Wilhelm Büchner Hochschule
University of Applied Sciences
Hilpertstraße 31
64295 Darmstadt

Masterthesis

Fachbereich Wirtschaftsingenieurwesen und
Technologiemanagement

Blockchain finality for real-world settlement

Lucas Konstantin Bärenfänger

Matrikelnummer: 903656
Postanschrift: Am Trauben 54, 63303 Dreieich
Studiengang: M.Sc. Innovations- und Technologiemanagement

Betreuer: Michael Best, LL.M.
Datum: 23.03.2021

Abstract

In the wake of the discussion on central bank digital currencies (CBDC), the applicability of blockchain technology in the realm of payments is subject of debate. However, an assessment of whether and how settlement finality can be achieved in blockchain systems is largely absent from the discourse, despite the direct impact of finality on settlement risks.

This thesis addresses this question. First, a definition of finality based on a literature review is provided. Then, as finality is traditionally regarded a legal concept, it is assessed whether blockchain-based payment systems are compatible with said notion of legal finality. Lastly, as the legal account of finality implicitly assumes certain technical properties, blockchain technology is analyzed with regards to this notion of technical finality.

It is concluded that Bitcoin-style blockchain technology is both incompatible with legislation on settlement finality and unable to achieve technical finality. However, changes in legislation on final settlement and the adoption of alternative blockchain technologies may enable blockchain-based payment systems to settle with finality.

Keywords: Payment systems, settlement systems, settlement finality, settlement risks, blockchain technology, distributed consensus, Bitcoin, proof of work, majority attacks, selfish mining, Byzantine agreement, Stellar consensus protocol

Table of contents

Chapter 1

Introduction

More than a decade after blockchain technology was originally introduced with Bitcoin, its applications go well beyond that of cryptocurrencies. Most notably, general-purpose blockchain systems, e.g., Ethereum, allow for the definition of smart contracts, which, in turn, enable the issuance of alternative tokens next to a blockchain system's native cryptocurrency token. Such tokens are used to digitally represent a plethora of assets, e.g., shares in companies or property rights.

At the same time, the idea of central bank digital currencies (CBDC), i.e., digital representations of legal tender issued by a central bank, is increasingly discussed in academia and in the banking community, with many central banks publishing working papers or even prototypes. More often than not, CBDCs are envisioned to be "on-chain," i.e., in the form of blockchain tokens.

However, digital money based on blockchain technology implies that payment systems, including wholesale payment systems, are blockchain-based as well. This has intensified the debate on whether blockchain technology is well-suited to underpin such systems.

While many facets of blockchain systems have been analyzed with regards to this, little work has been published on whether and how blockchain-based payment systems achieve final settlement. Settlement finality is, however, a crucial property of payment systems, as it is directly related to the settlement risks that participants incur and even to systemic risk.

This thesis aims to close this knowledge gap. To this end, the following three crucial questions are addressed:

How is finality defined in the literature? The first part of this thesis surveys the literature to derive a clear definition of finality, as the term has come to mean different things to different people.

Can blockchain systems provide legal finality? As finality is traditionally defined in a legal sense, the second part of this thesis examines whether blockchain-based payment systems can provide said legal finality.

Can blockchain systems provide technical finality? As there also is a technical aspect to finality that is traditionally assumed to "just work," the third part of this thesis examines whether blockchain-based payment systems can achieve this novel notion of technical finality.

Based on the insights resulting from addressing these questions, a comprehensive account of finality in the context of blockchain-based payment systems is given.

Chapter 2

Prerequisites

This chapter briefly introduces the terms and concepts on which this thesis is based and the knowledge of which is necessary for understanding it. Relevant parts from the banking domain of traditional payments are introduced, followed by relevant parts from the computer science domain of blockchain technology.

2.1 Traditional payments

In the following, relevant aspects from the banking domain of traditional payments are introduced, namely payment systems, settlement risks and foreign exchange.

2.1.1 Payment systems

A payment system is a "set of instruments, procedures and rules for the transfer of funds between or among participants," whose scope encompasses "the participants and the entity operating the arrangement."[1]

Payment systems are distinguished into those that facilitate retail payments, processing large volumes of low-value payments relating "to the purchase of goods and services by consumers and businesses," and those that facilitate wholesale payments, processing fewer large-value payments "between financial institutions."[2] This thesis is concerned with the latter, specifically, those that are systemically important, i.e., wholesale payment systems that represent "a major channel

[1] Committee on Payments and Market Infrastructures. *A glossary of terms used in payments and settlement systems*. 2006. URL: https://www.bis.org/dcms/glossary/glossary.pdf?scope=CPMI&base=term (visited on 03/16/2021), p. 13.

