Informatics

# Analyse von Konsens-Mechanismen in Smart Contract Plattformen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Wirtschaftsinformatik**

eingereicht von

**Michael Mayer, BSc**
Matrikelnummer 00925636

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ass.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Monika di Angelo

Wien, 12. Jänner 2020

Michael Mayer

Monika di Angelo

TU WIEN Informatics

# Analysis of Consensus Mechanisms of Smart Contract Platforms

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Business Informatics**

by

**Michael Mayer, BSc**
Registration Number 00925636

to the Faculty of Informatics

at the TU Wien

Advisor: Ass.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Monika di Angelo

Vienna, 12th January, 2020

Michael Mayer

Monika di Angelo

# Erklärung zur Verfassung der Arbeit

Michael Mayer, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 12. Jänner 2020

Michael Mayer

# Acknowledgements

The completion of this thesis could not have been possible without the participation and assistance of some people I would like to thank here.

First of all I wish to express my sincere appreciation to my supervisor, Ass.Prof. Dr. Monika di Angelo, for providing guidance and feedback throughout this project. Without her persistent help, the goal of this project would not have been realized.

I would like to especially thank my partner, who gave me the initial idea for choosing this topic and supported me with feedback, inputs and proofreading through the whole project.

I would also like to acknowledge the support of my family and friends. They kept me going on and this work would not have been possible without their input.

# Kurzfassung

Die Blockchain Technologie hat das Potential die Wirtschaft im Zeitalter der Digitalisierung zu revolutionieren. In zentralisierten Organisationen werden die Entscheidungen von einem Verantwortlichen oder einer Gruppe getroffen. In der Blockchain gibt es keinen bestimmten Entscheidungsträger. Um eine Entscheidung zu treffen, muss ein Konsens erreicht werden. Um diesen Konsens zu erreichen, wird von jeder Blockchain Plattform ein so genannter Konsens-Mechanismus eingesetzt. Damit Blockchains ihr volles ökonomisches Potential ausschöpfen können, muss eine gute Skalierbarkeit gegeben sein. Doch dazu müssen Kompromisse in anderen Bereichen eingegangen werden. Das Blockchain-Trilemma besagt, dass man die drei Haupteigenschaften Skalierbarkeit, Sicherheit und Dezentralität immer zulasten der anderen erreicht, bzw. dass alle drei Eigenschaften nicht gleichzeitig maximiert werden können. Die implementierten Konsens-Mechanismen versuchen dieses Dilemma mit unterschiedlichen Ansätzen zu lösen. Daher sind sie einer der wichtigsten Aspekte sowie Unterscheidungsmerkmale der Blockchain Plattformen. Diese Arbeit bietet einen Überblick sowie eine technische Analyse über die zur Zeit existierenden Konsens-Mechanismen. Darüber hinaus werden Smart Contract Plattformen analysiert und die dort eingesetzten Konsens-Mechanismen detailliert beleuchtet. Außerdem wird die Umstellung des Konsens-Mechanismus von Proof of Work zu Proof of Stake in der Ethereum-Plattform anhand Ihres Einflusses auf zukünftige Smart Contracts analysiert. Die Analyse liefert einen Überblick über 62 Konsens-Mechanismen und 21 Smart Contract-Plattformen die im Moment eingesetzt werden. Weiter werden Kriterienkataloge vorgestellt, welche für den Vergleich von Smart Contract-Plattformen sowie deren Konsens-Mechanismen eingesetzt werden können.

# Abstract

The blockchain technology has the ability to revolutionize the digital economy. In centralized organizations, all the decisions are taken by the leader or a board of decision makers. As the blockchain is decentralized system and therefore has no leader, this is not possible. To make decisions a consensus has to be reached. In order to get consensus, each blockchain platform has implemented a so-called "consensus mechanism". As blockchains must be scalable to achieve their full economic potential, this necessarily entails compromises. For a single blockchain it seems impossible to have the three desiderates of security, decentralization and scalability at maximum. This is the so-called "impossible triangle" that the blockchain protocol suffers from. Consensus mechanisms are trying to solve this trilemma with different approaches. So consensus mechanisms are one of the most important aspects and distinctions of the different platforms for blockchains. This work provides an overview and technical analysis about the existing consensus mechanisms that are currently in use. Furthermore, smart contracts platforms are reviewed and the consensus mechanisms used by these platforms are investigated in detail. The work further discusses how the change of the platform Ethereum from Proof-of-Work to Proof-of-Stake may influence future smart contracts and the challenges that arise. The analysis gives an overview of 62 consensus mechanisms and 21 smart contract platforms that were in use at the time of writing. This work provides an in-depth review and comparison in regard to important properties of the consensus mechanisms that were used in smart contract platforms. Furthermore two criteria catalogues for comparing smart contract platforms and their used consensus mechanisms are created.

# Contents

# CHAPTER 1

# Introduction

## 1.1 Motivation

With the rise of blockchain technology, Nick Szabo's concept of smart contracts got the technical background to be realised and implemented. [Sza97]

Originally adopted as the backbone of a public, distributed ledger system, the blockchain network processed asset transactions in form of digital tokens between Peer-to-Peer users. Blockchain networks are distinguished by their inherent characteristics of disintermediation, public accessibility of network functionalities (e.g. data transparency) and tamper-resilience. [DLZ+18]

Blockchain networks provide distributed consensus and are fundamental to orchestrating the global state machine for general purpose bytecode execution and are also envisaged for the emerging open-access, trusted virtual computers [KMS+16] for decentralized, transaction-driven resource management in communication networks and distributed autonomous systems [LCO+16], [YGA+18].

For these reasons, blockchain technologies have been proposed by both, the industry and academia as the fundamental "game changer" [Gla17] in decentralization of digital infrastructures. The area of application ranges from the financial industry [BMC+15] to a broad domain including Internet of Things (IoTs) [Ksh17] and self-organized network orchestration [N. 17].

Transactions in a blockchain are in an arbitrary order by cryptographically chaining the transaction subsets in the form of data "blocks" to their predecessors. Cryptographic references help to detect data tampering. The blockchain consensus mechanisms tackle the problem of replicated agreements [Ray10], [Sch90] on a single/canonical transaction history among trustless nodes. Furthermore, the consensus mechanisms are able to offer the agreement on the global blockchain-data state among a large number of trustless nodes with no identity authentication and low messaging overhead [BSAB$^+$19].

As the blockchain technologies evolve rapidly, the demand for the higher-level quality of blockchain-based services presents more critical challenges in designing blockchain protocols. Characteristics, such as data consistency, speed of consensus finality, robustness to arbitrarily behaving nodes (i.e., Byzantine nodes [Sch90]) and network scalability are significant for the performance of the blockchain network. These characteristics can be regulated by the implemented consensus algorithm. So the performance of the adopted consensus mechanism significantly controls the performance of the blockchain network.

The blockchain has the ability to revolutionize the digital economy. With smart contract applications based on blockchain technology there is the potential to decentralize many processes we know today.

Smart contracts are computer systems that manage assets similar to automatic teller machines and the Bitcoin system. In current systems transactions are usually conducted in a centralized form where you need a middleman both parties can trust. Smart contracts enable organizations, governments, legal bodies and individuals to exchange monetary values, properties, shares, bonds with a value and contracts in a way that avoids conflicts and without the need of a third party due to its automatically enforced obligations. Once the pre-defined rules have been met, the algorithm releases digital assets to all or some of the involved parties. [Ico18] So a smart contract is an executable code that runs on the blockchain. It is able to facilitate, execute and enforce the terms of an agreement between untrusted parties. [AvM17]

The fields where smart contracts can be used are widely spread, from financial issues, insurance premiums, contract breaches to property law and much more. Smart contract platforms have shown significant growth in the last years as more and more end-user, such as banks, governments, insurances, real estate businesses and so on saw the advantages of this new technology. [Ico18]

## 1.2 Preliminaries

### 1.2.1 How Smart Contracts Work

Loosely speaking, smart contracts, in the context of blockchains, are small programs that execute a pre-written algorithm, are stored and replicated on a distributed storage platform, are run and verified by a network of computers to ensure trustworthiness and can result in ledger updates. "If blockchains give us distributed trustworthy storage, then smart contracts give us distributed trustworthy calculations." [Lew15] Smart contracts can automate some clauses of traditional contracts. This is because computer code behaves in expected ways and doesn't have the linguistic nuances of human languages. So, there are less potential points of contention. [Lew15]

Characteristics of smart contracts: [Ros19]

- Autonomy:

  The need to rely on a third-party intermediary or facilitator is eradicated. This also reduces drastically the danger of manipulation by a third party, since execution is managed automatically by the network.

- Trust:

  All documents can be stored encrypted on a secured, shared ledger. Furthermore, you don't have to trust a third party, as the unbiased system of smart contracts essentially replaces trust.

- Backup:

  Nothing can get lost, as on the blockchain, everyone has a copy of the data. All documents are duplicated many times.

- Safety:

  The encryption used in smart contracts make them extremely difficult to hack. Thanks to this complex cryptography, the integrity of documents will be safe.

- Speed:

  Due to the automated tasks in smart contracts, a lot of time can be saved that would be needed to manually process documents, sending or transporting them to specific places.

- Savings:

  Since smart contracts render intermediaries dispensable, the fees associated with their services are saved.

- Accuracy:

  Errors that come from manually filling out forms can be avoided.

The correct execution of a smart contract is enforced by the consensus protocol [Sza97]. As you can see in figure 1.1 the consensus is enforced in one of the 4 blockchain network layers. A wide range of applications can be implemented in smart contracts, this includes financial instruments (e.g., sub-currencies, financial derivatives, savings wallets, wills) and self-enforcing or autonomous governance applications (e.g., outsourced computation [LTKS15], decentralized gambling). The code of a smart contract resides on the blockchain and the contract is identified by an address. Smart contracts can be invoked by users by sending transactions to the contract address. If this transaction is accepted by the blockchain, then all participants on the network execute the contract code. By participating in a consensus protocol, the network agrees on the output and the next state of the blockchain.

### 1.2.2 How Consensus Mechanisms Work

Each and every transaction that is made on a blockchain network is completely verified and secured. This is where blockchain consensus algorithms come into play, as there are several different ways in which various blockchain networks can both verify and secure a block of transactions on its network.

To get an understanding of consensus mechanisms we first have to define blockchains. A blockchain is a special type of a distributed database. They are termed trustless, because the most significant attribute is, that no one controls them. Users submit tasks in transactions that are grouped into blocks which are linked together to form chains.

To add blocks of transactions to the blockchain different tasks are involved. These tasks are defined by the used consensus mechanism and users that voluntarily perform these tasks get rewarded.

As a blockchain is a decentralized peer-to-peer system with no central authority, which creates a system that is devoid of corruption from a single source, it still has a major difficulty.

- How are any decisions made?
- How does anything get done?

When we think of a normal centralized organization, all the decisions are taken by the leader or a board of decision makers. As the blockchain has no leader, this is not possible. To make decisions a consensus has to be reached. For getting to this consensus, each platform on the blockchain has implemented a so called “consensus mechanism”. [Ame18]

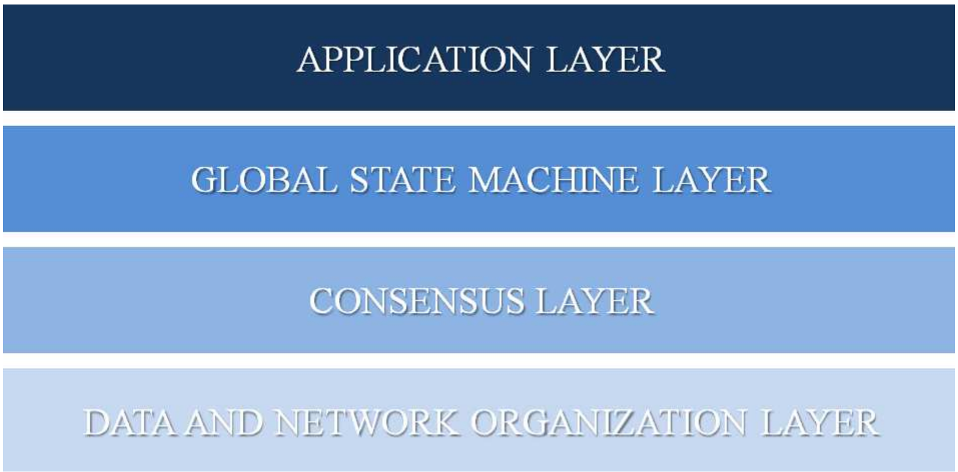

APPLICATION LAYER
GLOBAL STATE MACHINE LAYER
CONSENSUS LAYER
DATA AND NETWORK ORGANIZATION LAYER

Figure 1.1: blockchain network layers

The consensus ensures that the next block in a blockchain is the one and only version of the truth.

To reach this consensus the nodes need to agree on the transactions and the order in which these are listed on the newly-mined block. If no agreement would be reached, the blockchain will end up with forks where the nodes have different views of the world state and the network will no longer be able to maintain a unique authoritative chronology unless this fork is resolved. Therefore a consensus mechanism is needed in every blockchain network. Which consensus mechanism is used depends on the type of the blockchain network and the attack vector that the network operator adopts. [Gre15] Attack vectors that are faced by consensus mechanism are: [Wal19]

- Denial of Service:

  using lots of transactions to overload nodes.

- 51% Attack:

  controlling more than 50% of the nodes, network's mining hashrate or computing power.

- Sybil Attacks:

  one node tries to represent multiple identities.

- Cryptographic Attacks:

  break the underlying cryptography by finding a weakness in a code, cipher, cryptographic protocol or key management scheme.

The first big blockchain platform Bitcoin used Proof of Work to reach consensus. Meanwhile many consensus mechanisms have been implemented to face the triangle dilemma of blockchains. As blockchains must be scalable to achieve their full economic potential, this necessarily entails compromises. The trilemma claims that blockchain systems can only, at most, have two of the following three properties:

- Decentralization:

  The degree of diversification in ownership, influence and value in the blockchain.

- Security:

  The level of defensibility a blockchain has against attacks from external sources. Internally, it is a measure of how immutable the system is to change.

- Scalability:

  Determines the upper limit on how large a network can grow.

For a single blockchain it is impossible to have all three desiderates of security, decentralization and scalability. This is the so-called "impossible triangle" that the blockchain protocol suffers from. Consensus mechanisms are trying to solve this trilemma with different approaches. [Ico18]

Further, there is the CAP Theorem which states that in case of a partition, a distributed system can only preserve either consistency or availability.

- CONSISTENCY:

  All clients see current data regardless of update/delete

- AVAILABILITY:

  system continues to operate even with node failures

- PARTITION TOLERANCE:

  the system continues to operate despite network failures

The consensus algorithm plays a crucial role in maintaining the safety and efficiency of a blockchain. Using the right algorithm may bring a significant increase to the performance of blockchain applications. Each consensus algorithm has its own application scenario. There is no absolute good or bad. The choice of which consensus should be implemented on the blockchain depends on the type of network and data. [Wal19]

Currently, especially the widely used Proof of Work mechanism is criticized mostly, because, amongst other things, of its huge power consumption to reach consensus. [BGM16] One of the biggest blockchain platforms, Ethereum, wants to change its consensus mechanism from Proof of Work to Proof of Stake. [BS18]

So consensus mechanisms are one of the most important aspects and distinctions of the different platforms implemented on the blockchain.

At the time of writing this thesis, more than 60 different consensus mechanisms were used on different blockchain platforms.

With this large amount of mechanisms, there are many different approaches implemented to reach consensus in blockchains. Each of them has its advantages and disadvantages.

## 1.3 Problem Statement

To get an understanding of the differences, the consensus mechanisms mentioned above should be analysed and compared. A deeper analysis should then be done with the mechanisms that are used on smart contract platforms.

This analysis should help to get an overview of the used techniques and principles of the now used consensus mechanisms on smart contract platforms and provide valuable information for further researchers.

In order to analyse which effects the consensus mechanisms have on the different smart contract platforms, to gain insights into advantages and disadvantages of these mechanisms and understand why Ethereum is switching their consensus mechanism, the following research questions will be the topic of the work:

- RQ1: Which consensus mechanisms are currently used on the different blockchain platforms and how do they work?
- RQ2: Which purpose do currently used smart contract platforms have?
- RQ3: Which properties influence the choice of consensus mechanisms in smart contract platforms?
- RQ4: Which influences and challenges for future smart contracts do arise with the change of Ethereum from PoW to PoS?

## 1.4 Aim of the Work

The aim of this work is to provide an overview about the existing consensus mechanisms that are used on the blockchain and especially in the area of smart contract platforms. This overview should also include a technical analysis of the mechanisms used in these platforms.

The work should further identify how the change of the Ethereum platform from Proof of Work to Proof of Stake can influence future smart contracts and should also provide an overview of challenges that these changes may trigger.

Furthermore the analysis should provide valuable information for further researchers by analysing the techniques and principles of the now used consensus mechanisms on smart contract platforms.

## 1.5 Structure of the Work

In chapter 1 a short introduction about blockchain, smart contracts and consensus mechanisms is given and the motivation is described. After that, the research questions are named in the problem statement and furthermore the aim of the work is described.

In chapter 2 the methodology, used concepts and the literature review methods are described.

Chapter 3 gives an overview about the state of the art. Here a literature review about consensus mechanisms of blockchain platforms is conducted to get basic knowledge about the existing mechanisms and the platforms that use these techniques. This is also necessary to find out what the state of the art is and what the key performance indicators of these techniques are. The identified parameters are used to compare the techniques in the further parts. There is also an analysis and comparison of existing approaches.

Chapter 4, 5 and 6 is about the comparison and analysis of consensus mechanisms. It starts with a brief review of all consensus mechanisms that are used on different blockchain networks. Then smart contract platforms are also briefly described to get an outline about the existing approaches. Further, special focus is layed on consensus mechanisms that are used on these smart contract platforms. These consensus mechanisms are identified and used for further research in the following sections of chapter 6. This part contains the technical in-depth analysis, overview and comparison of the consensus mechanisms that are identified. It also includes a deeper look on the impacts that arise due to the changes that Ethereum will make on their platform in order to switch from PoW to PoS and discuss these and how they may influence future implementations of smart contracts.

The critical reflection in the seventh chapter show the conclusions and analyses of the results and compare them to related work. Also open issues for further research are discussed.

In the eighth chapter, the results are further evaluated and analysed based on the research questions.

