual combination. As explained in Section 4.2 the information on public permissionless blockchains are openly visible for anyone. Naturally, this openness of information conflicts with privacy regulations. Especially in industries like for example health-care and banking it is not unthinkable that there is a preference for keeping information private when required. As explained in section 4.2 not all the data on each blockchain is completely publicly visible to anyone since there is a distinction between private and permissioned blockchains. In these blockchains with adjustable access control, a potential conflict with privacy is easier to avoid than in public blockchains. Fortunately, there are several features that enable (public) blockchains to improve on the privacy of its user. Examples of these features are Zero-Knowledge Proofs, Zero-Knowledge Succint Non-Interactive Argument of Knowledge (zk-SNARK), ring-signatures and confidential signatures (Samman, 216; Coinbureau.com, 2018).
Zero-Knowledge Proofs are a cryptographic method in which one party (prover) assures another party (verifier) that they have knowledge of value X without revealing the actual value (Koens, Ramaekers, and Wijk, 2017). Zk-SNARKS are a variant of zero-knowledge proofs in which no interaction is necessary between prover and verifier. In practice, this entails that confidential information can stay confidential while still meeting an earlier specified goal. Ring-signatures make it possible to specify a set of possible signers of a transaction without revealing which member actually produced the signature, thus making the issuer of a transaction practically anonymous.

4.6 Tokenization

4.6.1 Bitcoin economic model

With the introduction of Bitcoin, a new economic model was created in the form of mining in the PoW consensus mechanism as explained in section 4.1 (Nakamoto, 2008). In order to keep the Bitcoin network running peers had to allocate their computational power. However, this mining is a resource-intensive task which requires significant capital investments in the form of powerful hardware. To provide these miners incentive to keep securing the network those who contributed received periodically rewards. This reward is based on a fixed number plus the transactions fees paid for by the users to execute their payments. These miner rewards were paid out in the native currency of the network, namely Bitcoins. However, the fixed portion of the reward gradually decreases over time towards zero. This makes the economic model of Bitcoin deflationary by nature and increases the incentive for miners to keep validating transactions. This is based on the assumption that following the laws of supply and demand, the value of Bitcoin will go up when faced with an increase in demand.

4.6.2 Token Classification Framework

At present Bitcoin is far from the only cryptographic coin available (CoinMarketCap.com, 2013). New coins continued emerging by either hard-forks from the Bitcoin source-code or entirely new blockchain projects. A lot of these new projects

are largely facilitated by blockchain platforms such as (mainly) Ethereum, NEO and Waves Platform. These platforms enable the creation of new cryptographic coins and tokens. The main difference between coins and tokens is the underlying structure. Coins are independent valuta and have their own blockhain, while tokens are created on existing blockchains. Tokens, therefore, depend on the functionality of the underlying blockchain. These tokens can also be classified in different token types under five different dimensions according to the Token Classification Framework (TCO) (Euler et al., 2018), being: *Technical Layer, Purpose, Underlying Value, Utility and Legal Status*.

The *Technical Layer* dimension refers to on which technical layer a token is implemented. Tokens can be the chain's native token, be a crypto economic protocol that sits on top of the non-native protocol or can be on the application level as dApp token. Bitcoin and Ethereum are for example the native tokens of their blockchain.
The *purpose* dimension refers to the token's main purpose and what it is designed to do. Tokens can be either classified as cryptocurrencies (medium of exchange or store of value), network token (provide functionality within a network) or investment token (promise owners a share of asset value in the issuing entity). An example of a cryptocurrency is Zcash (ZEC) which is used as a medium of exchange with an extensive set of privacy features for the user.
The *Underlying Value* dimension refers to the source of the token on which the value is determined. For example, Asset-backed tokens function as a claim on an underlying asset, such as precious metals like silver or gold. A token's value can also be tied to the value and development of a network as Network value tokens. These tokens are closely intertwined with key interactions of network participants. The most prominent example of tokens that derive its value from a network is Ether on the Ethereum blockchain. At last, a token can also derive its value from being the cryptographic equivalent of a share. These Share-like tokens could pay dividends or share profit with the token-holder.
The *Utility* a token provides can either be access to a digital service (Usage token), the right to contribute to a system (Work token) or a combination of both (Hybrid token). The REP token in the Augur dApp is an example of such a work token. This Augur dApp is a prediction market based on the principle 'wisdom of the crowd' and assumes a large group of people can predict the future better than a single entity (Peterson et al., 2018). The REP token provides the right to submit a certain prediction on this decentralized application.
The final dimension refers to the *Legal Status* of a token. The three types a token can be under this dimension are Utility tokens, Security tokens, and Cryptocurrencies. Again, Cryptocurrencies act as a medium of exchange or store of value. security tokens possess security-like features e.g. right to vote on decisions regarding the issuing entity, share profits or pay dividends and offers little to no utility. And utility tokens avoid these security-like features but are rather tied to the issuing network or application. Due to the volatility of the cryptocurrency environment the definition of these types can change significantly when new expected regulation emerges.

Archetypes

Based on the TCO (Euler et al., 2018) four token archetypes were extracted. These archetypes are based on rather obvious correlations between different token types. For example, when the *purpose* of a token is to be used within a network it also derives it *underlying value* from this network and thus is classified as network value token. And share-like tokens (*underlying value*) with as *purpose* to be an investment

are likely to be classified as Security tokens in the *legal status* dimension. These four identified archetypes with their respective properties are mentioned in Table 4.2. In the context of this research, a blockchain platform that doesn't allow for the creation of new tokens the respective token is classified under a certain token type per dimension according to the TCO. If a blockchain platform supports the creation of new tokens it provides all the token types under all dimensions next to the token types relevant to the blockchain itself. This is due to the simple fact that all these token types could be created relatively effortlessly in practice. For example Ether on the Ethereum blockchain is a Native token (*Technical Layer*, Network token (*Purpose*, Network Value token (*Underlying Value*, Hybrid token (*Utility* and both a Utility and Cryptocurrency (*Legal Status* (Ethereum-Foundation, 2014). However, all kinds of tokens belonging to different types categorized under the five dimensions were created on Ethereum, such as for example the REP token (Augur Application). Tokens created on the Ethereum blockchain are so-called ERC-20 tokens of whom more than 90.000 (roughly 500 publicly traded) have been created so far (Etherscan.io, 2018).

Archetypes	
Cryptocurrency	Used as store-of-value or means of payment; unit of account Not issued by a central authority Can be mineable or pre-mined
Tokenized Asset	Gives access to assets like gold, even in a micro transaction scale The underlying asset needs to be held by the issuing party Thus introduces counterparty risk, contrary to cryptocurrency
Tokenized Platform	Platform-like network, not owned & operated by a single entity Before users had limited roles in a platform, now roles are distributed and available to every network participant Value (financial/utility) flows freely through the network
Token–as-a-share	A tokenized instrument to invest in companies that has characteristics of stock and currency Shares on steroids: flexible, programmable via smart contract Currently a highly uncertain token class as regulatory frameworks are only beginning to emerge

TABLE 4.2: Four token classification archetypes (Euler et al., 2018)

4.7 Resilience and Security

Due to the immaturity of the blockchain technology and related innovations, security and resilience of blockchains remain a vexed topic. This is indicated by e.g. a denial-of-service attack on the IOTA network (Buntix, 2017), several Sybil attacks on the Verge blockchain (Sedgwick, 2018), the different Bitcoin hard-forks (Reiff, 2018) and the DAO-hack on the Ethereum blockchain (Siegel, 2016).

When a Sybil attack (also referred to as 51 percent attack) occurs a single adversary is controlling multiple nodes on a network. And for the rest of the network it is unknown that these nodes are controlled by the same adversarial entity (Douceur, 2002). For example, an adversary can control multiple virtual machines, IP addresses or computers. In centralized systems, these Sybil attacks are typically avoided through heuristics that do not provide cryptographic assurance of Sybil resilience. For example, a centralized entity may prevent Sybil attacks by disallowing individual IP addresses from creating more than a specific number of accounts in a given interval. In the Bitcoin blockchain, Sybil attacks are avoided by requiring the

ability to generate blocks proportional to the computational power available for the PoW consensus mechanism. An adversary would require more significant investments to control more than 50 percent of the computational power of the network.

A denial-of-service attack on a blockchain could occur when a large number of requests are sent to a node within the network making it so busy it cannot process normal transactions, thus clogging the network. The bitcoin network has some prevention built-in on the protocol level to avoid the less sophisticated denial-of-service attacks (Narayanan et al., 2016). Examples (but not limited to) of preventing this are: Banning misbehaving IP-addresses, not forwarding transactions or blocks twice to the same peer and restricting the maximum number of signature checks a transaction input may request. In blockchains where authentication and authorization is required malicious participants can simply be denied access beforehand or be kicked from the network, thus preventing Sybil and denial-of-service attacks.

With differences between blockchain platforms being resistant against these possible threats or not, these might be important features to consider in the selection process.

Chapter 5

Data Collection and Artifact Creation

This chapter describes the whole process of collecting the required data for the creation of the BPDSS artifact. According to the research method described in Chapter 3 the required information for the creation of the artifact were: Domain Features, Domain Alternatives, FA-mapping, SF-mapping and miscellaneous data such as Domain Feature definition and Numerical Feature parameters. This chapter will describe the gathering-process for all this data. All excel-sheet files mentioned in this chapter (as Appendices) can also be found anonymized online here.

5.1 Domain Features Gathering Process

5.1.1 Boolean Feature Process

Section 4.1 until section 4.7 describe the most prominent related technologies in the blockchain domain. From this literature, an initial set of domain features were extracted. This set of features is mentioned in Appendix A1.3 and the most left column in Appendix B. This initial set of domain features was used as a starting point in the expert interviews. This initial list of Domain Features consists of 76 different boolean features, only informally sorted by category. This list of features was included in the Interview Protocol. As described in the interview protocol (section 3.6.1) the opinion of 9 domain experts was asked for each of these features + additional features the experts came up with. Appendix B shows this process of adding and removing features after each interview. The names of the interviewees are anonymized in Appendix B. In the first interview, 13 new features were added, 5 removed. In the second interview, 7 new features were added and 3 removed. 6 new features were added after the third interview. After interviews 4, 5 and 6 a total of 22 features were removed and only 1 added. Based on the 7th, 8th and 9th interview no additional changes were made to the list of features. After this process, there were 75 boolean features (from which 10 higher level category-features) considered important enough for further analysis of importance. This further analysis was in practice analyzing the voice-recordings of the 9 interviews again. For each interviewee and feature combination, it was determined whether a feature should be included in the BPDSS or not. The anonymized results of this are shown in Appendix M. A green '1' for a feature/expert combination means the expert from that interview considered that feature generic and important enough. The average of these 9 interviews for each feature is calculated in column C. If the value of a feature was above 0.5 it was considered a generic feature. All these generic features are colored green in the third column of Appendix M. In the time lapsed between the different interviews a few

more possible important features (or renaming of features) were found through literature and documentation as well as a possible categorization of the features. These possible additions can be found in Appendix B, the most right column. The final step in gathering the final set of categorized boolean features was the opinion of one of the most knowledgeable domain experts on the draft set of categorized features. This expert was determined to be the most knowledgeable due to this expert's extensive scientific research in the blockchain domain. Based on this opinion the last small changes were made which resulted in the final set of categorized boolean features, shown in Appendix C.

5.1.2 Numerical Feature Parameters and mapping against Domain Alternatives

The numerical features (Maturity, Popularity, Transactions Speed and Innovation) in the BPDSS all required a set of parameters to determine the value for each of these numerical features. The starting point for this set of parameters was largely inspired by Farshidi et al. (2018). This section will further elaborate on each of these Numerical Feature Parameters as well as the data collection and mapping of these features against the Domain Alternatives.

Maturity

For maturity, the initial set of parameters were: Yearly revenue, founding year and the number of employees. A higher revenue, the earlier year the platform was founded and a larger number of employees would result in a higher maturity. Only relatively small changes were made to this set. Based on the Expert interviews the majority of the experts indicated that currently the maturity of a platform is highly correlated to the specific consensus-mechanism of a platform. For example, platforms using PoW would be considered the most mature, since this consensus-mechanism (although with its drawbacks) has been the most battle-hardened. Thus platforms using PoW would get the highest score with respect to consensus-mechanisms, BFT-variants slightly lower and all other consensus-mechanisms the lowest score. In the end, the four parameters for maturity were: Number of employees, Yearly Revenue, consensus-mechanism used and founding year. The yearly revenue data and the number of employees were largely gathered on Owler (and LinkedIn) and the founding year and used consensus-mechanism the in the white papers of the different Domain Alternatives. In the, end 6 Domain Alternatives received a high maturity score, 8 Domain Alternatives a medium maturity score and 14 Domain Alternatives a low maturity score.

Popularity

The starting set of parameters for popularity were similar to Farshidi et al. (2018), being: Number of Google, Yahoo, and Bing searches, Twitter Tweets, Followers and Following, and professionals mentioning it on StackOverFlow, Indeed and Simplyhired. In the final set of parameters, the Yahoo and Bing searches were removed and only monthly Google searches used as a parameter. The three twitter parameters were made just one, being the number of twitter followers. Indeed, Simplyhired and StackOverflow mentions were replaced by the number of people mentioning the platform on LinkedIn and the amount of Reddit-subscribers. Experts indicated that another way to measure the popularity of a platform would be the number of

daily transactions + operations executed for a platform and the current market capitalization. However, a platform can only have a market capitalization if it has a cryptocurrency. In the case of platforms not having a native cryptocurrency, this parameter was not included in the final score. In addition to this, for several private permissioned platforms no data was available about transactions + operations executed per day, thus was not included. A platform would receive a higher popularity score if it had a higher market capitalization (if applicable), higher transactions + operations executed per day, more Twitter followers, more Reddit subscribers, more monthly Google searches and more LinkedIn followers. The number of subscribers or followers were found on the respective social mediums (Twitter, Reddit and LinkedIn). The monthly Google searches data was gathered using the Keywordseverywhere Google-chrome add-on and searching for the relevant platforms. The market capitalization for each platform was gathered on Coinmarketcap. The daily transactions + operations data was mainly found on Bitinfocharts. In the end, 4 Domain Alternatives received a high popularity score, 12 Domain Alternatives a medium popularity score and 12 Domain Alternatives a low popularity score.

Innovation

Innovation was similarly to popularity largely inspired by Farshidi et al. (2018), but without the DBMS-specific innovations going on. Instead of the DBMS-specific innovations, these were replaced with the most prominent developments in the blockchain domain, being: Plasma technology, Sharding, Cross-chain interoperability and Zero-knowledge proofs. In addition to this internet-of-things, artificial intelligence, supply-chain management, and financial sector focus are important sectors/related technologies which could greatly benefit from blockchain cooperation according to expert interviews. The last addition that would determine whether how much innovation is going on on a platform is whether there is a consortium of companies supporting research. Several companies supporting a platform with financial aid and knowledge greatly benefits a platform according to the expert interviews. The data for the plasma, sharding, cross-chain interoperability and zero-knowledge proofs was gathered based on the whitepapers and roadmaps for each platform. The focus on certain sectors/technology was based on these whitepapers as well in addition to Medium blogs, Coindesk articles and expert interview knowledge. In the end, 4 Domain Alternatives received a high innovation score, 13 Domain Alternatives a medium innovation score and 11 Domain Alternatives a low innovation score.

Transaction-Speed

There was no initial starting point with respect to the parameters for the numerical feature transaction speed. Section 4.4 already describes scalability and transaction speed in large detail. Based on this and expert knowledge the following parameters were determined: confirmation time, the relative speed of consensus-mechanism and number of scalability technologies implemented. As stated in table 4.1 currently the chosen consensus-mechanism largely determines the transaction speed of a blockchain. Should the transaction speed of a consensus-mechanism be rather low it can still be offset by the implementation of the different scalability technologies as described in section 4.4. The data for these parameters were gathered from the relevant whitepapers of the blockchain platforms and several Medium blogs. However, the data for these parameters are often theoretical claims rather than strict factual

data. In the end, 12 Domain Alternatives received a high transaction speed score, 9 Domain Alternatives a medium transaction speed score and 7 Domain Alternatives a low transaction speed score.

5.2 Domain Alternatives Gathering Process

The gathering process of the Domain Alternatives was largely similar to the boolean Domain Feature gathering process. Initially, only five Domain Alternatives were selected, being: Ethereum, NEO, R3 Corda, Hyperledger and Ripple. These alternatives were selected based on being currently the largest blockchain platforms market-cap wise (Ethereum, NEO and Ripple based on (CoinMarketCap.com, 2013)) or being considered most noteworthy according to literature and documentation for the platforms lacking native cryptographic tokens (R3 Corda and Hyperledger). These Domain Alternatives were discussed, according to the Expert Interview Protocol (Appendix A). Based on the interviews, literature, and documentation more alternatives were added or removed. This process of adding and removing Domain Alternatives is shown in Appendix D. After all the interviews were conducted and a list of alternatives was gathered all the alternatives were checked against the selection criteria as defined in section 3.6.2. Alternatives that didn't meet these criteria were removed, as shown in column K of Appendix D. In the end, 29 Domain Alternatives met these criteria and would be used in the next steps of this research.

5.3 Mapping

This section describes the two mapping processes between the Features and respectively Sofware Quality Aspects (SF) Domain Alternatives (FA).

5.3.1 SF-Mapping

To create the SF-Mapping, the relationship between the final list of Domain Features (Appendix C) and the Software Quality Aspects from ISO/IEC 25010 Ext. ISO/IEC 9216 (Carvallo and Franch, 2006) had to be mapped. Determining these relationships would again be based on Domain Expert knowledge extracted from interviews. Four of these interviews were conducted, from which three experts also participated in the Domain Feature gathering process. These three experts were familiar with this research and the fourth expert was selected based on extensive experience with developing in the blockchain domain.

The individual data for each of these interviews can be found online here in the SF11, SF12, SF13 and SF14 sheets. The average of these interviews can be found in Appendix F and in the SF.Avg sheet here. A green '1' indicates a certain Domain Feature has a positive influence on a certain Software Quality Aspect. An apricot-colored '0' indicates a Domain Feature has no positive influence on a certain Software Quality Aspect. Explicit negative influences are not mapped since those are irrelevant in this decision model. When the average values for each Feature-Software Quality Aspect combination is at least 0.5 in the Sheet that displays the averages (so half the experts, 2/4) a green '1' is considered.

5.3.2 FA-Mapping

To create the FA-Mapping, the relationship between the final list of (boolean) Domain Features (Appendix C) and the Domain Alternatives (Appendix E had to be mapped. Determining these relationships was based mainly on analyzing documents and auxiliary domain expert knowledge. These documents consist mainly of whitepapers describing each specific blockchain platform, updates in blogs by blockchain platform developers on Medium and other grey literature such as e.g. benchmarks from consultancy firms (for example this benchmarking study by Hileman and Rauchs (2017) supported by Visa and Ernst and Young). The final FA-mapping is shown in Appendix G and online here in the FABPDSS sheet. A green '1' indicates that a specific Domain Alternative supports a certain (boolean) Domain Feature. An apricot-colored '0' indicates a specific Domain Alternative doesn't support a certain (boolean) Domain Feature. Features that were just mentioned on the road-map of a platform rather than fully implemented yet were marked as not supported, thus an apricot-colored zero. The process of mapping the Numerical Features against the Domain Alternatives was already described in section 5.1.2.