[2] M. Bech and J. Hancock. "Innovations in payments". In: *BIS Quarterly Review* (2020). URL: https://www.bis.org/publ/qtrpdf/r_qt2003f.pdf, p. 22.

by which shocks can be transmitted across domestic and international financial systems and markets."[3]

At the heart of a payment system are its clearing and settlement mechanisms (CSM). It is to be noted that although CSMs are oftentimes referred to as an entity, clearing and settlement are usually carried out by systems that are independent entities both technically and legally.

The functions of clearing cover "all activities from the time a transaction [...] is made until it is finally settled."[4] Clearing is, therefore, concerned with the "transmission of the [t]ransfer [o]rder message,"[5] i.e., with the transmission of information. The functions of settlement enable "the process of transferring funds to discharge monetary obligations between two or more parties."[6] Settlement is, therefore, concerned with the actual "transfer of value."[7]

Payment systems are further distinguished into those that settle on a net basis and those that settle on a gross basis. There also exist hybrid systems that combine both approaches.[8]

Settlement on a net basis is preceded by the process of netting, which is a clearing function. Netting is "[t]he offsetting of obligations between or among participants in the netting arrangement,"[9] which is commonly explained using an example similar to the following:[10]

1. At 10:00, participant A transfers 8,000€ to participant B.
2. At 13:00, participant A transfers 5,000€ to participant B.
3. At 16:00, participant B transfers 3,000€ to participant A.

At the three designated times, no actual value is transferred, as clearing is concerned with information only. Instead, a net settlement position, which is defined

[3] Committee on Payments and Market Infrastructures. "Core principles for systemically important payment systems". In: *CMPI Papers* (2001). URL: `https://www.bis.org/cpmi/publ/d43.htm`, p. 1.

[4] K. M. Löber. "The developing EU legal framework for clearing and settlement of financial instruments". In: *ECB Legal Working Paper Series* (2006). URL: `https://www.ecb.europa.eu/pub/pdf/scplps/ecblwp1.pdf`, p. 6.

[5] M. Vereecken and A. Nijenhuis. *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*. Kluwer Legal Publishers, 2003. ISBN: 9789013004878, p. 21.

[6] Bech and Hancock, "Innovations in payments", p. 23.

[7] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 21.

[8] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 27.

[9] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 12.

[10] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 39; J. P. Megue. *SEPA credit transfer: how to understand and add value to your SCT payment project*. Paiementor, 2018. ISBN: 9791094710012, pp. 21, 24.

as the "net position at settlement time,"[11] is calculated. Participant A ends up with a net debit position of 10,000€, participant B with a net credit position of 10,000€. Those are settled, i.e., value is transferred, at settlement time, which, "[t]ypically," is "at the end of the day,"[12] resulting in a total of two actual value transfers.

This is an example of bilateral netting, as there are only two participants. The offsetting of obligations between two or more participants is referred to as multilateral netting.

While settlement on a net basis is a form of batch processing, settlement on a gross basis corresponds to unitary processing, as gross settlement is "the settlement of transfer instructions [...] individually on a transaction-by-transaction basis for full value."[13] If the transactions in the previous netting example were to be settled on a gross basis, all three transfers would be settled individually, on or at some point after 10:00, 13:00 and 16:00, respectively, as settlement happens at "discrete intervals during the day."[14]

A settlement system that settles on a net basis is referred to as a deferred net settlement (DNS) system. One that settles on a gross basis is referred to as a real-time gross settlement (RTGS) system.[15] It is to be noted that "the term 'real-time' can be misleading, as the settlement is not [by definition] 'immediate.'"[16] The advantages and disadvantages of either approach are discussed later in this thesis.

As mentioned, the functions of a payment system are commonly carried out by independent parties. This is reflected in the definition of a payment system used in EU legislation, which considers such a system a "formal arrangement" between "participants."[17] Such participants are "institutions," which use the system to clear and settle transfer orders among each other, and "settlement agents," which provide accounts through which the transfer orders are settled.[18] Participants can, furthermore, be clearing houses and central counterparties, which need to be discussed in their own right, as they are crucial in the context of clearing.

[11] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 38.
[12] Ibid., p. 38.
[13] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 9.
[14] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 38.
[15] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 26.
[16] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 38.
[17] European Parliament and Council of the European Union. *Directive 98/26/EC on settlement finality in payment and securities settlement systems*. 1998. URL: `https://eur-lex.europa.eu/eli/dir/1998/26/oj` (visited on 03/15/2021).
[18] Ibid.