CHAPTER 2

# Methodology

## 2.1 Methodological Approach

This work provides a uniform view of consensus mechanisms and smart contract platforms on blockchain networks by presenting the actual implemented solutions in chapters 4 and 5 and revealing the interconnections between the different approaches in chapter 6.

The methodological approach to reach the expected results and to answer the research questions specified in chapter 1.3, comprises six steps:

- At first, a brief overview about blockchain networks, smart contracts and consensus mechanisms is provided and a systematic literature review about consensus mechanisms of blockchain platforms is conducted to get basic knowledge about the existing mechanisms and the platforms that use these techniques.
- The second step contains the basic analysis, overview and comparison of consensus mechanisms and smart contract platforms with the help of formed clusters and criteria catalogues that were created in the literature analysis of step one. The results of this are necessary to find out what the state of the art is and what the key performance indicators of these techniques are. The identified parameters are used to compare the techniques in the further steps. Special focus is on consensus mechanisms that are used on smart contract platforms. These are identified and used for further research in the following steps.
- The next step provides an in-depth review of the mechanisms that were identified in the previous step.

- The fourth step gives a deeper look on the changes that Ethereum will make on their platform to change from PoW to PoS and discuss these changes and the challenges that arise.

- In the last step, an outlook of the potential research directions is given and the results are summarized, evaluated and analysed. The results will also be discussed and interpreted here. This part concludes this analysis by summarizing the contributions.

## 2.2 Used Concepts

To get basic knowledge about the topic, research was started by searching and reading scientific work about the blockchain technology. After that, the specialization on smart contract platforms and consensus mechanisms was derived and scientific work about these topics was searched and analysed. To gather this information, a systematic literature review method was used.

To get an overview on the differences and similarities of the consensus mechanisms, the forming of clusters based on the gathered information from the initial literature review was started. After that, the creation of a catalogue of criteria for the analysis of smart contract platforms was conducted. Also a catalogue of criteria for the analysis of consensus mechanisms was created.

Based on these two catalogues a comparative analysis of smart contract platforms and the used consensus mechanisms was done.

### 2.2.1 Literature Review Method

To gain insights in the state of the art of the blockchain technology, the research started with searching and reading scientific work all around the topic. To find these, a systematic literature review method was used. [OS10] It gives a good foundation for research in information systems and strengthens information systems as a field. [WW02] This method for reviewing the literature is developed specifically for information-system research. Therefore, this method was chosen. The review is conducted in four phases. In phase 1 the purpose and review protocol of the study is designed. The review protocol is an essential element in conducting a systematic literature-review study and minimizes biases. [TK07] Furthermore, the purpose of the review is discussed and a protocol, a searching plan, the selection criteria, a data extraction method and data analyses are designed.

Phase 2 is about searching and practical screening of the literature. Therefore, a search for academic articles using the Google scholar database was conducted and was later expanded to the google and bing search engine. The keywords for the initial searches were: blockchain, smart contracts, consensus blockchain, smart contract consensus, consensus mechanism blockchain, consensus algorithm blockchain and consensus protocol blockchain. Further, more specific keywords, like "AION whitepaper" or "Delegated Proof of Stake blockchain type" were used for the in-depth review. From the results of these searches, an efficient screening process could be conducted, discarding articles not relevant to the study, duplicates and articles that not obtain the full text.

Phase 3 presents the quality appraisal and data extraction. In this execution phase data from eligible articles based on the research questions guiding this analyses could be extracted. Further information from articles could be collected to serve as raw material for the analyses. [OS10]

In Phase 4 the findings are analysed and the results are used for the further analysis.

# CHAPTER 3

# State of the Art

## 3.1 Literature Studies

Although blockchain technology is a relatively new field, much research has been done on this topic. There is a big number of articles and papers where the blockchain technology itself, various blockchain platforms or implementations on these, like smart contracts or digital currencies, are discussed and analysed. Furthermore, most of the different platforms offer whitepapers to define their approaches.

Many of these works touch the topic of consensus mechanisms or try to implement new forms that fit to their needs, but neither analysed or compared all existing consensus mechanisms regarding the needs of smart contract platforms.

## 3.2 Analysis

To analyse and compare the existing approaches, a systematic literature research on scientific work, papers, articles, whitepapers and public pages, like technology blogs, has been done. Most of the detailed information about the consensus mechanisms and smart contract platforms could be extracted from the technical whitepapers that are provided by the platforms.

For the creation of the catalogue of criteria for the analysis and comparison parts, the existing literature was examined for properties that indicate the important features of the consensus mechanisms or platforms. Since the diverse authors used different criteria to analyse blockchain technologies, the first step was to collect all criteria, that may be useful to compare and analyse specific technologies, like in this case the consensus mechanisms and smart contract platforms.

The criteria that could be found here, are described in the chapters 5 and 6.

## 3.3 Comparison and Summary of Existing Approaches

During the past decade, the application scope of blockchains has been widely expanded. The existing approaches that were found in the literature research in the field of blockchain can be divided into the following topics: cryptocurrencies, smart contract platforms and consensus mechanisms.

### 3.3.1 Cryptocurrencies related

Due to the hype about cryptocurrencies, most of the existing reviews and surveys on blockchains emphasize narrowly the scenarios of using blockchain networks as the backbone technologies for cryptocurrencies, especially the market-dominant ones such as Bitcoin and Ethereum. As both of them are using Proof of Work as their consensus protocol, most of the works are concentrating on this algorithm or a comparison with one other approach. [DLZ$^+$18], [TS16], [LCO$^+$16], [ZXD$^+$17], [CSLR18], [ABC17].

The work of Bentov, Gabizon and Mizrahi [BGM16] shows cryptocurrencies, that are using other consensus mechanisms as the widely spread Proof of Work. They analyse existing protocols with a "substantial amount of popularity". Furthermore they "offer novel constructions of pure Proof of Stake protocols that avoid depletion of physical scarce resources, and argue that our protocols offer better security than existing protocols."

In [Vuk16] a comparison between Proof of Work and BFT consensus is done. The author did a high-level comparison for a set of important properties between them and also defined these properties.

The work of Bonneau et al. [BMC$^+$15] focused on challenges for cryptocurrencies. One of these identified challenges is the consensus protocol of the platforms. For that reason they also analysed the Proof of Work protocol and mentioned alternative computational puzzles to get consensus. They stated that "it remains unclear if it is possible to design an alternate decentralized consensus system which can improve on Bitcoin.".

### 3.3.2 Smart Contract Platform related

In the article from Icorating [Ico18], smart contract platforms are compared to each other. They divided the platforms into "3 groups from the point of view of their interoperability and logic" to compare them. The focus here lies on the market performance of the platforms. They also mentioned the used consensus mechanisms of these platforms and did a short introduction with pros and cons about them.

Also the work from Alharby and van Moorsel [AvM17] studied and mapped smart contract platforms from a technical perspective. With their study they identified research gaps in the field of smart contracts and also mentioned that there is the need to reach consensus to create new nodes and also that the "studied topics are [...] proposing new consensus methods", but did not go into detail about the mechanisms.

Bartoletti and Pompianu [BP17] did an empirical analysis of smart contracts. They focused on Ethereum and Bitcoin and studied how the notion of smart contracts is interpreted. They also mentioned the used consensus mechanisms of each platform they analysed but did not focus on them.

Furthermore, there are the whitepapers of the smart contract platforms, like [Com18], [Car17], [EOS18], [Eth14]. These focus on the implementations and consensus mechanisms that are used for their specific approaches. Some of them also compare their consensus approach to Proof of Work or Proof of Stake and mention the differences and advantages.

### 3.3.3 Consensus related

The work "Consensus in Asynchronous Distributed Systems: A Concise Guided Tour" [GHM+00] studied the consensus problem. They showed that "The Consensus problem is a fundamental problem one has to solve when building reliable asynchronous distributed systems.". And "From a practical point of view, it is important to understand the central role played by the Consensus problem when building reliable distributed systems."

Bano et al. [BSAB+19] conducted "a systematization of knowledge of blockchain consensus protocols". They developed a systematization framework to group the protocols and analyse them in perspective to design, security and performance properties. They stated that "the wide-scale adoption of blockchains is constrained by their performance and scalability limitations, and is desperately in need of new and faster consensus protocols that can cater to varying requirements and use cases.".

In [Sky17] the Cypherium platform is introduced and also some consensus mechanisms are mentioned and described. The focus is on the implementation that is done on Cypherium.

In fact there are many papers that focus on one "new" consensus protocol. Like [Vas14] who proposed a new version of BlackCoin's Proof of Stake mechanism that solves potential security issues, [PPA+15] who implemented Proof of Space on his own cryptocurrency, Spacecoin, [Kin13] who tried to improve the classical Proof of Work mechanism with the help of prime numbers, in the work of [MHWK16] the Proof of Luck protocol is presented, [Bru13] who introduced Proof-Chains and [MJS+14] where Permacoin and Proof of Retrievability is presented.

### 3.3.4 Summary

The extensive analysis of blockchains based on both, the academic and online literature, showed that a large volume of research has been conducted with the aim of improving specific aspects of blockchain consensus mechanisms. However, in spite of the few above mentioned works, a comprehensive analysis on consensus mechanisms of smart contract platforms and the related problems is still missing.

The existing studies on the blockchain technology rarely provide a global view on the issues related to consensus protocols. Especially, there is a lack of a concise overview when it comes to relevant aspects of consensus mechanisms of smart contract platforms. This work aims to fill this gap by providing an analysis on this specific topic.

Due to the rapidly changing landscape of blockchains and smart contract platforms, technologies and consensus mechanisms are subject to change. New variants, algorithms and solutions are introduced on a frequent basis.

CHAPTER 4

# Clusters of Consensus Mechanisms

## 4.1 Short Description

In this section all consensus mechanisms that were in use at the time of writing this thesis will be described shortly. The mechanisms are clustered to get a structured overview. To get an overview of the analysed consensus mechanisms and the used clusters, you can see all mechanisms and clusters in figure 4.1, which also presents a summary of the protocols discussed in this chapter. The criteria to assign the mechanisms to the clusters, are described in the list below.

The clusters will be:

- Proof-of-Stake:

  This cluster includes all algorithms where stakeholders are relevant to validate new blocks.

- Proof-of-Work:

  In these mechanisms, a miner has to provide an answer to a specific computational challenge.

- Proof-of-Capacity/Space:

  This bundle of mechanisms, utilize the capacity or storage space of users hard drive.

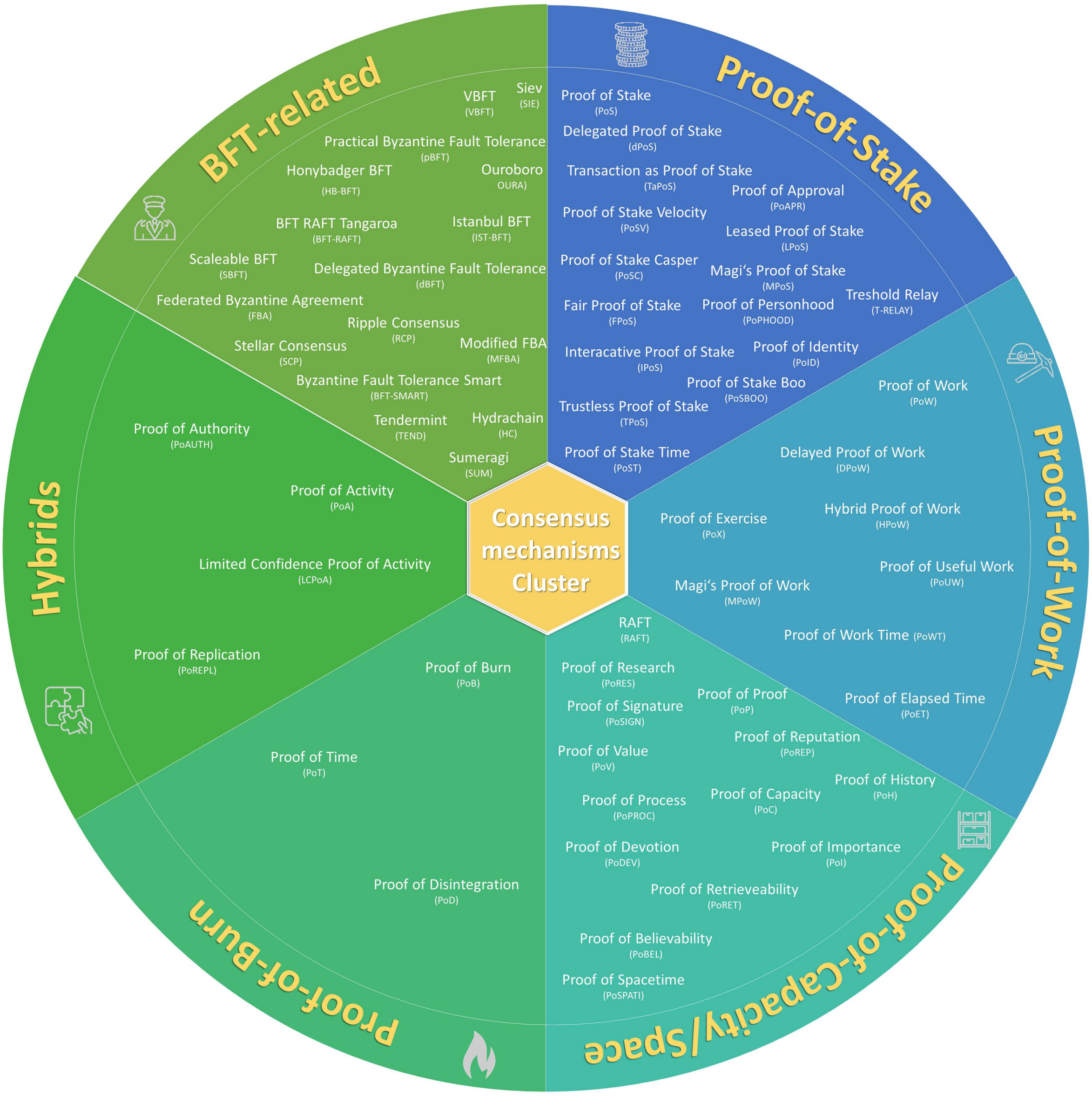

Consensus mechanisms Cluster
BFT-related
VBFT (VBFT)
Siev (SIE)
Practical Byzantine Fault Tolerance (pBFT)
Honybadger BFT (HB-BFT)
Ouroboro (OURA)
BFT RAFT Tangaroa (BFT-RAFT)
Istanbul BFT (IST-BFT)
Scaleable BFT (SBFT)
Delegated Byzantine Fault Tolerance (dBFT)
Federated Byzantine Agreement (FBA)
Ripple Consensus (RCP)
Stellar Consensus (SCP)
Modified FBA (MFBA)
Byzantine Fault Tolerance Smart (BFT-SMART)
Tendermint (TEND)
Hydrachain (HC)
Sumeragi (SUM)
Proof-of-Stake
Proof of Stake (PoS)
Delegated Proof of Stake (dPoS)
Transaction as Proof of Stake (TaPoS)
Proof of Approval (PoAPR)
Proof of Stake Velocity (PoSV)
Leased Proof of Stake (LPoS)
Proof of Stake Casper (PoSC)
Magi's Proof of Stake (MPoS)
Treshold Relay (T-RELAY)
Fair Proof of Stake (FPoS)
Proof of Personhood (PoPHOOD)
Interacative Proof of Stake (IPoS)
Proof of Identity (PoID)
Proof of Stake Boo (PoSBOO)
Trustless Proof of Stake (TPoS)
Proof of Stake Time (PoST)
Proof-of-Work
Proof of Work (PoW)
Delayed Proof of Work (DPoW)
Proof of Exercise (PoX)
Hybrid Proof of Work (HPoW)
Proof of Useful Work (PoUW)
Magi's Proof of Work (MPoW)
Proof of Work Time (PoWT)
Proof of Elapsed Time (PoET)
Proof-of-Capacity/Space
RAFT (RAFT)
Proof of Research (PoRES)
Proof of Proof (PoP)
Proof of Signature (PoSIGN)
Proof of Reputation (PoREP)
Proof of Value (PoV)
Proof of History (PoH)
Proof of Process (PoPROC)
Proof of Capacity (PoC)
Proof of Devotion (PoDEV)
Proof of Importance (PoI)
Proof of Retrieveability (PoRET)
Proof of Believability (PoBEL)
Proof of Spacetime (PoSPATI)
Proof-of-Burn
Proof of Burn (PoB)
Proof of Time (PoT)
Proof of Disintegration (PoD)
Hybrids
Proof of Authority (PoAUTH)
Proof of Activity (PoA)
Limited Confidence Proof of Activity (LCPoA)
Proof of Replication (PoREPL)

Figure 4.1: consensus mechanisms clusters

- Proof-of-Burn:

  Here all algorithms where something (coins, time,..) is burned, are clustered.

- Hybrids:

  This cluster includes combinations of consensus algorithms.

- BFT-related algorithms:

  Systems that tolerate the class of failures that belong to the Byzantine Generals' Problem are listed here.

## 4.2 Proof-of-Stake

In Proof-of-Stake related algorithms stakeholders are relevant to validate new blocks. Stakeholders are those having coins or smart contracts on the blockchain. Only they can participate. Those with high stakes are chosen to validate new blocks.