5.4 Other Data and Artifact

5.4.1 Feature Definitions and Alternatives information

This section briefly elaborates on the other data that was gathered during this research. For each of the features included in the SF and FA mapping, a definition/description was added based on scientific literature or documentation. These definitions will be used as input as well from the FA mapping into the BPDSS. Should a decision-maker which is utilizing the BPDSS be not completely familiar with all the Domain Features these descriptions could aid. Appendix N shows the list of Domain Feature definitions. The second column in Appendix N shows the reference number. All these references can be found in Appendix O. For the Alternatives, the relevant website URLs were added.

5.4.2 Artifact creation and explanation

The final step of the data collection phase was to create a functional working artifact, in the case of this research the Blockchain Platform Decision Support System (BPDSS). The required data used for this were mainly the FA-mapping, SF-mapping and a file to indicate the categories for the features. These files were handed over to S. Farshidi to create the initial valid Blockchain Selection Model in the Decision Model Studio. In an iterative process together with Farshidi bugs, errors and wrong data were resolved to end up with the final valid decision model. Screen shots of the DSS can be found in Appendix P. Figure P.1 shows the DSS with the 11 different categories collapsed on the bottom left. In this screen shot the case definition is empty, thus no features are selected or prioritized. Therefore in this screen shot, all the 29 Domain Alternatives are Feasible Solutions, as shown on the right in red font. In Figure P.2 a set of features has been selected and prioritized thus reducing the number of feasible solutions from 29 to 13. The must-have and won't have features act as hard-constraints. So should an alternative not provide a feature in the case of must-have it is excluded from the list of feasible solutions and vice-versa in the case of won't have features. Should have and Could have features act as soft-constraints,

thus only influencing the score based on a certain weighting. This weighting is determined based on the SF-mapping and Should Have features get a higher weighting than Could Have features in general. Figure P.3 shows this with the top 10 solutions with their score based on the selection and prioritization of the features. Figure P.4 is another visualization of these results. Figure P.4 shows some other novel functionality of this DSS like 'Comparable Alternatives' as well. Figure P.5 shows an in-depth analysis when pressing on the 'Comparable Alternatives' button. In the example of Figure P.5, the IOTA alternative was excluded as alternative because it provides several won't have features and doesn't provide several must-have features. In addition to the difference in scores between feasible solutions can be analyzed. For example in Figure P.5 the alternative Cosmos Network provides two should have features while NEO doesn't, thus receiving a higher score despite being similar in must have and won't have features.
A typical usage of the BPDSS would be to select first the valid blockchain selection model, then define a new case, select and prioritize a set of features and then view and analyze the results. Figure P.6 shows the decision tree for a certain case study as well.

Chapter 6

Case Study Description

To evaluate the created BPDSS it requires domain feature requirements as input. The BPDSS will be evaluated by three case studies. Each of these three case studies is being described in this chapter. The central theme in each of these case studies is the development of a solution utilizing a blockchain platform for development. In each of the case studies the context (including, but not limited to problem, stakeholders and technology), domain Feature requirements and Domain-Alternative prioritization will be elaborated on. And the means of how the information of each case study was gathered will briefly be discussed.

6.1 Case Study 1: ShareCompany BIQH PRIIPs, a Dutch Fintech Solution

6.1.1 Context

Following Regulation (EU) No 1286/2014 ('Regulation 1286/2014') (European-Union-Commission, 2017), issuers of packaged retail investment and insurance-based products (PRIIPs) are by means of legislation compelled to lay down uniform format on key information documents (KID), which are documents connected to PRIIPs. In wake of the legislation, at the request of one of their customers, ShareCompany BIQH developed an information system that would help banks accommodate the requirements put forth by the European Union. Packaged retail investment and insurance-based products constitute an intentionally broad category (for sake of regulation), and encompasses all packaged and publicly marketed financial products that have exposure to underlying assets such as stocks, bonds, treasuries, etc. They have many properties, whereof the KID is but one. The purpose of the KID is to present essential information to the buyer about the product, in a way that is as unambiguous as possible. In other words, the KID is what investors are left with when information about PRIIPs has been trimmed for perplexing financial jargon. The concerning products are difficult to understand, in this way they are made more approachable to the general public, so that more people may benefit from them. The PRIIP issuing entities must ensure the correct and most recent KID has been shown to the investor at the moment of purchase of the PRIIP.

After a successful deployment of a centralized solution, Sharecompany BIQH now wants to investigate distributed ledger technology (DLT), with the existing information system as the starting point and use case for researching the blockchain technology. Several driving forces can be identified for the development of a dApp to replace, or complement, the current system. First and foremost, the value of general inquiry into the technology, adding to the body of knowledge in this field which at this point in time is lacking in some respects (Yli-Huumo et al., 2016). Second,

BIQH recognizes the general societal need for innovative technological solutions that have potential beneficial effects. Third, in the context of this specific case, a characteristic of distributed ledger technology necessitates that there is no one party reaping all the rewards or monopolizing, thus the nature of the application is such that it is intended to be used by everyone for their convenience, as opposed to fraudulent approaches advocated by big business for the sake of individuals. In other words, the choice of technology in itself contributes to the product owner's credibility, which ultimately contributes to the adoptability of the system. To elaborate, if both the public and government could agree on a single system for this type of information provisioning; both issuers and the AFM would have an easier time carrying out their functions and responsibilities. Moreover, inherent to the blockchain technology, it ensures quality attributes such as security, anonymity, and data integrity enabled via timestamped digital signatures, hash-based consensus and asymmetric (Public key) cryptography, which tackle identified challenges with the current system.

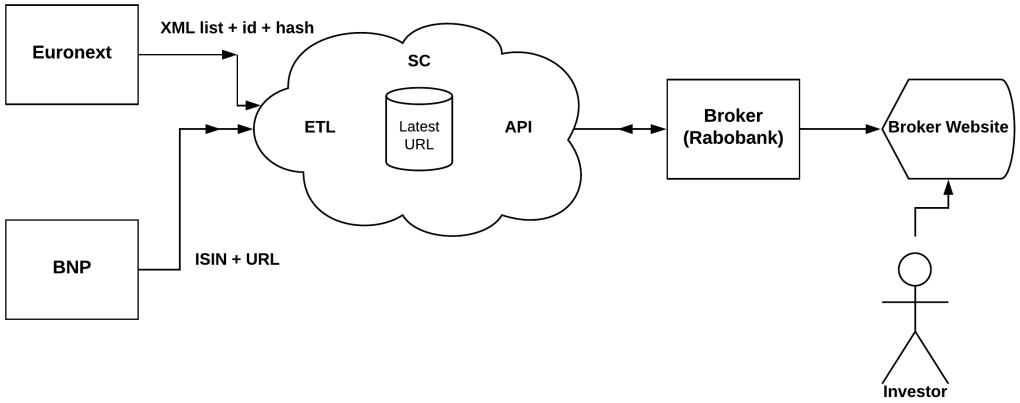


FIGURE 6.1: Current Centralized Solution by ShareCompany BIQH

Depicted in Figure 6.1 is the current information system developed by BIQH to streamline KID's from PRIIP issuers to investors. To the far left are the issuers who hold financial assets (PRIIPs) that are of interest to private investors - all of which have their associated KID. It is the issuer's responsibility to create and maintain this document, as they are the ones in charge of the product to which the document has relevance. The arrow from the issuer to ShareCompany (SC abbreviated) indicates a hand-over where a list of the issuers' financial assets in arbitrary format (e.g., XML) is sent. Given the varied nature of these lists, data transformation is necessary (ETL), as the end goal of the system is to pool and index the information and provide access to it by means of an API. The rightmost part of the diagram shows how the broker makes use of the system through an API, ultimately giving the broker's customers access to information relevant to them; upon PRIIP purchase especially, access to the KID. Here it is worth mentioning that although the issuer has responsibility for creation and maintenance of the document, it is the broker who must ensure that the customer reads, or agrees to have read, the KID upon purchase. Identified challenges of the current system includes (i) the need for data transformation given the lack of a standard form and format, (ii) errors in the actual documents due to the unique ways in which they are provided, (iii) and poor document version control as a result of inefficient channels for distribution. In the last case, this can result in a bank pointing to an outdated KID. However, ShareCompany has envisioned

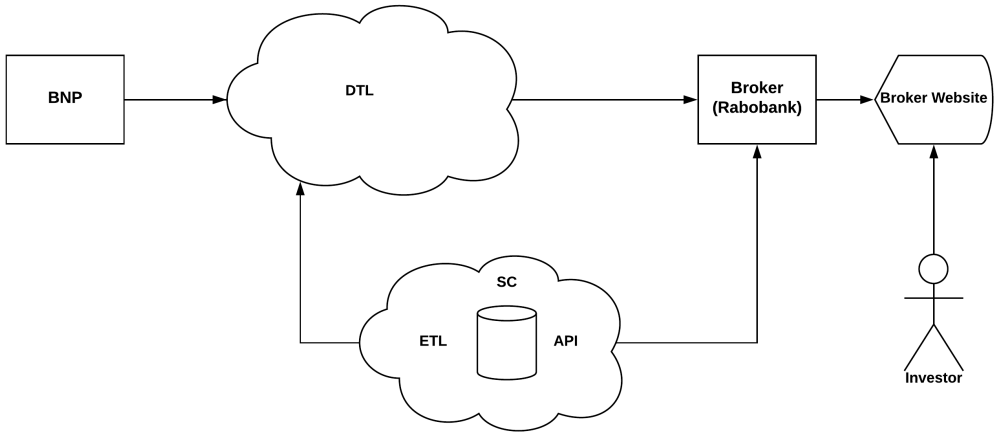


FIGURE 6.2: Envisioned decentralized application by ShareCompany BIQH

a decentralized application (dApp) solution as opposed to the current centralized solution. This decentralized solution is displayed in Figure 6.2.

The most notable change from Figure 6.1 to Figure 6.2 is the dis-intermediation, where we see that the role of ShareCompany (cloud at the bottom) has changed from essential to auxiliary. By utilizing distributed ledger technology, the information put on the blockchain is inherently structured and can be retrieved freely by anyone (with permission), meaning that services like data transformation and indexing are no longer strictly necessary. Instead, ShareCompany BIQH provides two distinct points of access (e.g. through an API) that can be used to input and retrieve data, by the information provider and receiver respectively. On the KID provider side (left-most) the issuer makes use of a client application designed for interaction with the blockchain. Here the issuer either puts in the necessary information through a form or uploads a file of a certain format, which is then parsed and put on the blockchain. In turn, the broker has another client application that extracts information from the blockchain and presents it to the user, possibly with the alternative to download in a specific file format. In Figure 6.2, there is no longer an inherent need for a third-party (e.g. ShareCompany BIQH) to perform ETL and data visualization, as these are services (say, a minimum of two minimal client applications for data input and retrieval) that would accompany the blockchain, the fact that the ledger is public does open up possibilities for third-parties to develop proprietary services based on the information found in the ledger.

6.1.2 Requirements and alternatives

Requirements and features

To fulfill the goals visualized in Figure 6.2 ShareCompany BIQH generated a list of Functional Requirements (Appendix H) for this project. These requirements were discussed with a ShareCompany representative and translated into feature requirements. In such a fashion that these feature requirements would fit as input for the BPDSS. Table 6.1 shows the required features prioritized based on the MoSCoW technique (DSDM-Consortium, 2014).

Must Have	Should Have	Could Have	Won't have
Permissioned	Golang	SNARKS	Proof-of-Work
Interoperability technologies	Private	Spam-attack resistant	Proof-Of-Stake
Smart Contracts	JavaScript	Virtual Machine	Directed Acyclic Graph
Java	Resilience Features	Turing-complete	
Sybil-attack resistant	Instant Transaction Finality	On-chain transactions	
Privacy Technologies	High Transaction Speed	Practical Byzantine Fault Tolerance	
Enterprise System Integration	Zero-knowledge Proof	Federated Byzantine Fault Tolerance	
Network Layer	High Maturity	Delegated Byzantine Fault Tolerance	
Application Layer	High Popularity		
Protocol Layer			

TABLE 6.1: Domain Feature Requirements for ShareCompany BIQH, prioritized based on the MoSCoW-technique (DSDM-Consortium, 2014)

Feature prioritization discussion

This section will discuss the prioritization of the Domain Feature Requirements as stated in Table 6.1. Not each feature will be discussed in depth as in many cases it comes down to a rather trivial decision in the case of many features categorized as 'could have' or are not categorized at all.

The envisioned solution requires extensive integration with current systems (e.g. APIs) therefore the features *Interoperability Characteristics* and *Enterprise System Integration* are grouped as must-have features. Since only a select number of participants should be authorized to make changes in the system a must-have feature is *Permissioned*. A *Private* system is not a hard-constraint since a lot of data is already publicly accessible, however it is still classified as a should-have feature. The *Protocol Layer*, *Network Layer* and the *Application Layer* are all categorized as must-have features. The *protocol layer* will support reaching consensus on the accuracy of the data, the *network layer* defines the communication between the different participants and the *Application Layer* will be used to build the required infrastructure to connect with current enterprise systems. The investors buying the PRIIP's should remain anonymous for participants such as BNP Paribas, MorningStar or even AMF since there is no valid reason why they should know a broker's customer by default. Therefore the envisioned solution requires *Privacy Technologies*. The must-have features *Smart-Contracts* in the *Java* programming language comes down to practical implications. The blockchain implementer is most proficient in programming in Java, thus this would prevent learning a whole new programming language. The final must-have feature is *Sybil-attack* resistant.

Since ShareCompany BIQH is operating in the financial data environment with large organizations such as the Rabobank the system should have both a *High Maturity* and a *High Popularity* in the market. This *High Maturity* and *High Popularity* should reduce unnecessary risks as much as possible in this new domain. A possible conflict emerges since the AMF has to check whether a broker has fulfilled his duty of showing an investor the right KID at the right moment. This would mean the AMF requires knowledge of a broker's customer. A feature to parry this potential dissension are *Zero-knowledge Proofs*, which is grouped as a should-have feature. Each morning it is possible that the KID document has been updated during closing hours of e.g. stock exchanges. So early in the morning, it is desirable that there is a *High Transaction speed* and *Instant Transaction Finality* to as quickly as possible process a large batch of changes/transactions. However, during the rest of the day the amount of changes regarding KID's is expected to be much lower and these features are deemed less critical. Thus, these two features are grouped as should-have

features rather than must-have features. In addition to Java the blockchain implementer is also quite proficient in *Golang* and *JavaScript* and acknowledges support for these programming languages should benefit the system. The way consensus will be reached is still undecided, however it won't be *Proof-of-Work* or *Proof-of-Stake* since there is no currency built into this system. A *Directed-Acyclic-Graph* is still deemed too experimental and immature. The most likely options for consensus-mechanisms are: *Practical Byzantine Fault Tolerance*, *Federated Byzantine Fault Tolerance* or *Delegated Byzantine Fault Tolerance*. These three options are all grouped as could-have features. The remaining features that are either not grouped, grouped as could-have features or grouped as won't have features all grouped based on trivial decisions not deemed important enough to specifically mention/categorize.

Case Participant Short-List Alternatives

ShareCompany BIQH selected two potential alternatives for developing a decentralized application appropriate for this case. Their first choice as a potential alternative is the Hyperledger Fabric project (Cachin, 2016). Strictly speaking, the Hyperledger project envelops several tools (Hyperledger Caliper, Cello, and Composer, Explorer and Quilt) and frameworks (Hyperledger Sawtooth, Fabric, Iroha, Indy and Burrow) from which the frameworks can be seen as independent blockchain platforms. However, for the sake of clarity, all these tools and frameworks are categorized under the term 'Hyperledger' in the BPDSS.

The second choice as a potential alternative is the Quorum blockchain platform from JPMorgan. Initially Ethereum would be the second choice behind Hyperledger, however it was identified that proof-of-work was undesirable and there would be no need for a token. Quorum did meet these criteria, thus was chosen as the second alternative. This is summarized in Table 6.2.

Domain Alternative	CP Rank
Hyperledger	1
JPMorgan Quorum	2

TABLE 6.2: ShareCompany's Short-List Alternatives

6.1.3 Data Collection

The context regarding the PRIIP/KID regulation was partially derived from official documents from the AFM and the European Union. The software architect from ShareCompany provided additional information in how this relates to them. This was explained until lower-level details. From this, the currently implemented solution was derived. Another representative from ShareCompany responsible for further exploring the possibilities of the blockchain technology suggested the envisioned solution. The feature requirements were generated by the software architect and the blockchain implementer. Together with the blockchain implementer, these requirements were translated into Domain Feature requirements prioritized according to the MoSCoW-technique (Table 6.1). In a similar way, together the Domain Alternatives short-list was established (Table 6.2). The blockchain implementer will also be the person evaluating the output from the BPDSS.

6.2 Case Study 2: Dienst Uitvoering Onderwijs, Dutch student financing system

6.2.1 Context

DUO is the administrative and executive agency of the Dutch government for managing the educational system (Rijksoverheid, 2010). The agency was established in 2009, when the CFI, a financial agency, and IB, an information technology agency, were merged. The goal of the agency was to improve the educational service, by decreasing the administrative burden and increasing the quality of technological services. DUO operates in the name of the Ministry of Education, Culture and Science and the Ministry of Social Affairs and Employment. DUO has eight different main functions with several activities as their core focus. However, this case study will merely focus on the process of student financing in the form of granting loans.

DUO (under their new system starting this year) gives out different kind of loans to students, being: Regular loans, Tuition fee loans, and supplementary grants. Secondary vocational education level or higher students that satisfy the following conditions are eligible for regular loans: Be enrolled in a full-time or dual study, be below 30 years old, have a Dutch nationality or have a residence permit type 1, 3, 4 or 5. However, in the exceptional case that a student earns more than 14.456 Euro in a year the student is not eligible anymore for a student loan. Should a student violate this rule, the excess money above the upper bound has to be paid back to DUO.
The maximum amount per student per month for the regular loan is determined at the beginning of the year, for 2018 this is 870,46 Euro per month. Students can increase (up until the maximum amount) or decrease the amount they borrow before the first of that month.
In addition to the regular loans all students are also eligible for tuition fee loan, used to support students paying their tuition fee. Therefore the height of the tuition fee loan is determined based on the tuition fee for that specific year.
Should a student's parent's income be below a certain threshold, students can apply for a supplementary grant as well. The height of this supplementary grant is higher when the income of the parents is lower, up to a maximum of 360 Euro per month.
The eligible student loan duration or supplementary grant depends on the type of bachelor and/or master. The maximum duration a student can receive a loan is 10 years. In the case the student acquires its degree within 10 years a certain amount of years of the supplementary grant can be remitted.
In 2017, around 714490 students were active in The Netherlands. 94 percent of these students received one or more types of student financing from DUO. On average a student received monthly 559 Euro, compared to 2015 this was 459 Euro monthly.