A clearing house (CH), as its name suggests, provides "clearing services" in a "multilateral [...] arrangement."[19] If netting is applied, one of these services is the calculation of the net settlement positions of the participants. Without a CH in between them, participants themselves would have to keep track of their positions regarding every other participant, resulting in an "extremely impractical," "fully-connected network topology."[20]

In this capacity, a CH oftentimes acts as a central counterparty (CCP) as it "interposes itself between counterparties [...], becoming the buyer to every seller and the seller to every buyer"[21] in a system. In other words, when a participant receives a credit transaction, it considers the CCP to owe it money, in case of a debit transaction, it considers that it owes money to the CCP.[22]

2.1.2 Settlement risks

Settlement finality or, for short, finality is commonly discussed in the context of settlement risks arising from the lack thereof, i.e., from the inability to achieve finality. Therefore, any discussion on the different accounts of finality in the literature must be preceded by a definition of settlement risks.

As a "general term," settlement risk is broadly defined as "the risk that settlement in a funds or securities transfer system will not take place as expected."[23] Settlement risk can, however, be differentiated into different kinds, as there are multiple "risks [that] can arise in payment systems"[24] that can be considered settlement risks.

Such risks include credit risk (or principal risk), which is "the risk that a party within the system will be unable to fully meet its financial obligations within the system either when due or at any time in the future," and liquidity risk, which is "the risk that a party within the system will have insufficient funds to meet financial obligations within the system as and when expected, although it may be able to do so at some time in the future."[25]

A downstream effect of the aforementioned settlement risks, systemic risk is "the risk that the inability of one of the participants to meet its obligations, or a disruption in the system itself, could result in the inability of other system participants [...]

[19] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 4.
[20] Megue, *SEPA credit transfer: how to understand and add value to your SCT payment project*, p. 23.
[21] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 3.
[22] Megue, *SEPA credit transfer: how to understand and add value to your SCT payment project*, p. 24.
[23] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 17.
[24] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 5.
[25] Ibid., p. 5.

to meet their obligations as they become due."[26] "Such a failure could [...] threaten the stability of the system or of financial markets."[27]

2.1.3 Foreign exchange

Furthermore, discussions on finality, including this thesis, commonly take a detour into the foreign exchange market, which must, therefore, be elaborated on as well.

The foreign exchange (FX) market "is the market in which foreign currency [...] is traded for domestic currency [...]," however, "[t]his 'market' is not in a centralized location; instead, it is a decentralized network that is [...] highly integrated via modern [IT] technology."[28] "At its core, settlement of a foreign exchange [...] trade requires the payment of one currency and the receipt of another."[29] Accordingly, an FX trade is made up of two legs, where one leg refers to the transfer of one currency from participant A to participant B and the other leg, inversely, refers to the transfer of another currency from participant B to participant A.

2.2 Blockchain technology

In the following, relevant aspects from the computer science domain of blockchain technology are introduced, namely the double-spending problem, proof of work, smart contracts and permissioned blockchains.

2.2.1 Double-spending problem

The very first blockchain system is Bitcoin and its creator, who operates under the pseudonym Satoshi Nakamoto and published the Bitcoin paper in 2008,[30] is considered the inventor of blockchain technology.

Before delving into how blockchain technology works, it is crucial to clarify what problem it actually solves, especially since so many misleading analogies and, frankly, falsehoods with regards to this, e.g., claims such as "works like gold mining," circulate among the wider public.

[26] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 5.
[27] Ibid., p. 5.
[28] J. A. Frankel. *Foreign exchange*. 2020. URL: https://www.econlib.org/library/Enc/ForeignExchange.html (visited on 12/09/2020).
[29] Committee on Payments and Market Infrastructures. "Settlement risk in foreign exchange transactions". In: *CMPI Papers* (1996). URL: https://www.bis.org/cpmi/publ/d17.htm, p. 4.
[30] S. Nakamoto. *Bitcoin: a peer-to-peer electronic cash system*. 2008. URL: https://bitcoin.org/bitcoin.pdf (visited on 03/15/2021).

The Bitcoin paper considers Bitcoin "a solution to the double-spending problem using a peer-to-peer network."[31] To be perfectly clear, it is that – previously unresolved – double-spending problem, a computer science problem, whose solution is the main contribution of Bitcoin and, consequently, blockchain technology.

As the double-spending problem arises in decentralized networks, it is necessary to first define those before introducing the double-spending problem itself.