### 4.2.1 Proof of Stake (PoS)

Proof of Stake, originally proposed for Peercoin, is the most popular alternative to proof of work. In contrast to proof of work, where a lot of energy is needed to achieve consensus, it relies on the nodes staking their coins to propose the blocks and secure the network. To get a chance of selection for creating the next block, a node has to hold a certain amount of tokens for a certain timespan (=coin age). It depends on the mix of these two variables to be selected as a block proposer. The block proposer is required to stake its coin age to append to the blockchain. If it acts maliciously, the stake of the node is slashed. The coin age is destroyed if the validator claims the reward for adding a new block to the chain. This allows other people of the community to "win the raffle". [KRD17]

### 4.2.2 Delegated Proof of Stake (dPoS)

In a Delegated Proof of Stake system, a number of witnesses are voted by the stakeholders to generate blocks. The roster of these witnesses gets reorganised with each maintenance interval, to give each witness a turn to produce a block at the fixed schedule of one block per n number of seconds. For each block that is produced the witnesses get paid. Witnesses may be voted out in future elections if they fail to produce a block after being elected. [Lar17]

### 4.2.3 Proof of Stake Velocity (PoSV)

In PoSV it depends on the wallet size and wallet activity if a node is selected as block leader. Traditional PoS protocols consider coin age as a linear product, whereas in PoSV new coins age quickly and old coins slowly. Thus it encourages users to both, stake and spend the tokens by using this exponential decay function for coinage. [Ren14]

### 4.2.4 Fair Proof of Stake (FPoS)

Instead of using a uniform distribution in random variable selection, Fair Proof of Stake uses exponential distribution to select the random variables. This improvement of traditional PoS should add a ’fair’ probability for creating a new block. [BK18]

### 4.2.5 Interactive Proof of Stake (IPoS)

To generate blocks IPoS requires communication among participants. Instead of one genesis block the blockchain starts with a certain number of genesis blocks in order to avoid breaking the ticket generation rules. To determine the generators of the next block, a unique seed value, known to all participants, from the block headers is used. A special formula that uses the seed value from current and previous block headers, public keys and balance of the accounts is used to generate tickets. These tickets from all participants are required by new blocks, after a node broadcasted a new block. [Che16]

### 4.2.6 Proof of Stake Boo (PoSBOO)

Based on PoS Casper, in PoSBOO a set of pre-selected master nodes takes part in consensus and block creation. A multiplication of fixed block reward and network weight gives the block reward. If a node tries to fork the chain by voting for two blocks at the same height, 25% of the staked coins are burned. For nodes voting on false blocks more than N number of times, further penalties are imposed. The finality is mainly determined by stake and risk factors. [Shi17]

### 4.2.7 Leased Proof of Stake (LPoS)

LPoS allows the users to ’lease’ their balances to other nodes. To produce the next block, nodes with a high number of leased balances have higher chances to be selected. This increases the number of electable members, by reducing the likelihood that the network is being controlled by a single group of nodes. The rewards are shared between miners and lenders. [RMC$^+$18]

### 4.2.8 Proof of Stake Time (PoST)

The PoST algorithm uses a non-linear proof function. At a given block this function accepts the distribution enhancing time and rejects the time that diminishes it. To achieve this, a periodic time-acceptance function, that is proportional to the coins held and relative to network strength, is used. Each stake has a unique idletime, which is defined as the fraction of age that no longer supports the distribution of consensus and instead begins to degrade it. It impacts the fraction of earnable matured interests via consensus and decreases the probability to meet proof. A node must stake actively to ensure passage through the Stake-Time window for all coins held, in order to maximize the probability of earning all matured interests and signing a block during a period of time. [DM15]

### 4.2.9 Proof of Stake Casper (PoSC)

Proposed as an alternative to PoW for Ethereum, PoSC was an early attempt at the ’nothing at stake’ problem where validators are penalized for malicious activities. It uses a checkpoint tree with checkpoint blocks whose height is an exact multiple of 100. Dynasties, consisting of parts of the validators, are defined as the number of finalized checkpoints from genesis block to the parent of the block. [BG17]

### 4.2.10 Magi’s Proof of Stake (MPoS)

MPos is designed on attraction repulsion models. The stake weight is conditionally proportional to the amount and the age of the coin. An increase in the coin count does not always increase the stake weight. There is a limit of seven days for offline staking. [The18]

### 4.2.11 Transaction as Proof of Stake (TAPoS)

In TAPoS all nodes that generate transactions, contribute to network security, because all transactions are required to carry their proof of validity with them. To inform the network that the user’s stake is on a particular fork, every transaction contains the hash of the most recent block. [Lar13]

### 4.2.12 Trustless Proof of Stake (TPoS)

In TPoS users are allowed to safely stake their offline coins from cold storage. The account owner can grant permission to a different address, merchant to stake on account holder’s behalf. These merchant nodes can only validate transactions but do not take part in block creation or convincing nodes to accept transactions. To take part in voting, generating blocks and to verify transactions, stakeholders have to run masternodes. [Sta18]

### 4.2.13 Proof of Approval (PoAPR)

PoAPR publishes blocks periodically at a pre-defined interval. Stake holders are given weighted privilege, but 'candidate blocks' can be proposed and broadcasted to the network by any node. A quorum of stake holders checks how close to the target timeslot the candidate block was received by them and scores the block. The qualified candidate blocks are ranked in a list in descending score. This list is broadcasted to the network. Candidate block creators with good score, can then create an approval block and broadcast it. The winning candidate block and the approval block are finally added to the blockchain. [Tak18]

### 4.2.14 Proof of Identity (PoID)

As a piece of cryptographic evidence PoID leads to the knowledge that any user knows a private key that compares to an authorized identity. As an added measure, the identity is also cryptographically attached to a specific transaction. [Sai18]

### 4.2.15 Proof of Personhood (PoPHOOD)

The objective of PoPHOOD is to verify people, rather than identify them. They are scalable collective signing, collective authority and linkable ring signatures, which give anonymity and accountability in the same context, to get to that goal. Pseudonym parties are organized to generate tokens. [BKKJ$^+$17]

### 4.2.16 Threshold Relay (T-RELAY)

T-Relay is used in the DFINITY blockchain and uses a beacon as its source of randomness for leader selection and leader ranking on the threshold relay technique. Based on the ranks of the leaders, who propose the blocks in the chain, a weight is attributed to the chain to select between competing chains. Further there is a notarization process which dramatically improves the time to finality and eliminates the nothing-at-stake and selfish mining attacks. [TW18]

## 4.3 Proof-of-Work

Each participant on the network can participate in the block generation. In order to confirm the transaction and enter a block into the blockchain, a miner has to provide an answer, or proof, to a specific computational challenge.

### 4.3.1 Proof of Work (PoW)

PoW was adopted by Satoshi Nakamoto as a consensus mechanism for Bitcoin. To add a block to the blockchain it requires the miners to rigorously find a nonce at a certain difficulty level. When a miner finds such a nonce, he creates the block and announces it to the network. Verification is easy and then done by the other nodes in the network. They compute the hash and verify that the requirements are met. This makes changes of block impossible without redoing the work, which is costly in terms of time, energy and computational power. [Nak08]

### 4.3.2 Delayed Proof of Work (DPoW)

DPoW is a mechanism which suites best for newly formed blockchains which are an easy target for attackers, because they do not have enough computational or staking power behind them. It proposes the use of established blockchains with high hash rate to secure the transactions. Elected notary nodes archive the data on the selected PoW blockchain. If transaction costs go substantially high or another PoW network offers greater hashing power, the notaries can also elect to switch to an alternative. In DPoW the longest chain rule is applied up to the most recent backup onto the chosen PoW network. [KC18]

### 4.3.3 Hybrid Proof of Work (HPoW)

Due to the high energy costs of PoW, HPoW is proposed as an energy considerate variant. By removing the profit incentive for miners it makes it impractical for mining farms to mine a network. This encourages solo miners with lower computational power to take part in the consensus. To randomize the winning node and guarantee that not the fastest node claims the reward, HPoW requires that a miner did not receive mining reward in the previous 60 blocks. Furthermore, the reward address should have a minimum of 1000 coinage and the last two characters of the hash of the reward address must match the last two characters of block hash value. [WC19]

### 4.3.4 Proof of Work Time (PoWT)

As another alternative for the waste of computational power in PoW, PoWT proposes a variable block creation rate that scales with the mining power. In PoW the difficulty level is regularly adjusted to create blocks at a regular interval, while the block creation rate in PoWT increases with the mining power, which also increases scalability and transaction speed. Simultaneously the waste of computational power, required to find the nonce that satisfies the target difficulty, is reduced. [Ver18]

### 4.3.5 Proof of elapsed time (PoET)

To finalize the block in PoET the leader gets randomly selected using a random leader election model or a lottery based election model based on Intel's Software Guard Extensions (SGX). This model is used by the algorithm to deal with untrusted nodes and open-ended participation of nodes in the consensus algorithm. The leader election has to be randomly distributed among all available participating nodes for the consensus to work correctly. Further it needs a secure way for other nodes to verify that a given leader was correctly selected without any scope for manipulation. To achieve this PoET is intended to run in a Trusted Execution Environment (TEE), which is used to guarantee the safety and randomness of electing a leader. In order to ensure that the leader role is randomly distributed among all validating nodes, there is randomness in generating wait times. PoET requires dedicated hardware which is the only drawback of this algorithm. [CXS+17]

### 4.3.6 Magi's Proof of Work (MPoW)

MPoW proposes a network dependent reward which limits the networks hash rate. Based on an attraction repulsion model, reward is continuously adjusted. To stimulate network activities during passive mining phases the rewards get incremented. During aggressive mining phases the rewards get decremented to mitigate redundant mining sources. This allows low end devices to take part in mining and makes it unsuitable for mining pools. On the negative side, it opens the network for an adversary to overcome the network hashing power and launch a 51% attack. [Pro18]

### 4.3.7 Proof of Exercise (PoX)

As an extension of PoW, PoX is based on the idea that problems that are solved by the miners should be useful. For that reason, miners are given a computation-expensive, real world matrix-based scientific problem. [Sho17]

### 4.3.8 Proof of Useful Work (PoUW)

As in PoX, PoUW requires miners to solve meaningful difficulties. Miners are required to solve Orthogonal Vectors, 3SUM, All-Pairs Shortest Path and any problem that reduces to them. Miners can grab problems from a public problem board, where delegators post the problems. The miners then attach the Proof of Useful Work to the newly mined block.

By checking that the problem has not been previously solved and checking the hash of the newly proposed block, it can be verified and added to the blockchain. [BRSN17]

## 4.4 Proof-of-Capacity/Space (PoCS)

Proof-of-Space, also called Proof-of-Capacity, is a means of showing that one has a legitimate interest in a service by allocating a non-trivial amount of memory or disk space to solve a challenge presented by the service provider.

### 4.4.1 Proof of Capacity (PoC)

In PoC the miner's capacity of storage is privileged. The goal is to reduce the waste of computational energy for hashing, as it is the case in PoW. PoC allows storing the list of possible solutions instead of calculating the hash in every block, even before mining the block. The more space a miner has, the more solutions can be stored. This technology, which was first introduced in Burstcoin, provides the miner an advantage to solve the block. [GVS17]

### 4.4.2 Proof of Importance (PoI)

First used in NEM cryptocurrency and similar to PoS, PoI is an advanced consensus mechanism. PoI introduces some new regulations which eliminate the drawback of PoS of the rich becoming richer. These new regulations include a score-based protocol known as the Proof of Importance score. Participants with a higher score have a higher chance to be selected as a validator. The three factors vesting, transaction partner and number of transactions in the previous 30 days are used to calculate the score. [Lai18]

### 4.4.3 Proof of Reputation (PoREP)

Introduced as an extension of PoAUTH, PoREP uses the reputation of nodes to select validators. The network operates as a PoAUTH network, once the validators are selected. If the participant acts maliciously, the reputation must be important enough that he faces serious financial and brand wise consequences. Block leaders are selected by round-robin lookup. [GoC18]

### 4.4.4 Proof of History (PoH)

To prove that a transaction happened before or after the event, PoH creates a high frequency variable delay function (VDF). In order to create this VDF, the collision resistance property of hashing functions is used. To provide a PoH sequence, which ensures a reliable global passage of time, a leader node is randomly chosen from the network. Transactions are ordered and signed by the leader and then broadcasted to the network. Verifier nodes use the current state of the VDF and execute the same transactions on their copies of the state and publish their signatures of state as confirmation. [Yak18]

### 4.4.5 Proof of Retrievability (PoRET)

In PoRET a prover is required to store some large dataset. Then he has to prove to a verifier that he possesses the dataset and that it is fully retrievable. To verify these properties a challenge response protocol is used. In this protocol the verifier issues random challenges and the prover provides responses which can be verified without possessing the response. In PoRET the network performs as a decentralized distributed cloud storage. [ML14]

### 4.4.6 Proof of Believability (PoBEL)

As a variation of PoS, PoBEL introduces a believability score. This score of a node is calculated at the beginning of an epoch. A score, for the long term added value to the community, is given to each user. Validators are divided into a believable league and a normal league. The PoBEL mechanism consists of two phases. In phase one, a block is proposed by validating and ordering a set of committed transactions from a believable validator that quickly processes the transactions. The second phase consists of normal validators that sample and verify the transactions. If the normal validator detects any misbehaviour, the user loses all its stake and reputation. [Fou18]

### 4.4.7 RAFT

Working on a state replication model, in RAFT all transactions are replicated across all participating nodes. The sequence of the transactions, regardless of crashes, keeps maintained while this replication. [OJ14]

### 4.4.8 Proof of Proof (PoP)

Similar to DPoW, PoP aims to enable a security inhering blockchain IB to inherit the security of another blockchain AB in a decentralized, trustless, permissionless, and transparent manner. The miners in PoP publish the current state of IB onto AB. [San18]

### 4.4.9 Proof of Research (PoRES)

PoRES is a combination of PoS and PoW. Miners are rewarded for performing computations to solve scientific problems and the blockchain is secured using PoS. PoRES is used in Gridcoin which contributes to Berkeley Open Infrastructure for Network Computing (BOINC). The BOINC project server stores and distributes project data to nodes running BOINC clients. The server rewards the nodes in BOINC credits which are converted to gridcoins to reward the participants. As in mining pools, miners are rewarded for their relative processing contribution to the project. [Gri18], [And04]

### 4.4.10 Proof of Signature (PoSIGN)

Relying on the authorized STATIC nodes, PoSIGN uses a VPN like network called VITALS. Whenever a transaction occurs on the network, a pulse signal is sent to each node to alert them to validate and sign the new transaction. Online STATIC nodes are rewarded with transaction fees for validating and signing each new block. If offline nodes come online, they do not sign the blocks but double check them. [Com17]

### 4.4.11 Proof of Value (PoV)

As a spinoff of PoREP, PoV enables peers to reach consensus about perceived value of contribution of an individual to a network. Users are rewarded for the derived value from projects. [Mem18]

### 4.4.12 Proof of Process (PoPROC)

The idea of PoPROC is that by combining the stages of a process into a single proof, every process can be proved. This single proof is called link hash. The stages that are combined are the message digest, the digital signature, the trusted timestamp and the hashchain. The proof of one process can be included into another process as a step, forming nested proof of processes. [Str16]

### 4.4.13 Proof of Devotion (PoDEV)

As a combination of PoS and PoI, PoDEV introduces a system where accounts with highest influence in the ecology and liquidity are selected. These accounts are given equal rights to create blocks. To become block validators the top ranked accounts voluntarily stake. Pseudo randomly chosen block proposers from the validator set which is divided into dynasties. Within an epoch of X blocks validators cannot change dynasties. To create the block, all validators from the dynasty participate in the round of BFT style, time bound voting. [Tea18a]

### 4.4.14 Proof of Spacetime (PoSPATI)

In PoSPATI provers have to store data for some time and verifiers have to verify that. To get verified provers generate short sequential proofs of storage. Verification is done without interacting with the provers storage. Miners put a collateral deposit and commit to store clients data. As a proof that they are storing the data for agreed time, miners generate PoSPATI and submit to the network. [Rei16], [BG18]

## 4.5 Proof-of-Burn

Participants have to proof that they burned something (coin, time,..) in order to get the chance to mine - e.g. for a coin, that it is sent to a verifiable unspendable address.

### 4.5.1 Proof of burn (PoB)

In order to gain mining privilege on the system, a node has to 'burn' some tokens. In order to burn tokens, the node has to send them to an irretrievable but verifiable address. The system can require that the native token or some other cryptocurrency, like bitcoin, is burnt. [P4T14]

### 4.5.2 Proof of Time (PoT)

In ChronoLogic time is considered as value. Based on the Ethereum blockchain a value token named DAY is used to store time. The only way additional DAY can be produced is via the passage of time. [Chr16]

### 4.5.3 Proof of Disintegration (PoD)

Proposed as an extension of PoB, in PoD coins are not burnt by sending them to an irretrievable & verifiable address. They get fully destroyed by disintegrating the coin, reducing the circulating amount and total supply of the coin. Compared to normal nodes, the used fundamental nodes yield more staking reward. [Tea17a]

## 4.6 Hybrids

Hybrid models are most of the time a combination of existing consensus algorithms, e.g. PoW and PoS.

### 4.6.1 Proof of activity (PoA)

PoA is a combination of PoW and PoS. In traditional manner of PoW, miners first try to solve a puzzle and claim their reward. With the difference that the blocks being mined do not contain transactions. Once this empty block header is mined, the system switches to PoS. The header information is used to select a random group of validators to sign the block. These stakeholders will be selected to sign the new block, the chance to be chosen depends on the stake a validator holds. All chosen validators have to sign the block in order to become an actual part of the blockchain. If the block remains unsigned by some of the chosen validators after a given time, it is discarded as incomplete and the next winning block is used and new validators are chosen. This continues until a winning block is signed by all of the validators. The winning miner and the validators who signed the block split the network fees upon them. [BLMR14]

### 4.6.2 Proof of authority (PoAUTH)

In PoAUTH a preselected set of authorities, which identity is verified both online and in public sector, has the right to propose blocks. In each step, a mining leader can create blocks. The authorities propose the blocks for each step on a round-robin basis. Once a block has been signed off by the majority of the authorized nodes, it is accepted on the blockchain. PoAUTH is best suited for private blockchains and consortiums as it becomes intrinsically centralized by identifying authorities. [DAB+18]

### 4.6.3 Limited Confidence Proof of Activity (LCPoA)

To limit the possibility of rewriting the history of the blockchain, this extension of PoA creates automatic checkpoints in the blockchain. With these checkpoints, attackers who try a 51% attack, would only be able to rewrite a small number of blocks. [Ned18]

### 4.6.4 Proof of Replication (PoREPL)

A PoREPL is a proof system where a prover is required to commit to store one or more retrievable replicas of some data and store the data in a dedicated storage. The prover has to convince a verifier that he is storing the unique physical copies instead of duplicating multiple copies of the data in the same storage space. In this sense PoREPL is a combination of PoCS and PoRET. [BDG17]

## 4.7 BFT-related

Byzantine Fault Tolerance is the characteristic which defines a system that tolerates the class of failures that belong to the Byzantine Generals' Problem and work as long as the number of traitors does not exceed one third of the generals.