Currently, students can apply for the different student loans through the website of DUO. However, Utrecht University student G. De Jonge explored the possibilities of utilizing the blockchain technology in this case for the fulfillment of his bachelor thesis (Jonge, 2018). According to the design science cycle of Wieringa (Wieringa, 2018) De Jonge created a Proof-Of-Concept for a decentralized application built utilizing the blockchain technology for the case of student financing by DUO. In this thesis project interviews were conducted with the relevant stakeholders, being representatives from: DUO Innovation Lab, LiteBit, Cyber Capital, Nibud and Foundation Forus. The Innovation Lab of DUO is the innovation unit from the extensively discussed DUO organization. LiteBit is a Dutch cryptocurrency exchange, Cyber Capital is a Dutch company that specializes in cryptocurrency investments, Nubid

is an independent consultancy agency in the Netherlands that researches financial matters of Dutch households and Foundation Forus is an independent foundation that develops blockchain applications. The goal of these interviews was to identify the stakeholders, map the functional requirements, Quality Requirements and constraints. Based on these interviews a design was proposed composing of a functional viewpoint, economic viewpoint, organizational viewpoint and a contextual viewpoint. In this contextual viewpoint, the different Domain Alternatives are discussed alongside arguments for and against developing on certain blockchain platforms. Based on this decision De Jonge created a proof-of-concept and a prototype that would align according to the features provided by this platform.

6.2.2 Requirements and alternatives

Stakeholders, Requirements, and features

De Jonge identified in his research the following three stakeholders: Students, Financier and Government. Students will be using the new system mainly to apply and receive their student loans. A student can choose whether they want to receive their loan in either cryptocurrency or fiat currency. In addition to this students can also repay their loans when they either completed or terminated their higher-education.

The financier will maintain the artifact, use it to handle student applications, grant loans to the students, pay the students, handle repayments of loans and maintain the artifact to be compliant with government regulations. The government determines the protocols and regulations that are being handled within the financier. An example of this is determining the most appropriate interest-rate on the student loans. The financier on his turn can and has to make these changes in the system. The government is the last responsible entity for both societal and financial consequences. Should in the rare occasion no active students be left anymore, the financier can kill the deployed smart-contract and retrieve all the assets that are still stored in the smart-contract.

A set of user stories as described by Lucassen et al. (2015) was created by De Jonge each belonging to one or more of these stakeholders (Appendix I). Based on the expert interviews and the set of user stories the functional requirements, quality requirements and constraints (Appendix I) were determined for the proof-of-concept.

Based on the requirements of the proof-of-concept, de Jonge himself prioritized the Domain Features based on the MoSCoW-technique which are shown in Table 6.3.

Feature prioritization discussion

This section will discuss the prioritization of the Domain Feature Requirements as stated in Table 6.3. Not each feature will be discussed in depth as in many cases it comes down to a rather trivial decision in the case of many features categorized as 'could have' or are not categorized at all.

The DUO financing artifact requires the three layers of a blockchain (*protocol, network and application layers*). The protocol layer will determine the rules of transactions and consensus within the system. However, there is no strict requirement for a specific *consensus-mechanism*. Therefore all the *consensus-mechanisms* are categorized under 'could have'. The *Network Layer* is required to accommodate all the users grouped under a certain stakeholder category. And the system itself is built on

Must Have	**Should Have**	**Could Have**	**Won't have**
Protocol Layer	Turing-complete	Proof-of-Work	Directed Acyclic Graph
Network Layer	JavaScript	Proof-of-Stake	
Application Layer	High Maturity	delegated Proof-of-Stake	
Smart-contracts	Native token	practical Byzantine Fault Tolerance	
On-chain transactions	Cryptocurrency (purpose)	federated Byzantine Agreement	
Cryptographic Tokens	Solidity	delegated Byzantine Fault Tolerance	
Sybil attack resistant		Proof-of-Authority	
Spam-attack resistant		Proof-of-Elapsed Time	
		Public	
		Private	
		Permissioned	
		Permissionless	
		Virtual Machine	
		Java	
		C++	
		Zero-knowledge Proof	
		SNARKS	
		Hard-fork resistant	
		Quantum resistant	
		Instant transaction finality	
		Medium Popularity	
		Medium Innovation	
		High Transaction speed	

TABLE 6.3: Domain Feature Requirements DUO, prioritized based on the MoSCoW-technique (DSDM-Consortium, 2014)

the *Application Layer* so the user can interact with it.*Smart-contract* support is a 'must have' features since this mainly influences the functionality of the system. For example, the *smart-contract* handles paying out the loans each month if a certain date has passed, grants regular loans to students if they meet the specified conditions or deny supplementary loans to students which try deceiving the system. These pay-out of loans can be either done by the system in fiat currency (Euro's) or in the form of *Cryptocurrency (purpose)* which act as *Native token* to the system. The transactions will be executed as *on-chain transactions*. Both the purpose of the token and the technical layer of the token are classified under the higher-level feature *Cryptographic Tokens*, which therefore is a must-have feature. The lower-level token classification is less important thus classified as should have features. Whether the system has to be *public* or *private* and *permissioned* or *permissionless* still has to be decided upon. One side of the argument is that the government might not prefer having everything public with privacy in mind but the other side of the argument is that it greatly increases transparency and possibly credibility, as indicated by Cyber Capital. Therefore these features are for the moment categorized as 'could have'. Programming the system in the *Solidity* language is a 'must have' feature since *Solidity* is currently the most common programming language to create *smart-contracts* and is specifically designed for it. The last two 'must have' features are *spam-attack resistant* and *Sybil attack resistant*. These two are required to guarantee a base level of security and resilience. *JavaScript* support should be nice to have however it is not completely necessary. The same applies to being *Turing-complete* and the platform having a *High Maturity*. DUO opts for a platform as mature as possible since the whole blockchain domain is still immature compared to other technologies. A medium or low maturity would bring additional unnecessary risks. The system won't use a *Directed Acyclic Graph* for now since it's considered too immature. The remaining features that are not discussed in detail are categorized by de Jonge as 'could have' features.

Case Participant Short-List Alternatives

Table 6.4 shows the Short-List for Alternatives considered by DUO to be viable to develop their solution on. The three main platforms that were considered were Ethereum, NEO and Hyperledger. Hyperledger, although deemed mature and offering a broad range of features (Cachin, 2016) there is no build-in cryptocurrency on this platform. Regarding development this would make things unnecessarily complicated, therefore Hyperledger was ranked third on the short-list. The other two alternatives (Ethereum and NEO) both offer this built-in cryptocurrency but differ on other aspects. Ethereum was considered to be the most developed of the two alternatives and therefore ranked as the most desirable solution on the short-list. However, NEO offers a higher scalability at the cost of less decentralization so was an interesting second choice as well.

Domain Alternative	CP Rank
Ethereum	1
NEO	2
Hyperledger	3

TABLE 6.4: DUO's Short-List Alternatives

6.2.3 Data Collection

For a large part, the data from this case study was initially collected by G. De Jonge in fulfillment of his bachelor thesis at Utrecht University. De Jonge created in collaboration with the important stakeholders for his system the user-stories. In addition to this, he derived the functional requirements, quality requirements and constraints for his system. The author of this paper together with De Jonge translated these different requirements into the domain feature requirements prioritized according to the MoSCoW-technique. The Domain alternatives were discussed in a similar fashion. The results of Tables 6.3 and 6.4 will be used as input for the BPDSS.

6.3 Case Study 3: Veris Foundation, USA Healthcare Claims Processing

6.3.1 Context

The Veris Foundation is an organization focusing on the American healthcare system. One of the most heavily regulated and fractured markets in existence is the current healthcare market in the United States (Plance and Lawlor, 2018). Unnecessary expenses are added for everyone (and especially patients) due to an abundance of redundant processes between different parties such as providers, insurers, and patients. These unnecessary expenses are estimated to be above 59 billion dollars per year as mentioned by the Veris Foundation. They are under the assumption this fragmentation is a result of the different stakeholders unwillingness to assume the risk associated with designating an intermediary to handle the processing of data related to healthcare services between all stakeholders. This means that all the stakeholders duplicate processes which could be executed by a central authority as well, thus reducing redundancy. However, moving these processes based on contemporary technologies would require an overwhelming amount of trust in this central authority. This is in the current American healthcare landscape no viable solution

for the reasons mentioned. A blockchain solution would allow users to interact with each other without relying on the trust of a single entity. All transactions would be completed with absolute certainty, thus allowing for a versatile system capable of replacing the numerous fragmented systems in the current situation. The main process that would be revised is processing claims.

Stakeholders, process, and motivation

The Veris Foundation discerned the different parties operating in the activity of processing claims, which is shown in Figure 6.3. In addition to this, they identified the motivation of each of those group, and how these can be satisfied in a mutually beneficial manner. The different parties can be grouped under: Providers, Insurers, Payers and Financial Institutions (Banks). In Figure 6.3 Providers establish the identity of the patient to be treated, thus the eligibility of the patient to be treated based on the patient's insurance plan. The result of this step is the ability to alert the patient to pay alongside for the service should this be required. In some specific cases provider request a pre-authorization for specific procedures. A payer can either confirm or deny the authorization for this procedure. Once the procedure has been performed by the provider the claim is submitted to the patient. The claim processing step is an interim step used to determine if a provider will receive payment for a service from a payer based on certain criteria. In the claim Payment step, the claim is paid by the patient to the provider, usually bundled together in a batch of transactions. The post-payment review step is an optional step depending on whether a payer wants to make adjustments to a payment after the fact. In most cases, a post-payment audit is triggered by a set of criteria run against all claims processed. With the implementation of smart contracts, this step can be completely replaced.

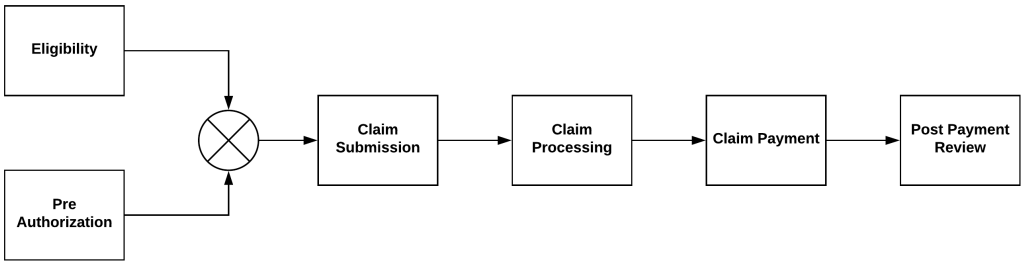


FIGURE 6.3: Claim processing procedure

Providers would benefit from a new decentralized system since currently 5-10 percent of healthcare providers their total expenses are processing claims. Veris can reduce this by creating free market forces when utilizing DLT. These free market forces will drive expenses down to a point where the market determines the added value of processing claims.
Payers are provided with greater detail and transparent data which can be used within their actuarial models and could lead to better forecasting. Insurers benefit in a similar way as providers do by cutting unnecessary expenses and increasing revenue. In the American healthcare industry these insurers are under heavy shareholder pressure to produce higher returns.

6.3.2 Requirements and alternatives

Requirements and features

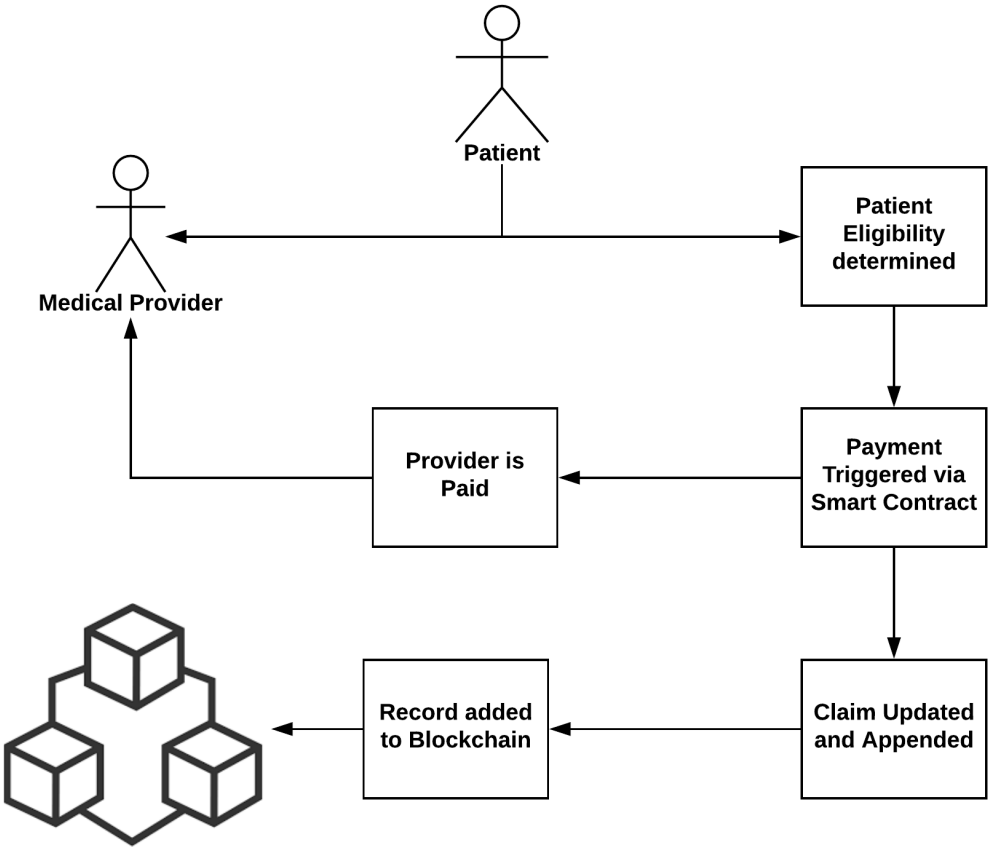


FIGURE 6.4: Envisioned Solution by the Veris Foundation

The Veris Foundation came up with a blockchain solution to replace the current process (Figure 6.3) with a new solution modeled in Figure 6.4. In this envisioned solution a lot fewer steps are required, and the ones required are largely automated. At the core of automating this solution are smart-contracts and a Proof-of-Stake consensus algorithm. In the Payment Triggered via Smart Contract step in Figure 6.4 the smart-contract either approves the payment based on information provided or flags the payment for review. After this, the claim is created and updated onto the blockchain by the smart-contract. Connected to the blockchain are all the relevant stakeholders (not visualized in Figure 6.4.

The utilized PoS implementation is forked from the NEO blockchain consensus-mechanism which is delegated Byzantine Fault Tolerance. But, Byzantine Fault Tolerance can be easily adapted to a PoS implementation like Veris did in this case. In this specific implementation (just like as in the NEO-blockchain) the stake/shares in the network are split from the network fees by creating two separate currencies; VeriStakes (VRS) and VeriCoins (VRCO). These VRS act as share within the Veris network, they pay out a dividend in the form of VRCO and offer the right to vote in the network on potential adaptations. The amount of VRCO awarded is directly related to the number of shares held as a proportion of the total VRS in existence.

These VeriCoins are used to execute the smart-contracts on the Veris network. In their solution Veris created six smart-contract types based on the MedRec Project at MIT Lab (Azaria et al., 2016), being: Identity contract (ID), Summary contract (SC), Patient Payer contract (PPaC), Patient Bank Contract (PBC), Payer/Provider contract (PaProC) and Patient/Provider Contract (PProC).
ID provides the public/private key pair for all the identities on the network and pointers to the SC. The SC is a list of all the contracts for which this identity has some level of participation, as well as the status of those contracts. the PPaC is a contract between a specific patient and a specific payer. This contract has all the permissions for retrieving the contract, as well as the queries necessary to access the providers Electronic Health Records. The PBC is a contract between a patient and their bank which contains all the rights for accessing the contract as well as access to the patient's Healthcare savings account (HSA). the PaProC is a contract between a payer and a provider which contains all the rights for accessing the contract, the database queries to access the payer's database and account information for processing payments to the provider. The last type of contract (PProC) is between a patient and a provider that is used to ensure the patient-provider relationship and authorizes access to the Payer/Provider Contract on-chain. Based on this envisioned solution the Domain Feature Requirements were established and are shown in Table 6.5.

Must have	**Should Have**	**Could Have**	**Won't have**
Permissioned	Private	Privacy Technologies	
Smart-Contracts	delegated Byzantine Fault Tolerance	Virtual Machine	
Cryptographic Token	Delegated Proof-of-Stake	Turing Complete	
Protocol Layer	Share-like token		
Network Layer	Security token		
Application Layer	Network token		
Interoperability technologies	Network value token		
Enterprise system integration	Work token		
On-chain transactions	Usage token		

TABLE 6.5: Domain Feature Requirements Veris Foundation, prioritized based on the MoSCoW-technique (DSDM-Consortium, 2014)

Feature prioritization discussion

This section will discuss the prioritization of the Domain Feature Requirements as stated in Table 6.5. Not each feature will be discussed in depth as in many cases it comes down to a rather trivial decision in the case of many features categorized as 'could have' or not categorized at all.

In the envisioned solution six types of *Smart-contracts* will be deployed, thus this is categorized as a must-have feature. Since the Veris implementation will be a forked version of the NEO blockchain it shares a lot of similar features. Most notably will be the *dBFT* consensus-mechanism on the *Protocol Layer*. Another option for the dBFT consensus-mechanism would be *Delegated Proof-of-Stake* which also enables bookkeeper nodes. Next to this layer the *Network Layer* defines the rules such as *Permissioned* authorization and the *Application Layer* is used to develop GUIs for the different stakeholders. Since the Veris solution doesn't operate in a vacuum but interacts with other entities such as banks it requires certain *Interoperability characteristics* and in particular *Enterprise system integration*. The VeriStakes and VeriCoins dual currency structure give rise to the must-have feature *Cryptographic tokens*. All the lower-level token-types are not strictly required, thus the following features are

should-have features: *Share-like token, Security token, Network token, Network value token, Work token and Usage token*. Normally *Work token* and *Usage token* together would be a *Hybrid-token*. However, the VRS is a *Work tokens* and the VRCO are *Usage tokens* and (although closely related) operate separately from each other.

Case Participant Short-List Alternatives

Veris Foundation thoroughly explains their decision process when selecting the right platform for their solution (Plance, 2017). They felt the three most important criteria for the creation of a success of their system would be: Technical capability, governance, and community. The governance and community will only briefly be discussed since the focus of this research is on technical capabilities in the form of features.