A decentralized network in the sense of the Bitcoin network is a peer-to-peer (P2P) network consisting of peers or nodes, i.e., computers, that form a network through point-to-point connections. However, not every node is connected to every other node directly, thus, a path between two nodes may include intermediary nodes. Most importantly, all of the nodes are equally privileged, i.e., all of them have the same rights and obligations. From this follows that no node (or cluster of nodes) acts as a central party (or trusted party) in any capacity, i.e., there is no single source of truth or gatekeeper. From the latter follows that nodes may join and leave the network freely at all times.

In such networks, network partitioning (or network split) may occur, which happens when there no longer is a connection between two or more clusters of nodes. A means of communication between nodes is network flooding, whereby one node broadcasts a message to all the nodes it is connected with or aware of, which, in turn, do the same, and so forth. Eventually, the message will have reached every node in the network, i.e., it will have propagated the network.

A centralized network is, by contrast, characterized by the existence of a central party, which, oftentimes, acts as a single source of truth and gatekeeper. Furthermore, in many cases, nodes communicate exclusively through said central party and not in a peer-to-peer fashion, forming a star-shaped network. The obvious criticism of centralized networks is that a central party represents a single point of failure, since the very functionality of such networks depends on it. Moreover, there is a temptation for those who control the central party to abuse this power.

The double-spending problem is probably best explained by first demonstrating how it does not arise in centralized systems. To this end, assume a simplistic payment system consisting of one central party, i.e., the bank, as well as participants, i.e., bank customers. The central party maintains a ledger, i.e., a record of all transactions between the participants. From the ledger, each participant's balance of tokens, which might, e.g., represent (crypto) currency or securities, can be derived. This model, obviously, corresponds to a centralized network, as the central party's node is the single source of truth (and, possibly, gatekeeper) and the participant's nodes are second-class nodes that communicate with the former.

Furthermore, assume a fraudulent participant A who intends to spend one and the same token twice. To this end, participant A prepares two transfer orders (or trans-

[31] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 1.

actions, which is the common terminology in the blockchain domain), each transferring said token to participants B and C, respectively. Participant A digitally, i.e., cryptographically, signs the transactions, thereby signaling their authenticity, allowing every other node to validate that the transactions truly come from participant A, the token's righteous owner.

Of course, both transactions are valid when considered individually but, obviously, may not both be executed. The node representing participant A sends the two transactions to the node representing the central party, which will process which ever one arrives first and update the ledger according to that one. When the other transaction arrives, it will simply be discarded, as the previously updated ledger indicates that participant A is no longer the owner of the token and, therefore, cannot – again – spend it. Thus, participant A's attempt to double spend is unsuccessful.

The attempt to double spend is, however, not so easily averted if said payment system is implemented Bitcoin-style, atop a decentralized network. In this case, the role of the central party, i.e., the bank, is represented by every node in the network, i.e., by the network as a whole. Accordingly, each node maintains its own copy of the ledger. In addition, each node assumes the role of a participant, i.e., a bank customer, as well, given that each node may broadcast its own transactions to the nodes it is aware of, which will then be forwarded until they have propagated the network.

In order to double spend, participant A, once again, sends out the two previously described transactions. Some of the nodes first receive the transaction that transfers the token to participant B and update their ledgers accordingly, while other nodes first receive the transaction that transfers it to participant C and update their ledger according to the latter transaction. Whether, at this point, participant A's attempt to double spend was successful is a matter of debate, what is clear, however, is that the bank or, rather, the payment system, which, as said, is represented by all nodes as a whole, is broken, since the individual ledgers it consists of are inconsistent.

The challenge to solve the double-spending problem, therefore, is really the challenge to get the nodes of a decentralized network, which – it must be emphasized – represent a fluent set of nodes, to reach agreement (or consensus) on one value. In the case of payment systems such as Bitcoin, the value on which consensus must be reached is the ledger, i.e., the order of transactions.

2.2.2 Proof of work

As mentioned before, the Bitcoin paper proposes the Bitcoin system as "a solution to the double-spending problem."[32] Conceptually, the Bitcoin system corresponds

[32] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 1.

to the decentralized payment system outlined before, although the Bitcoin system maintains coins (or bitcoins, spelled lowercase when referring to the cryptocurrency tokens rather than the system) and not generic tokens.

The double-spending problem is solved by leveraging "a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions." That timestamp server can be understood as a piece of software that runs on every node that participates in the Bitcoin system. Collectively, the individual instances of the timestamp server form the distributed timestamp server that establishes the desired system-wide order of transactions.