### 4.7.1 Practical Byzantine fault tolerant Mechanism (pBFT)

pBFT networks comprise a leader and validating peer nodes. The system implements a state machine replication and can tolerate faults. Block creation happens in rounds. Transactions are received, validated and broadcasted to the network by peers. The leader put the ordered transactions in a block at the end of each round. Block creation happens in 3 phases. In the pre-prepare phase, the leader broadcasts the proposed block to the peers. In the prepare and commit phases, the peers store the block locally and broadcast it to the other peers. Nodes will execute the commit phase and add the block to their current blockchain if 2/3 of validations from the peers are received. [CL02]

### 4.7.2 Delegated Byzantine Fault Tolerance (dBFT)

Instead of witnesses and delegates that are used in the similar DPoS mechanism, dBFT is composed of ordinary nodes and bookkeepers. Bookkeepers are elected by the ordinary nodes and the successful bookkeepers take part in the consensus on behalf of the ordinary nodes. To propose the next block, a random bookkeeper is selected. The block is added to the blockchain if more than 2/3 of the bookkeepers agree that the transactions are valid. [Tea15b]

### 4.7.3 Ouroboros (OUR)

Comprising of fixed time slots, OUR is a variant of PoS which operates in epoch. A group of 'qualifying' stake holders elect exactly one slot leader for each epoch. The slot leader is responsible for creating the block. The chances of being elected as block leader are proportional to the stake of a node. [KRD17]

### 4.7.4 Byzantine Fault Tolerance Smart (BFT-SMART)

The work on BFT-SMART started in 2009, long before the interest in permissioned blockchains surged around 2015. Therefore, there is a widespread agreement that BFT-SMART is the most advanced and most widely tested implementation of a BFT consensus protocol available. To make the system strictly crash fault tolerant, BFT-SMART supports a configuration parameter that can be activated if it is needed. [Aly14]

### 4.7.5 BFT RAFT Tangaroa (BFT-RAFT)

BFT-RAFT is inspired from RAFT and pBFT and aims to maintain safety, liveness and fault tolerance properties from RAFT. In BFT-RAFT nodes and users share the public keys with each other ahead of time. A node can be a leader, follower or a candidate. Leader election is done by voting and the leader serves for a fixed time term. Messages have to be signed by both nodes and users. Messages carrying invalid signatures are rejected. [Clo17]

### 4.7.6 Honeybadger BFT (HB-BFT)

As the first practical asynchronous BFT protocol, HB-BFT does not make any timing assumptions. It is optimized for scenarios where the network bandwidth is scarce, but computation is fairly ample. Pre-selected nodes with known identities have the goal to agree on the ordering of the input, given some transactions. The nodes store the received transactions in their transaction buffers. At the start of an epoch, nodes choose a subset of transactions from these buffers. This subset is provided as input to an instance of a randomized agreement protocol. The final set of transactions for the epoch is chosen at the end of the agreement process. The new set of transactions is than added to the committed log. [MXC+16]

### 4.7.7 Istanbul BFT (IST-BFT)

Inspired by pBFT, IST-BFT uses block proposers selected randomly from the validators in a round-robin fashion. A newly proposed block is broadcasted to the network with the pre-prepare message. Validators broadcast the prepare message after they enter the pre-prepare stage. After receiving the prepare messages from the validators, the block proposer enters the prepare state. In this stage he broadcasts the commit message with a proposal to insert the prepared block to the blockchain. After receiving commit messages, validators insert the block to their chains. [Yu-17]

### 4.7.8 Scaleable BFT (SBFT)

By binding all messages and tasks to a particular processor core, the parallelization scheme of SBFT enables BFT systems to scale with the number of available cores. Actors are organized in pillars and execute the replication protocol. Executed by a dedicated thread, each pillar is responsible for certain instances of consensus. Pillar numbers are kept in direct alignment with the number of cores and requests are managed at the same pillar level. [BDK14]

### 4.7.9 Federated Byzantine Agreement (FBA)

Considered as the most novel solution to the byzantine general problem, FBA introduces a list of important nodes that are trusted, maintained by each participant. When the majority of the trusted nodes agrees on a settlement, a transaction is considered settled. A transaction is only considered as settled by the trusted nodes, when the nodes they trust agree on the transaction. Finally, if the majority of the network agrees on the transaction it becomes immutable. Because of the selections of whom they can trust, made by the nodes, quorums and slices emerge. [MAZ15]

### 4.7.10 Modified Federated Byzantine Agreement (MFBA)

MFBA is a combination of FBA and PoS. Consensus is spread through overlapping nodes and takes place among quorums. Within a node users stake their coins and earn rewards on the stake. This serves as economic incentive to operate the node and also as a collateral if a node acts maliciously. [PPCC16]

### 4.7.11 VBFT (VBFT)

VBFT combines PoS, BFT and the variable random function (VRF). It can support scalability of consensus groups and guarantee the randomness and fairness of the consensus population generation through VRF. Further it ensures that state finality can be reached quickly. [Tea18b]

### 4.7.12 Stellar Consensus (SCP)

As a variant of FBA, SCP uses the same notion of quorums and quorum slices, instead of trusting the whole network. To achieve a consensus, SCP relies on a set of validator nodes. To reach consensus a set of nodes, called quorum, is needed. Quorum slices a subset of these quorums and can help to convince nodes about an agreement. In order to achieve broader consensus and finality quorums intersections are required. [MAZ15]

### 4.7.13 Ripple Consensus Protocol (RCP)

To maintain its ledger RCP relies on a trusted set of validating nodes. There are two forms of a ledger: the last-closed ledger and the open ledger. Transactions that do not acquire validation votes are discarded. Transactions with over 80% votes from the trusted nodes are considered valid and added to the last-closed ledger of Ripple network. Unverified transactions are kept in the open ledger until they meet 80% verification target. [Rip17]

### 4.7.14 Tendermint (TEND)

In Tendermint, all transactions are first broadcasted to a group of validators, which have some stake locked in the system. The validator nodes vote in three steps on the valid transactions. These steps are prevote, precommit and commit. 2 of 3 validator nodes have to sign a block to be committed. Block proposer is chosen in a round-robin fashion, with a proportion to their voting power, i.e. Stake. [Kwo14]

### 4.7.15 Sumeragi (SUM)

In SUM the nodes are divided into two sets and only nodes from set A take part in consensus, which is performed on every transaction. The transaction is verified and then ordered, signed and broadcasted by a lead validating peer. The remaining validating peers validate the signature of the transaction along with the contents and temporarily update the ledger. The merkle root and hash of the transactions content is signed and the finite ordered list of transactions is broadcasted. Until the roots match, the nodes keep sharing the valid parts of Merkle tree. [Tea17b]

### 4.7.16 Hydrachain (HC)

HC adds support for creating permissioned blockchains to the Ethereum platform. HC was inspired by Tendermint and is a BFT protocol that relies on a set of validators. These validators form quorums and validate the order of transactions. From the set of validators a block proposer is chosen randomly. Consensus is achieved via one or more rounds on the proposed block and a new round can only be started once more than 2/3 of the nodes have voted on the previous round. [Tea15a]

### 4.7.17 Sieve (SIE)

Sieve was implemented as a part of Hyperledger Fabric by the IBM Research group. Sieve executes the processes related to non-deterministic operations such as smart contracts and then compares the results. Processes are filtered out if they are detected to create divergence. If divergence is found among too many processes, the whole operation is sieved out of the sequence. [CSV17]

## 4.8 Directed Acyclic Graph (DAG)

The Directed Acyclic Graph or DAG, is another form of DLT (Distributed ledger technology). Some consider it to be a rival technology to blockchain, others an enabler. In this thesis I will focus on the blockchain technology. So the DAG is only mentioned here to complete the picture. Examples for DAG consensus algorithms are Tangle, Hashgrap, Block-lattic, SPECTR or PHANTO.

# CHAPTER 5

# Analysis of Smart Contract Platforms

## 5.1 Short Description

In this chapter the smart contract platforms that are used for the further detailed comparison of the consensus mechanisms are shortly described. In the figures 5.1, 5.2 and 5.3 you get an overview about the platforms and their characteristics that are described here.

The figures 5.1, 5.2 and 5.3 show the table of characteristics of the 21 smart contract platforms that were analysed. The characteristics that were used for this analysis are:

- NAME:

  Name of the platform

- IS LIVE:

  Indicates if the platform is already launched

- LAUNCH:

  Date of the launch

- TYPE:

  Shows if the platform can be publicly used, only for private purposes or even for both forms

- CONSENSUS:

  The used consensus mechanism

- CLUSTER:

  Shows the clusters that were described in chapter 4

- LANGUAGE:

  The language of the platform

- TURING:

  Shows if the platform supports a Turing-complete language

- TOKEN:

  Indicates if tokens are supported

- DEX:

  Shows if decentralized exchanges are supported

- PRIVACY:

  Shows if privacy is supported

- SIDECHAIN:

  Shows if sidechains are allowed

- ATOMIC SWAP:

  Shows if atomic swap is supported

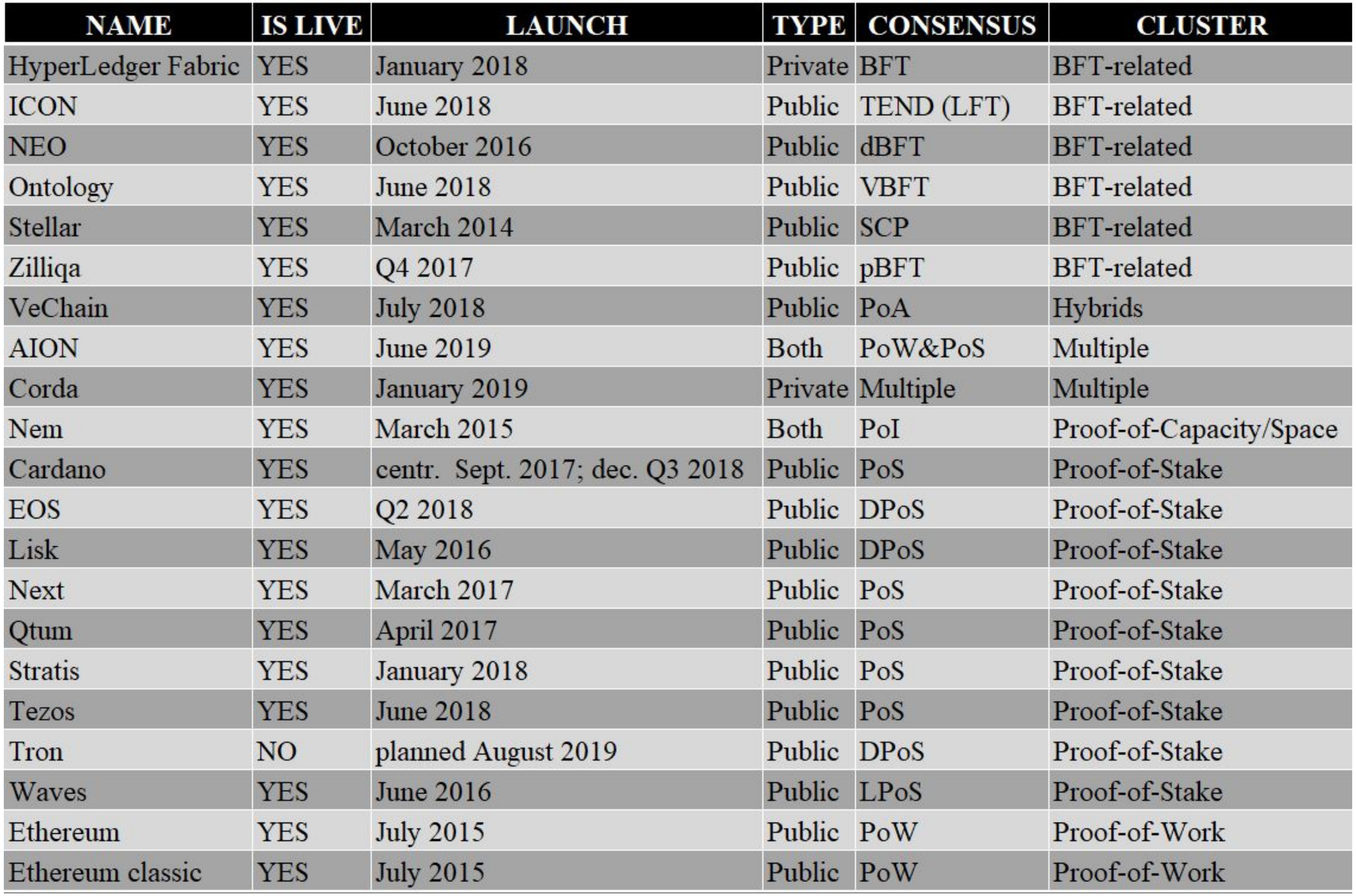
- BLOCKTIME:

  The time for creating new blocks

- TX/S:

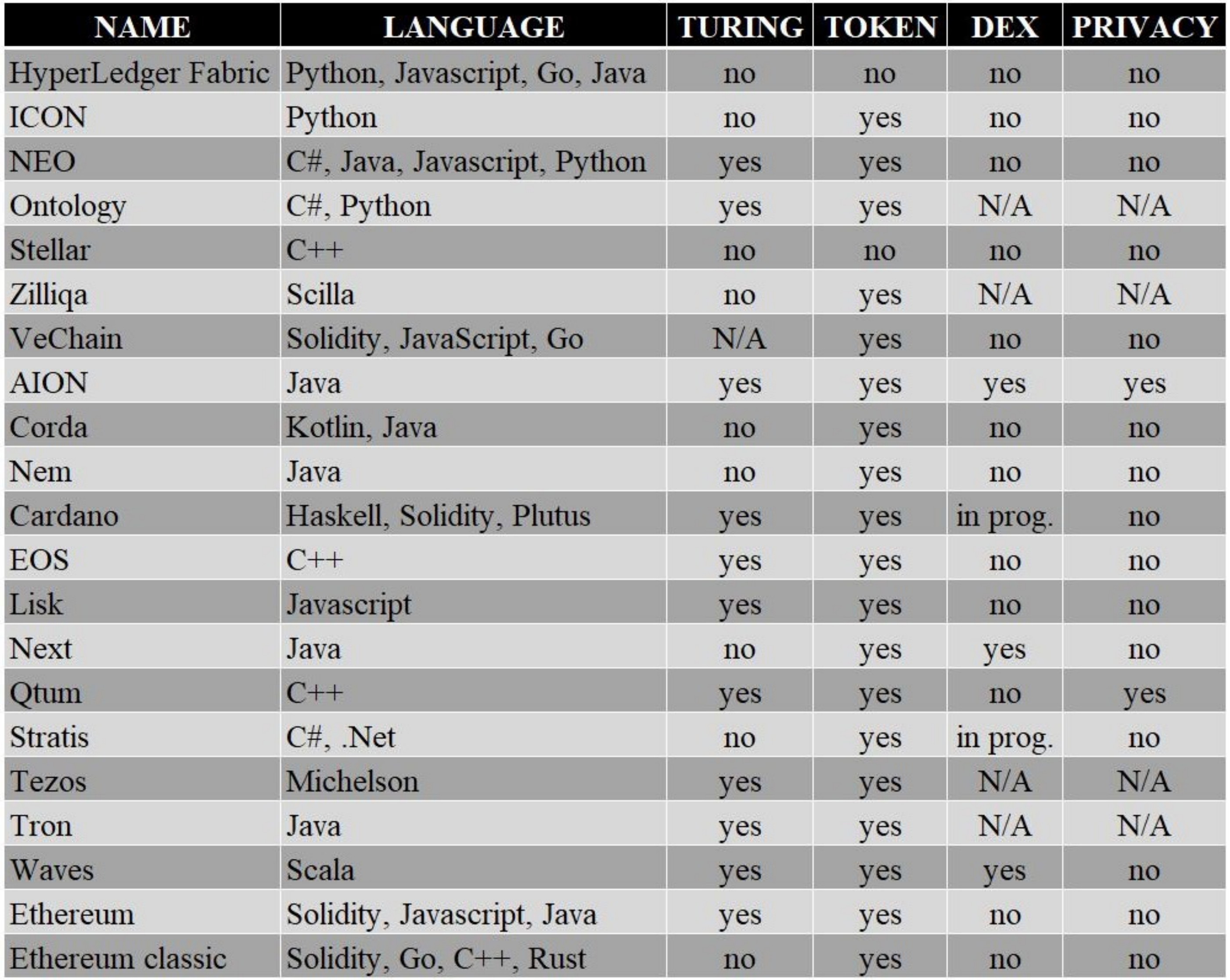
  Show how many transactions per seconds can be done

In the following sections the mentioned platforms are shortly described to give an insight of the platforms that are available especially in perspective to the used consensus mechanism.