Domain Alternative	CP Rank
NEO	1
Ethereum	2

TABLE 6.6: Veris Foundation Short-List Alternatives

Initially, Veris started of with five alternatives, of which the final two are mentioned in 6.6. Based on technical capabilities IOTA and CryptoNote could be immediately eliminated since they currently don't provide smart contract functionality. Bitcoin offers smart contract functionality with the RootStock add-on, but this is not a core functionality of Bitcoin. Veris determined that the risks attached to another party developing the smart contract with RootStock would be too high. Thus, in their decision process only Ethereum and NEO were left as alternatives. With respect to fundamental technical differences, Veris chose NEO. The first fundamental difference is that NEO allows for the use of bookkeeping nodes. These bookkeeping nodes become the gatekeeper between those who are holding coins and those who are creating insurance contracts on the chain. Veris feels this is critical to the success of their product. The second fundamental technical difference identified by Veris between Ethereum and NEO is the split of network fees from coins. Within the Ethereum network, the execution of smart-contracts requires ETH currency. This would reduce however the stakeholder's ETH after a prolonged time of usage. NEO solves this problem by splitting having a stake in the network and paying for network fees. The NEO currency generates GAS tokens and this GAS is used to execute smart-contracts and transactions on the NEO network. Even after a prolonged time of usage, the stake in the network of a stakeholder stays the same.
When comparing the communities, Ethereum obviously is the largest crypto-community at the moment. However, NEO has shown great growth past year and Veris acknowledges this potential as absolute size not being a limiting factor.

Chapter 7

Results, Analysis and Evaluation

This chapter will describe the results, the means of generating these results as well as analyzing these results. Based on the analysis of these results the artifact is evaluated as defined in the Research Method (Section 3.7).

7.1 Results

The following steps were taken to create the results of the DSS based on the case studies. Section 5.4.2 describes the process which was used to generate the results. As input for the features Table 7.1 was used, which is a summarized version of the Tables 6.1, 6.3 and 6.5. In addition to this the Tables 6.2, 6.4 and 6.6 were used for the ranking of the case participants beforehand. The output of this process and thus the results for each of the case studies is shown in Table 7.2. This table shows for each of the three case studies the Feasible Solutions, whether an alternative was on the case participants shortlist or not (if so including the rank) and the DSS score.

MoSCoW	ShareCompany BIQH		DUO		Veris Foundation	
Must Have	Permissioned, Smart Contracts, Sybil-attack resistant, etc.	10	Smart Contracts, Application Layer, Cryptographic Tokens, etc.	8	Permissioned, Cryptographic Tokens, Enterprise system integration, etc.	9
Should Have	Zero-knowledge Proof, High Maturity, High Popularity, Golang, Private, etc.	9	JavaScript, High Maturity, Solidity, Cryptocurrency (purpose), etc.	6	Delegated Proof-of-Stake, Work token, Security token, Usage token, etc.	9
Could Have	Turing-complete, Virtual Machine, SNARKS, Turing-complete, etc.	8	Permissioned, Permissionless, Java, Proof-of-Authority, C++, etc.	23	Privacy Technologies, Virtual Machine, Turing Complete	3
Won't have	Proof-of-Work, Proof-of-Stake, Directed Acyclic Graph	3	Directed Acyclic Graph	1	None	0

TABLE 7.1: Domain Feature Requirements for each case study

7.2 Analysis and Evaluation

7.2.1 Analysis

This section will analyze the results for each case study separate as presented in Table 7.2.

ShareCompany BIQH

Beforehand ShareCompany ranked Hyperledger first on their short-list and JPMorgan Quorum second. Hyperledger proved indeed to be the best scoring Feasible Solution by providing in all the must-have features and most of the should-have and could-have features. However, the alternative from JPMorgan Quorum was not second in the DSS results. R3 Corda scores slightly higher, mainly due to having a higher popularity in the market and a higher technology maturity compared to

Case Study	DSS Feasible solutions	CP Shortlist	DSS Score	CP Rank
ShareCompany BIQH	Hyperledger	Yes	99.39	1
	R3 Corda		68.13	-
	JPMorgan Quorum	Yes	61.92	2
	Chain		40.05	-
DUO	Ethereum	Yes	98.25	1
	Hyperledger	Yes	73.22	3
	Wanchain		64.68	-
	NEO	Yes	62.1	2
	Cosmos Network		51.1	-
	Stellar		37.91	-
	Komodo		37.65	-
	Waves Platform		37.25	-
	Chain		34.3	-
	VeChain		31.31	-
Veris Foundation	Cosmos Network		99.64	-
	NEO	Yes	69.42	1
	Ethereum	Yes	54.52	2
	Stellar		53.33	-
	Hyperledger		44.48	-
	Chain		44.48	-
	VeChain		30.27	-
	ICON		28.63	-
	Symbiont		28.16	-
	Neblio		21.37	-

TABLE 7.2: Feasible Solutions DSS score for each case study

Quorum. The main difference why Hyperledger scores significantly higher than the other feasible solutions is due to it supporting the should-have features JavaScript, Zero-knowledge Proofs, and Golang. ShareCompany BIQH found it rather interesting that both R3 Corda and JPMorgan Quorum are feasible solutions. Just like ShareCompany BIQH, both R3 Corda and JPMorgan Quorum focus on financial institutions (and financial data) thus for their solution they now consider R3 Corda as well besides Hyperledger and Quorum as a potential platform to be utilized.

DUO

Beforehand G. De Jonge ranked Ethereum as the most prominent platform, NEO second and Hyperledger third. When looking at the score of Ethereum this proved to be the right choice according to the DSS as well since Ethereum has the highest score. Wanchain was not on the case participant short-list, but since it is an Ethereum-based fork Wanchain scoring high is not too surprising. Despite Hyperledger scoring high, it should be noted however that the solution for DUO makes intensive use of cryptographic tokens. Hyperledger supports this feature, however Hyperledger has no native-token and token-based solutions are more troublesome on Hyperledger. Several of the should-have features are token-based, which Hyperledger doesn't support. Due to a large amount of could have features for this case study (and the should have feature High Maturity) Hyperledger does support, it scores quite high nevertheless. It was expected NEO would score slightly higher beforehand although the difference with Hyperledger's score is not too significant.

Veris Foundation

Beforehand the Veris Foundation had two main alternatives to develop their solution on, NEO as first choice and Ethereum as second. However, in the results from the DSS Cosmos Network would be the most appropriate platform. The reason Cosmos Network scores so high is due to the fact that it is rather flexible regarding different pluggable consensus-mechanisms and both allowing for any combination of permissioned/permissionless and public/private blockchains compared to both NEO and Ethereum. Hyperledger is a feasible solution once again, however the same possible difficulties as in the DUO case study could arise with a heavy reliance on different token-types which are harder to implement in practice. It is interesting to see that Chain (next to Hyperledger) is a feasible solution in all three case studies to develop their solutions on. Another interesting observation (based on these three case studies) is that it seems the main decision that has to be made is the choice between permissioned or permissionless platforms and whether cryptographic tokens are required or not.

7.2.2 Evaluation

This section will evaluate the DSS based on the results from section 7.1 and the analysis of these results described in section 7.2. This will be compared against the evaluation-metrics which were defined in section 3.7, being: efficacy, validity, and generality. Together these three metrics determine to which degree the goal dimension of the artifact is met. As mentioned in section 3.7 the relativeness of the evaluation is absolute since there are currently no comparable alternatives to this DSS available (yet).

Efficacy

The efficacy is the degree to which the artifact produces its desired effect (Venable, Pries-Heje, and Baskerville, 2012). The desired effect, as stated in the main research question, is to aid developers during the selection process between different blockchain platforms to develop their solution on. In the case of this research, the opinion of the participants from the three case studies was inquired on the efficacy of the artifact. The opinion on the efficacy of the artifact was rather positive in each of the three case studies. ShareCompany BIQH acknowledges that such a tool is highly crucial once blockchain starts becoming more adopted in organizations. Especially in the relatively early stages when knowledge about different alternatives is still lacking. G. de Jonge was satisfied with the results and content he had chosen the right alternative for DUO's envisioned solution. The Veris Foundation briefly indicated that the BPDSS could make future decisions for other organizations easier. Concluding, with respect to efficacy the BPDSS seems to perform sufficiently and might prove rather valuable in the future.

Validity

The validity metric is defined as the degree to which the artifact works correctly (Straub, Boudreau, and Gefen, 2004). Reliability is encompassed under validity as well. The validity of the artifact will be indicated in two ways. Besides analyzing the results from Table 7.2 (as described in section 7.2) also a Domain Expert has given his opinion on the validity of the artifact as mentioned in Table 3.2. Based on the analysis of the results, the BPDSS scored the highest ranked case participant solutions in two

of the three case studies highest as well. In the case of the ShareCompany BIQH case study, the platform ranked second Quorum from JPMorgan scored rather high. As mentioned in the analysis as well, Quorum is basically a permissioned version of Ethereum without a cryptographic token or mining-based consensus-mechanism. And R3 Corda scoring high for ShareCompany which is also an organization active in the financial domain increases the Validity of the BPDSS. So with respect to the validity of the results based on the three case studies, the DSS scores more than sufficient.

In addition to analyzing the DSS output, also one Domain Expert (who also participated in one of the interviews to identify the Domain Features) judged the BPDSS on validity. The reason for this is once again the immaturity of the blockchain domain. Experts from the case studies might be relatively less knowledgeable than case participants used in the creation of other DSSs by Farshidi et al. (2018). This expert indicated that overall the BPDSS is giving 'pretty valid' results but indeed acknowledges as well that it struggles with certain aspects. The first aspect the BPDSS struggles with is partially implemented features or features which are possible but tougher to make work in practice. A previous example of this are the cryptographic tokens within the Hyperledger platform. Another aspect the BPDSS struggles with according to this expert is the distinction between blockchain platforms having a focus on a specific industry (e.g. healthcare, finance, etc) or being industry agnostic

Generality

Generality is defined as the broader the goal addressed by the artifact the more general the artifact (Aier and Fischer, 2011). The goal addressed by the artifact is to aid developers which have decided to create a blockchain based solution. All the most prominent blockchain platforms are included in the artifact, which is a positive aspect regarding generality. However, the generality is reduced to being restricted to the blockchain domain. When a developer has decided a DBMS is more suited for his solution this artifact is deprived of its purpose. Fortunately, this BPDSS is just a small part of a larger research by S. Farshidi in which additional DSSs are created for other technology selection domains.

Comparison to other DSS by Farshidi

When comparing the results of the BPDSS to the results of DSSs for CSP and BDSM by Farshidi et al. (2018) a few things should be taken into consideration. Table 7.2 shows for each case study at least one feasible solution with a relatively low score such as Neblio only 21.37 in the case of the Veris Foundation. The most likely explanation for this is the total amount of Domain Features, Domain alternatives and the amount of selected must-have features as domain feature requirements. This on itself could most likely be explained by the immaturity of the domain and possible inefficiencies that go along with it. Should in the future there be more blockchain platforms the lower scoring alternatives would be filtered out obviously. In addition to this, when more should and could have features are being implemented as 'standard features' rather than slightly niche features only available on the most well-known platforms the variance between scores would be more insignificant as well.

Chapter 8

Conclusion, Limitations and Future Research

8.1 Conclusion

This first section of this chapter attempts to answer the Sub-Research Questions as well as the Main Research Question as described in Chapter 3.

8.1.1 Sub-Research Question 1:

What are the technologies related to the blockchain technology that are relevant for the creation of the artifact?

A literature study based on the snowballing-method was conducted to identify an initial set of Domain Features for the blockchain domain. Based on this literature study 76 Domain Features were initially identified. Most of these features belonged to one the following research topics: Consensus-mechanisms, Smart-Contracts, Scalability, Privacy, Tokenization, Resilience and Security, Maturity, Popularity, Innovation and Transaction Speed. These 76 features were included in the interview protocol.

8.1.2 Sub-Research Question 2:

What are the contemporary features and platforms in the blockchain domain?

The 76 initial Domain Features from Sub-Research Question 1 were discussed during 9 Domain Expert Interviews to identify the set of generic Domain Features. After these interviews, 75 generic Domain Features were identified alongside 11 categories under which these features were sorted. Based on these 9 Expert Interviews also 29 contemporary blockchain platforms (Domain Alternatives) were identified.

8.1.3 Sub-Research Question 3:

Which identified blockchain features have a positive influence on different software quality aspects?

The set of generic Domain Features from Sub-Research Question 2 were mapped against the Software Quality Aspects from ISO/IEC 25010 Ext. ISO/IEC 9216 based on four additional Domain Expert Interviews. These experts indicated whether a feature has a positive influence on a Software Quality Aspect or not. The deliverable of this Research Question was the SF-mapping.

8.1.4 Sub-Research Question 4:

Which features are offered by each of the available blockchain platforms?

The 29 Domain Alternatives from Sub-Research Question 2 were mapped against which of the 75 generic Domain Features from Sub-Research Question 2 they provide. Mapping this was based on a document analysis and Domain Expert knowledge. The deliverable of this Research Question was the FA-mapping.

8.1.5 Sub-Research Question 5:

Is the created decision support system applicable in the business environment?

Based on the SF-mapping from Sub-Research Question 3 and the FA-mapping from Sub-Research Question 4 the Blockchain Platform Decision Support System (BPDSS) was created in the Decision Model Studio. To evaluate the BPDSS the Domain Feature Requirements of three different case studies were applied to it. The BPDSS was evaluated in the goal-dimension on the efficacy, validity and generality metrics based on the results for the case studies and a Domain Expert evaluation. In each of the three evaluation-metrics the BPDSS artifact scored at least sufficient, thus is usable in the business environment. It should be noted however that the BPDSS has it's short-coming with respect to validity and accuracy of the results when facing partially implemented features.

8.1.6 Main Research Question:

How can an artifact be developed that assists during the selection process between different blockchain platforms from a developer's perspective?
In this research we've identified the choice between different blockchain platforms can be classified as a multi-criteria decision-making problem for technology selection. Based on Farshidi's (Farshidi et al., 2018) research we've created the Blockchain Platform Decision Support System (BPDSS). This BPDSS is a feature-based artifact which incorporates Software Quality Aspects from ISO/IEC 25010 Ext. ISO/IEC 9216 and feature-prioritization based on the MoSCoW-technique (DSDM-Consortium, 2014). In the current version of the BPDSS 75 generic features from the blockchain domain are included as well as 29 blockchain platforms which support these features. This BPDSS has been evaluated and validated with three case studies and an expert validation. The BPDSS was evaluated in the goal-dimension on efficacy, validity, and generality. Based on the results for these metrics we've concluded that the BPDSS is capable of assisting developers sufficiently during the selection process between different blockchain platforms.

8.2 Limitations

This section will briefly describe the limitations of this research. Some of these limitations are (closely) related to each other while other imperfections are more isolated. In addition to this, these limitations can roughly be divided into more theoretical focused limitations and more practical related limitations.

8.2.1 Theoretical

The first theoretical limitation is the somewhat limited scope of the BPDSS. The BPDSS assumes the blockchain technology is the appropriate technology. In practice however, this is still quite an issue with organizations utilizing the blockchain while a simple DBMS would've been sufficient as well when. Thus, it is assumed the decision maker knows the advantages and disadvantages of a blockchain and is capable of selecting the right technology. However, by choosing Farshidi's (Farshidi et al., 2018) model this was something inherent that could not have been avoided. Fortunately, Farshidi is working on a decision model for selecting the most appropriate *Software Architecture Patterns* that should alleviate this problem. Another issue that is closely related to this is that the decision-maker should have a pretty extensive knowledge of the blockchain domain when selecting Domain Feature Requirements. If the decision-maker lacks this knowledge even the Feature definitions in the BPDSS won't help much if everything is completely new. However, one could argue that someone should not be developing a software solution with a new immature technology without the required fundamental knowledge. On the plus side, it is likely general knowledge about the blockchain technology will increase in the coming years. If this will be indeed the case, both these knowledge related limitations will become trivial.

8.2.2 Practical

Despite all the thoroughly elaborated theoretical research methods, obviously, most research projects meet some practical implications down the road of execution. This research was no exception with a few practical limitations. The first limitation is related to the expert selection process and the expert interviews. In general, due to the immaturity and still highly unregulated nature of the blockchain domain selecting the experts proved quite a challenge. The experts with an academic background were rather easy to find, were benevolent towards participating in the research and provided valuable high-quality knowledge. Finding willing non-academic domain experts proved to be trickier in practice due to them: not responding, not willing to do pro-bono work or having other priorities. Eventually, the majority of these non-academic background domain experts willing to do an interview provided a lot of useful knowledge. However, one of the so-called selected experts proved to be the spokesperson/CEO of a blockchain platform which was unknown beforehand based on the expert's LinkedIn. During the interview, it became apparent this interviewee was rather biased towards features and alternatives. The information from this interview is still included in this research to make sure this research itself is not biased. Due to having 8 more interviews for identifying the generic Domain Features and Alternatives this bias is most likely been neutralized. A lesson to be learned from this is that there are still a lot of charlatans/biased people in this immature and unregulated domain and one should tread carefully.

Another limitation of this research is having the BPDSS evaluated in only three case studies, so results are not directly generalizable. Preferably at least one additional case study evaluation was conducted. Also, the Veris Foundation case study evaluation was not the most optimal solution for a case study. It was expected to be a case study at a large bank in the Netherlands instead, unfortunately this bank withdrew halfway from this project. Their main reason was that they were still in the middle of the decision-making process and couldn't release details about their case under any circumstances. When the bank decided to resign from this research the

time-factor started being a constraint so therefore the Veris Foundation was selected since they made all their information publicly available. The most important information publicly available was their thoroughly explained decision-making process between different blockchain platforms. Obviously, this is far from an ideal solution but given the time-constraints better than just two case study evaluations.

The last limitation of this research is that the created BPDSS can be outdated rather quickly without proper updates to it. The main reason for this is the rapid changes and developments in the blockchain domain. However, the BPDSS has been created in such a way that it can be updated rather easily when features/alternatives have to be added or removed from it. Another risk with respect to the BPDSS possibly not being future-proof is that one blockchain platform might emerge as the distinct dominant winner. Although unlikely with different platforms focusing on different industries and application areas it is still a possibility.

8.3 Future Research

During this research, new opportunities for future work were identified. This section describes a number of these future research possibilities roughly divided into two categories. Improvements/adaptations to the current artifact are the first category while suggestions for new artifacts serve as the second category.

8.3.1 Contemporary Artifact improvements

Future work building on the current artifact can be done in several different ways. As mentioned in Section 8.2 a limitation of the artifact is that after a while the artifact becomes outdated. Future work can be keeping the list of alternatives and features up to date so the artifact stays relevant. A possible way of doing this, as opposed to done is this research, is collecting these generic features could be done with natural language processing (NLP) techniques. Using NLP would save a lot of time compared to collecting all alternatives and features manually. Adding (or removing if needed) features or alternatives can be easily done due to the way the BPDSS was created on the Amuse-project site.

8.3.2 New Research

In regards to new research, there are a few possibilities when using this research as the foundation and starting point. Currently, the BPDSS makes barely a distinction between the application domain a blockchain might focus on. Currently, the only part where this is taken into consideration is the numerical feature innovation. Several application domains are parameters for the innovation feature, such as a focus on supply-chain management, finance or internet-of-things. New DSSs could be created for these domains, but also healthcare (like the Veris Foundation case study) or social media for example. These DSSs would be more specialized versions of the BPDSS, with fewer domain alternatives. For example, it is not unthinkable that all blockchains focusing on the finance sector have a higher throughput compared to non-finance focused blockchains but differ on other sub-features that might be more relevant for that sector. The creation of these DSSs could be done in a similar fashion as done is this research and utilize gathered features to find generic features for certain application domains.

