

# Decision Support for Blockchain Platform Selection: Three Industry Case Studies

Siamak Farshidi, Slinger Jansen, Sergio España , and Jacco Verkleij

## Abstract

Blockchain technology has received significant attention recently, as it offers a reliable decentralized infrastructure for all kinds of business transactions. Software-producing organizations are increasingly considering blockchain technology for inclusion into their software products. Selecting the best fitting blockchain platform requires the assessment of its functionality, adaptability, and compatibility to the existing software product. Novice software developers and architects are not experts in every domain, so they should either consult external experts or acquire knowledge themselves. The decision-making process gets more complicated as the number of decision-makers, alternatives, and criteria increases. Hence, a decision model is required to externalize and organize knowledge regarding the blockchain platform selection context.

Recently, we designed a decision support system to use such decision models to support decision-makers with their technology selection problems in software production. In this study, we introduce a decision model for the blockchain platform selection problem. The decision model has been evaluated through three real-world case studies at three software-producing organizations. The case-study participants asserted that the approach provides significantly more insight into the blockchain platform selection process, provides a richer prioritized option list than if they had done their research independently, and reduces the time and cost of the decision-making process.

## Index Terms

blockchain platform selection; blockchain decision model; technology selection; multi-criteria decision making; decision support system;

## I. INTRODUCTION

Blockchain technology offers a reliable decentralized infrastructure, which means that it does not have to be controlled by one central authority for business applications. In other words, blockchain technology can be employed as an application platform to build the underlying trust infrastructure of any distributed system. Since public blockchain platforms are open to the world, they can rapidly draw the attention of software development companies and communities to the strengths of blockchain technology.

Software-producing organizations are increasingly considering distributed ledger and blockchain technology for inclusion into their software products. The selection process refers to the steps involved in choosing and evaluating the best fitting blockchain platforms for software-producing organizations according to their preferences and requirements. The selection process is complicated because many criteria, such as security, interoperability, consensus mechanisms, and platform transaction speed, have to be considered and fitted to the needs of the project at hand. Additionally, as a software product is typically a long-living system, such decisions determine the future of the product and the costs associated with its development.

Numerous public blockchain platforms have emerged recently and utilized in diverse business applications. For instance, Hyperledger<sup>1</sup> and Ethereum<sup>2</sup> offer public blockchain platforms. The fundamental difference between Hyperledger and Ethereum is the goal they are designed for. Hyperledger is an open-source development project that offers multiple blockchain platforms and supports the collaborative development of blockchain-based distributed ledgers. On the other hand, Ethereum is an open-source distributed public blockchain platform that its smart contracts enable decentralized applications to be implemented and deployed on it. As the number of blockchain platforms in the market is increasing rapidly, blockchain platform selection is becoming a significant challenge for software-producing organizations. Hence, knowledge regarding blockchain platforms has to be collected, organized, stored, and quickly retrieved when it needs to be applied.

In literature, a variety of multi-criteria decision-making (MCDM) techniques has been introduced to address different technology selection problems for software-producing organizations. An MCDM problem deals with the evaluation of a set of alternatives and takes into account a set of decision criteria [1]. In our recent study [2], we introduced a technology selection framework that is used to build decision models for MCDM problems and assist decision-makers at software-producing organizations with the decision-making process. Furthermore, we have instantiated the framework to build two decision models for the Database Management System [3] and Cloud Service Provider [2] selection problem. In this study, the

S. Farshidi, S. Jansen, and S. España are with the Department of Information and Computer Science at Utrecht University, Utrecht, the Netherlands (e-mail: {s.farshidi, slinger.jansen, s.espana}@uu.nl)

J. Verkleij is with ABN AMRO Bank N.V. (e-mail: Jacco.verkleij@nl.abnamro.com)

<sup>1</sup><https://www.hyperledger.org>

<sup>2</sup><https://www.ethereum.org>

blockchain platform selection process is modeled as an MCDM problem, and the technology selection framework is employed to build a decision model for this MCDM problem.

Recently, we designed and implemented a Decision Support System (DSS) [4] for supporting decision-makers with their MCDM problems in software production. The DSS provides a decision model studio<sup>3</sup> for building decision models based on the technology selection framework. Moreover, such decision models can be uploaded to the knowledge base of the DSS to facilitate the decision-making process for software-producing organizations according to their requirements and preferences. The DSS provides a discussion and negotiation platform to enable decision-makers at software-producing organizations to make group decisions. Furthermore, the DSS can be used over the full life-cycle and can co-evolve its advice based on evolving requirements. During this research, we have built a decision model based on the technology selection framework for the blockchain platform selection problem, and then we have uploaded the decision model to the knowledge base of the DSS; finally, the outcomes of the DSS have employed in the case studies.

Please note that the proposed decision model in this study contains reusable knowledge regarding potential blockchain platforms and features. Such knowledge can educate and support the decision-makers to understand: 1) which blockchain platforms are available at the moment, 2) the capabilities of the blockchain systems, and 3) which features are fulfilled by which blockchain platforms.

The rest of this study is structured as follows: Section II describes our research method, which is based on design science and exploratory theory-testing case studies. Section III positions the proposed approach in this study among the other blockchain platform selection techniques in the literature. Section IV outlines a brief description of the DSS and the technology selection framework. This study has the following contributions:

- Section IV introduces a decision model, in the form of reusable knowledge, for the blockchain platform selection problem based on the technology selection framework [2].
- Section V describes the three conducted case studies that evaluate the effectiveness and usefulness of the approach to address the blockchain selection problem.
- Section VI analyzes the final results of the DSS and compares the outcomes of the DSS with the case-study participants shortlist of feasible blockchain platforms. The results show that the DSS recommended nearly the same solutions as the case-study participants suggested to their companies after extensive analysis and discussions, and does so more efficiently.

Section VII highlights barriers to the knowledge acquisition and decision-making process, such as motivational and cognitive biases, and argues how we have minimized these threats to the validity of the results. Finally, section VIII summarizes the proposed approach, defends its novelty, and offers directions for future studies.

## II. RESEARCH APPROACH

Rationality is the extent to which a decision-making process entails the compilation of information related to the domain of the problem and the degree of confidence upon analysis of this information in making the decision [5]. The decision-making process consolidates critical assessment of evidence and a structured process that requires time and conscious effort [6]. The decision-making process encourages decision-makers to establish relevant decision criteria, recognize a comprehensive collection of alternatives, and assess the alternatives accurately [7]. In other words, knowledge acquisition in the decision-making process is a time-consuming process in which the domain of the problem (decision context) should be interpreted accurately, and potential criteria and alternatives should be identified and compared precisely.

The decision-making process gets more complicated as the number of decision alternatives and criteria increases. Decision-makers at software-producing organizations are not experts in every domain, so they should either consult external experts or acquire knowledge themselves. In both cases, there is an investment of time (and eventually, money) that needs to be factored into the decision making process.

Recently, we introduced the technology selection framework [2] to build decision models for technology selection problems in software production. In addition, we designed and implemented a decision support system to employ such decision models to facilitate the decision-making process for decision-makers at software-producing organizations. The framework provides a guideline for decision-makers to model their technology selection problems as MCDM problems. The framework incorporates the following six-step decision-making process [8]: 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 other words, the framework is employed to build decision models for MCDM problems and find suitable alternatives for software-producing organizations based on their requirements and priorities.

The research approach for creating decision models for MCDM problems is Design Science, which addresses research through the building and evaluation of artifacts to meet identified business needs [9]. Knowledge engineering theories have been employed to design and implement the DSS and the technology selection framework. Thirteen experts (three DSS experts, four blockchain researchers from Dutch research institutes, two blockchain-developers, and four blockchain consultants/public-speakers) participated in this research to evaluate the DSS and the decision model for the blockchain platform selection problem.

<sup>3</sup>The decision model studio is available online on the DSS website: <https://dss-mcdm.com>

The experts were pragmatically selected according to their expertise and experience that they mentioned on their *LinkedIn* profile. Each of the interview series followed a semi-structured interview protocol and lasted between 45 and 90 minutes.

The DSS experts confirm that the DSS contains the main components of a standard DSS. Moreover, they asserted that the score calculation process of the DSS is not dependent on the knowledge-base facts and rules (i.e., the decision model). Therefore, the DSS would not generate invalid solutions if the decision models in the knowledge-base change or evolve during the time.

In this study, the primary source of knowledge to build a valid decision model is blockchain experts. Acquired knowledge during each interview typically propagated to the next to build and validate the decision model incrementally. Finally, the decision model was sent to the interview participants afterward for final confirmation.

The efficacy and effectiveness of the decision model have been evaluated through three exploratory theory-testing case studies. The unit of analysis is a unique blockchain platform selection for software-producing organizations. We performed three industry case studies at three blockchain-based software development companies to evaluate the decision model. The case studies typically consisted of (1) defining the blockchain feature requirements, (2) prioritizing them, and (3) comparing the DSS feasible solutions with the solutions that the experts had suggested. None of the interviewed experts mentioned above were in any way involved in the subsequent case studies.

### III. RELATED WORK

In this research, Snowballing was employed as the primary method to investigate the existing literature related to the techniques that address the blockchain platform selection problem for software-producing organizations.

In literature, some studies point out that benchmarking and performance testing can be employed to evaluate and compare a collection of blockchain platforms against each other. A subset of such studies is presented as follows:

Dinh et al. [10] present a benchmarking framework for evaluating private blockchain systems. The benchmark contains workloads for measuring the data processing performance and workloads for understanding the performance at different layers of the blockchain.

Hileman and Rauchs [11] provide an empirical overview of the current state of both enterprise and public sector use of blockchain and distributed ledger technology. They report the emergence and evolution of the distributed ledger technology ecosystem, explore actors and their business models and examine the current state of the industry in terms of use cases, network/application deployments, and fundamental challenges to broadly distributed ledger technology adoption.

Maple and Jackson [12] present the anatomy of blockchain platforms and analyze their essential technological features. Furthermore, the authors introduce a format for outlining generic blockchain building blocks. The anatomy ranges from permissions to consensus and can be referenced when evaluating blockchain platforms. Moreover, they represent a comparison among multiple blockchain platforms.

Kuo et al. [13] conducted a systematic literature study to identify healthcare applications of blockchain technology, besides the blockchain platforms that have been proposed or implemented by the healthcare blockchain studies. In addition, the authors considered ten blockchain platforms and 21 blockchain features, then compared the blockchain platforms based on their features.

Yabo [14] described the features of multiple blockchain platforms and compared them against each other. Macdonald et al. [15] discussed how the blockchain is employed in Bitcoin cryptocurrency, besides some potential applications in other domains. Furthermore, the authors presented a comparison of five general-purpose blockchain platforms based on eight criteria related to usability, flexibility, and performance.

A variety of MCDM approaches have introduced by researchers recently. A subset of selected MCDM methods is presented as follows:

The *Weighted Sum Model (WSM)* is an aggregation function that transforms multiple criteria into a single value by multiplying each criterion by a weighting factor and summing up all weighted criteria. Frauenthaler et al. [16] introduce a WSM-based framework to monitor and evaluate several blockchain platforms according to user-defined settings.

The *Analytic Hierarchy Process (AHP)* is a structured and well-known method for organizing and analyzing MCDM problems based on mathematics and psychology. This MCDM approach considers a hierarchical structure of objectives, criteria, and alternatives to make complex decisions. Maček and Alagić [17] present an AHP-based approach to evaluate the security characteristics of the Bitcoin cryptocurrency system in comparison to other widely used online transaction systems.

The *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)* is an MCDM approach that employs information entropy to assess alternatives. The purpose of this approach is to first come up with an ideal solution and a negative ideal solution and then identify a scenario which is nearest to the ideal solution and farthest from the negative ideal solution. Tang et al. [18] present a TOPSIS-based evaluation model to rank public blockchain platforms according to three dimensions, including technology, recognition, and activity.