Each node running the timestamp server continuously receives transactions and "collects" these "new transactions into a block" or, rather, a continuous stream of blocks.[33] For each new block, a node generates a hash, using a cryptographic hashing function. Such a function "maps a typically large, variable-length value [in this case, a block] to a typically small, fixed-length value" – the hash.[34] The hash function is a "one-way function," i.e., "it is practically impossible to infer from the hash the input value."[35] The hash that was generated for the very first block in the Bitcoin system is, e.g., `000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f`.[36] A block's hash (or block hash) can be seen as proof of the block's existence, as "the [block's] data must have existed at the time, obviously, in order to get [...] the hash [from it]."[37]

However, the hash that the timestamp server generates for each new block is not derived from the block's data alone but from the block's data as well as the hash of its preceding block. Consequently, each block's hash references the previous block's hash, resulting in "a chain"[38] of hashes, which represents the order of blocks as perceived by an individual node.

In the Bitcoin paper, block hashes are referred to as "timestamps"[39] – hence, the term timestamp server. One could argue that this is not the best choice of wording, as these do not capture time in the sense of date and time of day but rather establish a relative order of blocks. Be that as it may, such hashes or timestamps have an important property: As "[e]ach timestamp includes the previous timestamp in its hash," "each additional timestamp [is cryptographically] reinforcing the ones before it."[40] In other words, if one were to tamper with a transaction in one

[33] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.
[34] L. K. Bärenfänger. *Blockchain tokens: a review*. 2020. URL: `https://github.com/lkbaerenfaenger/blockchain-tokens-paper` (visited on 03/15/2021), p. 6.
[35] Ibid., p. 6.
[36] Blockchain.com Bitcoin Explorer. *Block: 0*. 2009. URL: `https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f` (visited on 12/17/2020).
[37] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 2.
[38] Ibid., p. 2.
[39] Ibid.
[40] Ibid., p. 2.

block, not only would the hash of that block have to be regenerated but that of all subsequent blocks would have to be. Thus, the more blocks are appended to one block, the more computationally expensive it becomes to tamper with it.

Obviously, the system outlined up to this point does not yet solve the double-spending problem, as each individual node works on its own chain of blocks, based on (the order of) which ever transactions that node receives. What is missing is the actual consensus algorithm, which enables agreement on which one of the different chains of blocks is the definitive one. In other words, Bitcoin's consensus algorithm facilitates the consensus required to designate the Bitcoin system's definitive blockchain. As a side note, it is to be pointed out that the term "blockchain" is not used in the Bitcoin paper but was coined after the fact.

Bitcoin's consensus algorithm is called proof of work (PoW) and is based on the challenge to find a specific number referred to as "nonce."[41] Contrary to what was said before, a block hash is not only derived from a block's data and the hash of its preceding block but also from said nonce. The nonce has to be "a value that when hashed [along with the other data], [...] the [resulting] hash begins with a number of zero bits."[42] (The attentive reader might have noticed that the hash of the first Bitcoin block does, in fact, begin with a number of zero bits.)

As hashing functions are one-way functions, though, a value satisfying this requirement cannot be computed, "it has to be guessed."[43] To this end, a node increments a nonce "until a value is found that gives the block's hash the required zero bits."[44] The number of required zero bits is referred to as difficulty, since the more zero bits are required, the more "average work,"[45] i.e., processing power, is required to solve the challenge to find the right nonce – to which, confusingly, the Bitcoin paper also refers to as proof of work.[46] In the live Bitcoin system, i.e., not the model described in the Bitcoin paper, "[t]he difficulty [...] is adjusted so as to limit the rate at which new blocks can be generated by the network to one every 10 minutes."[47]

Once a node finds a proof of work for a block it is currently working on, "it broadcasts the block [as well as, of course, the nonce and the block hash] to all nodes," which check "if all transactions in it are valid and not already spent" and, if so, "express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash."[48] Most importantly, "[n]odes always consider the longest chain [they are aware of] to be the

[41] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.
[42] Ibid., p. 3.
[43] Bärenfänger, *Blockchain tokens: a review*, p. 7.
[44] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.
[45] Ibid., p. 3.
[46] Ibid., p. 3.
[47] bitcoin.it Bitcoin wiki. *Proof of work*. 2020. URL: `https://en.bitcoin.it/wiki/Proof_of_work` (visited on 12/18/2020).
[48] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.

correct one" and switch towards extending it right away when learning about its existence.[49] This constitutes a solution to the double-spending problem.