Bibliography

Abdullah, L. (2013). "Fuzzy multi criteria decision making and its applications: A brief review of category". In: *The 9th International Conference on Cognitive Science*. Procedia - Social and Behavioral Sciences 97, pp. 131–136.

Aier, S. and C. Fischer (2011). "Criteria of progress for information system design theories". In: *Information systems and E-business Management* 9.1, pp. 133–172.

Azaria, A. et al. (2016). "MedRec: Using Blockchain for Medical Data Access and Permission Management". In: *In Open and Big Data (OBD), International Conference*, pp. 25–30. URL: https://www.media.mit.edu/publications/medrec/.

Back, A. et al. (2014). "Enabling Blockchain Innovations with Pegged Sidechains". In: URL: https://blockstream.com/sidechains.pdf.

Baliga, A. (2016). "The Blockchain Landscape". In: *Persistent Systems*. URL: https://pdfs.semanticscholar.org/c826/b333dfb04e3053a7c2cb3b881bff1d952942.pdf.

— (2017). "Understanding Blockchain Consensus Models". In: *Persistent Systems*. URL: https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf.

BBC (2017). "CryptoKitties craze slows down transactions on Ethereum". In: *BBC*. URL: http://www.bbc.com/news/technology-42237162.

Bowen, G.A. (2009). "Document Analysis as a Qualitative Research Method". In: *Qualitative Research Journal* 9, pp. 27–40. URL: https://www.emeraldinsight.com/doi/abs/10.3316/QRJ0902027.

Buntix, J.P. (2017). "IOTA Network Struggles Due to Lack of Full Nodes". In: *nulltx.com*. URL: https://nulltx.com/iota-network-struggles-due-to-lack-of-full-nodes/.

Buterin, V. (2015). "On Public and Private Blockchains". In: URL: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.

— (2018). "Sharding FAQ". In: URL: https://github.com/ethereum/wiki/wiki/Sharding-FAQ#but-doesnt-the-cap-theorem-mean-that-fully-secure-distributed-systems-are-impossible-and-so-sharding-is-futile.

Cachin, C. (2016). "Architecture of the Hyperledger Blockchain Fabric". In: URL: https://pdfs.semanticscholar.org/f852/c5f3fe649f8a17ded391df0796677a59927f.pdf.

Carvallo, J.P. and X. Franch (2006). "(2006). Extending the iso/iec 9126–1 quality model with non-technical factors for cots components selection." In: *Proceedings of the 2006 international workshop on software quality*. New York, pp. 9–14.

Castor, A. (2017). "A guide to Blockchain Consensus Protocols". In: URL: https://www.coindesk.com/short-guide-blockchain-consensus-protocols/.

Coinbureau.com (2018). "Everything You Need to Know About Zero Knowledge Proofs and zkSNARKs". In: *coinbureau.com*. URL: https://www.coinbureau.com/education/zero-knowledge-proofs-zksnarks/.

CoinMarketCap.com (2013). "CryptoCurrency Market Capitalizations". In: URL: https://coinmarketcap.com/.

Crosby, M. et al. (2016). "BlockChain Technology: Beyond Bitcoin". In: *Applied Innovation Review* 2. URL: http://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blockchain.pdf.

Deshpande, A. et al. (2017). "Understanding the landscape of Distributed Ledger Technologies/Blockchain". In: *RAND Europe*. URL: https://www.rand.org/pubs/research_reports/RR2223.html.

Douceur, J.R. (2002). "The Sybil Attack". In: *Peer-to-peer systems*, pp. 251–260. URL: https://link.springer.com/chapter/10.1007%2F3-540-45748-8_24.

DSDM-Consortium (2014). "The dsdm agile project framework". In: URL: https://www.agilebusiness.org/sites/default/files/the_dsdm_agile_project_framework_v1_11.pdf?token=yqzXtW1a1.

Ethereum-Foundation (2014). "Ethereum". In: URL: https://www.ethereum.org/.

Etherscan.io (2018). "Ethereum charts and statistics". In: URL: https://etherscan.io/charts.

Euler, T. et al. (2018). "The Token Classification Framework: A multi-dimensional tool for understanding and classifying crypto tokens." In: URL: http://www.untitled-inc.com/the-token-classification-framework-a-multi-dimensional-tool-for-understanding-and-classifying-crypto-tokens/.

European-Union (2016). "REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL". In: *Official Journal of the European Union*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2016:119:FULL.

European-Union-Commission (2017). "Guidelines on the application of regulation (eu) no 1286/2014 of the european parliament and of the council on key information documents for packaged retail and insurance-based investment products (priips)." In: *Official Journal of the European Union*.

Farshidi, S. et al. (2018). "A Decision Support System for Software Technology Selection". In: *19th Open Conference on the IFIP WG 8.3 on Decision Support Systems (IFIP DSS 2018*. URL: https://www.tandfonline.com/doi/full/10.1080/12460125.2018.1464821.

Geranio, M. (2016). "Fintech in the exchange industry: Potential for disruption?" In: *Masaryk University Journal of Law and Technology* 11.2, pp. 245–264. URL: 10.5817/MUJLT2017-2-3.

Greenspan, G. (2016). "Blockchains vs centralized databases)". In: *Private Blockchains*. URL: https://www.multichain.com/blog/2016/03/blockchains-vs-centralized-databases/.

Haswell, H. (2017). "Maersk and IBM Unveil First Industry-Wide Cross-Border Supply Chain Solution on Blockchain". In: URL: http://www-03.ibm.com/press/us/en/pressrelease/51712.wss.

Herrera-Joancomartí, J. and C. Pérez-Solà (2016). "Privacy in Bitcoin Transactions: New Challenges from Blockchain Scalability Solutions". In: *Modeling Decisions for Artificial Intelligence*. URL: 10.1007/978-3-319-45656-0_3.

Hevner, A.R. (2007). "A Three Cycle View of Design Science Research". In: *Scandinavian Journal of Information Systems* 19.2, pp. 87–92.

Hevner, A.R. et al. (2004). "Design Science Research in Information Systems". In: *MIS Quarterly* 28.1, pp. 75–105. URL: http://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf.

Hileman, G. and M. Rauchs (2017). "Global Blockchain Benchmarking Study". In: *Cambridge Centre for Alternative Finance*. URL: https://poseidon01.ssrn.com/delivery.php?ID=014117064098126088070103024068001069057020066018053053081100100094126120EXT=pdf.

Institute, Rand National Defense Researche (2009). "Data Collection Methods (Semi-Structured Interviews and Focus Groups)". In: URL: http://www.dtic.mil/get-tr-doc/pdf?AD=ADA512853.

ISO (2011). *Iso/iec 25010: 2011 systems and software engineering-systems and software quality requirements and evaluation-system and software quality models*. ISO.

Jalali, S. and C. Wohlin (2012). "Systematic Literature Studies: Database Searches vs. Backward Snowballing". In: *International Conference on Empirical Software Engineering and Measurement, ESEM'12, Lund, Sweden*. URL: http://www.diva-portal.org/smash/get/diva2:834640/FULLTEXT01.pdf.

Jonge, G. De (2018). "A student finance system based on blockchain technology and smart contracts". In: *Bachelor Thesis Utrecht University*. URL: https://drive.google.com/drive/folders/19khLt4Ud5F69-z1XJ5zldn4wdNjDW97E.

Koens, T., C. Ramaekers, and C. Wijk (2017). "Efficient Zero-knowledge Range Proofs in Ethereum". In: URL: https://www.ingwb.com/media/2122048/zero-knowledge-range-proof-whitepaper.pdf.

Koteska, B., E. Karafiloski, and A. Mishev (2017). "Blockchain Implementation Quality Challenges: A Literature Review". In: *6th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications*. URL: http://ceur-ws.org/Vol-1938/paper-kot.pdf.

Lamport, L., R. Shostak, and M. Pease (1982). *The Byzantine Generals Problem*. Vol. 4. ACM Transactions on Programming Languages and Systems 3.

Lewis, A. (2017). "Avoiding blockchain for blockchain's sake: Three real use case criteria". In: URL: https://bitsonblocks.net/2017/07/24/avoiding-blockchain-for-blockchains-sake-three-real-use-case-criteria/.

Lucassen, G. et al. (2015). "Forging high-quality user stories: towards a discipline for agile requirements". In: *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, pp. 126–135. URL: https://scholar.google.nl/citations?user=WFmLPOwAAAAJ&hl=nl#d=gs_md_cita-d&p=&u=%2Fcitations%3Fview_op%3Dview_citation%26hl%3Dnl%26user%3DWFmLPOwAAAAJ%26citation_for_view%3DWFmLPOwAAAAJ%3AzYLM7Y9cAGgC%26tzom%3D-120.

Luther, J. (2013). "Cryptocurrencies, network effects, and switching costs". In: URL: https://www.mercatus.org/system/files/Luther_CryptocurrenciesNetworkEffects_v1.pdf.

Mainelli, M. and M. Smith (2015). "Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology)". In: *EY Global Financial Services Institute* 3.3. URL: http://www.the-blockchain.com/docs/Journal%20of%20Financial%20Perspectives%20-%20Sharing%20Ledgers%20for%20Sharing%20Economies.pdf.

Majumder, M. (2015). *Multi criteria decision making*. Singapore: Springer.

Mazieres, D. and Stellar-Development-Foundation (2016). "The Stellar Consensus Protocol: A federated Model for Internet-level Consensus". In: URL: https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

Milani, F., L. Garcia-Banuelos, and M. Dumas (2016). "Blockchain and Business Process Improvement". In: URL: https://www.bptrends.com/bpt/wp-content/uploads/10-04-2016-ART-Blockchain-and-Bus-Proc-Improvement-Milani-Garcia-Banuelos-Dumas.pdf.

Morabito, V. (2017). "Business Innovation Through Blockchain". In: URL: https://link.springer.com/book/10.1007%2F978-3-319-48478-5.

Nakamoto, S. (2008). "Bitcoin: A peer-to-peer electronic cash system". In: URL: https://bitco.in/pdf/bitcoin.pdf.

Narayanan, A. et al. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduc- tion*. Princeton University Press.

NEO-Foundation (2017). "NEO White Paper; A distributed network for the Smart Economy". In: URL: https://pdfs.semanticscholar.org/f852/c5f3fe649f8a17ded391df0796677a599pdf.

Noble, D.F. (1995). "Progress without People: New technology, unemployment, and the message of resistance". In: *Between the Lines*.

Omri, M. (2013). "Are Cryptocurrencies 'Super' Tax Havens?" In: *112 Michigan Law Review First Impressions* 38. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2305863.

Pahl, C., N. El Ioini, and S. Helmer (2017). "A Decision Framework for Blockchain Platforms for IoT and Edge Computing". In: *3rd International Confernce on Internet of Things, Big Data and Security*. URL: https://www.researchgate.net/publication/323960587_A_Decision_Framework_for_Blockchain_Platforms_for_IoT_and_Edge_Computing?enrichId=rgreq-e7980dac585c20b0d587172a3809d8b4-XXX&enrichSource=Y292ZXJQYWdlOzMyMzk2MDU4NztBUzo2MDczMTE4NDg1NDIyMDhAMTUyMTgwNTgzNjU0OA%3D%3D&el=1_x_2&_esc=publicationCoverPdf.

Panetta, K. (2017). "Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017". In: URL: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/.

Perretta, H. (2017). "Private vs. Public and Permissioned vs. Permission-less". In: URL: http://blocktonite.com/2017/06/27/private-vs-public-and-permissioned-vs-permission-less/.

Peterson, J. et al. (2018). "Augur: a Decentralized Oracle and Prediction Market Platform". In: URL: https://www.augur.net/whitepaper.pdf.

Petri, C. (2010). "Decision Trees". In: URL: http://www.cs.ubbcluj.ro/~gabis/DocDiplome/DT/DecisionTrees.pdf.

Plance, C. (2017). "Three things to consider when choosing a blockchain for your project - or why veris chose NEO". In: URL: https://medium.com/verisfoundation/three-things-to-consider-when-choosing-a-blockchain-for-your-project-or-why-veris-chose-neo-b4483135c382.

Plance, C. and E. Lawlor (2018). "The Veris Foundation: Authorizatin, Eligibility, and Settlement for Healthcare Services via Smart contracts, and Proof of Stake to Reduce Expense, Decentralize the Process and Reduce Healthcare Expense". In: 7. URL: https://veris.docsend.com/view/fbqysxf.

Poon, J. and V. Buterin (2017). "Plasma: Scalable Autonomous Smart Contracts". In: URL: https://plasma.io/plasma.pdf.

Poon, J. and T. Dryja (2016). "The Bitcoin Lightning Network: Scalable Off-chain Instant Payments". In: URL: https://lightning.network/lightning-network-paper.pdf.

Prat, N., I. Comyn-Wattiau, and J. Akoka (2014). "ARTIFACT EVALUATION IN INFORMATION SYSTEMS DESIGN-SCIENCE RESEARCH – A HOLISTIC VIEW". In: URL: https://cedric.cnam.fr/fichiers/art_3208.pdf.

Prisco, G. (2017). "The Ethereum Killer is Ethereum 2.0: Vitalik Buterin's Roadmap". In: *Bitcoin Magazine*. URL: https://bitcoinmagazine.com/articles/ethereum-killer-ethereum-20-vitalik-buterins-roadmap/.

Reiff, N. (2018). "A History of Bitcoin Hard Forks". In: *investopedia.com*. URL: https://www.investopedia.com/tech/history-bitcoin-hard-forks/.

Rijksoverheid (2010). "Dienst Uitvoering Onderwijs (DUO)". In: URL: https://www.rijksoverheid.nl/ministeries/ministerie-van-onderwijs-cultuur-en-wetenschap/organisatie/organogram/dienst-uitvoering-onderwijs-duo.

Rikken, O. (2018). "Blockchain Choice". In: URL: http://www.blockchaincomparator.com/.

Rosic, A. (2017). "Blockchain Coding". In: URL: https://blockgeeks.com/guides/blockchain-coding/.

Saaty, T.L. (1990). "How to make a decision: the analytic hierarchy process". In: *European Journal of Operational Research* 48.1, pp. 9–26.

Sage, A. (1991). "Decision Support Systems Engineering". In: *Wiley-Interscience*.

Samman, G. (216). "The Trend Towards Blockchain Privacy: Zero Knowledge Proofs". In: *Coindesk*. URL: https://www.coindesk.com/trend-towards-blockchain-privacy-zero-knowledge-proofs/.

Schwartz, D., N. Youngs, and A. Britto (2014). "The Ripple Protocol Consensus Algorithm". In: URL: https://ripple.com/files/ripple_consensus_whitepaper.pdf.

Sedgwick, K. (2018). "Proof of Work Coins on High Alert Following Spate of 51Attacks". In: *news.bitcoin.com*. URL: https://news.bitcoin.com/proof-of-work-coins-on-high-alert-following-spate-of-51-attacks/.

Siegel, D. (2016). "Understanding The DAO Attack". In: *coindesk.com* 1. URL: https://www.coindesk.com/understanding-dao-hack-journalists/.

Straub, D., M.C. Boudreau, and D. Gefen (2004). "Validation Guidelines for IS Positivist Research". In: *Communications of the Association for Information System* 13.1, pp. 380–427.

Swan, M. (2015). "Blockchain: Blueprint for a New Economy". In: URL: http://w2.blockchain-tec.net/blockchain/blockchain-by-melanie-swan.pdf.

Szabo, N. (1994). "Smart Contracts". In: URL: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

Tapscott, D. and A. Tapscott (2016). *Blockchain Revolution: How the technology behind Bitcoin is changing money, business and the world*. Vol. 1. 375 Hudson Street New York: Penguin Random House LLC.

Triantaphyllou, E. et al. (1998). "Multi-criteria decision making: an operations research approach". In: *Encyclopedia of electrical and electronics engineering* 15.1998.

Venable, J., J. Pries-Heje, and R. Baskerville (2012). "A comprehensive Framework for Evaluation in Design Science Research". In: *Lecture Notes in Computer Science* 7286, pp. 423–438.

Vukolic, M. (2016). "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication". In: *Open Problems in Network Security InetSec 2015*, pp. 112–125.

Weerd, I. Van De and S. Brinkkemper (2008). "Meta-modeling for situational analysis and design methods". In: *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, pp. 38–58.

Wieringa, R. (2018). In: *ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering* 2, pp. 493–494. URL: https://dl.acm.org/citation.cfm?id=1810446.

Wohlin, C. (2014). "Guidelines for snowballing in Systematic Literature Studies and a Replication in Software Engineering". In: *EASE'14*. URL: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0031FEC6A3388A11E0723B4F32DC1DF7?doi=10.1.1.709.9164&rep=rep1&type=pdf.

Wust, K. and A. Gervais (2018). "Do you need a Blockchain?" In: p. 375. URL: http://eprint.iacr.org/2017/375.

Yli-Huumo, J. et al. (2016). "Where is Current Research on Blockchain Technology?-A systematic Review". In: *PLoS ONE* 11.10. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0163477.

Appendix A

Expert Interview Protocol

A.1 Blockchain Experts Interview Protocol

A.1.1 Introduction

- First of all i want to thank you for participating in this research, we highly appreciate it. Explain structure of interview: Introduction from both sides, main part your opinion on blockchain features, your opinion on my identified blockchain features.
- My introduction
- Based on Domain Features, Domain Alternatives, Software Quality aspects and the relationships between them the DSS will be created. Decision maker can assign priorities to different features.

A.1.2 Expert Opinion on Domain-Features

-This will be the main part of interview, identifying the generic Domain-Features and related Sub-Features.

Main-features

- What in your opinion are the main Features in the blockchain domain?
- Are there other features with respect to: Performance/Compatibility/Functional Suitability/Usability/Reliability/Security/Maintainability/Portability Note to self: For example categories like consensus mechanism

Sub-features

- What are sub-features for each of the main-features?
- What are in your opinion currently niche features but possibly prominent in the future?
Note to self: For example PoS or PoW which are different implementations of the consensus mechanism category, PoS or PoW can have sub-features as well such as in PoW SHA-256, Scrypt or CryptoNight

A.1.3 Expert opinion in initial Domain-features

-Consensus-mechanism (PoW, PoS, dPoS, Byzantine Fault Tolerance, dBFT, etc) (PoW hashing algorithms: SHA-256, Scrypt, X11, X13, Cryptonight, Dagger Hashimoto,KECCAK-256, SHA-1, Momentum, Ethash, Equihash, Multiple) (PoS: Casper, Decred, Ouroboros, SHA-512)
-Permission-type (Public, Hybrid, Private)
-Smart contracts

-Virtual Machine
-Programming language (Solidity, C-sharp, Golang, VB.Net, F-sharp, Java, Kotlin, Python, JavaScript, C, C++, Serpent, LLL, Viper, Ruby, Haskell, Ivy)
-Zero-knowledge proof
-Ring signature
-Access control
-Sybil attack resistant
-Spam attack resistant
-On-chain transactions
-Off-chain transactions
-Stateful/Stateless
-Tokenization
-Token-type (Utility, Security, (non)native-token, asset)
-(Cross-chain) Atomic-swaps
-Cross-chain interoperability
-Parallel-blockchain
-(Plasma)Side-chains
-Sharding
-Enterprise system integration
-Protocol layer
-Network layer
-Application layer
-(Hard)Fork-resistance

- Which features are in your opinion appropriate in this list and which are not? Which features are duplicates between this set and your set with different naming?