The *Boolean Decision Tree (BDT)* is an MCDM method to choose one of the available and feasible decision alternatives. Staderini et al. [19] propose a BDT-based requirements-driven methodology to support decision-makers to select suitable blockchain platform category (i.e., public or private, and permissionless or permissioned). Moreover, their proposed method assists the decision-makers with the configuration of selected blockchain platforms. Pahl et al. [20] introduce a BDT-based

framework to guide decision-makers on whether to use blockchain technology or not. Furthermore, they categorize blockchain platforms into three categories (public permissionless, public permissioned, and private) and compare them against each other based on a set of decision criteria. Wüst and Gervais [21] present a BDT approach to assist decision-makers on whether to select one of blockchain platforms or centralized databases. Additionally, they provide a comparison between permissionless, permissioned blockchain platforms, and centralized databases. Koens and Poll [22] suggest three questions to find out if blockchain is the best fitting technology (Should you use a blockchain? If so, which blockchain variant is best? If not, which alternative is best?) and if so which type of blockchain platforms should be employed. Moreover, they introduce a BDT-based scheme for determining which type of database is appropriate such as public permissionless blockchain, distributed database, and central database.

**TABLE I:** this table compares selected studies from the literature that address the blockchain platform selection problem. The first and second columns (Studies and Years) refer to the considered studies and their publication years. The third column (Decision-making technique) indicates the decision-making approach that the studies have employed to address the blockchain platform selection problem. The fourth column (MCDM) denotes whether the corresponding decision-making technique is an MCDM approach. The fifth column (Pairwise comparison) indicates whether the MCDM approach applied pairwise comparison as a weight calculation method or not. The sixth column (Quality Attributes) determines the type of quality attributes. The seventh and eighth columns (Criteria and Alternatives) signify the number of criteria and alternatives that were considered in the selected studies.

Studies	Years	Decision-making technique	MCDM	Pairwise comparison	Quality Attributes	Criteria	Alternatives
[10]	2017	Benchmarking	No	N/A	Domain specific	4	3
[12]	2018	Benchmarking	No	N/A	Not Defined	3	6
[13]	2019	Benchmarking	No	N/A	Not Defined	21	10
[14]	2016	Benchmarking	No	N/A	Not Defined	28	25
[15]	2017	Benchmarking	No	N/A	Domain specific	8	5
[16]	2019	WSM	Yes	No	Domain specific	8	4
[17]	2017	AHP	Yes	Yes	Domain specific	6	4
[18]	2019	TOPSIS	Yes	Yes	Domain specific	14	30
[19]	2018	BDT	Yes	No	Domain specific	8	4
[20]	2018	BDT	Yes	No	Domain specific	6	6
[21]	2018	BDT	Yes	No	Domain specific	6	4
[22]	2018	BDT	Yes	No	Domain specific	9	8
This study		DSS	Yes	No	ISO/IEC 25010 EX. ISO/IEC 9216	121	28

Table I summarizes the selected studies that discuss the blockchain platform selection problem. Performance testing methods [10] are time-consuming approaches and mainly applicable to a limited set of alternatives (blockchain platforms), as their implementation requires in-depth knowledge of blockchain platforms (such as APIs).

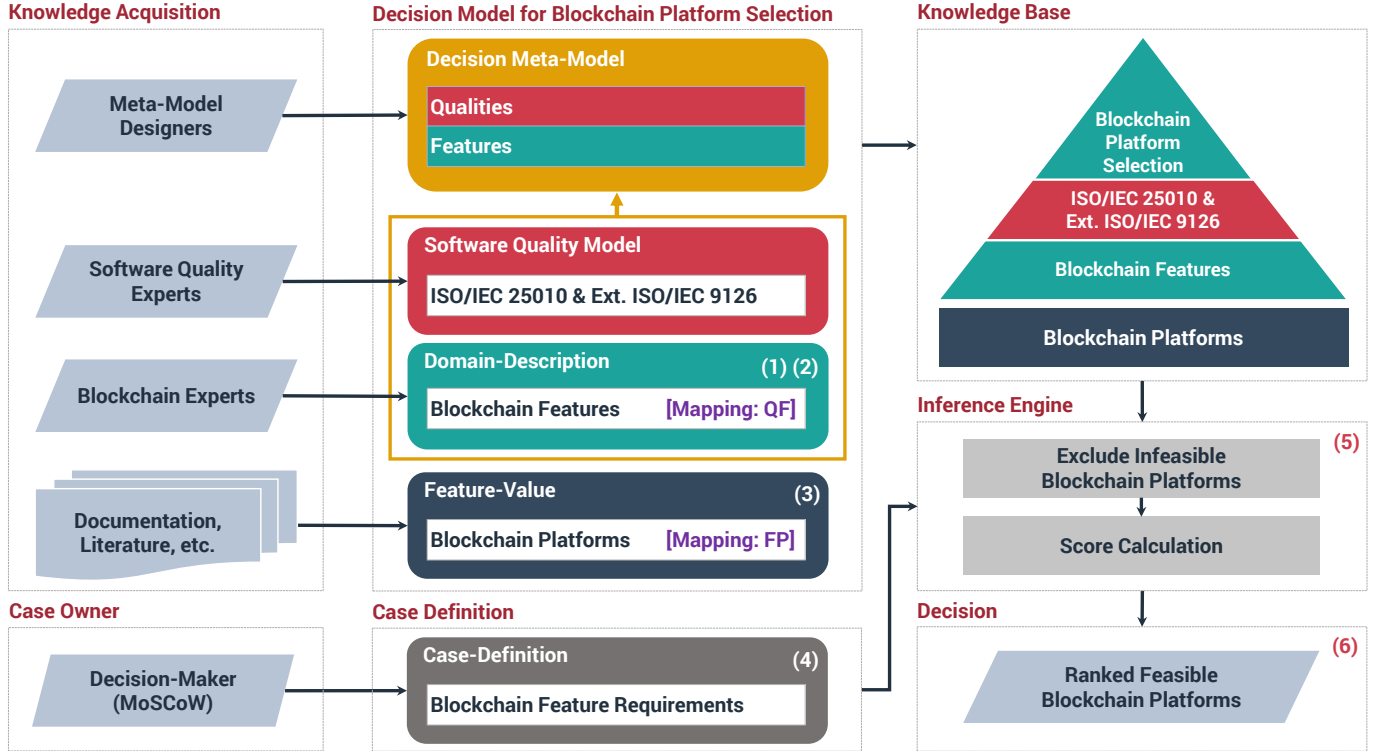
As blockchain is a relatively new and fast-evolving technology, so documentation is often out of date or not available; therefore, studies based on documentation and reports [12], [13], [14], [15] are likely to become outdated soon and should be kept up to date continuously.

The majority of the MCDM techniques in literature define domain-specific quality attributes to evaluate the alternatives. Such studies are mainly appropriate for specific case studies. Furthermore, the results of MCDM approaches are valid for a specified period; therefore, the results of such studies, by blockchain technology advances, will be outdated. Note that, in our proposal, this is also a challenge, and we propose a solution for keeping the knowledge base up to date, in section VII.

The number of criteria of BDT-based approaches [19], [20], [21], [22] is limited, i.e., under 10, as processing the large decision-trees is time-consuming and complicated. BDT-based approaches suggest only one solution at the end of each evaluation. Furthermore, decision-makers cannot prioritize decision criteria based on their preferences. Some of the MCDM techniques in the literature use pairwise comparison as the main method to assess the weight of criteria. For a problem with  $n$  number of criteria  $\frac{n(n-1)}{2}$  number of comparisons are needed [23]. It means that the pairwise comparison is a time-consuming process, and gets exponentially more complicated as the number of criteria increases. Some of the methods, such as AHP and TOPSIS, are not scalable, so in the case of modifying the list of alternatives or criteria, the whole process of evaluation should be redone. Therefore, these methods are costly and applicable to only a small number of criteria and alternatives. Please note that, in this study, we have considered 121 criteria and 28 alternatives to building a decision model for the blockchain platform selection problem.

In contrast to the mentioned MCDM approaches in the literature, the cost of creating, evaluating, and applying the proposed decision model is not penalized exponentially by the number of criteria and alternatives, because it is an evolvable and expandable approach that splits down the decision-making process into four maintainable phases [3]. Moreover, we introduce several parameters to measure the values of non-Boolean criteria, such as the cost and popularity of the blockchain platforms. The proposed decision model addresses main knowledge management issues, including capturing, sharing, and maintaining knowledge. Furthermore, it uses the ISO/IEC 25010 [24] as a standard set of quality attributes. This quality standard is a domain-independent software quality model and provides reference points by defining a top-down standard quality model for software systems.

Recently, we built two decision models based on the technology selection framework [2] to address the Database Management System [3] and Cloud-Service Provider [2] selection problems. In both studies, several case studies were conducted to evaluate



**Fig. 1:** This figure is adapted from our previous study [2] and shows the main building blocks of the decision support system beside the proposed decision model for the blockchain platform selection problem.

the effectiveness and usefulness of the DSS to address MCDM problems. The results showed that the DSS performed well to solve the Database Management System and Cloud-Service Provider selection problems for software-producing organizations. We believe that the technology selection framework can be considered as a reference framework to build decision models for MCDM problems in software-producing organizations.

#### IV. MULTI-CRITERIA DECISION-MAKING FOR BLOCKCHAIN PLATFORM SELECTION

We formulate the blockchain selection problem as an MCDM problem. Let  $Platforms = \{p_1, p_2, \dots, p_{|Platforms|}\}$  be a set of blockchain platforms in the market (i.e., Hyperledger and Ethereum). Moreover,  $Features = \{f_1, f_2, \dots, t_{|Features|}\}$  be a set of blockchain features (i.e., supporting JavaScript, Spam-attack resistant, and Sybil-resistant) of the blockchain platforms, and each  $p \in Platforms$  supports a subset of the set  $Features$ . The goal is finding the suitable blockchain platform  $p$ , which supports a set of required blockchain features (set  $Requirements$ ), where  $Requirements \subseteq Features$ . In other words, a blockchain platform  $p$  is the suitable one that supports blockchain feature requirements and satisfies the preferences of the decision-maker. Typically, a unique optimal solution for an MCDM problem does not exist, and it is necessary to employ decision-makers' preferences to differentiate between solutions [25]. The MCDM proposed in this article, therefore, provides a prioritized list of options for decision-makers.

##### A. Decision Model for Blockchain Platform Selection

In a previous study, we designed and implemented a DSS<sup>4</sup> [4] that comprises of standard DSS components [26], and introduced the technology selection framework [2] that applies the six-step decision-making process [25] to build maintainable and evolvable decision models for MCDM problems in software production. In this study, we follow the technology selection framework as modeled in Fig. 1 to build a decision model for the blockchain platform selection problem. Generally speaking, a decision model for an MCDM problem contains decision criteria, alternatives, and relationships among them. Fig. 1 illustrates the main building blocks of the decision support system besides the proposed decision model for the blockchain platform selection problem.

<sup>4</sup>We implemented an online Decision Model Studio (<https://dss-mcdm.com>) to build decision models for MCDM problems in software-producing organizations.

#### Decision Meta-Model:

As introduced in [4], the *Decision Meta-Model* is a simplified view of decision models and highlights the fundamental structure of decision models. Furthermore, it provides ontological descriptions of MCDM problems. The Decision Meta-Model has two sets, namely, *Qualities* and *Features*. Software quality attributes such as interoperability, maturity, and performance of blockchain platforms are kept in the set *Qualities*. Additionally, Blockchain features such as *Smart-contracts* and *on-chain transactions* are listed in the set *Features*.