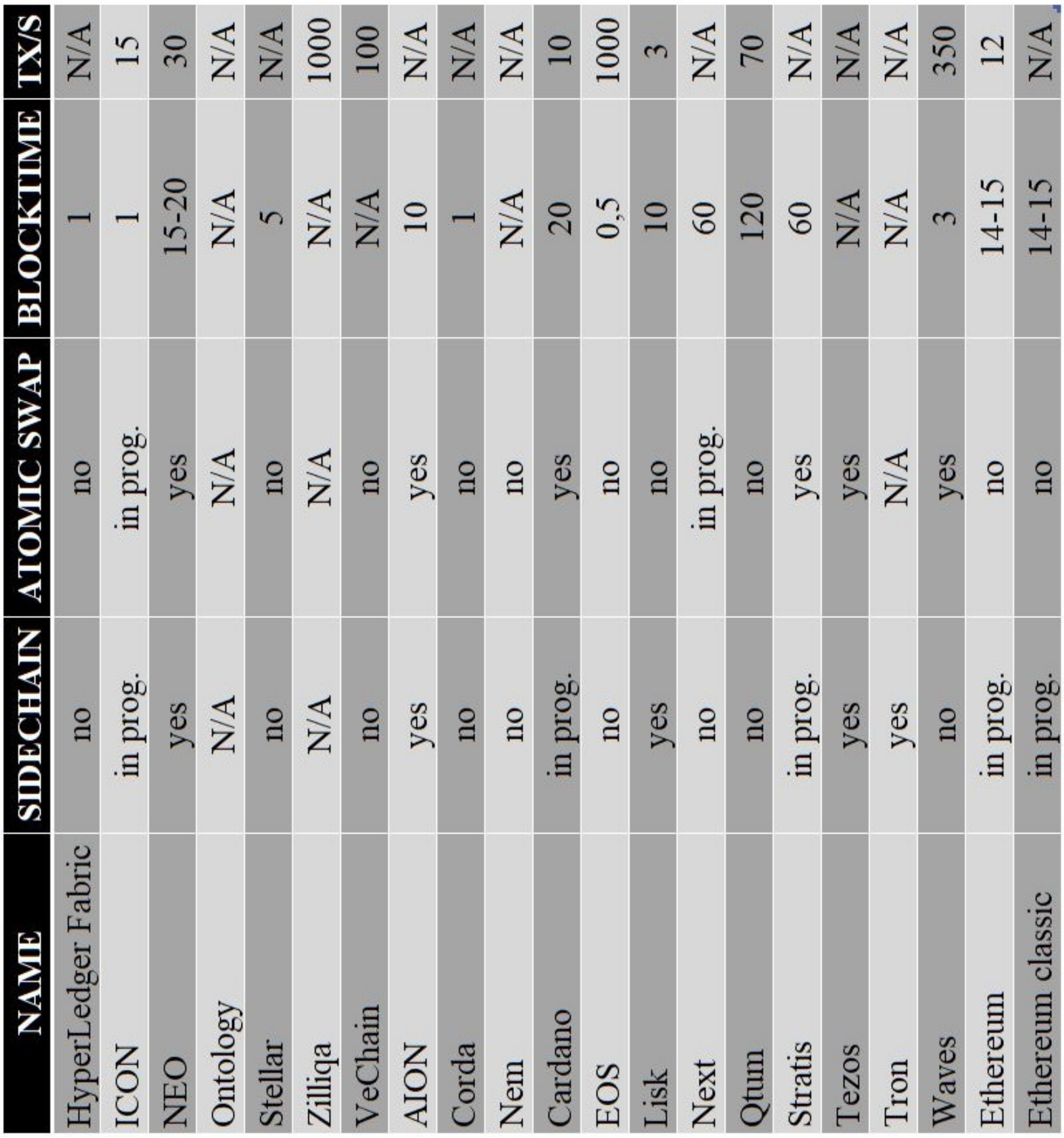
| NAME | IS LIVE | LAUNCH | TYPE | CONSENSUS | CLUSTER |
|---|---|---|---|---|---|
| HyperLedger Fabric | YES | January 2018 | Private | BFT | BFT-related |
| ICON | YES | June 2018 | Public | TEND (LFT) | BFT-related |
| NEO | YES | October 2016 | Public | dBFT | BFT-related |
| Ontology | YES | June 2018 | Public | VBFT | BFT-related |
| Stellar | YES | March 2014 | Public | SCP | BFT-related |
| Zilliqa | YES | Q4 2017 | Public | pBFT | BFT-related |
| VeChain | YES | July 2018 | Public | PoA | Hybrids |
| AION | YES | June 2019 | Both | PoW&PoS | Multiple |
| Corda | YES | January 2019 | Private | Multiple | Multiple |
| Nem | YES | March 2015 | Both | PoI | Proof-of-Capacity/Space |
| Cardano | YES | centr. Sept. 2017; dec. Q3 2018 | Public | PoS | Proof-of-Stake |
| EOS | YES | Q2 2018 | Public | DPoS | Proof-of-Stake |
| Lisk | YES | May 2016 | Public | DPoS | Proof-of-Stake |
| Next | YES | March 2017 | Public | PoS | Proof-of-Stake |
| Qtum | YES | April 2017 | Public | PoS | Proof-of-Stake |
| Stratis | YES | January 2018 | Public | PoS | Proof-of-Stake |
| Tezos | YES | June 2018 | Public | PoS | Proof-of-Stake |
| Tron | NO | planned August 2019 | Public | DPoS | Proof-of-Stake |
| Waves | YES | June 2016 | Public | LPoS | Proof-of-Stake |
| Ethereum | YES | July 2015 | Public | PoW | Proof-of-Work |
| Ethereum classic | YES | July 2015 | Public | PoW | Proof-of-Work |

Figure 5.1: smart contract platforms comparison - Part 1



| NAME | LANGUAGE | TURING | TOKEN | DEX | PRIVACY |
|---|---|---|---|---|---|
| HyperLedger Fabric | Python, Javascript, Go, Java | no | no | no | no |
| ICON | Python | no | yes | no | no |
| NEO | C#, Java, Javascript, Python | yes | yes | no | no |
| Ontology | C#, Python | yes | yes | N/A | N/A |
| Stellar | C++ | no | no | no | no |
| Zilliqa | Scilla | no | yes | N/A | N/A |
| VeChain | Solidity, JavaScript, Go | N/A | yes | no | no |
| AION | Java | yes | yes | yes | yes |
| Corda | Kotlin, Java | no | yes | no | no |
| Nem | Java | no | yes | no | no |
| Cardano | Haskell, Solidity, Plutus | yes | yes | in prog. | no |
| EOS | C++ | yes | yes | no | no |
| Lisk | Javascript | yes | yes | no | no |
| Next | Java | no | yes | yes | no |
| Qtum | C++ | yes | yes | no | yes |
| Stratis | C#, .Net | no | yes | in prog. | no |
| Tezos | Michelson | yes | yes | N/A | N/A |
| Tron | Java | yes | yes | N/A | N/A |
| Waves | Scala | yes | yes | yes | no |
| Ethereum | Solidity, Javascript, Java | yes | yes | no | no |
| Ethereum classic | Solidity, Go, C++, Rust | no | yes | no | no |

Figure 5.2: smart contract platforms comparison - Part 2



| NAME | SIDECHAIN | ATOMIC SWAP | BLOCKTIME | TX/S |
|---|---|---|---|---|
| HyperLedger Fabric | no | no | 1 | N/A |
| ICON | in prog. | in prog. | 1 | 15 |
| NEO | yes | yes | 15-20 | 30 |
| Ontology | N/A | N/A | N/A | N/A |
| Stellar | no | no | 5 | N/A |
| Zilliqa | N/A | N/A | N/A | 1000 |
| VeChain | no | no | N/A | 100 |
| AION | yes | yes | 10 | N/A |
| Corda | no | no | 1 | N/A |
| Nem | no | no | N/A | N/A |
| Cardano | in prog. | yes | 20 | 10 |
| EOS | no | no | 0,5 | 1000 |
| Lisk | yes | no | 10 | 3 |
| Next | no | in prog. | 60 | N/A |
| Qtum | no | no | 120 | 70 |
| Stratis | in prog. | yes | 60 | N/A |
| Tezos | yes | yes | N/A | N/A |
| Tron | yes | N/A | N/A | N/A |
| Waves | no | yes | 3 | 350 |
| Ethereum | in prog. | no | 14-15 | 12 |
| Ethereum classic | in prog. | no | 14-15 | N/A |

Figure 5.3: smart contract platforms comparison - Part 3

## 5.2 AION

AION provides a trustless mechanism for cross-chain interoperability and it is designed to support custom blockchain architectures. AION is the world's first dedicated public enterprise blockchain that introduces a new paradigm of security and fair representative crypto-economic incentives. As consensus mechanism AION uses a Hybrid of PoW and PoS. [Com18]

## 5.3 Cardano

Cardano is an open-source project with a decentralized public blockchain. The goal is to develop a smart contract platform which delivers more advanced features than any protocol previously developed. Cardano is developed by a team that consists of a large global collective of expert engineers and researchers and evolved out of a scientific philosophy and a research-first driven approach. Cardano uses PoS as consensus protocol. [Car17]

## 5.4 Corda

Built on proven, familiar technologies, Corda is the blockchain for Java developers. Further, Corda uses enterprise technologies such as relational databases and AMQP. The architecture focuses on a lower cost of ownership, greater resilience and easier deployments. In Corda, multiple consensus algorithms are supported. [Hea16]

## 5.5 EOS

EOS is proposed as a performance-based and self-governing blockchain. It provides an operating system-like construct for building large-scale distributed applications. EOS.IO applications are designed around event (aka action) handlers, that respond to user actions. The sender, the receiver or the currency application itself can process or potentially reject this event. The developers of applications can decide what actions users can take and which handlers may or must be called in response to those events. Due to its high scalability, zero transaction fee and C++ as smart contract language, it is considered as a good competitor to Ethereum. As dPoS is used as consensus mechanism and other optimizations such as parallel execution and partial evaluation are implemented, transactions are fast. [EOS18]

## 5.6 Ethereum

Ethereum provides a decentralized blockchain with a built-in Turing-complete programming language. The platform is public and open-source and allows anyone to build and deploy smart contracts and decentralized applications. Programmers can create their own arbitrary rules for ownership, transaction formats and state transition functions. Ethereum provides a set of integral features, like user authentication, fully customizable payment logic, DDOS resistant up-time, no-fuss storage and interoperability. Ethereum still uses PoW as consensus mechanism, but wants to change to a form of PoS in near future. [Eth14]

## 5.7 Ethereum Classic

Ethereum Classic is a secure, censorproof, reliable, public trustless and decentralized blockchain platform featuring smart contract functionality. It uses the Ethereum Virtual Machine to execute Turing-complete smart contracts. Ethereum Classic came into existence because some members rejected the hard fork of Ethereum after the DAO attacks in 2016. Ethereum Classis uses PoW as consensus mechanism. [Bec17]

## 5.8 Hyperledger

Hyperledger was started by the Linux Foundation in 2015 and is an umbrella project of open source blockchains and related tools. It is meant to support the collaborative development of blockchain-based distributed ledgers. Hyperledger can use different consensus mechanisms, the default one is BFT. [Tea17b]

## 5.9 ICON

Icon focuses on interoperability between enterprise and public blockchains. It is a scalable smart contract enabled blockchain platform with an innovative consensus mechanism and decentralized governance structure. A smart contract in ICON is written in python and is called SCORE. An enhanced BFT-based algorithm known as Loop Fault Tolerance (LFT), a governance mechanism known as Delegated Proof of Contribution (DPoC) to reward those that contribute most to the ICON Network and a concept known as "Virtual Step" allowing SCORE operators to cover user transaction fees, are used. [Ico17]

## 5.10 Lisk

Lisk is a public blockchain with its own currency. It supports the execution of Turing-complete smart contracts written in JavaScript or in Node.js. Because the determinism of executions is not ensured by the language, programmers must take care of it. Each smart contract is executed on a separated blockchain, beside the main blockchain. To avoid double spending, users can deposit or withdraw currencies from a contractor to the main chain. Contract owners can customise their blockchain before deploying their contracts, e.g. choosing which nodes can participate to the dPoS consensus mechanism. [Lis17]

## 5.11 NEM

The New Economy Movement, short NEM, platform is written in Java and implemented as a dual-layer blockchain. It is the world's first Smart Asset blockchain, supports multiple ledgers and delivers a platform for management of almost any kind of asset: currencies, supply chains, notarizations, ownership records and more. The nodes on the NEM blockchain process API calls. NEM also has an encrypted P2P messaging system, multisignature accounts, and an Eigentrust++ reputation system. Its native currency XEM, is 'harvested' using the PoI (Proof of Importance) algorithm. [Lai18]

## 5.12 NEO

NEO states itself as a distributed network for the smart economy. Its mission is to improve and revolutionize the way e-commerce is done. The NEO blockchain will implement P2P networking, digital certificates, cross-chain interoperability, Superconducting Transactions and dBFT consensus technologies to effectively execute the management of smart assets within a safe and legally binding framework. [Tea15b]

## 5.13 Next (NXT)

NXT includes many core-level features, such as a Decentralized Asset Exchange, Marketplace and Voting system, all in additional to the NXT digital currency itself. NXT is easy-to-use, permissionless and gives its users complete freedom in many ways. Template smart contracts in NXT were created in a Turing-complete scripting layer, but NXT smart contracts are not Turing-complete themselves. Users can choose templates and adjust them for their use-case. Smart contracts from these templates should cover most business applications. NXT uses PoS to get consensus. [NXT14]

## 5.14 Ontology

Ontology is a blockchain that builds the infrastructure for a peer-to-peer trust network which is cross-chain, cross-system, cross-industry, cross-application and cross-device. It combines distributed identity system, distributed data exchange, distributed data collaboration, distributed procedure protocols, distributed communities, distributed attestation and various industry-specific modules. VBFT is used as consensus mechanism. [Ont18]

## 5.15 Qtum

Qtum is a decentralized platform and currency. It claims to combine a version of the bitcoin core with the Ethereum Virtual Machine and was created with the purpose of providing a platform for the development of DApps. The platform functions on a unique PoS algorithm. [DMEN17]

## 5.16 Stellar

Stellar uses a decentralized protocol for digital currency to fiat currency transfers which allows cross-border transactions between any pair of currencies. The Stellar protocol is open-source and is supported by the non-profit Stellar Development Foundation. Stellar uses it own consensus protocol, the Stellar Consensus Protocol (SCP), which is a construction for FBA. [MAZ15]

## 5.17 Stratis

Stratis smart contracts use the .NET core to be executed. And, they are using the full C# package supplied by Microsoft. Because smart contracts must execute deterministically, they cannot use all capabilities of the C# language or all the .NET Core libraries. Stratis includes also a validation tool that checks self-written smart contracts for any non-deterministic elements. They also use the concept of gas, like Ethereum does, and support PoS as consensus algorithms. [Chr17]

## 5.18 Tezos

With a new smart contracting language designed specifically to facilitate formal verification of on-chain code, Tezos plans to greatly improve security. This new language, called Michelson is a low level, stack-based, Turing-complete programming language that is directly interpreted by the Tezos virtual machine. Furthermore it includes high-level constructs such as maps, sets, lambdas, cryptographic primitives and contract-specific operations to make it easier for humans to read and write. To simplify the construction of correctness proofs and eliminate several types of vulnerabilities that have afflicted Solidity contracts, it is purely functional, strongly typed and statically type-checked. Tezos uses PoS as consensus mechanism. [Goo16]

## 5.19 TRON

As one of the largest blockchain-based operating system in the world, the TRON protocol offers public blockchain support. Characteristics of this platform are a high throughput, high scalability and high availability for all DApps in the TRON ecosystem. To get consensus TRON uses the dPoS algorithm. [TRO18]

## 5.20 VeChain

Build as a business ecosystem platform, VeChain and the VeChainThor Blockchain aim to build a trust-free platform to enable transparent information flow, efficient collaboration and high-speed value transfer. VeChain uses PoA as consensus mechanism. [VeC18]

## 5.21 Waves

Waves is built in Scala, is non-inflationary and there are no block rewards. Waves chooses a miner in advance and allow them to add new transactions to the next block as soon as they arrive. Therefore, blocks are confirmed in around 1 minute and transactions are added as quickly as network latency allows. Smart contracts in Waves are implemented with simple account controls such as balance freezing and multi-sig transactions. Turing-complete transactions, as in Ethereum, will follow, but these will have fixed fees and the resources required to execute them will be known in advance. This should avoid some of the potential attack vectors and edge cases that have affected Ethereum. Waves decentralized exchange uses a combination of centralized Matcher nodes with blockchain settlement to allow highly secure and private trading, in real-time, without having to wait for the next block for orders to execute. Waves uses a form of LPoS for consensus. [Wav17]

## 5.22 Zilliqa

ZILLIQA introduces the idea of sharding and proposes an innovative special-purpose smart contract language and execution environment. Sharding means, that the mining network is divided into smaller shards, each capable of processing transactions parallel. ZILLIQA further provides a large scale and highly efficient computation platform. The smart contract language follows a dataflow programming style which makes it ideal for running large-scale computations that can be easily parallelized. It uses PoW for miner verification to prevent Sybil attacks and Practical Byzantine Fault Tolerance for consensus. [Tea17c]

CHAPTER 6

# Detailed Comparison of Consensus Mechanisms of Smart Contract Platforms

## 6.1 Description

In this chapter the consensus mechanisms that are used in the big smart contract platforms are technically analysed and compared. An in-depth analysis of the properties will show how these properties influence the choice of consensus mechanisms in smart contract platforms. In the figures 6.1 and 6.2 you can see the consensus mechanisms and the characteristic properties that are identified in the criteria catalogue and are analysed here.

The consensus mechanisms that were identified in the previous chapter are compared in the following sections:

- Proof of Stake (PoS)
- Delegated Proof of Stake (dPoS)
- Proof of Work (PoW)
- Proof of Authority (PoAUTH)
- Practical Byzantine Fault Tolerance (pBFT)

- Delegated Byzantine Fault Tolerance (dBFT)
- VBFT

The properties taken into account for the comparison are blockchain type, transaction finality, transaction rate, scalability of peer network, energy saving, token needed, cost of participation, trust model, adversary tolerance, node identity management, verification speed, throughput, latency and validation.

## 6.2 Technical Analysis

The properties mentioned above are key indicators for the platforms to choose a consensus mechanism. To understand why they are important, the following description will give an overview about them.

The figures 6.1 and 6.2 show the comparison of the 7 consensus mechanisms that were analysed in detail in perspective to these properties. The criteria that were used for this analysis and are shown in the figures are named and described in the following sections.

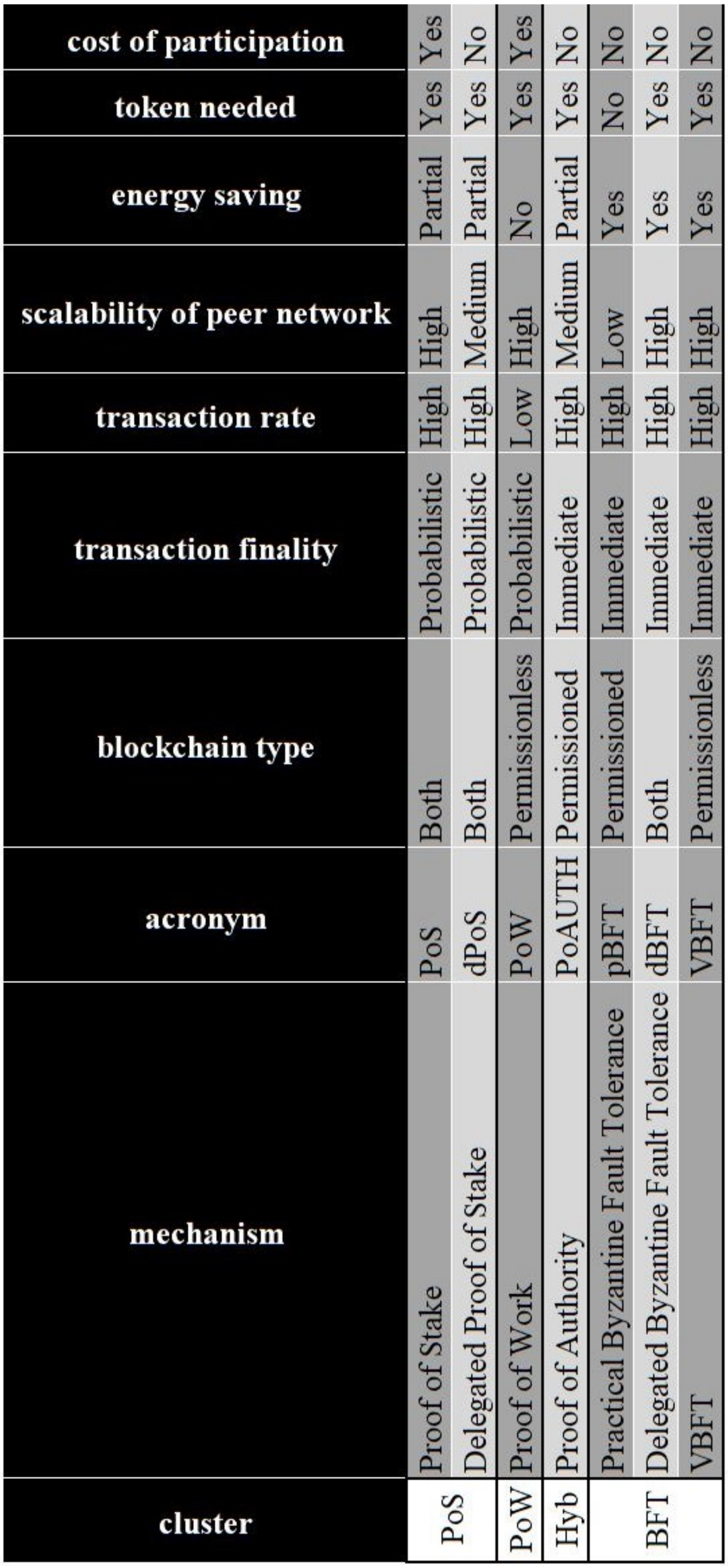
### 6.2.1 blockchain type

The two main types of blockchain platforms are permissioned and permissionless. Publicly available platforms, like Ethereum, are permissionless. In this type, any node can conduct transactions and take part in the consensus process. Private platforms, like Hyperledger, use permissioned systems. They are aimed at consortiums where participation is close-ended. This property indicates the type of blockchain in which the consensus mechanism can be used. [Bal17]

Where PoW and VBFT only support permissionless blockchains, PoAUTH and pBFT are for permissioned ones. PoS, dPos and dBFT can be used in both forms and are therefore universally applicable.