A.1.4 Expert Opinion on most prominent Domain Alternatives

-Which (five) available blockchain solutions are currently the most prominent ones in your opinion?
-What are some less well-known blockchain alternatives available to the market in your opinion?

A.1.5 Expert opinion on most prominent initial Domain Alternatives

-Hyperledger Fabric
-R3 Corda
-Ethereum Enterprise Alliance
-NEO
-Ripple
-What is your opinion (if known) on each of these alternatives?

A.1.6 Conclusion

-Ask if experts wants to participate in the mapping of the Features against the Software Quality Aspects as well after all the Alternatives and Features are identified.
-Thank the expert for participating in this interview.

Appendix B

Domain Feature Process

Added Feature after interview										
Removed Feature after interview										
Inital List Domain Features Interview Protocol	**Interview 1**	**Interview 2**	**Interview 3**	**Interview 4**	**Interview 5**	**Interview 6**	**Interview 7**	**Interview 8**	**Interview 9**	**Possible additions based on documentation + Literature**
Consensus-mechanism	Federated Byzantine Agreement	Proof-of-Elapsed Time	md-5	Scrypt	VB.Net	RFID connectability				SIEVE (Baliga, 2017)
Proof-of-Work	Directed Acyclic Graphs	Docker	ASIC-algorithm	X11	C	IoT connectability				Cross-Fault Tolerance (Baliga, 2017)
Proof-of-Stake	Permissioned	Usage token	Proof-of-Burn	X13	Kotlin					Transaction finality (Baliga, 2017)
practical Byzantine Fault Tolerance	Permissionless	Hybrid token	Proof-of-Luck	Casper	LLL					SNARKS (Kosba et al, 2016)
delegated Byzantine Fault Tolerance	Permission-type	Work token	Quantum resistant	Dagger	Serpent					.NET (NEO Foundation, 2017)
Hashing algorithm	Cryptocurrency	Per-chain consensus	Wallet	Hashimoto	Viper					RingCT
Scrypt	delegated Proof-of-Stake	Ring signature		KECCAK-256	Ruby					State Channels
SHA-256	Block-size	TOR		Momentum	Time-stamping					
Scrypt	Spam attack resistant	Turing-complete		Ethash	Ivy					
X11	Sybil attack resistant	Global-data diffusion		Equihash						
X13	Transaction irreversibility			Decred						
Cryptonight	Transaction speed			Proof-of-Authority						
Dagger	Block-size									
Hashimoto	RFID connectability									
KECCAK-256	Data-streams self-reconciliation									
SHA-3	Querying									
Momentum	API queries									
Ethash	Multi-party computation									
Equihash										
Casper										
Decred										
SHA-512										
Permissiontype										**Added features after expert opinion**
Public										Market Cap (popularity numerical feature)
Hybrid										SIEVE
Private										Cross-Fault Tolerance
Smart contracts										Transaction finality
Programming Language										(D)App Token
Solidity										Share-like Token
C#										Work token (not removed)
Golang										Hybrid token (not removed)
VB.Net										Investment token
Java										RingCT
Kotlin										State-Channels
Python										
JavaScript										
C										
C++										
Serpent										
Viper										
Ruby										
Haskell										
Ivy										
LLL										
Zero-knowledge Proof										
TOR										
Access control										
Multi-party computation										
On-chain transactions										
Off-chain transactions										
Global data diffusion										
Stateful										
Stateless										
Virtual Machine										
Tokenization										
Utility										
Security										
Native token										
non-native token										
Asset										
Atomic-swaps										
Cross-chain interoparability										
Plasma side-chains										
Side-chains										
Parallel-blockchain										
Sharding										
Per-chain consensus										
Enterprise system integration										
Data-streams self reconciliation										
Time-stamping										
Querying										
API queries										
Protocol layer										
Network Layer										
Application layer										
Hardfork resistant										

Appendix C

Final Domain Features

Category/Feature	(Sub)Feature
Consensus Mechanism	Proof-of-Work
Category	Proof-of-Stake
	delegated Proof-of-Stake
	practical Byzantine Fault Tolerance
	federated Byzantine Agreement
	delegated Byzantine Fault Tolerance
	Proof-of-Authority
	Proof-of-Elapsed Time
	SIEVE
	Cross Fault Tolerance
	Directed Acyclic Graph (variants)
Layers	Protocol Layer
Category	Network Layer
	Application Layer
Authorization and Authentication	Public
Category	Private
	Permissioned
	Permissionless
Contracts	Smart-contracts
Feature	Virtual Machine
	Turing-complete
Programming Language	Solidity
Smart Contract Support	Python
Category	Golang
	JavaScript
	Java
	C++
Tokens	Tokenization
Feature	Native
	Non-native
	Cryptocurrency
	Utility
	Security
	Usage
	Asset
	Work
	Hybrid
Scalability Technologies	On-chain transactions
Category	Off-chain transactions
	Side-chains
	Sharding
	Plasma-chains

Appendix D

Domain-Alternatives Process

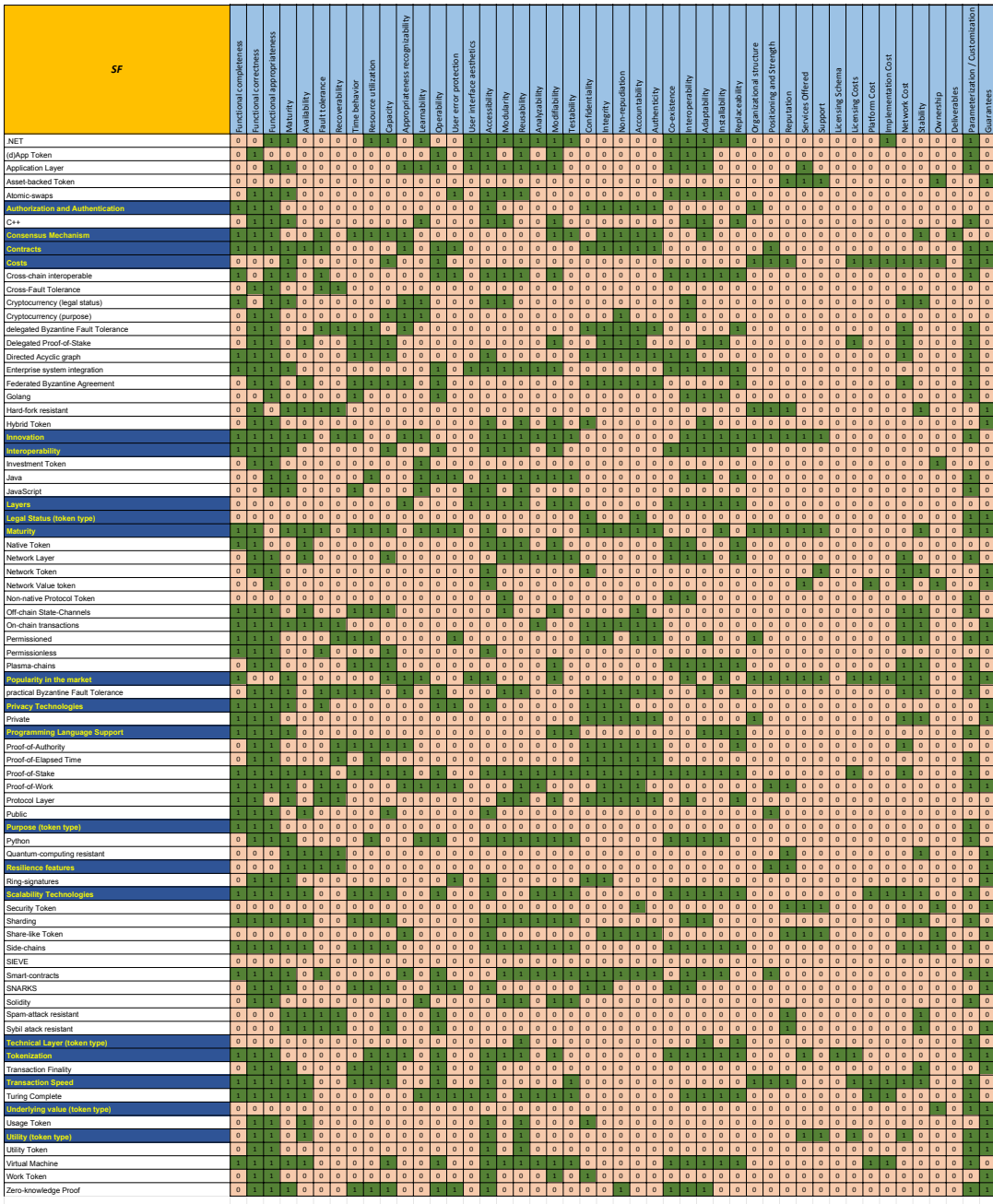
Appendix E

Final domain Alternatives

Final Domain Alternatives	Additional Information
Ethereum (Enterprise Alliance)	https://www.ethereum.org/
R3 Corda	https://docs.corda.net/
JPmorgan Quorum	https://www.jpmorgan.com/global/Quorum
Hyperledger Fabric	https://www.hyperledger.org/projects/fabric
Hyperledger Sawtooth	https://www.hyperledger.org/projects/sawtooth
Hyperledger Indy	https://www.hyperledger.org/projects/hyperledger-indy
Hyperledger Burrow	https://github.com/hyperledger/burrow
Hyperledger Iroha	https://www.hyperledger.org/projects/iroha
BigChainDB	https://www.bigchaindb.com/
MultiChain	https://www.multichain.com/
HydraChain	https://github.com/HydraChain/hydrachain
Chain	https://chain.com/
Symbiont	https://symbiont.io/
Azure BaaS	https://azure.microsoft.com/nl-nl/features/blockchain-workbench/
OpenChain	https://www.openchain.org/
NEO	https://neo.org/
Cardano	https://www.cardano.org/en/home/
Stellar	https://www.stellar.org/
Ripple	https://www.ripple.com/?gclid=Cj0KCQjwuMrXBRC_ARIsALWZrIhbd3yaPNWOUzUXxzCdmho_ovnT3qYhhknr7rIupa8sK0vi8PosRdAaAu8iEALw_wcB
Bitshares	https://bitshares.org/
QTUM	https://qtum.org/en/
ICON	https://icon.foundation/?lang=en
VeChain	https://www.vechain.org/
IOTA	https://www.iota.org/
Factom	https://www.factom.com/
Zilliqa	https://www.zilliqa.com/
Cosmos	https://cosmos.network/
LISK	https://lisk.io/
Waves	https://wavesplatform.com/
Wanchain	https://wanchain.org/
Stratis	https://stratisplatform.com/
Komodo	https://komodoplatform.com/

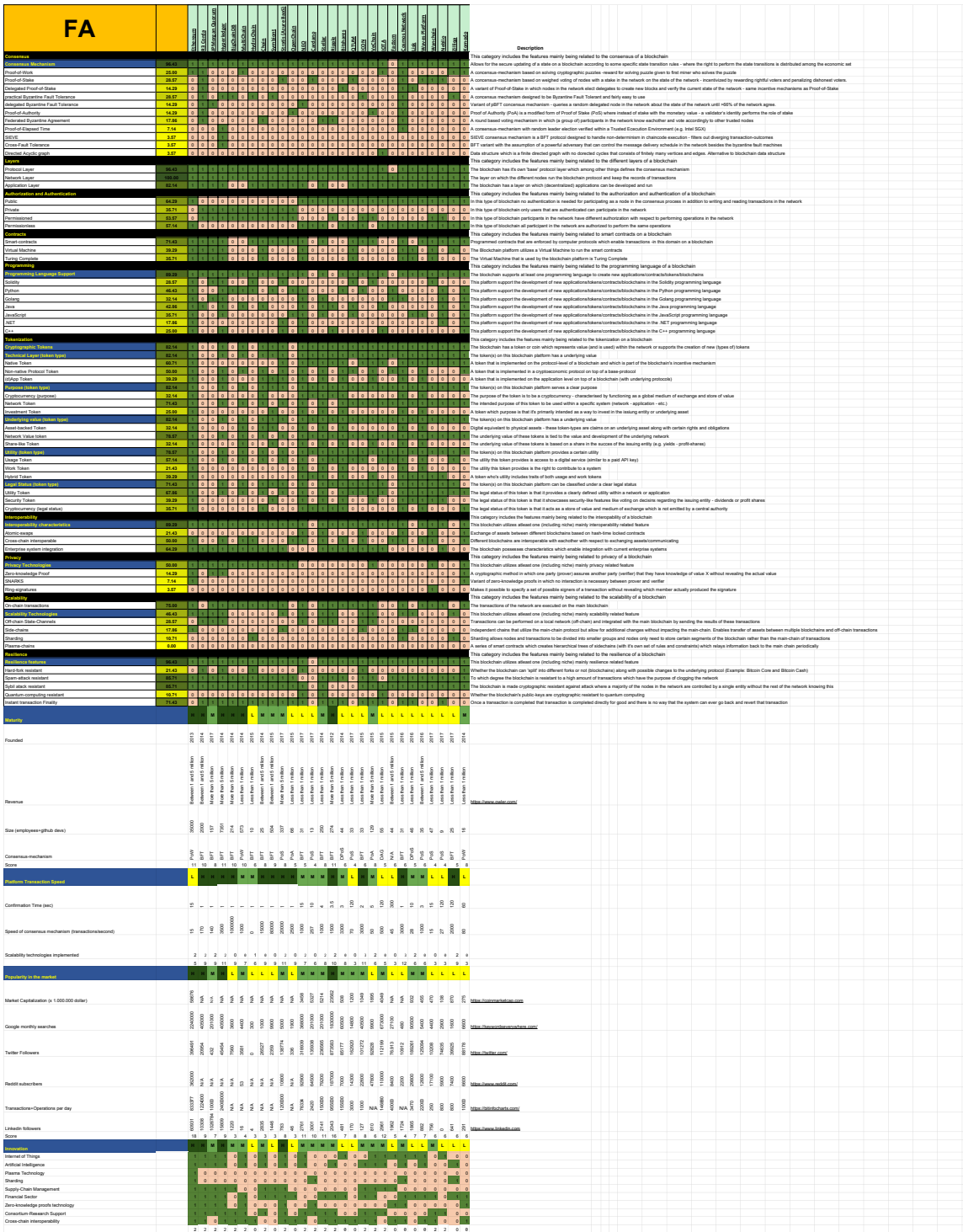
Appendix F

SF-Mapping



Appendix G

FA-mapping



Appendix H

ShareCompany BIQH Functional Requirements

Functional Requirements			
p	**#**	**Short Name**	**Description**
fr	1	Input KID	As a bank, I want to input/upload KID to the blockchain (to adhere to legislation).
fr	2	View KID	As an investor, I want to view the KID (based on ISIN code) associated to a given PRIIP.
fr	3	Update KID	As a bank, I want to update/overwrite an existing KID if its contents change (reliability of content).
fr	4	Inspect KID provisioning	As the AFM, I want to make sure that the way KIDs are created (e.g., with the form) is sufficient in terms of what is asked in the EU legislation.
fr	5	Identify	As the AFM, in case of a dispute, I want to establish the identities of the parties involved in the dispute (investor and PRIIP issuer).
fr	6	Leave trace	As an investor, whenever I (agree to) read a KID, I want to leave a trace. In this way, *when* and *what version* of the KID I read can be traced.
fr	7	Inspect transaction	As the AFM, in case of a dispute, I want as much information as possible on the relevant transaction.
fr	8	Update KID notification	As an investor, I want to be notified should the contents of a KID relevant to me change.

Appendix I

DUO User Stories + Requirements

User story 1: As a student, I only want to be able to subscribe for loans I am eligible for, so that I do not have to check what loans I am eligible for myself.
User story 2: As a student, I want to automatically be unsubscribed for loans I am not eligible for anymore, so that I do not risk having to repay money and/or fines.
User story 3: As a student, I want to be able to loan according to the terms and agreements of a subscribed loan for as long as its duration, so that I can loan what I am entitled to.
User story 4: As a student, I want to see an overview of my current loans and debts, so that I can get a hold of my current financial situation
User story 5: As a student, I want to automatically receive gifts, so that I do not have to check for and request gifts by myself.
User story 6: As a student, I want to choose what personal information from my account is shown to the financier, so that I do not have to share more information than necessary.
User story 7: As a student, I want to be able to repay money to the financier at any time, so that I can repay my debts in my own pace.
User story 8: As an agency that provides student financing, I want to be able to select the terms and agreements of a loan, so that I can decide what and how students can loan.
user story 9: As an agency that provides student financing, I want to be able to select what loans are still available, so that I can manage what loans students are allowed to subscribe for.
User Story 10: As an agency that provides student financing, I want to be able to change the interest rate yearly, so that they correspond to our current regulations.
User story 11: As an agency that provides student financing, I want to have a clear overview of all performed transaction, so that I can get a hold of the financial situation.
User story 12: As an agency that provides student financing, I want to be send and withdraw money to the system's money pool, so that I can manage the amount of money available for paying student loans.

Functional requirement 1: The user should be able to log in using a username and password.
Functional requirement 2: The user should be able to change his/her password.
Functional requirement 3: The user should be able to select what data is shared with the financier.
Functional requirement 4: The user should be able to see an overview of all loans he/she is eligible for.

Functional requirement 5: The user should be able to accept the terms and agreements of a loan he/she is eligible for.
Functional requirement 6: A student should be able to view the the loan terms and agreements that apply to him/her.
Functional requirement 7: Once a student accepted the terms and agreements of a loan, he or she is able to loan according to these agreements for as long as the loan's duration.
Functional requirement 8: The user should be able to select whether a loan should be provided ac- cording to revolving or installment credit agreements.
Functional requirement 9: In case a loan is requested using revolving credit agreements, a student should be able to request and receive money to his/her personal wallet at any time, as long as the month's maximum loan amount is not exceeded.
Functional requirement 10: In case a loan is requested using installment credit agreements, a student should be able to request a loan amount up to the loan's maximum monthly amount, being paid at the 1st of each month.
Functional requirement 11: The user should be able to select whether his/her loan is being paid in fiat money or a cryptocurrency.
Functional requirement 12: In case a user is not eligible for a loan anymore, the loan should auto- matically be terminated.
Functional requirement 13: The user should be able to repay money to the financier at any time, as long as they do not repay more than their current debt.
Functional requirement 14: In case a student has not repaid their minimum repayment amount at the end of the month, the remaining amount should be withdrawn from his/her wallet automatically.
Functional requirement 15: In case the balance of a student is to low to pay the minimum repayment amount of last month, the student will be added to a list of defaulters.
Functional requirement 16: The financier should be able to select the terms and agreements of a loan.
Functional requirement 17: The financier should be able to select what loans are currently available and which are not.
Functional requirement 18: The financier should not be able to send money which a student had not requested.
Functional requirement 19: When a student is eligible for a gift, this should be processed in his or her debt automatically.
Functional requirement 20: A miner should receive a small compensation for processing transactions.
Functional requirement 21: The financier should be able to view a list of all defaulters, including the repayment amount they lack behind.
Functional requirement 22: The financier should be able to reproduce data about a student, for up to 10 years after he/she has repayed his/her debt.
Functional requirement 23: The financier should be able to wipe out all data of a student who repayed his/her debt.