#### Software Quality Model:

The *Software Quality Model* is a set of characteristics, and relationships between them, which provides a structure for specifying quality requirements and assessing them [4]. The Software Quality Model supports the specification of quality requirements, assess blockchain features, or predict the quality of a blockchain platform. The decision model for the blockchain platform selection problem employs the *ISO/IEC 25010* standard [24] and *extended ISO/IEC 9126* standard [27] in order to define the set *Qualities*. Such domain-independent quality models suggest standard hierarchical quality models for software systems. The elements of the *Software Quality Model* are used to analyze blockchain features based on their impact on quality attributes of blockchain platforms.

#### Domain-Description:

As aforementioned in [4], the *Domain-Description* determines the first and second steps, indicated by *Identifying the objective* and *Selection of the features*, of the decision-making process. As it is clear, the objective of the decision-making process in this study is *Blockchain Platform Selection*. Blockchain experts are the primary source of knowledge to identify the best fitting set of blockchain features, even though documentation and literature study of blockchain platforms can be employed to come up with an initial hypothesis about the blockchain feature set. Each blockchain feature has a data type, such as *Boolean* and *non-Boolean*. For example, the data types of blockchain features like the *popularity in the market* and supportability of *Smart-contracts* of a Blockchain Platform can be considered as *non-Boolean* and *Boolean*, respectively.

In this study, the initial set of blockchain features is extracted from online documentation of blockchain platforms. Then, a list of prominent blockchain features were identified during nine blockchain expert interviews (three researchers from Dutch research institutes, two blockchain developers, and four blockchain consultants/public-speakers). Finally, 71 Boolean and 4 non-Boolean blockchain features<sup>5</sup> identified and validated by the blockchain experts. Table II shows the identified blockchain features based on blockchain experts' opinion. Check marks (✓) denote the blockchain features suggested by the corresponding blockchain experts and cross marks (✗) signify that the blockchain experts did not express the blockchain features. Note, we excluded the blockchain features that were not considered at least by two blockchain experts.

The mapping ( $QF$ ) between the sets *Qualities* and *Features* is identified based on blockchain experts' knowledge. Four blockchain experts participated in this phase of the research to map the considered blockchain features to the set *Qualities* based on a Boolean adjacency matrix ( $Qualities \times Features \rightarrow Boolean$ ). For instance, *consensus-mechanisms* as a blockchain feature influences the *Fault-tolerance* quality aspect. The *Domain Description* does not enforce a blockchain feature to present in a single quality aspect; Blockchain features can be part of many quality aspects. For example, *Spam-attack resistant* as a blockchain feature might connect to multiple quality aspects such as *Recoverability* and *Availability*.

#### Feature-Value:

As we discussed in our previous study [4], the *Feature-Value* represents the third step, shown by *Selection of the alternatives*, of the decision-making process. Accordingly, a list of blockchain platforms (set *Platforms*) should be defined. Well-known blockchain platforms, websites, related forums, and blockchain experts are the primary source of knowledge to specify the list of blockchain platforms. In this study, 28 blockchain platforms (i.e., Hyperledger, Ethereum, and Chain) have been considered.

Blockchain features can be either Boolean or non-Boolean. A Boolean blockchain feature ( $Feature^B$ ) is a feature that its supportability by the blockchain platforms is investigated. Moreover, a non-Boolean blockchain feature ( $Feature^N$ ) assigns a non-Boolean value to a particular blockchain platform, for example, the maturity level of a blockchain platform. Therefore, the blockchain features in this study is a collection of Boolean and non-Boolean features, where  $Features = Feature^B \cup Feature^N$ .

The mapping  $BFP : Feature^B \times Platforms \rightarrow \{0, 1\}$  defines the supportability of the Boolean blockchain features by the blockchain platforms. So that  $BFP(f, p) = 0$  means that the blockchain platform  $p$  does not support the blockchain feature  $f$  and  $BFP(f, a) = 1$  signifies that the platform supports the feature.

The mapping  $BFP$  is defined based on documentation of the blockchain platforms and expert interviews. One of the principal challenges is the lack of standard terminology among blockchain platforms. Different blockchain platforms refer to the same concept by different names, or even worse, the same name might stand for different concepts in different blockchain platforms. Discovering conflicts in the *Feature-Value* is essential to prevent semantic mismatches throughout the blockchain platform

<sup>5</sup>The entire list of the blockchain features and supportability of considered blockchain platforms are available and accessible on the "Blockchain Platform Selection" website (<https://dss-mcdm.com>)

**TABLE II:** this table shows the considered blockchain features by the interviewees. Checkmarks (✓) denote the blockchain features suggested by the corresponding blockchain experts, and cross marks (✗) signify that the blockchain experts did not express a need for the blockchain features in the decision model. The *Agreement* columns denote the agreements among the blockchain experts regarding considering blockchain features. Note, this is not the final list of blockchain features. The full list of the blockchain features besides their definitions are available in the appendix section of this study. (<https://dss-mcdm.com>)

Blockchain features	Agreement	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7	Interview 8	Interview 9	Blockchain features	Agreement	Interview 1	Interview 2	Interview 3	Interview 4	Interview 5	Interview 6	Interview 7	Interview 8	Interview 9
Proof-of-Work	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Solidity	88.89%	✓	✓	✓	✓	✗	✓	✓	✓	✓
Proof-of-Stake	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	C#	22.22%	✗	✓	✓	✗	✗	✗	✗	✗	✗
delegated Proof-of-Stake	55.56%	✓	✓	✓	✓	✗	✗	✗	✗	✗	Golang	55.56%	✗	✓	✗	✗	✗	✗	✗	✗	✗
practical Byzantine Fault Tolerance	77.78%	✓	✓	✓	✗	✗	✗	✓	✓	✓	JavaScript	55.56%	✗	✓	✗	✗	✗	✓	✓	✓	✓
federated Byzantine Agreement	77.78%	✓	✓	✓	✗	✗	✗	✓	✓	✓	Java	77.78%	✗	✓	✓	✓	✗	✓	✓	✓	✓
delegated Byzantine Fault Tolerance	77.78%	✓	✓	✓	✗	✗	✗	✓	✓	✓	C++	55.56%	✗	✓	✓	✓	✗	✗	✗	✗	✗
Proof-of-Authority	55.56%	✗	✗	✗	✓	✓	✓	✓	✓	✗	Python	55.56%	✗	✓	✓	✗	✗	✓	✓	✓	✗
Proof-of-Elapsed Time	55.56%	✗	✓	✓	✗	✗	✗	✓	✓	✗	Haskell	22.22%	✗	✗	✗	✗	✗	✗	✓	✗	✗
Proof-of-Burn	11.11%	✗	✗	✗	✗	✗	✗	✗	✗	✗	Tokenization	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓
Proof-of-Luck	11.11%	✗	✗	✓	✗	✗	✗	✗	✗	✗	Native	55.56%	✗	✗	✗	✗	✗	✗	✓	✓	✓
Directed Acyclic Graph (variants)	55.56%	✓	✗	✗	✗	✗	✓	✓	✓	✓	Non-native	55.56%	✗	✓	✗	✓	✗	✗	✓	✓	✓
SHA-256	44.44%	✗	✗	✓	✗	✗	✗	✗	✓	✓	Cryptocurrency	77.78%	✓	✓	✓	✓	✗	✓	✓	✗	✓
SHA-3	44.44%	✗	✗	✓	✗	✗	✗	✗	✓	✓	Utility	66.67%	✓	✓	✗	✗	✗	✗	✓	✓	✓
md-5	22.22%	✗	✗	✓	✗	✗	✗	✗	✓	✗	Security	55.56%	✓	✓	✓	✗	✗	✗	✗	✗	✓
SHA-512	33.33%	✗	✗	✗	✗	✗	✗	✗	✗	✓	Usage	55.56%	✗	✓	✓	✓	✗	✗	✗	✗	✓
ASIC-algorithm	33.33%	✗	✗	✗	✗	✗	✗	✗	✗	✓	Asset	66.67%	✓	✓	✗	✗	✗	✓	✓	✓	✓
Cryptonight	22.22%	✗	✗	✗	✗	✗	✗	✗	✗	✗	Work	33.33%	✗	✓	✗	✗	✗	✗	✓	✗	✓
Protocol Layer	77.78%	✓	✗	✗	✓	✓	✓	✓	✓	✓	Hybrid	33.33%	✗	✗	✗	✗	✗	✗	✓	✗	✓
Network Layer	77.78%	✓	✗	✗	✓	✓	✓	✓	✓	✓	On-chain transactions	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓
Application Layer	88.89%	✓	✗	✓	✓	✓	✓	✓	✓	✓	Off-chain transactions	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓
Public	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Side-chains	88.89%	✓	✗	✓	✓	✓	✓	✓	✓	✓
Private	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Sharding	88.89%	✓	✗	✓	✓	✓	✓	✓	✓	✓
Permissioned	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Plasma-chains	66.67%	✗	✗	✗	✓	✓	✓	✓	✓	✓
Permissionless	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Atomic-swaps	66.67%	✗	✗	✓	✓	✗	✓	✓	✓	✓
Smart-Contracts	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Cross-chain interoperable	66.67%	✓	✗	✗	✓	✓	✓	✗	✓	✓
Virtual Machine	66.67%	✗	✓	✓	✗	✗	✓	✓	✓	✓	Enterprise system integration	77.78%	✓	✓	✓	✗	✓	✓	✗	✓	✓
Docker	11.11%	✗	✓	✗	✗	✗	✗	✗	✗	✗	Zero-knowledge Proof	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓
Turing-Complete	66.67%	✗	✓	✓	✗	✗	✓	✓	✓	✓	Ring-signatures	55.56%	✗	✗	✓	✓	✗	✗	✓	✓	✓
Transaction Speed	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Hard-fork resistant	55.56%	✓	✓	✓	✗	✗	✗	✗	✗	✗
Block-Size	44.44%	✓	✓	✗	✗	✓	✓	✗	✗	✗	Spam-attack resistant	77.78%	✓	✓	✓	✗	✗	✓	✓	✓	✓
Maturity	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Sybil attack resistant	88.89%	✓	✓	✓	✓	✓	✓	✓	✓	✓
Platform Transaction Speed	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Quantum resistant	55.56%	✗	✗	✓	✓	✗	✗	✗	✗	✗
Popularity in the market	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Transaction Irreversibility	55.56%	✓	✗	✓	✓	✓	✗	✗	✗	✓
Innovation	100.00%	✓	✓	✓	✓	✓	✓	✓	✓	✓	Wallet	22.22%	✗	✗	✓	✓	✗	✗	✗	✗	✗

selection process. Table III shows a sample of the *BFP* mapping between the **Boolean blockchain Features** and **Platforms** in the knowledge base of the DSS.

In this study, the non-Boolean blockchain features are *Blockchain Platform Maturity*, *Popularity in the Market*, *Transaction Speed*, and *Innovation*. The assigned values to these non-Boolean blockchain features for a specific blockchain platform is a 3-point Likert scale, where  $NFP : Features^N \times Platforms \rightarrow \{High, Medium, Low\}$ , based on several predefined parameters.

Table IV illustrates the non-Boolean blockchain features besides their parameters. The blockchain experts assigned the 3-point Likert scale values to these non-Boolean blockchain features according to the corresponding values of the parameters.

## B. Knowledge Base

Each decision model defines a decision structure for an MCDM problem systematically based on the technology selection framework [2]. The *Knowledge Base* is a collection of decision models, which are groups of rules and facts. The blockchain decision model has been uploaded to the knowledge base of the DSS to facilitate the decision-making process for software-producing organizations for any blockchain platform selection.