Nodes are rewarded for expending work to create new blocks. Once a block that a node created is included into the blockchain, the node benefits in two ways. First, it gets to keep any transaction fees that were attached to the block's transactions[50] as participants may voluntarily pay transaction fees as an incentive to get their transactions processed faster, i.e., to motivate nodes to put them into a block rather than other transactions. Second, "the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block."[51]

Consequently, with each new block, a set amount of new bitcoins come into existence, as the block-creating node puts a special transaction into the new block that transfers bitcoin(s) to it – without specifying a debtor. To ensure scarcity, this amount is reduced by 50% roughly every four years, with the result being "that the number of bitcoins in existence will not exceed slightly less than 21 million."[52]

The process of expending work to extend the blockchain and being rewarded for it with bitcoins is commonly referred to as "mining," and the nodes that engage in it are called "miners." Note that this terminology is not mentioned in the Bitcoin paper but emerged later, just like the term blockchain itself. Furthermore, note that this incentive model is specific to Bitcoin. Other ways to incentivize nodes to collectively maintain a blockchain are conceivable.

As the Bitcoin system is based on a decentralized network, individual nodes or entire clusters of nodes may learn about new blocks/blockchains with delay, e.g., due to long paths or temporary network partition. However, as the longest chain will at some point have propagated the network, all nodes will eventually accept and extend it. Thus, Bitcoin is said to feature "eventual consistency."[53]

Since the inception of Bitcoin and, along with it, blockchain technology, a variety of other blockchain systems and, hence, consensus algorithms have been proposed, some of which are featured later in this thesis. However, as blockchain technology is commonly discussed in reference to Bitcoin, Bitcoin and PoW can be considered the reference model of blockchain technology and consensus algorithms, respectively.

Unless explicitly noted otherwise, in this thesis, the term "blockchain system" refers to Bitcoin-style blockchain systems based on PoW.

[49] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.
[50] Ibid., p. 4.
[51] Ibid., p. 4.
[52] bitcoin.it Bitcoin wiki. *Controlled supply*. 2020. URL: `https://en.bitcoin.it/wiki/Controlled_supply` (visited on 01/03/2021).
[53] I. Bashir. *Mastering Blockchain*. Packt Publishing, 2020. ISBN: 9781839213199, p. 35.

2.2.3 Smart contracts

As outlined previously, blockchain technology enables the nodes of a decentralized system to reach consensus. In Bitcoin, specifically, nodes reach consensus on a single transaction history, from which each participant's balance of bitcoins, i.e., Bitcoin's cryptocurrency token, can be derived. Blockchain technology and its ability to facilitate agreement can, however, be used for a variety of use cases beyond cryptocurrency.

Most notably, blockchain systems enable the decentralized enforcement of "smart contracts." The American computer scientist and legal scholar Nick Szabo is commonly credited with conceiving the concept of smart contracts as early as 1994, long before blockchain technology came around, defining a smart contract as "a computerized transaction protocol that executes the terms of a contract."[54]

Szabo later describes smart contracts in the context of the quest towards "the formalizations of our relationships" in order to achieve "ideal security."[55] He notes that "[m]any kinds of contractual clauses (such as collateral, bonding, delineation of property rights, etc.) can be embedded in the hardware and software."[56]

As a "canonical real-life example," he cites "the humble vending machine," as "the machine takes in coins, and via a simple mechanism, [...] dispense[s] change and product according to the displayed price," thereby constituting "a contract with bearer: anybody with coins can participate in an exchange with the vendor."[57]

Smart contracts, however, "go beyond the vending machine in proposing to embed contracts in all sorts of property that is valuable and controlled by digital means."[58] He, furthermore, adds that smart contracts "reference [...] property in a dynamic, often proactively enforced form" and, thereby, "provide much better observation and verification."[59]

After several iterations of "refinement," Szabo presents an exemplary "smart lien protocol," i.e., a smart contract, to formalize how a "car is being used to secure credit" – a matter that, traditionally, "would create a headache for the creditor," as the "the repo man would no longer be able to confiscate a deadbeat's car":[60]

[54] N. Szabo. *Smart contracts*. 1994. URL: `https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html` (visited on 03/16/2021).
[55] N. Szabo. *The idea of smart contracts*. 1997. URL: `https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html` (visited on 03/16/2021).
[56] Ibid.
[57] Ibid.
[58] Ibid.
[59] Ibid.
[60] Ibid.

1. A [car] lock to selectively let in the owner and ex[c]lude third parties;
2. A back door to let in the creditor;

3.a. Creditor back door switched on only upon nonpayment for a certain period of time; and

3.b. The final electronic payment permanently switches off the back door.

Blockchain technology lets these remarks on smart contracts shine in a new light, as it enables the nodes of a decentralized network to execute such formalized contracts by facilitating agreement between them with regards to said execution. While Bitcoin aims at eradicating "the need for a trusted third party" in monetary systems by proposing a decentralized "one-CPU-one-vote" system that is open-access and "based on cryptographic proof instead of trust,"[61] the same underlying mechanism can be used to collectively enforce smart contracts in a decentralized, transparent and cryptographically verifiable way.