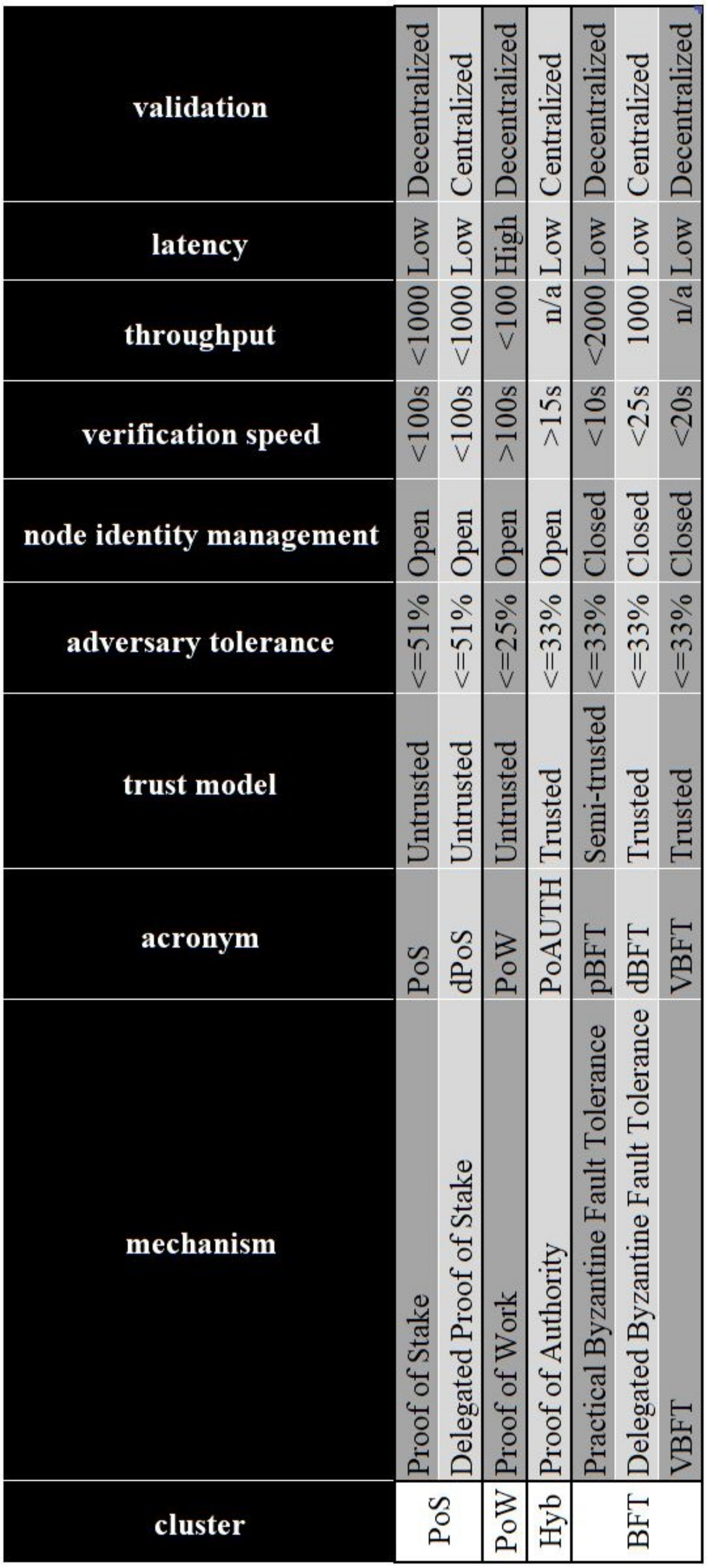
### 6.2.2 transaction finality

A transaction once added to a block in the blockchain can be considered final immediately or probabilistically. Due to their model of leader election in combination with network latencies, consensus mechanisms, like PoW, PoS and dPoS carry the risk of multiple blocks being mined at the same time. This can generate temporary forks where previously confirmed blocks get rejected. So clients have to wait much longer for transactions to be confirmed and finalized, which leads to a probabilistic transaction finality. [Bal17]



| cluster | mechanism | acronym | blockchain type | transaction finality | transaction rate | scalability of peer network | energy saving | token needed | cost of participation |
|---|---|---|---|---|---|---|---|---|---|
| PoS | Proof of Stake | PoS | Both | Probabilistic | High | High | Partial | Yes | Yes |
| | Delegated Proof of Stake | dPoS | Both | Probabilistic | High | Medium | Partial | Yes | No |
| PoW | Proof of Work | PoW | Permissionless | Probabilistic | Low | High | No | Yes | Yes |
| Hyb | Proof of Authority | PoAUTH | Permissioned | Immediate | High | Medium | Partial | Yes | No |
| BFT | Practical Byzantine Fault Tolerance | pBFT | Permissioned | Immediate | High | Low | Yes | No | No |
| | Delegated Byzantine Fault Tolerance | dBFT | Both | Immediate | High | High | Yes | Yes | No |
| | VBFT | VBFT | Permissionless | Immediate | High | High | Yes | Yes | No |

Figure 6.1: detailed comparison of consensus mechanisms - Part 1



| cluster | mechanism | acronym | trust model | adversary tolerance | node identity management | verification speed | throughput | latency | validation |
|---|---|---|---|---|---|---|---|---|---|
| PoS | Proof of Stake | PoS | Untrusted | <=51% | Open | <100s | <1000 | Low | Decentralized |
| | Delegated Proof of Stake | dPoS | Untrusted | <=51% | Open | <100s | <1000 | Low | Centralized |
| PoW | Proof of Work | PoW | Untrusted | <=25% | Open | >100s | <100 | High | Decentralized |
| Hyb | Proof of Authority | PoAUTH | Trusted | <=33% | Open | >15s | n/a | Low | Centralized |
| BFT | Practical Byzantine Fault Tolerance | pBFT | Semi-trusted | <=33% | Closed | <10s | <2000 | Low | Decentralized |
| | Delegated Byzantine Fault Tolerance | dBFT | Trusted | <=33% | Closed | <25s | 1000 | Low | Centralized |
| | VBFT | VBFT | Trusted | <=33% | Closed | <20s | n/a | Low | Decentralized |

Figure 6.2: detailed comparison of consensus mechanisms - Part 2

The BFT and PoAUTH models with immediate transaction finality confirm a transaction at the moment it is included in the block. Once it is included it will not be rolled back.

### 6.2.3 transaction rate

The transaction rate shows how fast transactions can be confirmed and consensus can be reached. The transaction rate is usually higher in platforms with immediate transaction finality and a low latency. [Bal17]

The PoW approach has to spend much time solving the cryptographic puzzle and therefore provides a low transaction rate. BFT based approaches, PoAUTH, PoS and dPoS can confirm transactions fast and they are expected to support high transaction rates.

### 6.2.4 scalability of peer network

Scalability is one of the most important problems in blockchains and has been the focus of both, industry practitioners and academic researchers, since Bitcoin was born.

Generally and for most computer systems, "scalability" refers to the system's capability to handle a growing amount of work [Bon00]. In the domain of blockchain, the word "scalability" has a much broader range of meanings. Here we talk about the "scalability of peer network", which indicates the ability to reach consensus when the number of peering nodes is increasing.

In poorly scalable networks, like the ones using pBFT, it is recommended to keep the number of peers less than 20. Especially for networks with a large amount of contracts, a high scalability is very important. [Bal17] This high scalability of the peer network comes with PoS, PoW, dBFT and VBFT. dPoS and PoAUTH support medium scalability of the peer network.

### 6.2.5 energy saving

Electricity spent to reach consensus is increasing constantly for the last years. This is one of the biggest points of criticism when you are talking about consensus mechanisms. In consensus mechanisms like PoW, where "mining" is done and rewarded, the amount of electricity required to process has reached an immense scale and is therefore not energy saving. [ZXD+17]

In PoAUTH and PoS based algorithms, there are still calculations that cost energy, but it is reduced due to the limited search space. BFT-related consensus algorithms save the most energy, as there is no mining in the consensus process.

### 6.2.6 token needed

The existence of a cryptographic token is required by design for some consensus models. There the token is needed for the consensus mechanism to function. [Bal17]

This can be applied to PoS, dPoS, PoW, PoAUTH, dBFT and VBFT. pBFT does not require a token for consensus to function.

### 6.2.7 cost of participation

To participate in the consensus some models require some resource to be spent. PoW needs an external resource, expending energy, to participate. In PoS nodes are required to buy some initial tokens to generate a security deposit for declaring interest and bonding with the platform. [Bal17]

The other consensus mechanisms beside PoW and PoS, which are compared here, do not need a resource to participate in the consensus.

### 6.2.8 trust model

The trust model indicates whether nodes participating in the consensus have to be known or trusted.

In untrusted models, like PoS, dPoS and PoW, the mechanism to reach consensus is based on computational work or security deposits. Consensus decisions will be intact as long as more than 25-50% of the network is not adversarial. In trusted models, like PoAUTH, dBFT and VBFT, the peering nodes have to be known to participate in the consensus decision. The consensus process will not be affected as long as more than 30% of the nodes are not compromised. Semi-trusted models, like pBFT, are often used for consortium networks. [Bal17]

### 6.2.9 adversary tolerance

This is the fragment of the network which can be compromised while the consensus is still intact. [Bal17]

In PoW the total computational (hashing) power, controlled by the adversary, matters. PoW has the worst adversary tolerance with under 25% [EE16]. The BFT related models and PoAUTH tolerate less than 1/3 corrupted nodes [DLS88]. PoS and dPoS have the highest adversary tolerance. The threshold in these two mechanism lies on 51%.

### 6.2.10 node identity management

Node identity management is one of the most fundamental differences in the different consensus protocols.

While PoS, PoW and Hybrid forms feature an entirely decentralized, open, identity management, BFT-based protocols typically use the closed variant. Open node identity management means that everybody can participate in the protocol, knowing only a single peer to start with. In contrast, the closed approach typically requires every node to know the entire set of its peer nodes participating in consensus. This can be helpful for smart contracts where the identity of nodes have to be known for legal and compliance reasons. [Vuk16]

### 6.2.11 verification speed, throughput and latency

The three properties verification speed, throughput and latency can be summarized under the term performance, as they depend on each other. If you want to boost the throughput by increasing the block size, the latency will also increase. So, the performance of a blockchain network ultimately depends on the consensus mechanism that is chosen. Verification speed means the time a transaction needs to be verified. Throughput is expressed in terms of blocks appended to the blockchain per second. Latency refers to the time a user has to wait till his transaction is processed. [Bal17]

PoW comes with slow verification speed, a low throughout and high latency. The PoS based protocols are mid-fast in the verification phase, with high throughput and a low latency. BFT related protocols are typically very fast in verifying transactions, have high throughput and low latency.

### 6.2.12 validation

Unaffected by the blockchain type, the validation can be performed in a centralized or decentralized way.

Centralized validation, as used by dPoS, PoAUTH and dBFT, is performed by one particular or several nodes that are responsible to validate data and avoid to much overhead and allow low latency. Decentralization is characterized by the fact that all nodes in the network are able to validate data. PoS, PoW, pBFT and VBFT use this validation type. Typically, permissionless models support decentralized validation whereas permissioned ones use centralized validation. But every other combination or hybrid form is possible. [Bal17]

## 6.3 Challenges of Ethereum in Switching its Consensus Mechanism

As mentioned in the section 1.3, the largest smart contract platform, Ethereum, has been looking to shift their consensus protocol from PoW to PoS since years. [BS18]

More precisely, the founder of Ethereum, Vitalin Buterik, mentioned already 2014 that "in the long term, it seems like either pure proof of stake or hybrid PoW/PoS are the way that blockchains are going to go." [But14]

The wish to change arised mainly due to the increasing energy costs it takes to maintain the PoW-based network, the centralization risks and security issues against different types of 51% attacks. The high energy consumption is due to the mining process that is done in PoW. Ethereum validators currently have to spend a lot of electricity to run the nodes with the Proof of Work consensus protocol. In addition with the ETH price and Gas costs, that need to be spent to execute a transaction on the Ethereum network, in some cases it is not worth it to run transactions. Vitalik Buterin noticed this problem and determined PoS as a viable alternative. Without PoW there is no need for miners and the high hardware and energy resources it takes to mine. Therefore the PoS model would put an end to mining on the Ethereum blockchain. [Blo19]

The development of Ethereum was planned over four different stages. Each stage included "hard forks", that change the functionality of the network, and also introduced more features and fixes of problems. In blockchains, a fork means a radical change in transaction validation. After a hard fork, old-ruled software needs to be upgraded to see blocks produced in accordance to new rules as valid. Changes of hard forks are not backward compatible.

To lay the groundwork to the switch from PoW to PoS, the Ethereum Team has planned major milestones since 2016. Originally, the Serenity Update with "pure Proof of Stake" was planned for 2019. To do this switch, the Constantinople Update should lay the pieces to allow the transition of the consensus mechanism. It was originally supposed to include a Hybrid PoW/PoS model. Shortly before the release of this version should have been done, security issues were found and so the update was cancelled. Beside other issues, there was a huge problem with centralisation because of the planned amount of Ether that would have been necessary to take part in the PoS consensus. As the developers decided to scratch the hybrid idea, a delay in the roadmap occurred and the Serenity update was forced to be pushed back. [Dex19]

The following hard forks were done or are planned:

- Frontier

  Initial development stage of Ethereum

- Ice Age

  Introduced an exponential difficulty increase for Proof of Work. The aim was to motivate the transition to Proof of Stake.

- Homestead

  The second state of Ethereum

- DAO

  Reimbursed victims of the DAO hack and caused the split of the network into Ethereum and Ethereum Classic.

- Tangerine Whistle

  Changed the gas calculation for certain I/O heavy operations. Cleared the accumulated state after a denial-of-service (DoS) attack.

- Spurious Dragon

  Addressed more DoS attack vectors and another state clearing. Also, a replay attack protection mechanism.

- Metropolis Byzantium

  The third stage of Ethereum development.

- Constantinople

  Included changes that fix security issues codenamed Petersburg.

- Istanbul

  Included security fixes and incentives to move away from Proof of Work to Proof of Stake algorithm.

- Serenity — Ethereum 2.0

  Serenity will be the next stage of Ethereum development and includes the change to Proof of Stake. It will break into different sub-stages:

  Phase 0: Beacon Chain

  Phase 1: Shard Chains

  Phase 2: eWASM (New Ethereum Virtual Machine)

  Phase 3: Continued Improvement

### 6.3.1 Ethereum 2.0

As stated Ethereum 2.0 will bring several new features to the Ethereum network and is not backwards compatible. In Ethereum 2.0 mining will be replaced by staking. So this hard fork will put an end to Ethereum's old Proof of Work mechanism.

The overall primary design goals of Ethereum 2.0 are [Eth19]:

- Decentralisation

  Allow for more low-end devices to participate in the network as validators.

- Resilience

  The network should still be live even when lots of nodes go offline or through major network partitions.

- Security

  Utilize crypto and design techniques that allow for the massive participation of validators in total and per unit time.

- Simplicity

  Minimise complexity, even at the cost of some efficiency losses.

- Longevity

  Make components either quantum secure or easily swappable for quantum secure counterparts when available. This will mean preparing the network for a future where Quantum computing is fully accessible.

As the changes of the consensus mechanisms will happen in Phase 0, I will only attend to this phase.

#### Beacon Chain - Phase

This is the phase where the Proof of Stake consensus mechanism will be rolled out. The designers intend the beacon chain to become the hub of Ethereum 2.0's ecosystem.

Once deployed, the Beacon Chain will introduce the new Proof of Stake-based algorithm called Casper and it will be a separated blockchain from the main Ethereum blockchain which still uses Proof of Work. This early iteration of the beacon chain is designed to be as simple as possible, which is why Phase 0 will not support smart contracts, accounts, asset transfers and will not include any shards. [Eth19]

### 6.3.2 Challenges

Beside the technical issues and problems that arise with the switch, Ethereum suffers from other more human issues that make the change difficulty. One of these is to keep the users from changing to other platforms. Many miners have invested a lot of money in hardware and they may likely move to a different blockchain and mine there. Another point of criticism is that a lot of smart contracts will break with the changes that will be done. In September 2019 the CEO of the Aragon platform said that 680 smart contracts will break with the next update of Ethereum. [Coi14] So the challenges for Ethereum 2.0 are the community's uncertainty, unclear details of the transition and its future design.