## C. Case-Definition

As discussed in [4], the *Case-Definition* defines the fourth step, denoted by *Selection of the weighing method*, of the decision-making process. Decision-makers prioritize the blockchain feature requirements using the *MoSCoW* technique.

Suppose  $W_{MoSCoW} = \{w_{Must}, w_{Should}, w_{Could}, w_{Won't}\}$  is the set of priority weights according to the definition of the *MoSCoW* [28]. Blockchain feature requirements with *Must Have* or *Won't Have* priorities act as hard constraints and blockchain feature requirements with *Should Have* and *Could Have* priorities act as soft constraints. In other words, a case definition, based on the decision-maker preferences, is a way to define blockchain feature requirements and assign priorities to them. Note, we could have used other prioritization techniques, but we purposely wanted to keep it simple.

**TABLE III:** this table lists a sample of the **BFP** mapping between the **B**oolean **b**lockchain **F**eatures and **P**latforms. The first row and column of the table designate the blockchain features and platforms, respectively. Furthermore, 1s on each row indicates that the corresponding platforms support the blockchain feature of that row. Conversely, 0s mean that the corresponding platforms do not support that blockchain feature. Note, the *Coverage* column denotes the percentage of blockchain platforms that support each feature. The complete list of the blockchain features and platforms are available in the appendix. We plan to keep the list up-to-date, so the list is available on the following website as well: <https://dss-mcdm.com>

BFP Boolean blockchain Features and Platforms	Coverage	Ethereum	R3 Corda	JPMorgan Quorum	Hyperledger	BigChainDB	Multichain	HydraChain	Chain	Symbiont	Stratis (Azure Baas)	OpenChain	NEO	Cardano	Stellar	Ripple	Bitshares	QTUM	ICON	VeChain	IOTA	Factom	Cosmos Network	Lisk	Waves Platform	Wanchain	Neblio	Zilliqa	Komodo	
		<b>Consensus Mechanism</b>																												
Proof-of-Work	25.00%	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1
Proof-of-Stake	28.57%	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0
Delegated Proof-of-Stake	14.29%	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	
Practical Byzantine Fault Tolerance	28.57%	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0
Delegated Byzantine Fault Tolerance	14.29%	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Proof-of-Authority	14.29%	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
Federated Byzantine Agreement	17.86%	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Proof-of-Elapsed Time	7.14%	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SIEVE	3.57%	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cross-Fault Tolerance	3.57%	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Directed Acyclic graph	3.57%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
<b>Layers</b>																														
Protocol Layer	96.43%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Network Layer	100.00%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Application Layer	82.14%	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>Authorization and Authentication</b>																														
Public	64.29%	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Private	35.71%	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Permissioned	53.57%	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0
Permissionless	57.14%	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
<b>Contracts</b>																														
Smart-contracts	71.43%	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	1	1	1
Virtual Machine	39.29%	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0
Turing Complete	35.71%	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0
<b>Programming Language Support</b>																														
Solidity	28.57%	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0
Python	46.43%	1	0	0	1	1	1	1	0	1	0	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	1
Golang	32.14%	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1

Decision-makers specify desirable values for numeric blockchain feature requirements. For example, a decision-maker could be interested in prioritizing blockchain platforms with the *Blockchain Platform Maturity* above average. Therefore, the *Blockchain Platform Maturity* above average is considered as a *should have* blockchain feature.

Fig. 2 shows a decision structure from a case definition. The DSS generated the decision structure and inferred the solutions based on the proposed decision model for the blockchain platform selection problem. At the top of the figure (Domain), the domain of the decision structure is shown (*ShareCompany BIQH Blockchain Platform (BP) Selection*). The next level of decision structure (Qualities) illustrates the characteristics and sub-characteristics of the ISO/IEC 25010 [24] plus ISO/IEC 9126 [27] standards, respectively. The relationship among the quality attributes is based on the definitions of these two quality models. The third part of the decision structure (Features) presents the blockchain feature requirements based on decision-makers' priorities and preferences. As we mentioned earlier, the decision model utilizes the *MoSCoW* prioritization technique to assign blockchain features weights. In this figure, the colors identify the blockchain features priorities. Finally, the lowest segment of the decision structure (Platforms) shows the final results for the specific case definition according to the decision-makers' blockchain features requirements and priorities. The mapping QF, which is based on blockchain experts knowledge (tacit knowledge), demonstrates the relationship between sub-characteristics of the ISO/IEC standards and blockchain Features. Moreover, the mapping *FP* partially illustrates the mapping between the blockchain features and platforms. The mapping between the sets of Boolean blockchain features and platforms is denoted by *BFP*, and the mapping between the sets of non-Boolean blockchain features and platforms is signified by *NFP*. The mappings *NFP* and *BFP* are two subsets of the mapping *FP*, where  $FP = NFP \cup BFP$ .

#### D. Inference Engine

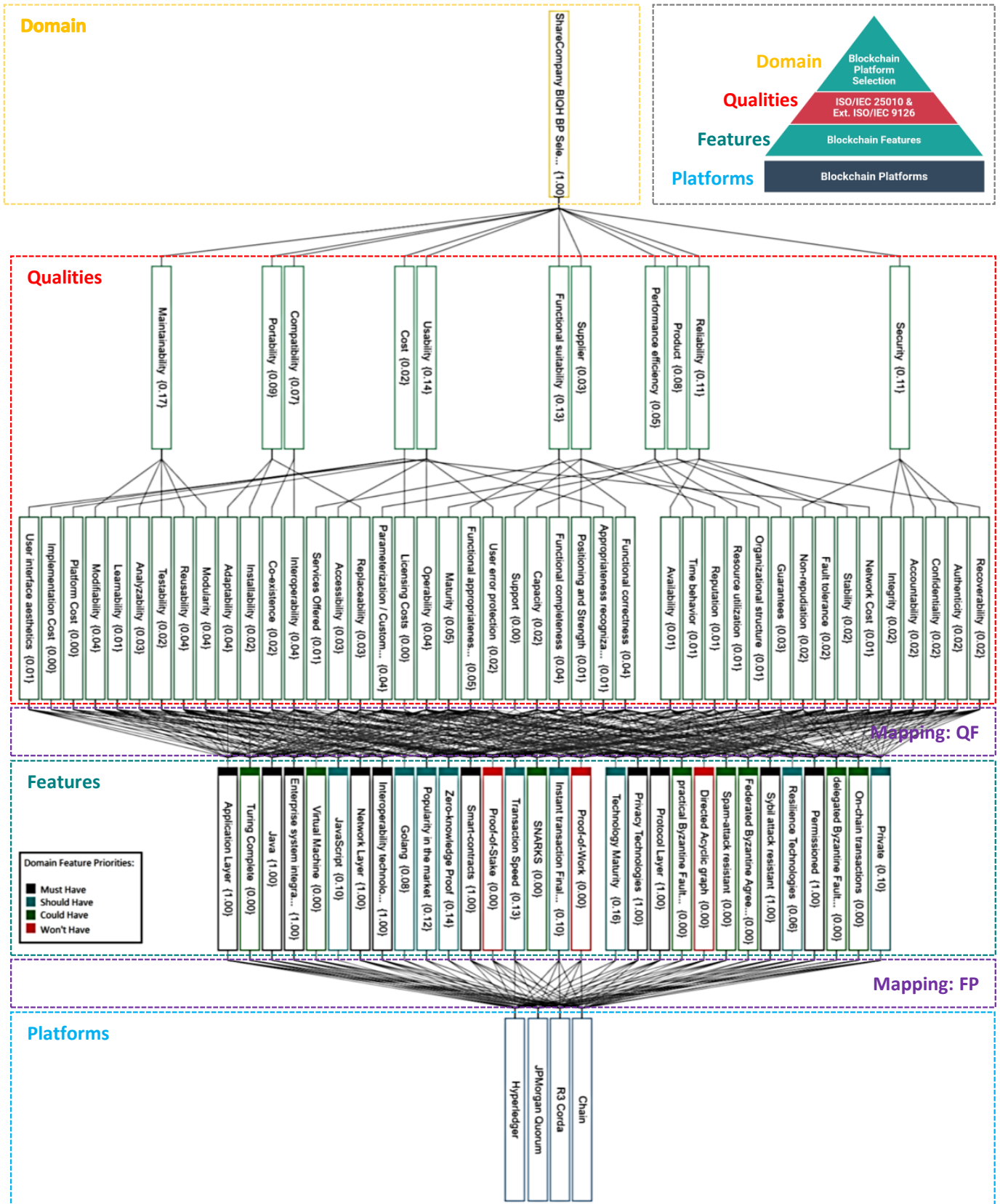
The *Inference Engine* comprises two steps: the fifth step, i.e., *Applying the method of aggregation* and the sixth step, i.e., *Decision making based on the aggregation results*, of the decision-making process [4]. The *Inference Engine* makes logical deductions about knowledge assets, intending to find the best fitting blockchain platforms.

A feasible blockchain platform must support all blockchain feature requirements with *Must-Have* priorities, and must not support all blockchain feature requirements with *Won't-Have* priorities.



**TABLE IV:** this table shows the **NFP** mapping between the Non-Boolean blockchain **F**eatures and **P**latforms. Note, the *Platform Transaction Speed*, *Popularity in the market*, *Innovation*, and *Blockchain Platform Maturity* are the Non-Boolean blockchain features that were considered in this study. The parameters of these features are listed below each features, for example, *Founded*, *Revenue*, *Size*, and *Consensus-mechanism* are the parameters of the *Blockchain Platform Maturity*.