However, for the sake of correctness, it is to be pointed out that in the aforementioned examples given by Szabo, smart contracts "reference"[62] and affect physical, i.e., "off-chain," objects, which, obviously, cannot be achieved by software alone. Physical objects can very well be represented by "on-chain" tokens, though, and these can be affected, e.g., transferred, in the context of the blockchain-based execution of smart contracts.

At this point, it is necessary to refine the previous usage of the term token. In a technical sense, it describes "an entity that is maintained and transferred within a blockchain-based system."[63] E.g., bitcoin is the token of the Bitcoin system, as it is maintained and transferred within it. In a legal sense, a token can "be regarded as a [metaphorical] 'container'" that may embody legal rights.[64]

While it remains subject of debate whether a bitcoin has intrinsic value, it is clear that it neither represents a claim on a central bank, unlike, e.g., euro banknotes, nor does it represent any other legal rights, like, e.g., property rights to a gold bar. Thus, to continue the analogy of containers, a bitcoin must be considered an empty container or, rather, empty token. However, as mentioned, blockchain systems are perfectly able to manage tokens that are not empty containers, i.e., tokens that do, in fact, embody rights, e.g., property rights to physical objects.

---

[61] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, pp. 1, 3.
[62] Szabo, *The idea of smart contracts*.
[63] Bärenfänger, *Blockchain tokens: a review*, p. 15.
[64] Bärenfänger, *Blockchain tokens: a review*, p. 15; Liechtenstein Government. *Report and application of the government to the parliament of the principality of Liechtenstein concerning the creation of a law on tokens and TT service providers (Tokens and TT Service Provider Act; TVTG) and the amendment of other laws*. 2019. URL: `https://www.naegele.law/files/Downloads/2019-07-12_BuA_TVTG_en_full_report.pdf` (visited on 03/15/2021), pp. 55, 57.

The previously hinted at question that arises from this is how synchrony between the digital world and the physical world is enforced. If, e.g., a token representing property rights to a gold bar is transferred to a new owner in a blockchain-based system, how does this translate to the physical world? To this end, the Liechtenstein Blockchain Act, the first European legislation to regulate the blockchain domain, introduces the role of a physical validator, which is tasked with ensuring the very synchrony "between the object and the token that represents rights to it."[65] In the remainder of this thesis, synchrony between "on-chain" tokens and the real-world objects they represent is assumed.

Arguably, it was Ethereum, a blockchain system featuring the cryptocurrency ether as its native token, that revisited the idea of smart contracts and popularized them in the blockchain domain. Thus, today, smart contracts are commonly discussed in reference to Ethereum's model of smart contracts.

The Ethereum whitepaper was proposed by the Russian-Canadian software engineer Vitalik Buterin in 2013.[66] Its title, "Ethereum white paper: a next generation smart contract & decentralized application platform," confirms that Ethereum aims to be understood as a decentralized platform for smart contracts. The following excerpt of the Ethereum whitepaper is emblematic for Ethereum's view on smart contracts:[67]

> Commonly cited applications [of blockchain technology] include [...] "smart contracts" – systems which automatically move digital assets according to arbitrary pre-specified rules. For example, one might have a treasury contract of the form "A can withdraw up to X currency units per day, B can withdraw up to Y per day, A and B together can withdraw anything, and A can shut off B's ability to withdraw."

Ethereum is, in many ways, similar to Bitcoin, most notably, Ethereum also leverages PoW to facilitate consensus, although, as part of the Eth2 upgrades,[68] it switches to an alternative consensus algorithm called proof of stake (PoS).[69] One of the crucial differences between Bitcoin and Ethereum is that Bitcoin is

[65] Liechtenstein Government, *Report and application of the government to the parliament of the principality of Liechtenstein concerning the creation of a law on tokens and TT service providers (Tokens and TT Service Provider Act; TVTG) and the amendment of other laws*, p. 67.

[66] V. Buterin. *Ethereum white paper (original version by Vitalik Buterin)*. 2013. URL: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (visited on 03/15/2021); V. Buterin. *Ethereum white paper (updated version by the Ethereum Foundation)*. 2021. URL: https://ethereum.org/en/whitepaper (visited on 03/15/2021).

[67] Buterin, *Ethereum white paper (original version by Vitalik Buterin)*, p. 1.

[68] Ethereum Foundation. *The Eth2 upgrades: upgrading Ethereum to radical new heights*. 2021. URL: https://ethereum.org/en/eth2/ (visited on 03/15/2021).