# CHAPTER 7

# Critical Reflection

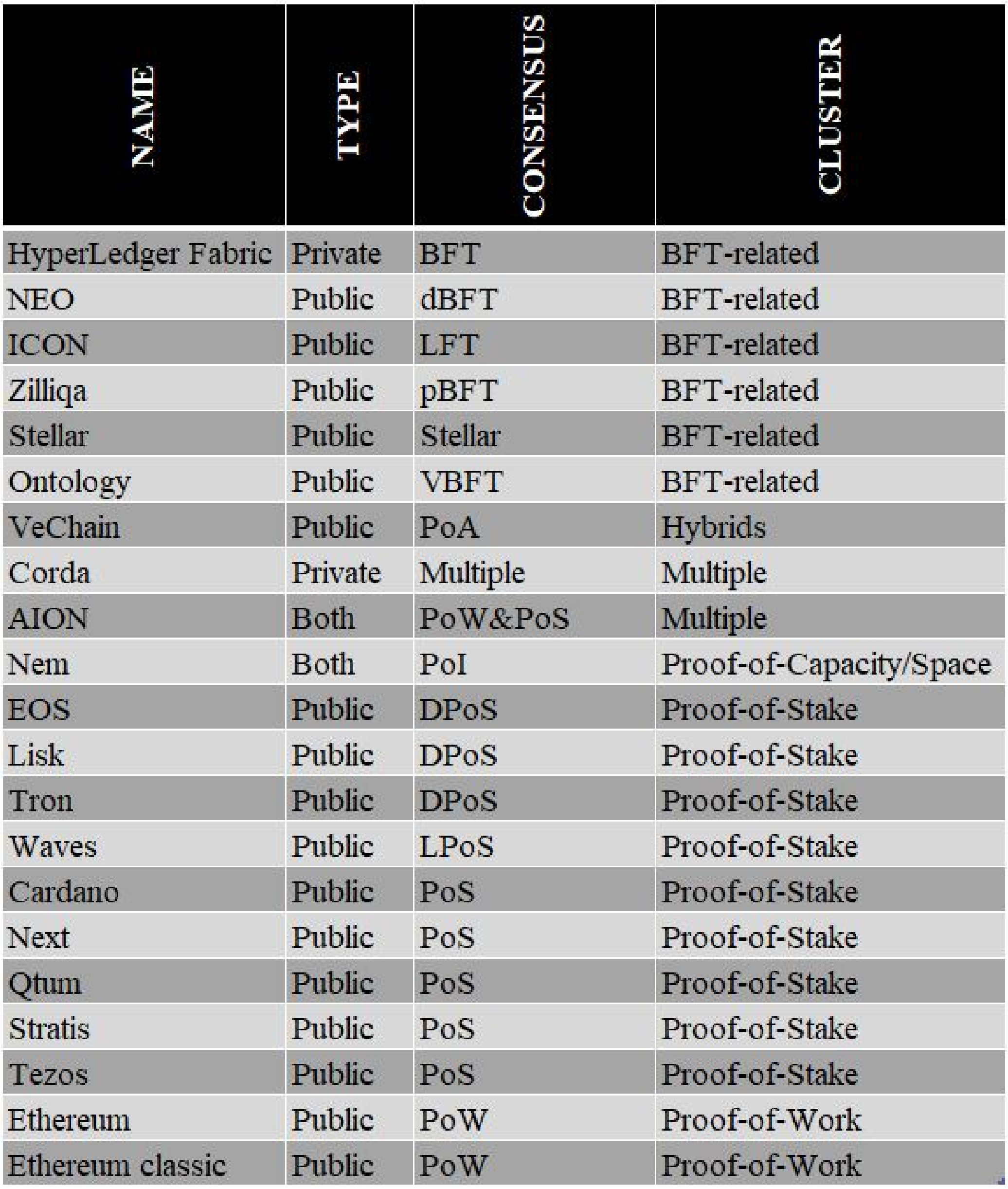
## 7.1 Conclusion

With the blockchain technology all transactions that have ever occurred in a network can be recorded on a distributed database. One of the main features is that transactions can be done without the need of a trusted third party. To get to an agreement the parties have to reach a consensus. This process is defined by the used consensus mechanism of the blockchain. With smart contract platforms on blockchains, agreements between untrusted parties can be facilitated, executed and enforced.

The aim of this work was to analyse these consensus mechanisms especially in perspective to the smart contract platforms. By doing so, I was able to identify 62 consensus mechanisms and could create 6 clusters to group them. After that 21 smart contract platforms could be identified and compared especially in perspective to their used consensus mechanisms. For this comparison of the platforms a criteria catalogue was created and used. This led to the detailed comparison where another criteria catalogue was created and used for the analysis of the consensus mechanisms that were used on the smart contract platforms.

As you can see in figure 7.1, the chosen category to get consensus is mainly BFT-related or in the Proof-of-Stake cluster. Actually there are only two platforms, Ethereum and Ethereum Classic, that still use PoW.

Ethereum Classic will soon be the only smart contract platform that remains on PoW, as Ethereum's change to PoS will happen soon. This long announced wish to change the consensus mechanism arised mainly due to the increasing cost of energy it takes to



| NAME | TYPE | CONSENSUS | CLUSTER |
|---|---|---|---|
| HyperLedger Fabric | Private | BFT | BFT-related |
| NEO | Public | dBFT | BFT-related |
| ICON | Public | LFT | BFT-related |
| Zilliqa | Public | pBFT | BFT-related |
| Stellar | Public | Stellar | BFT-related |
| Ontology | Public | VBFT | BFT-related |
| VeChain | Public | PoA | Hybrids |
| Corda | Private | Multiple | Multiple |
| AION | Both | PoW&PoS | Multiple |
| Nem | Both | PoI | Proof-of-Capacity/Space |
| EOS | Public | DPoS | Proof-of-Stake |
| Lisk | Public | DPoS | Proof-of-Stake |
| Tron | Public | DPoS | Proof-of-Stake |
| Waves | Public | LPoS | Proof-of-Stake |
| Cardano | Public | PoS | Proof-of-Stake |
| Next | Public | PoS | Proof-of-Stake |
| Qtum | Public | PoS | Proof-of-Stake |
| Stratis | Public | PoS | Proof-of-Stake |
| Tezos | Public | PoS | Proof-of-Stake |
| Ethereum | Public | PoW | Proof-of-Work |
| Ethereum classic | Public | PoW | Proof-of-Work |

Figure 7.1: clustered comparison of smart contract platforms

maintain the PoW-based network. This is due to the mining process that is done in PoW. Ethereum founder Vitalik Buterin noticed this problem and determined PoS as a viable alternative to this energy consumption problem. Without PoW there is no need for miners and therefore the PoS model would put an end to mining on the Ethereum blockchain. This change would not only affect the Ethereum network. Because miners, who previously mined on the Ethereum blockchain, would probably move their hardware to a different blockchain and mine there. This move can create an influx of hashing power to alternative blockchain networks. [Blo19] A few platforms try other approaches or even allow multiple consensus mechanisms to be used.

Which consensus mechanism suits best for smart contract platforms cannot be answered in general. It depends on the desired use case of the platform. In particular the number of participants, how they can access the platform and if you desire anonymity determine which consensus mechanism fits best for your purpose. Also security and performance issues should be considered when choosing the right consensus protocol. Each of them has weaknesses that put the network at risk. However, it is worth considering consensus mechanisms only as tools for ensuring network stability. By combining protocols together, or by creating hybrid protocols, you can get significantly better results.

This paper has summarized blockchain consensus protocols, especially focusing on their properties in perspective to smart contract platforms. It further identified criteria catalogues for comparing and analysing these platforms and their consensus mechanisms. The general overview of consensus mechanisms and their properties contributes to this effort, by establishing a common ground for formal protocol reviews and more technical comparisons.

## 7.2 Comparison with Related Work

As mentioned in chapter 3 there are many papers that compare specific "new" approaches to existing ones and a few surveys that include comparisons of some consensus mechanisms.

In addition to these comparisons, my analysis focused on the detailed comparison of consensus mechanisms of smart contract platforms. Furthermore the creation of two criteria catalogues for comparing smart contract platforms and their used consensus mechanisms could be achieved. Additionally a short insight in the challenges of Ethereums plans to change their consensus mechanism could be given.

## 7.3 Discussion of Open Issues

The detailed comparison of consensus mechanisms in this work was limited to consensus mechanisms that were in use in one of the available smart contract platforms at the time of writing. Future work could further analyse how the other presented consensus algorithms could fit for smart contracts.

As we can see on the proposed version of Ethereum's Proof of Stake, especially platforms that focus on smart contracts, try to implemented specific adaptions of the classic consensus mechanisms that fit their needs. With more systems become publicly available and more widely used, it will be interesting to compare their performance through benchmarks and to observe their resilience to actual attacks or network incidents. With rising number of deployed smart contracts on the mentioned platforms, there will also be the possibility to do meaningful mathematical measurements and comparisons.

As Ethereums switch to PoS is still not fulfilled, there will also be the possibility to do further research and comparisons when the change is finally done. If this change could be done successfully it will be interesting to see, if other smart contract platforms will follow the example of Ethereum and also try to switch to more energy saving alternatives.

As the number of consensus mechanisms and smart contract platforms is still rising, there will be much potential to do further studies and comparisons in this exciting area of the blockchain technology.

CHAPTER 8

# Summary

This analysis showed an overview including a description of the consensus mechanisms that are used on blockchain platforms at the time of writing. Furthermore, smart contract platforms were introduced and analysed with the created criteria catalogue. Also the used consensus mechanisms of these platforms were analysed and compared, again with a criteria catalogue that was created. Further the challenges that arise due to the switch of the consensus mechanism in Ethereum could be pointed out.

With the knowledge of this, the research questions of chapter 1.3 can be answered.

- RQ1: Which consensus mechanisms are currently used on the different blockchain platforms and how do they work?

  At the time of writing this thesis, 62 consensus mechanisms were in use in the different blockchain platforms and analysed in chapter 4. To get a structured overview, I created clusters for the mechanisms.

  These clusters, based on the general approaches for achieving consensus, are Proof-of-Stake, Proof-of-Work, Proof-of-Capacity/Space, Proof-of-Burn, Hybrids and BFT-related algorithms.

- RQ2: Which purpose do currently used smart contract platforms have?

  Smart contract platforms that where launched or planned to launch till fall 2019 were analysed especially in perspective to the used consensus mechanisms. With the help of existing literature I created a criteria catalogue to analyse and compare the platforms.

Here 21 smart contract platforms were identified and analysed in terms of their purpose. These platforms were analysed and compared based on criteria like supported blockchain type, used consensus mechanism, used programming language, Turing-completeness, supports token, supports decentralized exchange, privacy issues, supports sidechains, supports atomic swap, needed time to create blocks and transactions per seconds in chapter 5.

- RQ3: Which properties influence the choice of consensus mechanisms in smart contract platforms?

  After the smart contract platforms were analysed, 7 consensus mechanisms that were in use on these platforms could be identified. The mechanisms that were compared are Proof of Stake (PoS), Delegated Proof of Stake (dPoS), Proof of Work (PoW), Proof of Authority (PoAUTH), Practical Byzantine Fault Tolerance (pBFT), Delegated Byzantine Fault Tolerance (dBFT) and VBFT.

  In the literature 12 properties that are important characteristics of consensus algorithms could be found and were used for the creation of a criteria catalogue. These properties were taken in account for the comparison.

  The identified properties are blockchain type, transaction finality, transaction rate, scalability of peer network, energy saving, token needed, cost of participation, trust model, adversary tolerance, node identity management, verification speed, throughput, latency and validation. The detailed analysis and comparison of the properties of consensus mechanisms used in smart contract platforms can be found in chapter 6.

- RQ4: Which influences and challenges for future smart contracts do arise with the change of Ethereum from PoW to PoS?

  The long announced wish to change the consensus mechanism arised mainly due to the increasing cost of energy it takes to maintain the PoW-based network. Without PoW there is no need for miners and therefore the PoS model would put an end to mining on the Ethereum blockchain.

  Beside technical issues and problems that arise with the switch, Ethereum has to keep the users from changing to other platforms. Many miners have invested a lot of money in hardware and they may likely move to a different blockchain and mine there. Another point of criticism is that a lot of smart contracts will break with the changes that will be done. So this change would not only affect the Ethereum network and this move may create an influx of hashing power to alternative blockchain networks.

How the switch from PoW to PoS could affect Ethereum, the challenges that arise and the influence on future smart contracts is described in section 6.3 and chapter 7.1.

# CHAPTER 9

# Appendix

## List of Figures

# Bibliography

[ABC17] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on Ethereum smart contracts (SoK). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10204 LNCS(July):164–186, 2017.

[Aly14] Eduardo E. P. Alchieri Alysson Bessani, João Sousa. State Machine Replication for the Masses with BFT-SMART. Technical report, 2014. https://www.di.fc.ul.pt/ bessani/publications/dsn14-bftsmart.pdf, Accessed: 2019-04-02.

[Ame18] Ameer Rosic. Basic Primer : Blockchain consensus protocol. https://blockgeeks.com/guides/blockchain-consensus/, 2018. Accessed: 2018-12-15.

[And04] David P. Anderson. BOINC: A system for public-resource computing and storage. *Proceedings - IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004.

[AvM17] Maher Alharby and Aad van Moorsel. Blockchain-based Smart Contracts: A Systematic Mapping Study. In *AIS, CSIT, IPPR, IPDCA - 2017*, pages 125–140, 2017.

[Bal17] Arati Baliga. Understanding Blockchain Consensus Models. Technical Report April, 2017. https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf, Accessed: 2019-03-02.

[BDG17] Juan Benet, David Dalrymple, and Nicola Greco. Proof of Replication. Technical report, 2017. https://filecoin.io/proof-of-replication.pdf, Accessed: 2019-04-09.

[BDK14] Johannes Behl, Tobias Distler, and Rüdiger Kapitza. Scalable BFT for Multi-cores: Actor-based Decomposition and Consensus-oriented Parallelization. In

*Proceedings of the 10th USENIX Conference on Hot Topics in System Dependability*, page 9, 2014. http://dl.acm.org/citation.cfm?id=2696558.2696567, Accessed: 2019-04-04.

[Bec17] Matthew Beck. Into the Ether with Ethereum ClassicThe Store-of-Value Commodity to Power the Internet of Things. Technical report, 2017. www.grayscale.co, Accessed: 2019-04-02.

[BG17] Vitalik Buterin and Virgil Griffith. Casper the Friendly Finality Gadget. Technical report, 2017. http://arxiv.org/abs/1710.09437, Accessed: 2019-03-30.

[BG18] Juan Benet and Nicola Greco. Filecoin: A Decentralized Storage Network. Technical report, 2018. https://filecoin.io/filecoin.pdf, Accessed: 2019-04-07.

[BGM16] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies Without Proof of Work. In Cormac Herley, P. van Oorschot, and Andrew Patrick, editors, *LNCS 9604 International Conference on Financial Cryptography and Data Security*, pages 142–157. Springer, 2016.

[BK18] A Begicheva and A Kofman. Fair Proof of Stake. Technical report, 2018. https://docplayer.net/135847953-Fair-proof-of-stake-a-begicheva-a-kofman-may-18-2018.html, Accessed: 2019-04-15.

[BKKJ$^+$17] Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Proof-of-personhood: Redemocratizing permissionless cryptocurrencies. In *Proceedings - 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017*, number April, pages 23–26, 2017.

[BLMR14] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of Activity. *ACM SIGMETRICS Performance Evaluation Review*, 42(3):34–37, 2014.

[Blo19] Blockbasemining.com. How the switch from PoW to PoS could affect Ethereum mining. https://blockbasemining.com/how-the-switch-from-pow-to-pos-could-affect-ethereum-mining/, 2019. Accessed: 2019-11-02.

[BMC$^+$15] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. SoK: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Proceedings - IEEE Symposium on Security* *and Privacy*, volume 2015-July, pages 104–121, Piscataway, NJ, USA, 2015. IEEE.

[Bon00] André B. Bondi. Characteristics of scalability and their impact on performance. In *Proceedings Second International Workshop on Software and Performance WOSP 2000*, pages 195–203, 2000.

[BP17] Massimo Bartoletti and Livio Pompianu. An empirical analysis of smart contracts: platforms, applications, and design patterns. In *10323 LNCS, Financial Cryptography and Data Security*, pages 494–509. Springer, 2017.

[BRSN17] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of Useful Work. *IACR Cryptology ePrint Archive*, 2017:203, 2017.

[Bru13] Jd Bruce. Purely P2P Crypto-Currency With Finite Mini-Blockchain. Technical Report May, 2013. http://cryptonite.info/files/mbc-scheme-rev2.pdf, Accessed: 2019-03-10.

[BS18] Thomas Bocek and Burkhard Stiller. Smart Contracts - Blockchains in the Wings. In Claudia Linnhoff-Popien, Ralf Schneider, and Michael Zaddach, editors, *Digital Marketplaces Unleashed*, pages 169–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[BSAB$^+$19] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consensus in the age of blockchains. In *AFT 2019 - Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, number Section 4, pages 183–198, 2019.

[But14] Vitalik Buterin. Long-Range Attacks: The Serious Problem With Adaptive Proof of Work. https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/, 2014. Accessed: 2019-12-15.

[Car17] Cardano. Cardano Whitepaper. https://www.cardano.org/en/academic-papers/$%0A, 2017. Accessed: 2019-04-02.

[Che16] Alexander Chepurnoy. Interactive Proof-of-stake. Technical Report 1, 2016. http://arxiv.org/abs/1601.00275, Accessed: 2019-04-02.

[Chr16] Chronologic. Temporal Innovation on the Blockchain. Technical report, 2016. https://chronologic.network/uploads/Chronologic_Whitepaper.pdf, Accessed: 2019-05-02.

[Chr17] Nicolas Dorier Chris Trew, Guy Brandon. Stratis White Paper. Technical report, 2017. https://stratisplatform.com/files/Stratis_Whitepaper.pdf, Accessed: 2019-05-21.

[CL02] Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems*, 20(4):398–461, 2002.

[Clo17] John Clow. A byzantine fault tolerant raft. 2017.

[Coi14] Coindesk. Long-Range Attacks: The Serious Problem With Adaptive Proof of Work. https://www.coindesk.com/ethereums-istanbul-upgrade-will-break-680-smart-contracts-on-aragon, 2014. Accessed: 2019-12-15.

[Com17] Xtrabytes Community. Xtrabytes Non-Technical Whitepaper. Technical report, 2017. https://xtrabytes.global/build/files/whitepaper.pdf, Accessed: 2019-05-01.