Quality requirement 1: All personal information should only be visible to the financier and the student itself.
Quality requirement 2: Everyone should be able to mine (and process transactions of) the network.
Quality requirement 3: The system should offer a user-friendly interface, usable by students with- out knowledge about blockchain and/or smart contracts technology.

Quality requirement 4: A user's password should have a length of at least 8 characters, using 3 or more special characters.
Quality requirement 5: The application should have an availability of at least 99.9 percent.
Quality requirement 6: The correctness of all processed transactions should be a 100 percent

Constraint 1: The application interface should be compatible with at least the 5 most used web browsers.
Constraint 2: The application should be able to operate on a smart phone
Constraint 3: The application should be able to operate on a laptop
Constraint 4: Optional: The application should not be using a public blockchain

Appendix J

Process-Deliverable Diagram

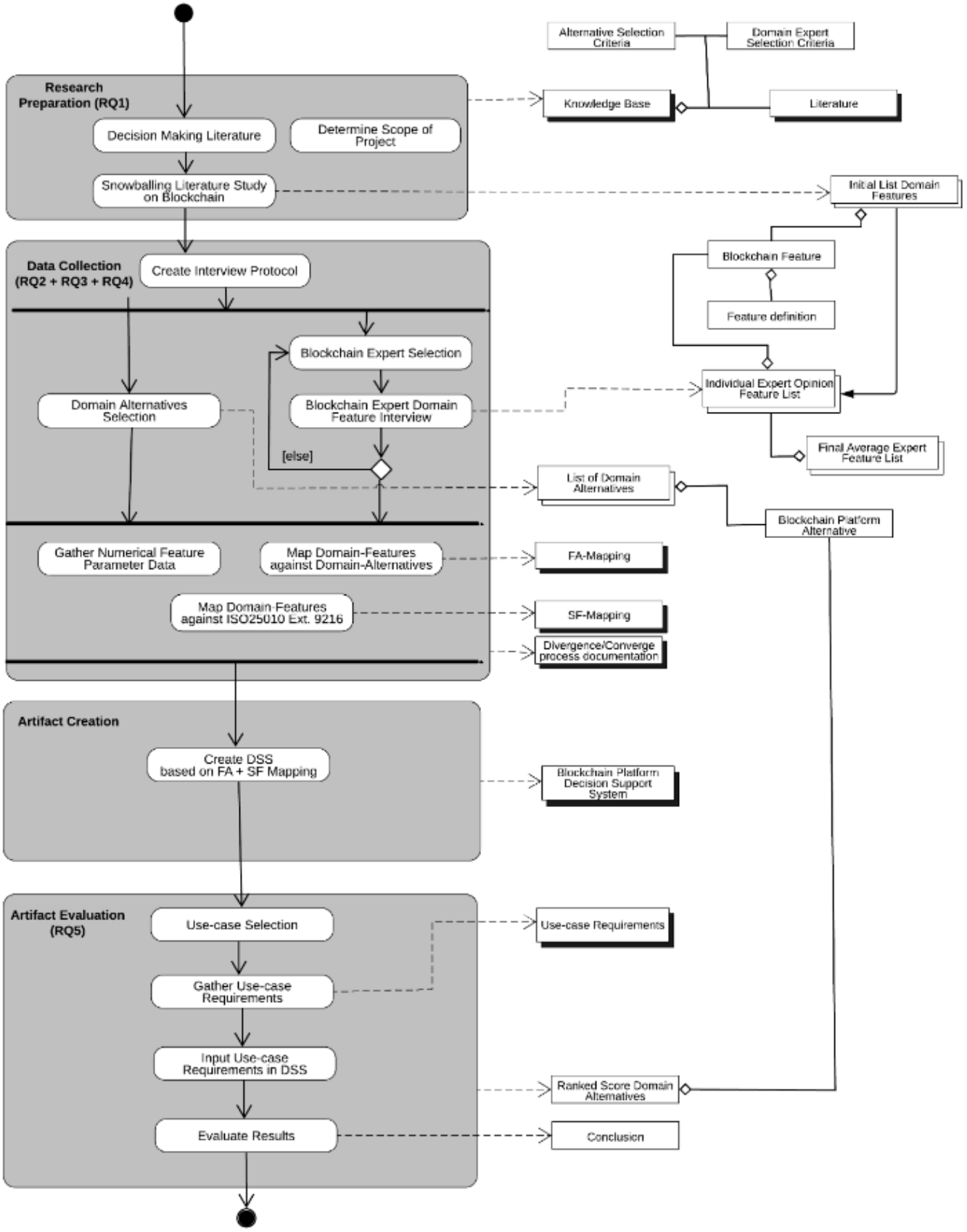


FIGURE J.1: Process-Deliverable Diagram of this research

Appendix K

Expert Invitation Letter Dutch

Universiteit Utrecht

Postbusadres
P.O. Box 90 089
3508 TB Utrecht
The Netherlands

Afzender
Department of Information and Computing Sciences

Bezoekadres
Princetonplein 5 Kamer BBL-884
3584 CC Utrecht

Telefoon +31 30 253 1454
E-mail Slinger.jansen@uu.nl
Website https://slingerjansen.nl/
Datum 30-01-2018
Onderwerp Uitnodiging blockchain onderzoek

Geachte,

De Universiteit Utrecht onderzoekt de rol die de blockchain technologie kan spelen in de toekomst. In dit onderzoek zal centraal staan hoe bedrijven de blockchain technologie kunnen toepassen in hun dagelijkse processen. Op dit moment is namelijk nog één van de grote hindernissen omtrent de blockchain technologie de gelimiteerde hoeveelheid wetenschappelijk onderzoek. Daarnaast is het toepassen van de blockchain technologie nog vaak onbekend terrein.

Dit onderzoek is onderdeel van het AMUSE-project: Adaptable Model-based and User-specific Software Ecosystems. Dit AMUSE-project is een academische samenwerking tussen de Universiteit Utrecht, Vrije Universiteit Amsterdam en AFAS Software.
Het doel van dit specifieke onderzoek is om een Decision-Support System te creëren die bedrijven helpt in hun beslissingsproces bij de keuze tussen de verschillende beschikbare blockchain-oplossingen.
Dit Decision Support System zal gecreëerd worden op basis van kennis van verschillende blockchain experts. Door middel van deze brief wil ik u uitnodigen deel te nemen aan dit onderzoek aangezien u een expert bent in uw domein. Deze kennis zal besproken worden tijdens een semigestructureerd interview met een Junior Researcher. De verwachting is dat het interview ongeveer een uur kost en kan via Skype (of een vergelijkbaar alternatief), of in persoon.
Mocht u verdere vragen hebben omtrent dit onderzoek dan horen wij dit graag.
We kijken uit naar uw positieve reactie!

Met vriendelijke groet,

Dr. R.L. Jansen
Assistant professor Department of Information and Computer Science Utrecht University

Jacco Ronaldo Quirinus Verkleij
Junior Researcher

Appendix L

Expert Invitation Letter English

Universiteit Utrecht

PO Box Address
P.O. Box 90 089
3508 TB Utrecht
The Netherlands

Sender
Department of Information and Computing Sciences

Visiting address
Princetonplein 5 Room BBL-884
3584 CC Utrecht

Phone +31 30 253 1454
E-mail Slinger.jansen@uu.nl
Website https://slingerjansen.nl/
Date 30-01-2018
Topic Invitation blockchain research

Dear,

Utrecht University is researching the future role of the blockchain technology. The core of this research is dedicated to researching how companies can utilize the blockchain technology in their daily processes. Currently one of the major obstacles with respect to the blockchain technology is the limited amount of scientific research that has been performed.
In addition to this using the blockchain technology in practice is still largely unknown territory.

This research is part of the AMUSE project: Adaptable Model-based and User-specific Software Ecosystems. This AMUSE project is an academic collaboration between Utrecht University, Free University of Amsterdam and AFAS Software.
The aim of this specific research is to create a Decision Support System that aids companies in their decision process when choosing a blockchain-solution.
This Decision Support system shall be created based on the knowledge of several blockchain experts. Through the means of this letter I would like to invite you to participate in this research since you are considered an expert in your domain. This knowledge will be discussed during a semi-structured interview with a Junior Researcher. The interview is expected to take about an hour and can be done through Skype (or another video chat platform), or in person.
Should you have further questions with respect to this research then we're glad to answer those questions. We look forward to your positive reply!

Yours sincerely,

Dr. R.L. Jansen
Assistant professor Department of Information and Computer Science Utrecht University

Jacco Ronaldo Quirinus Verkleij
Junior Researcher

Appendix M

Expert Feature Opinion

Category/Feature	(Sub)Feature	Average	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7	Interview 8	Interview 9
Boolean											
Consensus Mechanism	Proof-of-Work	1	1	1	1	1	1	1	1	1	1
Category	Proof-of-Stake	1	1	1	1	1	1	1	1	1	1
	delegated Proof-of-Stake	0.555555	1	1	1	1	0	0	0	1	0
	practical Byzantine Fault Tolerance	0.777777	1	1	1	0	0	1	1	1	1
	federated Byzantine Agreement	0.777777	1	1	1	0	0	1	1	1	1
	delegated Byzantine Fault Tolerance	0.777777	1	1	1	0	0	1	1	1	1
	Proof-of-Authority	0.555555	0	0	0	1	1	1	1	1	0
	Proof-of-Elapsed Time	0.555555	0	1	1	0	0	1	1	1	0
	Proof-of-Burn	0.111111	0	0	1	0	0	0	0	0	0
	Proof-of-Luck	0.111111	0	0	1	0	0	0	0	0	0
	Directed Acyclic Graph (variants)	0.555555	1	0	0	0	0	1	1	1	1
Hashing Algorithm	SHA-256	0.444444	0	0	1	0	0	1	0	1	1
Category	SHA-3	0.444444	0	0	1	0	0	1	0	1	1
	md-5	0.222222	0	0	1	0	0	0	0	1	0
	SHA-512	0.333333	0	0	0	0	0	1	0	1	1
	ASIC-algorithm	0.333333	0	0	1	0	0	0	0	1	1
	Cryptonight	0.222222	0	0	1	0	0	0	0	1	0
Layers	Protocol Layer	0.777777	1	0	0	1	1	1	1	1	1
Category	Network Layer	0.777777	1	0	0	1	1	1	1	1	1
	Application Layer	0.888888	1	0	1	1	1	1	1	1	1
Identity	Public	1	1	1	1	1	1	1	1	1	1
Category	Private	1	1	1	1	1	1	1	1	1	1
	Permissioned	1	1	1	1	1	1	1	1	1	1
	Permissionless	1	1	1	1	1	1	1	1	1	1
Contracts	Smart-contracts	1	1	1	1	1	1	1	1	1	1
Feature	Virtual Machine	0.666666	0	1	1	0	0	1	1	1	1
	Docker	0.111111	0	1	0	0	0	0	0	0	0
	Turing-complete	0.666666	0	1	1	0	0	1	1	1	1
Programming Language	Solidity	0.888888	1	1	1	1	0	1	1	1	1
Smart Contract Support	C#	0.222222	0	1	1	0	0	0	0	0	0
Category	Golang	0.555555	0	1	0	0	0	1	1	1	1
	JavaScript	0.555555	0	1	0	0	0	1	1	1	1
	Java	0.777777	0	1	1	1	0	1	1	1	1
	C++	0.555555	0	1	1	1	0	0	1	0	1
	Python	0.555555	0	1	1	0	0	1	1	1	0
	Haskell	0.222222	0	1	0	0	0	0	1	0	0
Tokens	Tokenization	1	1	1	1	1	1	1	1	1	1
Feature	Native	0.555555	0	1	0	1	0	0	1	1	1
	Non-native	0.555555	0	1	0	1	0	0	1	1	1
	Cryptocurrency	0.777777	1	1	1	1	0	1	1	0	1
	Utility	0.666666	1	1	0	1	0	0	1	1	1
	Security	0.555555	1	1	1	0	0	1	0	0	1
	Usage	0.555555	0	1	1	1	0	0	1	0	1
	Asset	0.666666	1	1	0	0	0	1	1	1	1
	Work	0.333333	0	1	0	0	0	0	1	0	1
	Hybrid	0.333333	0	1	0	0	0	0	1	0	1
Scalability Technologies	On-chain transactions	1	1	1	1	1	1	1	1	1	1
Category	Off-chain transactions	1	1	1	1	1	1	1	1	1	1
	Side-chains	0.888888	1	0	1	1	1	1	1	1	1
	Sharding	0.888888	1	0	1	1	1	1	1	1	1
	Plasma-chains	0.666666	0	0	0	1	1	1	1	1	1
Interoperability	Atomic-swaps	0.666666	0	0	1	1	0	1	1	1	1
Category	Cross-chain interoperable	0.666666	1	0	0	1	1	1	0	1	1
	Enterprise system integration	0.777777	1	1	1	0	1	1	0	1	1
Privacy Technologies	Zero-knowledge Proof	1	1	1	1	1	1	1	1	1	1
Category	Ring-signatures	0.555555	0	0	1	1	0	0	1	1	1
Resilience features	Hard-fork resistant	0.555555	1	1	1	0	0	1	0	1	0
Category	Spam-attack resistant	0.777777	1	1	1	0	0	1	1	1	1
	Sybil attack resistant	0.888888	1	1	1	1	0	1	1	1	1
	Quantum resistant	0.555555	0	0	1	1	0	1	0	1	1
	Transaction Irreversibility	0.555555	1	0	1	1	0	0	0	1	1
Utility	Wallet	0.222222	0	0	1	1	0	0	0	0	0
Numerical	Transaction speed	1	1	1	1	1	1	1	1	1	1
	Block-size	0.444444	1	1	0	0	1	1	0	0	0

Appendix N

Domain Features Definition

Feature	Reference number	Definition
Consensus-mechanism	1	Allows for the secure updating of a state on a blockchain according to some specific state transition rules, where the right to perform the state transitions is distributed among the economic set
Proof-of-Work	1	A concensus-mechanism based on solving cryptographic puzzles, reward for solving puzzle given to first miner who solves the puzzle
Proof-of-Stake	3	A concensus-mechanism based on weighed voting of nodes with a stake in the network on the state of the network, incentivized by rewarding rightful voters and penalizing dishonest voters.
Delegated Proof-of-Stake	3	A variant of Proof-of-Stake in which nodes in the network elect delegates to create new blocks and verify the current state of the network, same incentive mechanisms as Proof-of-Stake
practical Byzantine Fault Tolerance	3	A concensus mechanism designed to be Byzantine Fault Tolerant, membership for network participation set by central authority
federated Byzantine Agreement	3	A round based voting mechanism in which (a group of) participants in the network know eachother and vote accordingly to other trusted nodes
delegated Byzantine Fault Tolerance	3	Variant of pBFT concensus mechanism, queries a random delegated node in the network about the state of the network until >66% of the network agree.
Proof-of-Authority	23	Proof of Authority (PoA) is a modified form of Proof of Stake (PoS) where instead of stake with the monetary value, a validator's identity performs the role of stake
Proof-of-Elapsed Time	3	A consensus-mechanism with random leader election verified within a Trusted Ececution Environment (e.g. Intel SGX)
Directed Acyclic Graph	24	Data structure which is a finite directed graph with no dorected cycles that consists of finitely many vertices and edges. Alternative to blockchain data structure
Proof-of-Burn	9	A consensus mechanism that shows that verifiers in the network have put in effort, but without expending real resources like electricity
Proof-of-Luck	25	Consensus-mechanism which is highly comparable to Proof-of-Elapsed Time besides some minor changes
Cross-fault tolerance	3	BFT variant with the assumption of a powerful adversary that can control the message delivery schedule in the network besides the byzantine fault machines
SIEVE	3	SIEVE consensus mechanism is a BFT protocol designed to handle non-determinism in chaincode execution, filters out diverging transaction-outcomes
SHA-256	1	PoW 256 bits hashing algorithm based on the Merkle-Damgard structure, GPU intensive
Scrypt	33	CPU optimized hashing algorithm version of SHA-256
Cryptonight	33	CPU optimized PoW hashing algorithm which is ASIC-resistant
KECCAK-256 (SHA-3)	34	KECCAK PoW hashing algorithm standardized to SHA-3 hashing algorithm standards
SHA-512	34	512 bits hashing algorithm variant of SHA-256
md-5	35	128 bit based hashing algorithm
ASIC-algorithm	32	Application-specific integrated circuit algorithm, opposed to a CPU or GPU specific hardware is required to utilize this algorithm
Public	2	In this type of blockchain no authentication is needed for participating as a node in the consensus process in addition to writing and reading transactions in the network
Private	2	In this type of blockchain only users that are authenticated can participate in the network
Permissionless	4	In this type of blockchain all participant in the network are authorized to perform the same operations
Permissioned	4	In this type of blockchain participants in the network have different authorization with respect to performing operations in the network
Hybrid/Consortium	2	This type of blockchain has a set of pre-selected semi-trusted and verified nodes which participate in the consensus process, nodes can have different write/read rights
Acces control	2	Granting of permission and different rights in the network
Smart contract	5	Programmed contracts that are enforced by computer protocols which enable transactions, in this domain run on a blockchain (platform)
Solidity	10	Smart contracts can be developed and run in the Solidity programming language
C#	13	Smart contracts can be developed and run in the C# programming language
Golang	13	Smart contracts can be developed and run in the Golang programming language
JavaScript	11	Smart contracts can be developed and run in the JavaScript programming language
Java	13	Smart contracts can be developed and run in the Java programming language
C++	11	Smart contracts can be developed and run in the C++ programming language
Python	11	Smart contracts can be developed and run in the Python programming language
.NET	13	Smart contracts can be developed and run in the .NET framework
Haskell	12	Smart contracts can be developed and run in the Haskell programming language
Virtual Machine	30	The Blockchain platform utilizes a Virtual Machine to run the smart contracts
Docker	31	A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.
Turing Complete	30	The Virtual Machine that is used by the blockchain platform is Turing Complete
Zero-knowledge proof	15	A cryptographic method in which one party (prover) assures another party (verifier) that they have knowledge of value X without revealing the actual value
Ring signatures	20	Makes it possible to specify a set of possible signers of a transaction without revealing which member actually produced the signature (Rivest, 2001)
Sybil attack resistant	3	The blockchain is made cryptographic resistant against attack where a majority of the nodes in the network are controlled by a single entity without the rest of the network knowing this
Spam attack resistant	29	To which degree the blockchain is resistant to a high amount of transactions which have the purpose of clogging the network
Transaction finality	28	Whether the transaction is executed directly or is probalistic
Cross-chain interoperability	21	Different blockchains are interoperable with eachother with respect to exchanging assets/communicating
On-chain transactions	1	Once a transaction is completed that transaction is completed directly for good and there is no way that the system can ever go back and revert that transaction
Off-chain transactions	17	Transactions can be performed on a local network (off-chain) and integrated with the main blockchain by sending the results of these transactions
Side-chains	16	Independent chains that utilize the main-chain protocol but allow for additional changes without impacting the main-chain. Enables transfer of assets between multiple blockchains and off-chain transactions
Plasma-chains	7	A series of smart contracts which creates hierarchical trees of sidechains (with it's own set of rules and constraints) which relays information back to the main chain periodically
Sharding	8	Sharding allows nodes and transactions to be divided into smaller groups and nodes only need to store certain segments of the blockchain rather than the main-chain of transactions
Hard-fork resistance	3	Whether the blockchain can 'split' into different forks or not (blockchains) along with possible changes to the underlying protocol (Example: Bitcoin Core and Bitcoin Cash)
Quantum resistant	26	Whether the blockchain's public-keys are cryptographic resistant to quantum computing
Enterprise system integration	21	The blockchain possesses characteristics which enable integration with current enterprise systems
Protocol Layer	18	The blockchain has it's own 'base' protocol layer which among other things defines the consensus mechanism,
Network Layer	14	The layer on which the different nodes run the blockchain protocol and keep the records of transactions
Application Protocol Layer	19	The blockchain has a layer which defines the cryptoeconomic rules of the application layer, utilizes the base protocol
Application Layer	14	The blockchain has a layer on which (decentralized) applications can be developed and run
Atomic-swaps	16	Exchange of assets between different blockchains based on hash-time locked contracts
Cryptographic Tokens	6	The blockchain has a token or coin which represents value (and is used) within the network (Token classicifcation framework, 2018)
Native-Token	6	A token that is implemented on the protocol-level of a blockchain and which is part of the blockchain's incentive mechanism
Non-native Protocol Token	6	A token that is implemented in a cryptoeconomic protocol on top of a base-protocol
(d)App-token	6	A token that is implemented on the application level on top of a blockchain (with underlying protocols)
Cryptocurrency (purpose)	6	The purpose of the token is to be a cryptocurrency, characterised by functioning as a global medium of exchane and store of value
Network Token	6	The intended purpose of this token to be used within a specific system (network, application, etc.)
Investment Token	6	A token which purpose is that it's primarily intended as a way to invest in the issiung entity or underlying asset
Asset-backed Token	6	Digital equivelant to physical assets, these token-types are claims on an underlying asset along with certain rights and obligations
Network Value token	6	The underlying value of these tokens is tied to the value and development of the underlying network
Share-like Token	6	The underlying value of these tokens is based on a share in the succes of the issuing entity (e.g. dividens, profit-shares)
Usage Token	6	The utility this token provides is access to a digital service (similar to a paid API key)
Work Token	6	The utility this token provides is the right to contribute to a system
Hybrid Token	6	A token who's utility includes traits of both usage and work tokens
Utility token	6	The legal status of this token is that it provides a clearly defined utility within a network or application
Security Token	6	The legal status of this token is that it showcases security-like features like voting on decisins regarding the issuing entity, dividends or profit shares
Cryptocurrency (legal status)	6	The legal status of this token is that it acts as a store of value and medium of exchange which is not emitted by a central authority
Transaction speed	27	The amount of transactions which the blockchain is capable of processing per time unit (numerical value)
Block size	1	The size of the blocks on the blockchain (in MB) in which the transactions are stored