<b>NFP</b> Non-Boolean blockchain <b>F</b> eatures and <b>P</b> latforms		Ethereum	R3 Corda	JPMorgan Quorum	Hyperledger	BigChainDB	MultiChain	HydraChain	Chain	Symbiont	Stratis (Azure Baas)	OpenChain	NEO	Cardano	Stellar	Ripple	Bitshares	QTUM	ICON	VeChain	IOTA	Factom	Cosmos Network	Lisk	Waves Platform	Wanchain	Neblio	Zilliqa	Komodo	Source of Knowledge	
Transaction Speed		Low	High	High	High	High	Medium	Medium	High	High	High	High	Medium	Medium	Medium	High	High	Low	High	Medium	Low	Low	High	Medium	Medium	Low	Low	High	Low	Domain Experts	
Confirmation Time (sec)		15	1	1	1	1	1	1	1	1	1	1	15	10	4	3.5	3	120	2	5	120	300	1	28	3	15	120	120	The platform website		
Speed of consensus mechanism (transactions/second)		15	170	140	3500	1000000	1000	10	15000	80000	20000	2500	1000	257	1000	1500	3300	70	3000	50	500	45	3000	2	1000	15	27	2000	80	The platform website	
# Scalability technologies implemented		2	15	2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	3	2	2	3	2	2	2	2	The platform website	
Popularity in the market		High	High	Medium	High	Low	Medium	Low	Low	Low	Medium	Low	Medium	Medium	Medium	High	Medium	Medium	Medium	Low	Medium	Low	Low	Medium	Medium	Low	Low	Low	Low	Domain Experts	
Market Capitalization (x 1.000.000 dollar)		58676	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	3458	5327	5214	23562	508	1200	1049	1895	4049	N/A	N/A	932	455	470	108	870	275	<a href="https://coinmarketcap.com">https://coinmarketcap.com</a>	
Google monthly searches		2240000	405000	201000	405000	3600	4400	300	1000	9900	5000	1900	368000	201000	201000	1830000	60500	14800	40500	9900	673000	27100	480	90500	5400	4400	2900	1600	6600	<a href="https://keywordeverywhere.com">https://keywordeverywhere.com</a>	
Twitter Followers		396491	20954	432	45454	7560	3581	0	2627	2359	138774	336	318939	135938	236565	87583	85177	162920	101272	92828	112199	76913	10912	189261	125094	10208	74635	39925	88178	<a href="https://twitter.com">https://twitter.com</a>	
Reddit subscribers		362000	N/A	N/A	N/A	N/A	53	N/A	N/A	N/A	10800	N/A	92900	64500	79200	187000	70000	14300	22600	47800	110000	8400	2200	29800	12600	17100	5900	7400	6600	<a href="https://www.reddit.com">https://www.reddit.com</a>	
Transactions+Operations per day		833377	1224000	10000	24000000	N/A	N/A	N/A	N/A	N/A	1200000	N/A	76334	2420	150000	955000	155000	1000	1000	N/A	146880	40000	N/A	3470	22000	250	800	800	10000	<a href="https://bitinfocharts.com">https://bitinfocharts.com</a>	
Linkedin followers		60931	10308	1026784	15809	1220	16	4	2835	1446	783	46	2761	3001	2141	2043	481	170	127	810	2961	1962	1724	1865	882	756	0	641	291	<a href="https://www.linkedin.com">https://www.linkedin.com</a>	
Innovation		High	High	Medium	High	Medium	Medium	Low	Medium	Low	High	Low	Medium	Medium	Medium	Medium	Low	Low	Medium	Medium	Medium	Low	Medium	Low	Low	Medium	Low	Low	Low	Domain Experts	
Internet of Things		1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	The platform website
Artificial Intelligence		1	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	The platform website
Plasma Technology		1	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	The platform website
Sharding		1	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	The platform website
Supply-Chain Management		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	The platform website
Financial Sector		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	The platform website
Zero-knowledge proofs technology		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	The platform website
Consortium-Research Support		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	The platform website
Cross-chain interoperability		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	The platform website
Blockchain Platform Maturity		High	High	High	High	Medium	Low	Medium	High	High	High	Low	Low	Low	Medium	High	Low	Low	Low	High	Low	Medium	Low	Low	Medium	Low	Low	Low	Low	Domain Experts	
Founded		2013	2014	2017	2014	2014	2014	2015	2014	2015	2017	2015	2017	2017	2014	2012	2014	2017	2015	2015	2015	2015	2016	2016	2016	2017	2017	2017	2014	The platform website	
Revenue		Between 1 and 5 million	Between 1 and 5 million	More than 5 million	More than 5 million	More than 5 million	Less than 1 million	Less than 1 million	Between 1 and 5 million	Between 1 and 5 million	More than 5 million	Less than 1 million	Less than 1 million	Less than 1 million	Less than 1 million	More than 5 million	Less than 1 million	Less than 1 million	Less than 1 million	More than 5 million	Between 1 and 5 million	Between 1 and 5 million	Between 1 and 5 million	Between 1 and 5 million	Between 1 and 5 million	Less than 1 million	Less than 1 million	Less than 1 million	Less than 1 million	Less than 1 million	<a href="https://www.owler.com">https://www.owler.com</a>
Size (employees+github devs)		35000	2000	157	7351	214	573	10	25	504	337	66	31	13	250	274	44	33	33	129	55	44	31	46	35	47	9	25	16	<a href="https://www.owler.com">https://www.owler.com</a> <a href="https://github.com">https://github.com</a>	
Consensus-mechanism		Proof-of-Work	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Proof-of-Work	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Proof-of-Stake	Proof-of-Authority	Byzantine Fault Tolerance	Proof-of-Stake	Byzantine Fault Tolerance	Byzantine Fault Tolerance	Delegated Proof-of-Stake	Proof-of-Stake	Byzantine Fault Tolerance	Proof-of-Authority	Directed Acyclic Graph	Proof-of-Work	Byzantine Fault Tolerance	Delegated Proof-of-Stake	Proof-of-Stake	Proof-of-Stake	Proof-of-Stake	Byzantine Fault Tolerance	Proof-of-Work	The platform website	



**Fig. 2:** this figure illustrates a decision structure based on a case definition (ShareCompany BIQH Blockchain Platform Selection). The DSS has automatically generated the decision structure. The first level of the decision structure (Domain) indicates the goal of the decision structure. The second level (Qualities) denotes the relevant quality attributes that have impacts on the prioritized blockchain platform feature requirements, which are signified in the third level (Features). The last level (Platforms) shows the list of feasible blockchain platforms. Note, the mapping FP and QF define the relationship between the qualities, features, and platforms.

The *Inference Engine* excludes infeasible blockchain platforms, calculates the scores of the feasible blockchain platforms, and finally suggests a sorted shortlist of them. The score calculation process is based on the *Weighted Sum Model*, as described in our previous work [2]. The scores of feasible blockchain platforms are non-negative integers, so by sorting the feasible blockchain platforms in descending order of their scores, the final ranked list of feasible blockchain platforms will be provided as the result of the DSS.

#### E. Group Decision-Making

The DSS provides a discussion and negotiation platform to enable decision-makers to make group decisions. The DSS asks decision-makers to define individual blockchain feature requirements based on the *MoSCoW*. Next, it collects the individual prioritized blockchain feature requirements of decision-makers and considers the maximum the *MoSCoW* priority for each blockchain feature requirement [4]. It detects and highlights the conflicts in the assigned priorities to the blockchain feature requirements by decision-makers, and asks them to resolve disagreements.

### V. EMPIRICAL EVIDENCE: THE CASE STUDIES

Three industry case studies at three software development companies are conducted to evaluate and signify the usefulness and effectiveness of the decision model. The case-study participants have identified a number of potentially feasible blockchain platforms for their organizations through multiple internal expert meetings and investigation into blockchain platforms before participating in this research. Moreover, the case-study participants have employed the DSS to analyze, document, track, and prioritize their blockchain feature requirements. The remaining sections describe the case studies and discuss the outcome of the DSS.

#### A. Case Study 1: ShareCompany BIQH

ShareCompany BIQH, a FinTech company in the Netherlands, supports two well-known Dutch banks with accommodating the requirements put forth by the European Union regarding packaged retail investment and insurance-based products (PRIIP/KID regulation). ShareCompany BIQH is now interested in investigating the possibility of deploying its current centralized financial system on a blockchain platform. Some of the envisioned system requirements and corresponding blockchain feature requirements that were asserted by the case-study participants are listed as follows.

- The envisioned system requires extensive integration with the current system (e.g., APIs), so *Enterprise System Integration* is considered as a *Must-Have* feature. Since only a limited number of end-users are authorized to make changes in the system, *Permissioned* is a *Must-Have* feature.
- The current system and its data are already publicly accessible, so a *private platform* is not a necessary blockchain feature and considered as a *Should-Have* feature. The *Protocol Layer*, *Network Layer*, and the *Application Layer* are all prioritized as *Must-Have* features. The protocol layer supports reaching consensus on the accuracy of the data; the network layer defines the communication among system end-users, and the application layer employs to build the required infrastructure to connect with current enterprise systems.
- The clients of the system should remain anonymous. So, the envisioned system should follow the General Data Protection Regulation and force its employed technologies to support data privacy regulation. Therefore, supporting *Smart Contracts* in the Java programming language have been prioritized as a *Must-Have* feature.
- The selected blockchain platform must be *Sybil-attack resistant* to prevent such attacks in peer-to-peer networks.
- ShareCompany BIQH is operating in a financial data environment with major internationally-operating banks, so the selected blockchain platform *Should-Have* both a *High Maturity* and a *High Popularity* in the market to reduce potential risks.
- The envisioned system should provide anonymous and secure transactions. Thus, *Zero-knowledge Proof* feature is prioritized as a *Should-Have* feature.
- The speed of transactions in the envisioned system is not a crucial issue; therefore, *High Transaction speed* can be considered as a *Should-Have* blockchain feature.
- ShareCompany BIQH has professional Golang and JavaScript developers, however, coding in other programming languages is not a major obstacle for them. Thus, supporting *Golang* and *JavaScript* programming language are two *Should-Have* blockchain features.
- ShareCompany BIQH does not decide on a consensus mechanism. As they are not working with cryptocurrency and not interested in the consensus mechanisms designed for a cryptocurrency, they prioritized *Proof-of-Work* or *Proof-of-Stake* as two *Won't-Have* blockchain features. Three remaining considered consensus-mechanisms in the decision model are *Practical Byzantine Fault Tolerance*, *Federated Byzantine Fault Tolerance*, and *Delegated Byzantine Fault Tolerance* prioritized as *Could-Have* blockchain features.
- The case-study participants mentioned that *Directed-Acyclic-Graph* is too experimental and immature yet, so they considered it as another *Won't-Have* blockchain feature.

Finally, the case-study participants themselves concluded that *Hyperledger* and *JPMorgan Quorum* are two potential blockchain platforms that meet all their requirements.

### B. Case Study 2: DUO

DUO is the administrative and executive agency of the Dutch government for managing the educational system. 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. This case study will merely focus on the process of student financing in the form of granting loans. DUO is interested in building a decentralized application based on blockchain technology to address the requirements of the student financing activities. Some of the envisioned system requirements and corresponding blockchain feature requirements that were asserted by the case-study participants are listed as follows.

- The DUO financing system requires the three layers of a blockchain platform (including the protocol, network, and application layers). Therefore, these three layers are considered as three *Must-Have* blockchain features.
- The system has no strict requirement for a specific consensus-mechanism. Therefore, all considered consensus-mechanisms in the decision model are prioritized as *Could-Have* blockchain features.
- Supporting *Smart Contracts* is a *Must-Have* blockchain feature, as it mainly influences the functionality of the system. For example, the *Smart Contracts* handle paying out the loans each month if a specific date has passed, grant conventional loans to students if they meet the specified conditions, or deny additional loans to students who try deceiving the system.
- The payout of credits can be either done by the system in fiat currency (Euro) or in the form of *Cryptocurrency*, which acts as Native token to the system. Thus, the *Cryptocurrency* is a *Should-Have* blockchain feature.
- The DUO financial system executes transactions as *On-chain transactions*, moreover, it utilizes *Cryptographic Tokens*. Therefore, both of them are prioritized as *Must-Have* blockchain features.
- The DUO financial system has to be deployed on a public, private, permission, or permissionless blockchain platform. The Dutch government prefers not to utilize public blockchain platforms. However, selecting a public blockchain platform that follows privacy regulations dramatically increases transparency and possibly credibility, as indicated by Cyber Capital. Therefore, *Permission* and *Permissionless* are considered as two *Could-Have* blockchain features.
- Currently, Solidity is the most common programming language to create smart contracts and specifically designed for it. Thus, *Solidity* prioritized as a *Must-Have* blockchain feature.
- *Spam-attack resistant* and *Sybil attack resistant* are *Must-Have* blockchain features from the case-study participants to guarantee a base level of security and resilience.
- Supporting *JavaScript*, being *Turing-complete* are two *Should-Have* blockchain features.
- Selecting a blockchain platform with *High Maturity* decreases potential unnecessary risks as much as possible, so it as a *Should-Have* blockchain feature in this case study.
- The case-study participants asserted that they would not utilize a *Directed Acyclic Graph* for now since it is still too immature; therefore, they prioritized it as a *Won't-Have* blockchain feature.

Finally, the case-study participants selected three blockchain platforms as their main potential solutions, namely *Ethereum*, *NEO*, and *Hyperledger*.

### C. Case Study 3: Veris Foundation

The Veris Foundation is an organization focusing on the American healthcare system. The Veris Foundation addresses the problem of bringing healthcare service providers, insurers, and banks together to authorize the provisioning and payment for healthcare services. The Foundation is a nonprofit entity whose core objective is the establishment of a platform to reduce the cost of healthcare and make it more affordable to patients. Traditional, centralized healthcare systems are slow, redundant, and expensive, because, service providers and payers employ their staff and separate software stacks to facilitate their medical claims processes. These isolated systems make the sharing of necessary information complicated, costly, and prone to errors and fraud. The Veris Foundation is interested in finding the best fitting blockchain platform, as they believe that creating decentralized databases enables all parties to securely access and share data within and across organizations, eliminating the need to hire and maintain expensive third-party information systems. Some of the requirements and corresponding blockchain feature requirements that were stated by the case-study participants are listed as follows.