[Com18] AION Community. AION Network Whitepaper. Technical report, 2018. https://aion.network/media/en-aion-network-technical-introduction.pdf, Accessed: 2019-04-28.

[CSLR18] Mauro Conti, Kumar E. Sandeep, Chhagan Lal, and Sushmita Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys and Tutorials*, 20(4):3416–3452, 2018.

[CSV17] Christian Cachin, Simon Schubert, and Marko Vukolić. Non-determinism in Byzantine fault-tolerant replication. In *Leibniz International Proceedings in Informatics, LIPIcs*, volume 70, pages 24.1–24.16, 2017.

[CXS+17] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On Security Analysis of Proof-of-Elapsed-Time (PoET). In Paul Spirakis and Philippas Tsigas, editors, *Stabilization, Safety, and Security of Distributed Systems*, pages 282–297, Cham, 2017. Springer International Publishing.

[DAB+18] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain. In *CEUR Workshop Proceedings*, volume 2058, pages 1–11, 2018.

[Dex19] Shawn Dexter. Ethereum Roadmap Update [2019]: Casper & Sharding Release Date. https://www.mangoresearch.co/ethereum-roadmap-update/, 2019. Accessed: 2019-12-15.

[DLS88] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the Presence of Partial Synchrony. *Journal of the Association for Computing Machinery, Vol. 35, No. 2, April 1988, pp. 288-323.*, 35(2):288–323, 1988.

[DLZ$^+$18] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385, 2018.

[DM15] Douglas Pike, Patrick Nosker, David Boehm, Daniel Grisham, Steve Woods and Joshua Marston. PoST White Paper. Technical report, 2015. https://www.vericoin.info/downloads/VeriCoinPoSTWhitePaper10May2015.pdf, Accessed: 2019-04-28.

[DMEN17] Patrick Dai, Neil Mahi, Jordan Earls, and Alex Norta. Smart-contract value-transfer protocols on a distributed mobile application platform. Technical report, 2017. https://qtum.org/uploads/files/cf6d69348ca50dd985b60425ccf282f3.pdf, Accessed: 2019-05-13.

[EE16] Ittay Eyal and G Emin. Majority is not Enough: Bitcoin Mining is Vulnerable. Technical report, 2016. https://www.cs.cornell.edu/ ie53/publications/btcProcFC.pdf, Accessed: 2019-02-15.

[EOS18] EOS. Documentation_TechnicalWhitePaper EOS.IO. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, 2018. Accessed: 2019-04-02.

[Eth14] Ethereum Community. White Paper · Ethereum. https://github.com/ethereum/wiki/wiki/White-Paper, 2014. Accessed: 2019-04-02.

[Eth19] EthHub. Ethereum Roadmap. https://docs.ethhub.io/ethereum-roadmap/ethereum-2.0/eth-2.0-phases/, 2019. Accessed: 2019-12-21.

[Fou18] The Internet of Services Foundation. Internet of Services: The Next-generation, Secure, Highly Scalable Ecosystem for Online Services. https://github.com/iost-official/Documents/commits/master/Technical_White_Paper/EN/Tech_white_paper_EN.md, 2018. Accessed: 2019-03-01.

[GHM$^+$00] Rachid Guerraoui, Michel Hurfin, Achour Mostefaoui, Riucarlos Oliveira, Michel Raynal, Andre Schiper, Michel Hurfinn, Achour Mostefaoui, Riucarlos Oliveira, Michel Raynal, and Andre Schiper. Consensus in Asynchronous Distributed Systems: A Concise Guided Tour. In Sacha Krakowiak and Santosh Shrivastava, editors, *LNCS 1752: Distributed Systems*, pages 33–47, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[Gla17] Florian Glaser. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis. In *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, pages 1543–1552, 2017.

[GoC18] GoChain. GoChain : Blockchain at Scale. Technical report, 2018. https://neironix.io/documents/whitepaper/3901/gochain-whitepaper-v1.pdf, Accessed: 2019-04-28.

[Goo16] L.M. Goodman. Tezos — a self-amending crypto-ledger. Technical report, 2016. https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf, Accessed: 2019-04-29.

[Gre15] Gideon Greenspan. Avoiding the pointless blockchain project | MultiChain. https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project/, 2015. Accessed: 2019-03-07.

[Gri18] Gridcoin. GRIDCOIN WHITE PAPER - the Computation Power of a Blockchain. Technical report, 2018. https://gridcoin.us/assets/img/whitepaper.pdf, Accessed: 2019-05-10.

[GVS17] Seán Gauld, Franz Von Ancoina, and Robert Stadler. The Burst Dymaxion An Arbitrary Scalable, Energy Efficient and Anonymous Transaction Network Based on Colored Tangles. Technical report, 2017. https://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf, Accessed: 2019-04-15.

[Hea16] Mike Hearn. Corda: A distributed ledger (Whitepaper). Technical report, 2016. https://docs.corda.net/_static/corda-technical-whitepaper.pdf, Accessed: 2019-05-02.

[Ico17] Icon. Hyperconnect the World Icon whitepaper. Technical Report January, 2017. http://docs.icon.foundation/ICON-Whitepaper-EN-Draft.pdf, Accessed: 2019-05-01.

[Ico18] Icorating. Smart Contract Platforms Review. *Icorating*, pages 1–35, 2018.

[KC18] Komodo-Community. Komodo: An Advanced Blockchain Technology, Focused on Freedom. Technical report, 2018. https://docs.komodoplatform.com, Accessed: 2019-05-12.

[Kin13] Sunny King. Primecoin: Cryptocurrency with Prime Number Proof-of-Work. Technical report, 2013. http://betouchi.free.fr/doc_et_ebook/cryptocoins/primecoin-paper.pdf, Accessed: 2019-05-08.

[KMS$^+$16] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 839–858, Piscataway, NJ, USA, 2016. IEEE.

[KRD17] Aggelos Kiayias, Alexander Russell, and Bernardo et al. David. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, volume 1919, pages 1–27, 2017.

[Ksh17] Nir Kshetri. Can Blockchain Strengthen the Internet of Things? By: Nir Kshetri Kshetri, Nir (2017). "Can Blockchain Strengthen the Internet of Things?". *IEEE IT Professional*, 19(4):68–72, 2017.

[Kwo14] Jae Kwon. TenderMint : Consensus without Mining. Technical report, 2014. tendermint.com/docs/tendermint.pdf, Accessed: 2019-04-28.

[Lai18] Y. Lai. NEM White paper. Technical Report 159679, 2018. https://www.cryptoground.com/nem-white-paper, Accessed: 2019-05-03.

[Lar13] Daniel Larimer. Transactions as Proof-of-Stake! Technical report, 2013. https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf, Accessed: 2019-04-02.

[Lar17] Daniel Larimer. DPOS Consensus Algorithm - The Missing White Paper, 2017.

[LCO$^+$16] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making Smart Contracts Smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, pages 254–269, 2016.

[Lew15] Lewis Anthony. A gentle introduction to smart contracts – Bits on Blocks, 2015.

[Lis17] Lisk. Lisk SDK Overview :: Documentation. https://lisk.io/documentation/lisk-sdk/index.html, 2017. Accessed: 2019-04-05.

[LTKS15] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying Incentives in the Consensus Computer. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 706–719, New York, NY, USA, 2015. ACM.

[MAZ15] DAVID MAZIERES. The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus. Technical report, 2015. https://www.stellar.org/papers/stellar-consensus-protocol.pdf, Accessed: 2019-04-26.

[Mem18] AI Crytpo Members. AI Crypto: A Blockchain for Decentralized Economy. Technical report, 2018. https://www.aicrypto.ai/AIC_WhitePaper_Eng.pdf, Accessed: 2019-05-13.

[MHWK16] Mitar Milutinovic, Warren He, Howard Wu, and Maxinder Kanwal. Proof of Luck: an Efficient Blockchain Consensus Protocol. Technical report, 2016. http://delivery.acm.org/10.1145/3010000/3007790/a2-milutinovic.pdf?ip=128.130.60.94&id=3007790&acc=CHORUS&key=9074-CF143665B1C6.97709C79A94C9E0F.4D4702B0C3E38B35.6D218144511F3437-&___acm___=1518623925_947ef80135d534d802d3586534d0086a, Accessed: 2019-04-02.

[MJS$^+$14] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. Permacoin: Repurposing bitcoin work for data preservation. *Proceedings - IEEE Symposium on Security and Privacy*, pages 475–490, 2014.

[ML14] Andrew Miller and Joseph J LaViola Jr. *Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin*. PhD thesis, 2014.

[MXC$^+$16] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The Honey Badger of BFT protocols. In *Proceedings of the ACM Conference on Computer and Communications Security*, volume 24-28-Octo, pages 31–42, 2016.

[N. 17] S. Secci N. Bozic, G. Pujolle. Securing virtual machine orchestration with blockchains. In *2017 1st Cyber Security in Networking Conference (CSNet)*, pages 1–8, 2017.

[Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, page 9, 2008.

[Ned18] Andrey Nedobylsky. LCPoA — universal as PoW, economical as PoS – IZZZIO – Medium. https://medium.com/@izzzio/lcpoa-universal-as-pow-economical-as-pos-c26f6ba90017, 2018. Accessed: 2019-03-01.

[NXT14] COMMUNITY NXT. Nxt Whitepaper. Technical report, 2014. https://www.dropbox.com/s/cbuwrorf672c0yy/NxtWhitepaper_v122_rev4.pdf, Accessed: 2019-04-28.

[OJ14] Diego Ongaro Ousterhout and John. *In Search of an Understandable Consensus Algorithm (Extended Version)*. PhD thesis, 2014.

[Ont18] Ontology. Ontology Introductory Whitepaper. Technical report, 2018. https://ont.io/wp/Ontology-Introductory-White-Paper-EN.pdf, Accessed: 2019-05-20.

[OS10] C. Okoli and K. Schabram. A guide to conducting a systematic literature review of information systems research, 2010.

[P4T14] P4Titan. A Peer-to-Peer Crypto-Currency with Proof-of-Burn “Mining without Powerful Hardware". Technical report, 2014. www.slimcoin.org, Accessed: 2019-04-12.

[PPA$^+$15] Sunoo Park, Krzysztof Pietrzak, Joel Alwen, Georg Fuchsbauer, and Peter Gazi. Spacecoin: A Cryptocurrency Based on Proofs of Space. *IACR Cryptology ePrint Archive*, pages 1–26, 2015.

[PPCC16] Han-kyul Park, Changki Park, Yezune Choi, and Jake Hyunduk Choi. The BOScoin White Paper. Technical Report 1, 2016. http://boscoin.net/BOScoinWhitePaperv20170121.pdf, Accessed: 2019-04-12.

[Pro18] The Coin MAGI Project. MAGI _ Coin MAGIpow. https://www.m-core.org/resources/mining.html#mpow-mining, 2018. Accessed: 2019-03-14.

[Ray10] Michel Raynal. Communication and Agreement Abstractions for Fault-Tolerant Asynchronous Distributed Systems. *Synthesis Lectures on Distributed Computing Theory*, 1(1):1–273, 2010.

[Rei16] Christian Reitwiessner. zkSNARKs in a Nutshell. Technical report, 2016. https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/, Accessed: 2019-05-24.

[Ren14] Larry Ren. Proof of Stake Velocity: Building the Social Currency of the Digital Age. Technical report, 2014. http://reddcoin.com/papers/PoSV.pdf, Accessed: 2019-03-26.

[Rip17] Ripple. Solution Overview A comprehensive business overview for financial institutions on RippleNet. Technical Report October, 2017. https://ripple.com/files/ripple_solutions_overview.pdf, Accessed: 2019-04-27.

[RMC⁺18] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems*, 88(June):173–190, 2018.

[Ros19] Ameer Rosic. What Are Smart Contracts? [Ultimate Beginner's Guide to Smart Contracts], 2019.

[Sai18] Vaibhav Saini. ConsensusPedia: An Encyclopedia of 30+ Consensus Algorithms - By. https://hackernoon.com/consensuspedia-an-encyclopedia-of-29-consensus-algorithms-e9c4b4b7d08f, 2018. Accessed: 2019-03-01.

[San18] Maxwell Sanchez. Proof-of-Proof: A Decentralized, Trustless, Transparent, and Scalable Means of Inheriting Proof-of-Work Security. Technical report, 2018. https://res.tuoluocaijing.cn/20190403161103-y8xs.pdf, Accessed: 2019-03-21.

[Sch90] Fred B. Schneider. Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial. *ACM Computing Surveys (CSUR)*, 22(4):299–319, 1990.

[Shi17] Shield. PoS Boo introduction - SHIELD - Medium. https://medium.com/@shieldxsh/pos-boo-introduction-b012d7546d60, 2017. Accessed: 2019-03-22.

[Sho17] Ali Shoker. Sustainable blockchain through proof of exercise. In *2017 IEEE 16th International Symposium on Network Computing and Applications, NCA 2017*, volume 2017-Janua, pages 1–9, 2017.

[Sky17] Sky Guo. Cypherium: A Scalable and Permissionless Smart Contract Platform. Technical report, cypherium.io, 2017. https://www.cypherium.io/wp-content/uploads/2017/03/cypherium_whitepaper.pdf, Accessed: 2019-05-20.

[Sta18] Stakenet. Stake Net Whitepaper. Technical Report July, 2018. https://stakenet.io/Whitepaper_Stakenet_V3.0_EN.pdf, Accessed: 2019-05-29.

[Str16] Stratumn. Proof of Process. Technical report, 2016. https://stratumn.com/proof-of-process.html, Accessed: 2019-03-02.

[Sza97] Nick Szabo. Formalizing and Securing Relationships on Public Networks. *First Monday*, 2(9):28, 1997.

[Tak18] Shunsai Takahashi. Proof-of-Approval: A Distributed Consensus Protocol for Blockchains. Technical report, 2018. https://github.com/Takanium/doc/blob/bf8f7934bac20d4ce15d8b611cc3525-e18618651/research/proof-of-approval.pdf, Accessed: 2019-04-02.

[Tea15a] Hydrachain Team. HC Consensus. https://github.com/HydraChain/hydra-chain/commits/develop/hc_consensus_explained.md, 2015. Accessed: 2019-03-15.

[Tea15b] NEO Team. NEO White Paper. https://docs.neo.org/docs/en-us/basic/whitepaper.html#consensus-mechanism-dbft, 2015. Accessed: 2019-03-15.

[Tea17a] B3Coin Team. B3 Coin - Proof of Disintegration and Fundamental Node. https://b3coin.io/#info, 2017. Accessed: 2019-03-15.

[Tea17b] Hyperledger Team. Hyperledger Architecture, Volume 1. Technical report, 2017. https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf, Accessed: 2019-06-02.

[Tea17c] The ZILLIQA Team. The Zilliqa Technical Whitepaper. Technical report, 2017. https://docs.zilliqa.com/whitepaper.pdf, Accessed: 2019-05-11.

[Tea18a] Nebulas Team. Nebulas Technical White Paper. Technical report, 2018. https://nebulas.io/docs/NebulasTechnicalWhitepaper.pdf, Accessed: 2019-06-01.

[Tea18b] The Ontology Team. Ontology Launches VBFT, a Next-Generation Consensus Mechanism, Becoming one of the First VRF-Based Public Chains. https://medium.com/ontologynetwork/ontology-launches-vbft-a-next-generation-consensus-mechanism-becoming-one-of-the-first-vrf-based-91f782308db4, 2018. Accessed: 2019-03-16.

[The18] The Coin MAGI Project. MAGI | Coin MAGI, 2018.

[TK07] P. Brereton B. A. Kitchenham D. Budgen M. Turner and M. Khalil. Lessons from applying the systematic literature review process within the software engineering domain, 2007.

[TRO18] TRON Foundation. Advanced Decentralized Blockchain Platform Whitepaper Version : 2 . 0 TRON Protocol Version : 3 . 2 TRON Foundation. Technical report, 2018. https://tron.network/static/doc/white_paper_v_2_0.pdf, Accessed: 2019-05-20.

[TS16] Florian Tschorsch and Björn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys and Tutorials*, 18(3):2084–2123, 2016.

[TW18] Mahnush Movahedi Timo Hanke and Dominic Williams. DFINITY Technology Overview Series Consensus System. Technical report, 2018. https://arxiv.org/pdf/1805.04548.pdf, Accessed: 2019-04-28.

[Vas14] Pavel Vasin. BlackCoin's Proof-of-Stake Protocol v2. Technical report, 2014. https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf, Accessed: 2019-02-27.

[VeC18] VeChain. VeChain - DEVELOPMENT PLAN AND WHITEPAPER. Technical report, 2018. https://cdn.vechain.com/vechainthor_development_plan_and_whitepaper-_en_v1.0.pdf, Accessed: 2019-06-02.

[Ver18] Vericoin. Proof-of-Work-Time - VeriCoin & Verium Wiki. https://wiki.vericoin.info/index.php?title=Proof-of-Work-Time, 2018. Accessed: 2019-03-15.

[Vuk16] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. 2016.

[Wal19] Cèdric Walter. Blockchain Consensus, 2019. https://www.tokens-economy.com/wp-content/uploads/2019/02/Major-Blockchain-consensus-Infographics.pdf, Accessed: 2019-04-02.

[Wav17] Waves. WAVES whitepaper. https://blog.wavesplatform.com/waves-whitepaper-164dd6ca6a23, 2017. Accessed: 2019-04-10.

[WC19] Ben Wilson and Alexis Carreiro. Lynx Technical White Paper 1.1. Technical report, 2019. http://cdn.getlynx.io/2018-06-18_Lynx_Whitepaper.pdf, Accessed: 2019-06-02.

[WW02] J. Webster and R. T. Watson. Analyzing the past to prepare for the future: Writing a literature review, 2002.

[Yak18] Anatoly Yakovenko. *Solana: A new architecture for a high performance blockchain*. 2018.

[YGA+18] Kimchai Yeow, Abdullah Gani, Raja Wasim Ahmad, Joel J.P.C. Rodrigues, and Kwangman Ko. Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues. *IEEE Access*, 6:1513–1524, 2018.

[Yu-17] Yu-Te Lin. Istanbul Byzantine Fault Tolerance · Issue #650 · ethereum/EIPs · GitHub. https://github.com/ethereum/EIPs/issues/650, 2017. Accessed: 2019-03-01.

[ZXD+17] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain Challenges and Opportunities : A Survey Shaoan Xie Hong-Ning Dai Huaimin Wang. *International Journal of Web and Grid Services*, 14(4):1–24, 2017.