Appendix O

Domain Features Definition References

ber	Reference
1	Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved from https://bitco.in/pdf/bitcoin.pdf
2	Buterin, V. (2015) On Public and Private Blockchains. Retrieved from https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/
3	Baliga, A. (2017). Understanding Blockchain Consensus Models. Retrieved from https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf
4	Perretta, H. (2017, June 27). Private vs. Public and Permissioned vs. Permission-less [Blog post]. Retrieved March 20, 2018, from http://blocktonite.com/2017/06/27/private-vs-public-and-permissioned-vs-permission-less/
5	Szabo, N. (1994). Smart Contracts. Retrieved from http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html
6	Euler, T. (2018.) The Token classification Framework: A multi-dimensional tool for understanding and classifying crypto tokens. Retrieved from http://www.untitled-inc.com/the-token-classification-framework-a-multi-dimensional-tool-for-understanding-and-classifying-crypto-tokens/
7	Poon, J., Buterin, V. (2017) Plasma: Scalable Autonomous Smart Contracts. Retrieved from https://plasma.io/plasma.pdf
8	Buterin, V. (2017) Sharding FAQ. Retrieved from https://github.com/ethereum/wiki/wiki/Sharding-FAQ
9	Baliga, A. (2016) The blockchain Landscape. Retrieved from https://pdfs.semanticscholar.org/c826/b333df5b04e3053a7c2cb3b881bff1d952942.pdf
10	Ray, J. (2016) Programming languages intro. Retrieved from https://github.com/ethereum/wiki/wiki/Programming-languages-intro
11	Rosic, A. (2017) Blockchain Coding. Retrieved from https://blockgeeks.com/guides/blockchain-coding/
12	Kiayias, A., Russell, A., David, B., Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol
13	NEO Foundation (2017) NEO White Paper: A distributed network for the Smart Economy, retrieved from http://docs.neo.org/en-us/index.html
14	Everts, M. (2017) De betekenis van blockchain. Retrieved from https://esb.nu/esb/20032301/de-betekenis-van-blockchain
15	Koens, T., Ramaekers, C., Wijk, van C. (2017) Efficient Zero-Knowledge Range Proofs in Ethereum
16	Back et al (2014) Enabling Blockchain Innovations with Pegged Sidechains. Retrieved from http://kevinriggen.com/files/sidechains.pdf
17	Bitcoinwiki, accessed on 23-04-2018 from https://en.bitcoin.it/wiki/Off-Chain_Transactions
18	Swan, M. (2015). Blockchain: Blueprint for a New Economy. Sebastopol, United States of America: O'Reilly Media
19	Monegro, J. (2016). Fat Protocols. Retrieved from http://www.usv.com/blog/fat-protocols
20	Zhang, L., Zhang, F., Wu, W. (2007) A provably Secure Ring Signature Scheme in Certificateless Cryptography. Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-75670-5_7
21	Korpela, K., Hallikas, J., Dahlberg, T. (2017) Digital Supply chain Transformation toward Blockchain Integration
22	Buterin, V. (2016) Chain Interoperability. From https://www.r3.com/wp-content/uploads/2017/06/chain_interoperability_r3.pdf
23	POA Network (2017). Proof of Authority: consensus model with Identity at Stake
24	Witherspoon (2018). A hitchhiker's Guide to Consensus Algorithms. https://hackernoon.com/a-hitchhikers-guide-to-consensus-algorithms-d81aae3eb0e3
25	Milutinovic et al (2017). Proof of Luck: an Efficient Blockchain Consensus Protocol. retrieved from https://arxiv.org/pdf/1703.05435.pdf
26	Feo, De L., Jao, D., Plut, J. (2014) Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies
27	Kiayias, A., Panagiotakos, G. (2015). Speed-Security Tradeoffs in Blockchain Protocols. retrieved from https://pdfs.semanticscholar.org/7deb/ffbb85a020aa8f162dd862336e9416550f3.pdf
28	Buterin, V. (2016) On settlement Finality. retrieved from https://blog.ethereum.org/2016/05/09/on-settlement-finality/
29	Room, C. Blockchain consensu. retrieved from https://devopedia.org/blockchain-consensus
30	Ethereum Github. Retrieved from http://ethdocs.org/en/latest/introduction/what-is-ethereum.html#ethereum-virtual-machine
31	Taylor, G. (2016) BUILDING PRIVATE ETHEREUM NETWORKS WITH DOCKER COMPOSE. Retrieved from https://capgemini.github.io/blockchain/ethereum-docker-compose/
32	https://en.wikipedia.org/wiki/Application-specific_integrated_circuit
33	http://www.bitcoinlion.com/cryptocurrency-mining-hash-algorithms/
34	https://keccak.team/index.html
35	Kasgar et al. (2013). A review Paper of Message Digest 5 (MD5). Volume 1, Issue 4, December 2013. retrieved from http://ijmemr.org/Publication/V1I4/IJMEMR-V1I4-005.pdf

Appendix P

BPDSS Screenshots

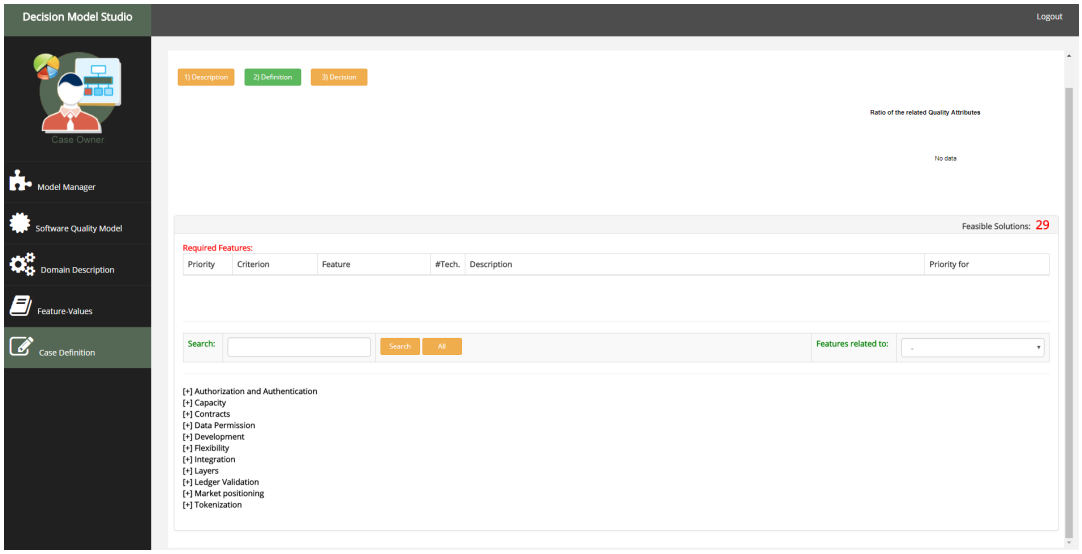


FIGURE P.1

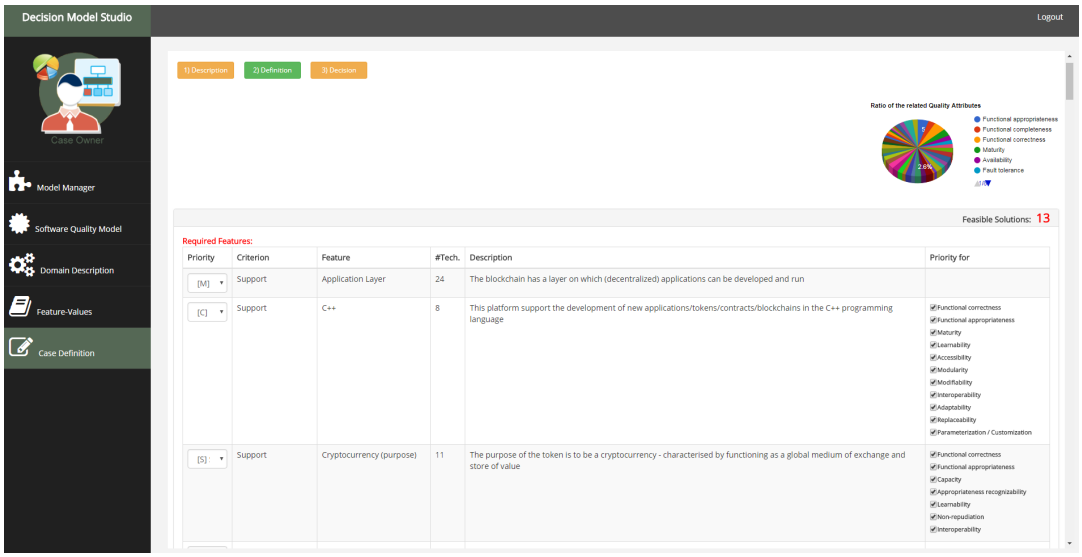


FIGURE P.2

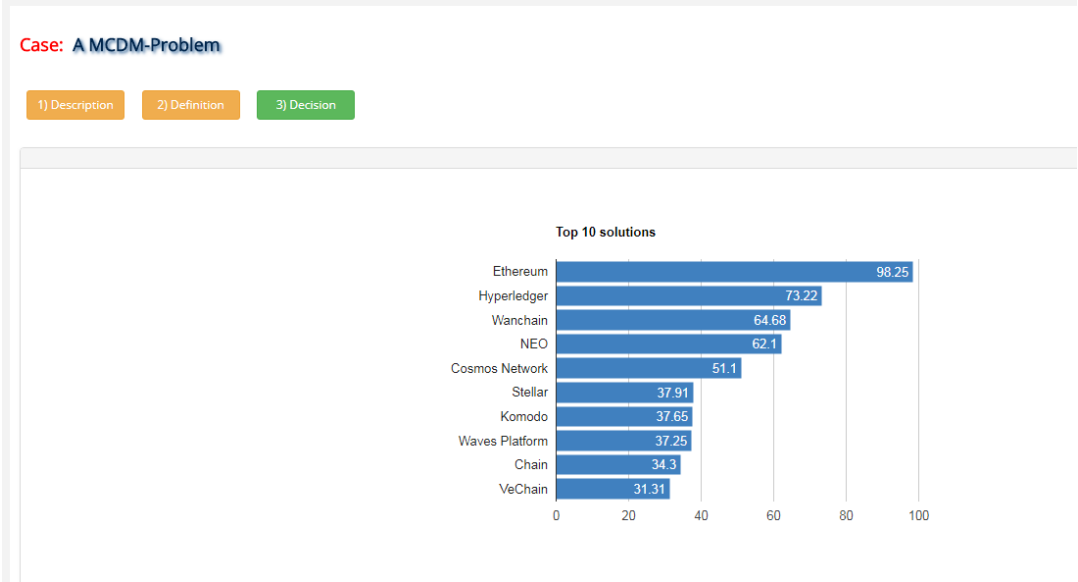


FIGURE P.3

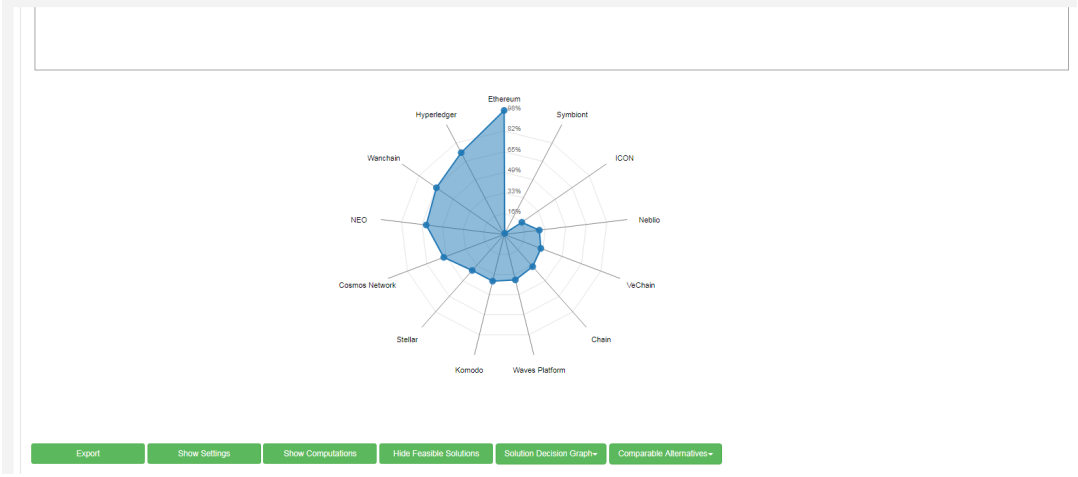


FIGURE P.4

Top-10 Technologies: 1. NULL 99.64	IOTA	Cosmos Network	NEO
[M] Application Layer	■	■	■
[M] Cryptographic Tokens	■	■	■
[S] delegated Byzantine Fault Tolerance		■	■
[S] Delegated Proof-of-Stake		■	
[W] Directed Acyclic graph	■	■	■
[M] Enterprise system integration	■	■	■
[M] Interoperability technologies	■	■	■
[M] Network Layer	■	■	■
[S] Network Token	■	■	■
[S] Network Value token	■	■	■
[M] On-chain transactions	■	■	■
[M] Permissioned	■	■	■
[C] Privacy Technologies			
[S] Private		■	
[W] Proof-of-Work	■	■	■
[M] Protocol Layer	■	■	■
[S] Security Token		■	■
[S] Share-like Token		■	■
[M] Smart-contracts	■	■	■

FIGURE P.5

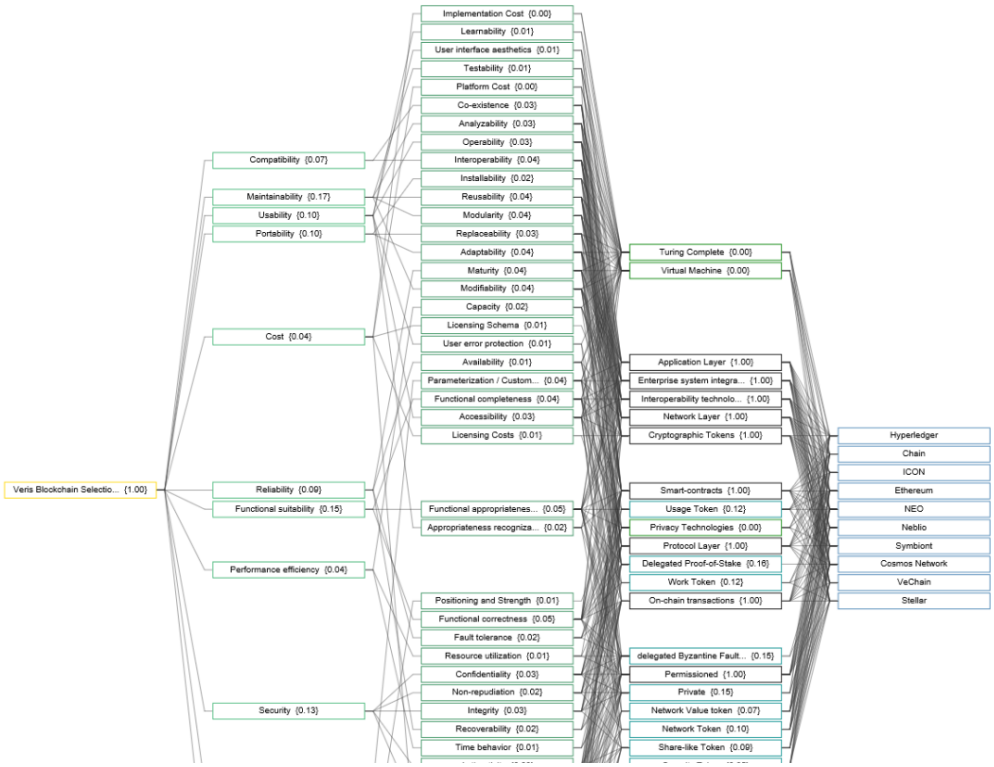


FIGURE P.6