- case-study participants prioritized supportability of *smart contracts* as a *Must-Have* blockchain feature, because smart contracts define the rules and penalties related to agreements among parties and automatically enforce those obligations.
- the Veris platform is currently a forked version of the NEO blockchain platform, so the *Protocol Layer*, *Network Layer*, and the *Application Layer* are all prioritized as *Must-Have* features. Note, the NEO blockchain platform already supports most of the Veris blockchain feature requirements.
- case-study participants stated that the *delegated Byzantine Fault Tolerance* and *Proof-of-Stake* are two consensus mechanisms that can be employed interchangeably, so that they are two *Should-Have* blockchain features.
- The Veris platform provides different Graphical User Interfaces for its stakeholders; moreover, their authority is required to provide the validation of blocks of transactions. Therefore, case-study participants prioritized *Permissioned* as a *Must-Have* blockchain feature.
- The platform interacts with other parties, such as banks, so it requires a specific type of interoperability, and in particular *Enterprise system integration*. Thus, the case-study participants have considered it as a *Must-Have* blockchain feature.

**TABLE V:** this table presents the entire list of blockchain feature requirements of the considered three industry case studies. The blockchain feature requirements are defined based on the MoSCoW prioritization technique.

MoSCoW	ShareCompany BIQH	DUO	Veris Foundation
Must-Have	Permissioned Platform Interoperability technologies Smart Contract Java Sybil-attack resistant Privacy Technologies Enterprise System Integration Network Layer Application Layer Protocol Layer	Protocol Layer Network Layer Application Layer Smart contracts On-chain transactions Cryptographic Tokens Sybil attack resistant Spam-attack resistant	Permissioned Blockchain Smart Contracts Cryptographic Token Protocol Layer Network Layer Application Layer Interoperability technologies Enterprise system integration On-chain transactions
Should-Have	Golang Private Platform JavaScript Resilience technologies Instant Transaction Finality High Transaction Speed Zero-knowledge Proof High Maturity High Popularity	Turing-complete JavaScript High Maturity Native token Cryptocurrency (purpose) Solidity	Private Blockchain delegated Byzantine Fault Tolerance Delegated Proof-of-Stake Share-like token Security token Network token Network value token Work token Usage token
Could-Have	zk-SNARKS Spam-attack resistant Virtual Machine Turing-complete On-chain transactions Practical Byzantine Fault Tolerance Federated Byzantine Fault Tolerance Delegated Byzantine Fault Tolerance	Proof-of-Work 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 Public Platform Private Platform Permissioned Platform Permissionless Platform Virtual Machine Java C++ Zero-knowledge Proof zk-SNARKS Hard-fork resistant Quantum resistant Instant transaction finality Medium Popularity Medium Innovation High Transaction speed	Privacy Technologies, Virtual Machine, Turing Complete
Won't-Have	Proof-of-Work Proof-of-Stake Directed Acyclic Graph	Directed Acyclic Graph	(None)

- The dual-currency structure of the Veris platform gives rise to discuss the *Cryptographic tokens* as a *Must-Have* blockchain feature.
- The Veris platform has not decided on a special type of token. Therefore, *Share-like token*, *Security token*, *Network token*, *Network value token*, *Work token*, and *Usage token* are considered as *Should-Have* blockchain features.

In this study, the case-study participants selected *Ethereum* and *NEO* as two potential blockchain platforms for their system.

## VI. RESULTS AND ANALYSIS

The case-study participants specified their blockchain features requirements according to the MoSCoW priorities (table V), so three industry cases are defined and stored in the knowledge base of the DSS. Next, the Inference Engine of the DSS generated feasible solutions for each case definition.

### A. The DSS Results

Table VI shows the deduced feasible blockchain platforms along with their calculated scores by the DSS. Moreover, it compares the case-study participants' shortlists and their ranks, which are the results of internal meetings and investigations, with the outcomes of the DSS.

1) *ShareCompany BIQH*: The case-study participants at ShareCompany BIQH considered *Hyperledger* and *JPMorgan Quorum* as the first and second potential solutions in their shortlist. *Hyperledger* supports all the *Must-have* and most of the *Should-have* and *Could-have* blockchain feature requirements (such as JavaScript programming language, Zero-knowledge

Proofs, and Golang), therefore, the DSS assigned the highest score to this blockchain platform. However, the second DSS feasible blockchain platform is *R3 Corda*, which has higher values for the non-Boolean blockchain feature requirements, such as *Popularity in the market* and *technology maturity*, compared to *JPMorgan Quorum*.

2) *DUO*: The case-study participants at DUO ranked *Ethereum*, *NEO*, *Hyperledger* as their three potential blockchain platform solutions. *Ethereum* has gained the highest score among the top-10 DSS feasible blockchain platforms for DUO according to their blockchain feature requirements. *Wanchain* was not on the case-study participant shortlist, but since it is an *Ethereum-based* blockchain platform, its high score is not surprising. Despite *Hyperledger* reaching the second place in the DSS feasible blockchain platforms, it forces DUO to make intensive use of *cryptographic tokens*, so it is not as suitable as the previous two platforms. Moreover, *Hyperledger* does not support *native-tokens*, so it is not a suitable blockchain platform for token-based applications. Also, several blockchain feature requirements with *Should-have* priority are *token-based*. The case-study participants at DUO considered 23 blockchain feature requirements with *Could-have* priority, and *Hyperledger* supports all of them, so *Hyperledger* received the second-highest score among the DSS feasible blockchain platforms. *NEO* does not support all of the blockchain feature requirements with *Should-have* and *Could-have* priorities. However, the gap between the calculated scores of *NEO* and *Hyperledger* is not too much.

3) *Veris Foundation*: The case-study participants at the Veris Foundation considered *NEO* and *Ethereum* as the first and second potential blockchain platform in their shortlist. *Cosmos Network* has gained the highest score among the DSS feasible blockchain platforms for the Veris Foundation, because *Cosmos Network* is flexible regarding different *pluggable consensus mechanisms* and supports any combinations of *permission/permission-less* and *public/private* blockchain platforms 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.

## B. Analysis of the results

Table VI states that *Chain* is a feasible blockchain platform for all three case studies, which means that this blockchain platform at least supports all of the blockchain feature requirements with *Must-have* priority and does not support the blockchain feature requirements with *Won't-Have* priority. None of the case-study participants considered *Chain* as a potential feasible blockchain platform for their company, which demonstrates that the DSS potentially can come up with more feasible blockchain platforms than human experts.

Another interesting observation is that the main decision that has to be made is the choice between *permission* or *permission-less* blockchain platforms and whether *cryptographic tokens* are required or not.

Concerning effectiveness, the case-study participants asserted that the updated and validated version of the decision model is useful and valuable in finding the shortlist of feasible blockchain platforms. Moreover, the DSS reduces the time and cost of the decision-making process. The case-study participants expressed that the DSS enabled them to meet more detailed blockchain feature requirements. Furthermore, they were surprised to find what their primary concerns seem to be, especially when the opinions of different experts are combined.

The validity metric defined as the degree to which an artifact works correctly. There are two ways to measure validity: 1) the results of the DSS compared to the predefined case-study participant shortlist of potential feasible blockchain platforms, and 2) according to the blockchain experts' opinion.

The case-study participants confirm that the DSS provides effective blockchain platforms to help software-producing organizations in their initial decisions for selecting blockchain platforms. In other words, the DSS recommended nearly the same blockchain platforms as the case-study participants suggested to their companies after extensive analysis and discussions. However, the DSS offers a short ranked list of feasible blockchain platforms; therefore, software-producing organizations should perform further investigations, such as performance testing, to find the best fitting blockchain platform for their software products.

## VII. DISCUSSION

The DSS assists requirements engineers in the requirements elicitation activity by offering a list of prominent blockchain features. Software-producing organizations have different perspectives on their blockchain feature requirements in different phases of the software development life-cycle. Requirement engineers (decision-makers) might want to consider generic blockchain features in the early phases of the life-cycle, whereas they are interested in more technical blockchain features as their development process matures. For instance, *Consensus Mechanism* could be prioritized as a *Should Have* blockchain feature in the design phase, but in the implementation phase, one of its sub-features (more technical blockchain feature), e.g., *Proof-of-Work*, might be selected instead. Furthermore, blockchain features' priorities could be changed in different phases. Therefore, the DSS might come up with various blockchain platforms for a software-producing organization in different phases of its software development life-cycle. As the blockchain feature requirements for each *Case Definition* are stored in the knowledge base of the DSS, it does not cost a significant amount of time to rerun the decision-making process. Therefore, the DSS, as a requirements management tool, provides a platform to enable decision-makers to analyze, document, track collaboratively, and prioritize their blockchain features requirements.

**TABLE VI:** this table presents the outcomes of the DSS for ShareCompany BIQH, DUO, and Veris Foundation based on their blockchain feature requirements' priorities. The columns *DSS Feasible Solutions* and *CP Shortlist* demonstrate the deduced feasible solutions by the DSS and the shortlist of potential solutions by the case-study participants. Moreover, the Columns *CP Rank* and *DSS score* show the order of potential feasible solutions based on the case-study participants' opinions and the calculated scores of the feasible solutions by the DSS.

Case Study	DSS Feasible Solutions	CP Shortlist	DSS Score	CP Rank
ShareCompany BIQH	Hyperledger	✓	99.39	1
	R3 Corda		68.13	-
	JPMorgan Quorum	✓	61.92	2
	Chain		40.05	-
DUO	Ethereum	✓	98.25	1
	Hyperledger	✓	73.22	3
	Wanchain		64.68	-
	NEO	✓	62.10	2
	Cosmos Network		51.10	-
	Stellar		37.91	-
	Komodo		37.65	-
	Waves Platform		37.25	-
	Chain		34.30	-
	VeChain		31.31	-
Veris Foundation	Cosmos Network		99.64	-
	NEO	✓	69.42	1
	Ethereum	✓	54.52	2
	Stellar		53.33	-
	Hyperledger		44.48	-
	Chain		44.48	-
	VeChain		30.27	-
	ICON		28.63	-
	Symbiont		28.16	-
Neblio		21.37	-	

Biases, such as motivational and cognitive [29], arise because of shortcuts or heuristics that decision-makers use to solve problems and perform tasks. The Hawthorne effect, which is the tendency for decision-makers to change their behavior when they are being observed, is a form of cognitive bias. The case-study participants might have been more careful in the observational setting than they would be in the real setting because they are being observed by scientists judging their selected blockchain feature requirements and priorities. Moreover, the Bandwagon effect, which is the tendency to do or believe things because many other decision-makers do or believe the same, is another form of cognitive bias. The Bandwagon effect typically shows up in group decisions. To mitigate the Hawthorne and Bandwagon effects, individual *and* group interviews have been conducted. The DSS provides a discussion and negotiation platform to enable requirement engineers to make group decisions. It detects and highlights the conflicts in the assigned priorities to the blockchain feature requirements by decision-makers and asks them to resolve disagreements. Thus, the DSS supports requirements engineers in the Requirements verification and validation activity by avoiding conflict between blockchain feature requirements and generating feasible solutions according to the blockchain feature requirements. Moreover, the DSS is a communication tool among the decision-makers to facilitate the requirements specification activity.

We define DSS success when it, in part, aligns with the case-study participants shortlist and when it provides new suggestions that are identified as being of interest to the case-study participants. Using the case-study participants' opinion as a measurement instrument is risky, as the case-study participants may not have sufficient knowledge to make a valid judgment. We counter this risk by conducting more than one case study, by assuming that the case-study participants are handling in their interest, and by applying the DSS to other problem domains, where we find similar results [2], [3], [4].

## VIII. CONCLUSION AND FUTURE WORK

Blockchain technology is evolving rapidly, and the number of blockchain platforms in the market is growing rapidly. Furthermore, software-producing organizations are increasingly considering blockchain technology for inclusion in their products. Therefore, a decision model is required to externalize and organize knowledge regarding the current state of blockchain technology, besides, to assist decision-makers at software-producing organizations with selecting right blockchain platforms based on their preferences and requirements.

In this study, the blockchain platform selection process is modeled as a multi-criteria decision-making problem that deals with the evaluation of a set of alternatives, and taking into account a set of decision criteria [1]. Moreover, we introduced a decision model for the blockchain selection problem based on the technology selection framework [2]. We have designed and implemented a decision support system for supporting decision-makers with their technology selection problems in software production. The decision support system provides a modeling studio to build such decision models for technology selection problems. The decision models can be uploaded to the knowledge base of the decision support system to facilitate the decision-making process for software-producing organizations. The proposed decision support system addresses common knowledge management issues, including capturing, sharing, and maintaining knowledge.

The novelty of the approach is that it provides knowledge about blockchain platforms to support uninformed decision-makers while contributing a sound decision model to knowledgeable decision-makers. Furthermore, it incorporates deeply embedded requirements engineering concepts, such as the ISO software quality standards and the MoSCoW prioritization technique, besides knowledge engineering theories, to develop the decision support system. We conducted three case studies to evaluate the usefulness and effectiveness of the decision support system to address multi-criteria decision-making problems. Our website<sup>6</sup> is up and running to keep the knowledge base of the decision support system up-to-date and valid. We aim to create a community around the platform that will regularly update the curated knowledge base with new blockchain platform features.

Probing more in-depth, the decision model presented in this paper also provides a foundation for future work in multi-criteria decision-making problems. We intend to build trustworthy decision models to address *Software Architecture Pattern* and *Model-Driven Development Platform* selection problems as our (near) future work.

<sup>6</sup><https://dss-mcdm.com>



## APPENDIX

In this section, we listed the considered blockchain features and platforms.

**TABLE VII: Blockchain Types**

#	Blockchain Feature	Description:
1	<b>Public Platforms</b>	are open networks that permit anyone to participate in the network. Such a network relays on the number of participants for its success and supports public participation through an incentivization mechanism. An example of a public blockchain is Bitcoin, where participants in the network are rewarded with BTC tokens.
2	<b>Private Platforms</b>	protect the privacy and security of data. Participation in a private blockchain requires an invitation. Such invitations are validated by the network starter or a collection of rules. Private blockchains can also restrict participant activity such that individual transactions can only be carried out by specific participants, even though they are on the network. Such characteristic requires an added layer of privacy.
3	<b>Permissioned Platforms</b>	manage an access control layer to allow specific actions to be performed only by individual identifiable participants. Such blockchains platforms differ from the public as well as private blockchains. Permissioned blockchains function as closed ecosystems, where users are not freely able to join the network or issue their transactions. Permissioned blockchains are preferred by centralized organizations, which leverage the power of the network for their own internal business operations. Typically, company consortiums hire private blockchains to record transactions securely, and exchange information between one another.
4	<b>Permissionless Platforms</b>	allow every user to create a personal address and begin interacting with the network, by submitting transactions and appending entries to the ledger. Additionally, all parties have the choice of running a node on the system or applying the mining protocols to verify transactions.

**TABLE VIII: Consensus Mechanisms**

#	Blockchain Feature	Description:
1	<b>Proof-of-Work (PoW)</b>	is a consensus mechanism that is employed to confirm transactions and generate new blocks to the chain. With PoW, miners compete against each other to perform transactions on the network and gain rewards.
2	<b>Proof-of-Stake (PoS)</b>	is a consensus algorithm by which a cryptocurrency blockchain network aims to achieve distributed consensus. In PoS-based cryptocurrencies, the producer of the next block is chosen based on diverse combinations of random selection, wealth, or age.
3	<b>Delegated Proof of Stake (DPoS)</b>	is a consensus mechanism for maintaining the final agreement across a blockchain network, validating transactions, and performing as a form of digital democracy.
4	<b>Practical Byzantine Fault Tolerance (pBFT)</b>	is a consensus algorithm and works efficiently in asynchronous (no upper bound on when the response to the request will be received) systems. It is optimized for low overhead time.
5	<b>Delegated Byzantine Fault Tolerance (dBFT)</b>	is a consensus algorithm and performs similarly to a country's governance system, having its citizens, delegates, and speakers to ensure that the country (blockchain network) is functional. It is closer to PoS rather than PoW by employing a voting system to determine delegates and speaker.
6	<b>Proof-of-Authority (PoA)</b>	is a replacement for PoW, which can be utilized for private blockchains. It does not depend on nodes solving arbitrarily severe mathematical problems, but instead uses a set of "authorities" - nodes that are explicitly permitted to generate new blocks and secure the blockchain.
7	<b>Federated Byzantine Agreement (FBA)</b>	is a form of Byzantine fault tolerance where each byzantine general is responsible for their blockchain. An FBA has high throughput, scalability, and low transaction costs. Notable cryptocurrencies using the FBA include Stellar and Ripple. Stellar was the first cryptocurrency that implemented a secure FBA.
8	<b>Proof of elapsed time (POET)</b>	is a consensus algorithm that prevents high resource utilization and high energy consumption and keeps the process more efficient by following a fair lottery system.
9	<b>SIEVE</b>	is a consensus algorithm and enables a blockchain network to identify and eliminate possible non-deterministic requests, and also gain consensus on the output of the suggested transactions.
10	<b>Cross-Fault Tolerance (XFT)</b>	is a consensus mechanism and smartly cuts a corner and ignores possible attacks that are deemed to be costly. The XFT protocol simplifies the attack model and makes Byzantine Fault tolerance achievable and efficient for practical scenarios. The XFT can assist with building reliable protocols that can tolerate both the crash machine faults regardless of the number of network faults and non-crash machine faults when the number of machines that are either faulty or partitioned is within a threshold.
11	<b>Directed Acyclic Graph (DAG)</b>	is a directed graph data structure that uses a topological ordering. The blockchain combination with DAG comes from the idea of sidechains. Different types of transactions can run on different chains simultaneously.

**TABLE IX: Blockchain Tokens**

#	Blockchain Feature	Description:
1	<b>Cryptographic Tokens</b>	are a particular kind of cryptocurrency tokens that exist on blockchain networks and represent an asset or utility. For instance, one can have a cryptographic token that signifies some customer loyalty points on a blockchain platform that is employed to handle such details for a retail chain.
2	<b>Native Tokens</b>	are essential elements of the incentive scheme of a blockchain network. Such tokens encourage a group of users who do not know or trust each other to organize themselves around the purpose of a particular blockchain application.
3	<b>Non-native Protocol Tokens</b>	are implemented on top of a blockchain platform and utilize it to process transactions. Such tokens are built in a crypto-economic protocol rather than directly on the token. They are tracked on an underlying blockchain to which it is not integral.
4	<b>dApp Tokens</b>	are multipurpose utility tokens and enable an ecosystem of utilities, resources, and services mainly assisting with the requirements of dApp developers developing dApps.
5	<b>Cryptocurrency Tokens</b>	are the original (and most straight-forward) form blockchain-derived tokens. Such tokens are created uniquely as a means of payment for assets and services external to the blockchain platform running the tokens.
6	<b>Network Tokens</b>	are utilized to make the network fault-tolerant besides attack and collision-resistant.
7	<b>Investment Tokens</b>	are data units that symbolize the rights of an investor to participate in an investment in any project or business.
8	<b>Asset-backed Tokens</b>	express economic rights of real-world assets, such as art and real estate. In other words, such tokens denote ownership of an asset, imagine them in the hands of Platform Owners.
9	<b>Network Value Tokens</b>	are tied to the value of a particular blockchain network and not the value of an issuing entity. The value is generated and exchanged on the network.
10	<b>Share-like Tokens</b>	have features such as ownership of an entity, voting rights, dividends, profit shares, or interest in the development of an entity.
11	<b>Usage Tokens</b>	are required to utilize the value generated on a blockchain network; for instance, a peer participant can use these tokens to pay for services coming from other peers.
12	<b>Work Tokens</b>	are required to provide work on a blockchain network; for instance, a peer partner can use them to prove she has the right to contribute.
14	<b>Utility Tokens</b>	are digital assets designed to be consumed within a specific blockchain ecosystem. In other words, such tokens empower future access to the products or services of a company. Accordingly, utility tokens are not designed to be an investment.
13	<b>Hybrid Tokens</b>	include elements of utility tokens. Such tokens can be utilized to unlock the utility from a decentralized application, thereby acting as a means of exchange within a blockchain network. Similar to utility tokens, the lack of utility for token holders in such applications is mitigated by the additional possibility for financial profits through a hybrid token's value when the network becomes broadly adopted.
15	<b>Security Tokens</b>	signify full or fractional ownership in an asset (such as shares in a company, or a real-estate asset). Security tokens bring significant transparency over traditional paper shares through blockchain networks.

**TABLE X: Blockchain Layers**

#	Blockchain Feature	Description:
1	<b>Protocol Layer</b>	specifies a collection of rules that manage a blockchain network. Blockchain protocols usually incorporate rules about consensus, transaction validation, and network participation. Blockchain holds the majority of its value in the base, protocol layer with only a fraction of that value distributed along at the applications layer.
2	<b>Network Layer</b>	is characterized by the peer-to-peer network topology, client attachments and communication strategies, and user behaviors. This layer deals with the distribution of transactions and block data among accessible peers in the network, the reliability of the network, and local validation of data.
3	<b>Application Layer</b>	introduces capabilities that provide application interfaces on top of the blockchain, both out-of-the-box functionality and custom implementations. These capabilities explain how the digital ledger is implemented, how smart contracts can be built and operate on the blockchain, and how third-party applications can interact with the digital ledger and smart contracts.

**TABLE XI: Cryptocontract**

#	Blockchain Feature	Description:
1	<b>Smart-Contract</b>	manages the transfer of cryptocurrencies or assets between parties under particular conditions. A smart contract not only determines the rules and penalties regarding an agreement in the same way that a traditional contract does, but it can also automatically enforce those obligations.
2	<b>Virtual Machine</b>	is a runtime environment for smart contracts in blockchain technology. It is not only sandboxed but fully isolated, which implies that code running inside the virtual machine has no access to network, filesystem, or other processes. In other words, smart contracts have restricted access to other smart contracts.
3	<b>Turing Completeness</b>	indicates that a blockchain platform is implemented with a Turing-complete language. In other words, a Turing Complete blockchain platform enables users to write programs (contracts) that can (for the most part) solve any feasible computational problem.

**TABLE XII: Programming Languages**

#	Blockchain Feature	Description:
1	<b>Solidity</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
2	<b>Python</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
3	<b>Golang</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
4	<b>Java</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
5	<b>JavaScript</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
6	<b>.NET</b>	is a programming language framework developed by Microsoft that runs initially on Microsoft Windows. Some programming languages, such as <i>C#.net</i> , that support .Net framework and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.
7	<b>C++</b>	is a programming language and can be supported by a blockchain platform to develop applications, tokens, or smart contracts.

**TABLE XIII: Privacy in Blockchain**

#	Blockchain Feature	Description:
1	<b>Zero-knowledge Proof/Protocol</b>	allows one party, called PROVER, to prove another party, called VERIFIER, that PROVER knows some facts (such a secret or a proof of a theorem) without revealing to the VERIFIER ANY information about PROVER's knowledge (secret, proof, etc.).
2	<b>zk-SNARK</b>	is an acronym for Zero-Knowledge Succinct Non-interactive Argument of Knowledge. zk-SNARKs verify the correctness of a computation without executing the computation itself. The computation process can be unknown — just the ability to check that a computation was done correctly. zk-SNARKs is mainly implemented when privacy is an issue; in such a situation, the details are not required to verify something, so zk-SNARKs bring trust in trustless environments.
3	<b>Ring signatures</b>	are digital signatures that can be executed by any member of a group of users that each of them has some keys. Accordingly, a message signed with a ring signature, which is validated by someone in a specific group of users. One of the security characteristics of a ring signature is that it should be infeasible to reveal which of the group members' keys was used to generate the signature. Ring signatures have two essential characteristics: 1) there is no way to eliminate the anonymity of a specific signature, 2) any group of users can be employed as a group without extra setup.

**TABLE XIV: Interoperability in Blockchain**

#	Blockchain Feature	Description:
1	<b>Atomic Swap</b>	is a peer-to-peer exchange of digital currencies from one party to another, without going through a third-party service like a crypto exchange. During this whole process, users have full control and ownership of their private keys.
2	<b>Cross-chain technology</b>	enables transmission of value and information among different blockchain networks. In other words, cross-chain technology allows for interoperability by proposing a platform that lets different blockchains interact with each other, and in principle, operate as a single chain.
3	<b>Enterprise system integration</b>	is integration with current and legacy systems by employing the least disruptive path.

**TABLE XV: Resilience in Blockchain**

#	Blockchain Feature	Description:
1	<b>Hard fork resistant</b>	prevents Hard forks. A hard fork is a radical change to the protocol that makes previously invalid blocks/transactions valid (or vice-versa). This needs all nodes or users to upgrade to the latest version of the protocol software.
2	<b>Spam attack resistant</b>	prevents spam attacks. A spam attack occurs in the form of empty invocations, admitting a considerable number of transactions to be submitted without any actual transfer of funds.
3	<b>Sybil attack resistant</b>	prevents Sybil attacks. Sybil Attack is an attack that happens in peer-to-peer networks in which a node in the network runs multiple identities actively at the same time and threatens the authority/power in blockchain systems.
4	<b>Quantum attack resistant</b>	refers to cryptographic algorithms that are thought to be protected against an attack by a quantum computer.
5	<b>Instant transaction Finality</b>	guarantees transaction finality. A block is 'final' when it cannot be revised any more. Since Nakamoto consensus does not provide such revisions, it is not considered to be consensus safe.

TABLE XVI: Scalability in Blockchain

#	Blockchain Feature	Description:
1	<b>On-chain transactions</b>	refer to cryptocurrency transactions that occur on a blockchain platform and remain dependent on the state of the blockchain for their validity. Such transactions are considered to be valid when the blockchain is modified to reflect the transactions on public ledger records.
2	<b>Off-chain transactions</b>	refer to transactions occurring on a cryptocurrency network which move the value outside of the blockchain. Because of their zero/low cost, off-chain transactions are getting more popular, particularly among large participants.
3	<b>Off-chain channels</b>	<b>state</b> are necessarily a mechanism by which blockchain interactions and typically occur on the blockchain instead get conducted off of the blockchain. This is done securely without raising the risk of any participant while providing notable improvements in cost and speed.
4	<b>Sidechains</b>	improve the performance of blockchain networks. Instead of computing everything in one big blockchain, a side chain can be employed to take some things out of the computation while guaranteeing the correctness of the computation. Sidechains are formed by locking values in their parent chains and transmitting them to secured distributed ledgers.
5	<b>Sharding</b>	is a method that provides scalability besides maintainability, privacy, and security in a blockchain platform. Sharding is a way of partitioning to spread out the computational and storage workload across a blockchain network so that each node is not responsible for processing the entire network's transactional load. Instead, each node only keeps information regarding its partition or shard.
6	<b>Plasma-chain</b>	is a set of smart contracts executing on top of a root-chain forming a trusted sidechains-of-sidechains network connected to a root-chain in a hierarchical and tree-like structure with enforced blockchain histories and MapReducible computations performed into a set of merkle proofs.

TABLE XVII: Blockchain Platforms

#	Blockchain Platform	Open Source	Public	Private	Permissioned	Permissionless	URL
1	<b>Ethereum</b>	✓	✓	✓	✓	✓	www.ethereum.org
2	<b>R3 Corda</b>	✓	✗	✓	✓	✗	www.corda.net
3	<b>JPMorgan Quorum</b>	✓	✗	✓	✓	✗	www.jpmorgan.com
4	<b>Hyperledger</b>	✓	✗	✓	✓	✗	www.hyperledger.org
5	<b>BigChainDB</b>	✓	✗	✓	✓	✗	www.bigchaindb.com
6	<b>MultiChain</b>	✓	✗	✓	✓	✗	www.multichain.com
7	<b>HydraChain</b>	✓	✗	✓	✓	✗	www.github.com/HydraChain
8	<b>Chain</b>	✓	✗	✓	✓	✗	www.chain.com
9	<b>Symbiont</b>	✓	✗	✓	✓	✗	www.symbiont.io
10	<b>Stratis (Azure BaaS)</b>	✓	✗	✓	✓	✗	www.stratisplatform.com
11	<b>OpenChain</b>	✓	✗	✓	✓	✗	www.openchain.org
12	<b>NEO</b>	✓	✓	✗	✗	✓	www.neo.org
13	<b>Cardano</b>	✓	✓	✗	✗	✓	www.cardano.org
14	<b>Stellar</b>	✓	✓	✗	✗	✓	www.stellar.org
15	<b>Ripple</b>	✓	✓	✗	✓	✗	www.ripple.com
16	<b>Bitshares</b>	✓	✓	✗	✗	✓	www.bitshares.org
17	<b>QTUM</b>	✓	✓	✗	✗	✓	www.qtum.org
18	<b>ICON</b>	✓	✓	✗	✓	✓	www.icon.foundation
19	<b>VeChain</b>	✓	✓	✗	✓	✗	www.vechain.org
20	<b>IOTA</b>	✓	✓	✗	✗	✓	www.iota.org
21	<b>Factom</b>	✓	✓	✗	✗	✓	www.factom.com
22	<b>Cosmos Network</b>	✓	✓	✗	✗	✓	www.cosmos.network
23	<b>Lisk</b>	✓	✓	✗	✗	✓	www.lisk.io
24	<b>Waves Platform</b>	✓	✓	✗	✗	✓	www.wavesplatform.com
25	<b>Wanchain</b>	✓	✓	✗	✓	✓	www.wanchain.org
26	<b>Neblio</b>	✓	✓	✗	✓	✓	www.nebl.io
27	<b>Zilliqa</b>	✓	✓	✗	✗	✓	www.zilliqa.com
28	<b>Komodo</b>	✓	✓	✗	✗	✓	www.komodoplatform.com

## REFERENCES

- [1] Evangelos Triantaphyllou, B Shu, S Nieto Sanchez, and Tony Ray. Multi-criteria decision making: an operations research approach. *Encyclopedia of electrical and electronics engineering*, 15(1998):175–186, 1998.
- [2] Siamak Farshidi, Slinger Jansen, Rolf de Jong, and Sjaak Brinkkemper. A decision support system for cloud service provider selection problem in software producing organizations. In *2018 IEEE 20th Conference on Business Informatics (CBI)*, volume 1, pages 139–148. IEEE, 2018.
- [3] Siamak Farshidi, Slinger Jansen, Rolf de Jong, and Sjaak Brinkkemper. A decision support system for software technology selection. *Journal of Decision Systems*, 27(sup1):98–110, 2018.
- [4] Siamak Farshidi, Slinger Jansen, Rolf de Jong, and Sjaak Brinkkemper. Multiple criteria decision support in requirements negotiation. In *the 23rd International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2018.
- [5] James W Dean Jr and Mark P Sharfman. Procedural rationality in the strategic decision-making process. *Journal of management Studies*, 30(4):587–610, 1993.
- [6] Dinora R Fitzgerald, Susan Mohammed, and Gül Okudan Kremer. Differences in the way we decide: The effect of decision style diversity on process conflict in design teams. *Personality and Individual Differences*, 104:339–344, 2017.
- [7] Lutz Kaufmann, Sebastian Kreft, Matthias Ehrhoff, and Felix Reimann. Rationality in supplier selection decisions: The effect of the buyer’s national task environment. *Journal of Purchasing and Supply Management*, 18(2):76–91, 2012.
- [8] Mrinmoy Majumder. Multi criteria decision making. In *Impact of urbanization on water shortage in face of climatic aberrations*, pages 35–47. Springer, 2015.
- [9] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28(1):6, 2008.
- [10] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1085–1100. ACM, 2017.
- [11] Garrick Hileman and Michel Rauchs. Global blockchain benchmarking study, 2017.
- [12] Carsten Maple and Jack Jackson. Selecting effective blockchain solutions. In *European Conference on Parallel Processing*, pages 392–403. Springer, 2018.
- [13] Tsung-Ting Kuo, Hugo Zavaleta Rojas, and Lucila Ohno-Machado. Comparison of blockchain platforms: a systematic review and healthcare examples. *Journal of the American Medical Informatics Association*, 26(5):462–478, 2019.
- [14] Pablo Yabo. Comparison of cryptocurrency developments. key metrics of blockchain platforms.
- [15] M Macdonald, L Liu-Thorold, and R Julien. The blockchain: a comparison of platforms and their uses beyond bitcoin. *COMS4507-Adv. Computer and Network Security*, 2017.
- [16] Philipp Frauenthaler, Michael Borkowski, and Stefan Schulte. A framework for blockchain interoperability and runtime selection. *arXiv preprint arXiv:1905.07014*, 2019.
- [17] Davor Maček and Dino Alagić. Comparisons of bitcoin cryptosystem with other common internet transaction systems by ahp technique. *Journal of Information and Organizational Sciences*, 41(1):69–87, 2017.
- [18] Huimin Tang, Yong Shi, and Peiwu Dong. Public blockchain evaluation using entropy and topsis. *Expert Systems with Applications*, 117:204–210, 2019.
- [19] Mirko Staderini, Enrico Schiavone, and Andrea Bondavalli. A requirements-driven methodology for the proper selection and configuration of blockchains. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, pages 201–206. IEEE, 2018.
- [20] Claus Pahl, Nabil El Ioini, and Sven Helmer. A decision framework for blockchain platforms for iot and edge computing. In *International Conference on Internet of Things, Big Data and Security*, 2018.
- [21] Karl Wüst and Arthur Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54. IEEE, 2018.
- [22] Tommy Koens and Erik Poll. What blockchain alternative do you need? In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 113–129. Springer, 2018.
- [23] Thomas L Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.
- [24] ISO. Iec25010: 2011 systems and software engineering—systems and software quality requirements and evaluation (square)—system and software quality models. *International Organization for Standardization*, 34:2910, 2011.
- [25] Mrinmoy Majumder. *Multi Criteria Decision Making*, pages 35–47. Springer Singapore, Singapore, 2015.
- [26] A.P. Sage. *Decision support systems engineering*. Wiley series in systems engineering. J. Wiley, 1991.
- [27] Juan Pablo Carvallo and Xavier Franch. 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*, pages 9–14. ACM, 2006.
- [28] DSDM Consortium et al. The dsdm agile project framework. <http://www.dsdm.org/dig-deeper>, 2014.
- [29] Gilberto Montibeller and Detlof Winterfeldt. Cognitive and motivational biases in decision and risk analysis. *Risk Analysis*, 35(7):1230–1251, 2015.