[69] Ethereum Foundation. *Proof-of-stake (PoS)*. 2020. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/ (visited on 12/29/2020).

transaction-based (or based on the unspent transaction output (UTXO) model) while Ethereum is account-based (or based on the account/balance model).

In Bitcoin, there is no notion of accounts, the state of the system consists solely of bitcoins, each of which is represented by the transactions that transfer ownership of that particular bitcoin. Thus, in order to calculate a participant's balance of bitcoins, one must – simplified speaking – traverse all transactions of all bitcoins and sum up the value of those transactions that transfer (fractions of) bitcoins to the participant in question, while, obviously, only counting those transactions that the participant has not themselves spent yet – hence, the term unspent transactions. Ethereum, by contrast, does feature an intuitive model of accounts as "the [system's global] state is made up of objects called 'accounts.'"[70]

With regards to smart contracts, it is to be pointed out that Ethereum features two types of accounts: "externally-owned accounts," as in owned by a participant, and "contract accounts," which are "controlled [or owned] by their contract code."[71] An account in Ethereum is made up of "fields," including "[t]he account's current ether balance," "[t]he account's contract code" and "[t]he account's storage," where the latter two fields only apply to contract accounts and refer to the source code of an account's smart contract and the "internal storage" that is available to it for maintaining mutable state, respectively.[72]

In Ethereum, the execution of a smart contract is initiated by a transaction sent to the address of its account and is carried out as "part of the block validation algorithm." Thus, "if a transaction is added into block B the code execution spawned by that transaction will be executed by all nodes, now and in the future, that download and validate block B."[73]

Ethereum has proposed multiple programming languages to define smart contracts. The code examples from the original Ethereum whitepaper are implemented in the Python-based programming language Serpent (which has since been deprecated[74]). Today, the "default choice" to define smart contracts in Ethereum is the JavaScript-like programming language Solidity, for "close-to-the-metal optimizations" there is the pythonic programming language Vyper, which, however, "is still experimental."[75]

[70] Buterin, *Ethereum white paper (original version by Vitalik Buterin)*, p. 13.
[71] Ibid., p. 13.
[72] Ibid., pp. 13, 14.
[73] Buterin, *Ethereum white paper (updated version by the Ethereum Foundation)*.
[74] V. Buterin. *Tweet declaring Serpent deprecated*. 2017. URL: `https://twitter.com/VitalikButerin/status/886400133667201024` (visited on 12/29/2020).
[75] V. Buterin and other contributors. *Readme file of the GitHub repository of the Serpent programming language*. 2017. URL: `https://github.com/ethereum/serpent/blob/develop/README.md` (visited on 12/29/2020).

As part of a seminar paper on blockchain tokens[76] that I wrote in 2020, some contributors from the Ethereum community on StackExchange[77] and I designed an Ethereum smart contract as an example of an archetypal application of the technology. To this end, the smart contract had to be simple, it had to place a task that is traditionally performed by a trusted third party into the hands of the decentralized network that is Ethereum and, most importantly, it had to be a formalization of a contractual agreement in the sense of Szabo's remarks.

The resulting smart contract, dubbed `SplitPot`, represents the basic agreement between three participants to evenly split up all funds, i.e., ether, sent to the smart contract's account, which, traditionally, would most likely have involved some kind of trustee:[78]

```
1  pragma solidity ^0.6.0;
2
3  contract SplitPot {
4    address payable[] beneficiaries = [
5      0x498898b3F52DAba1bB304a4b4D2EA31a111484B1,
6      0xAcb19c763EB67ea757Efd8Cd8b6ecceb28F1284B,
7      0xD5d3f3650C4DdE7B8034671129443A596Ce8ed57
8    ];
9
10   receive() external payable {
11     uint individualAmount = msg.value / beneficiaries.length;
12
13     for (uint i = 0; i < beneficiaries.length; i++) {
14       beneficiaries[i].transfer(individualAmount);
15     }
16   }
17 }
```

A smart contract representing a `SplitPot`

In line 1, the version `pragma` is set to prevent compilation with compiler versions that might be incompatible with the stated version of the Solidity programming language. In lines 4–8, the addresses of the (external) accounts of the three participants are stored in an array (`beneficiaries`). In lines 10–16, the `receive()` function is defined, which is executed whenever a transaction is sent to the contract account. It calculates each beneficiary's due (`individualAmount`) by dividing the amount of received ether (`msg.value`) by the number of beneficiaries (`beneficiaries.length`), i.e., three. Lastly, a `for` loop iterates

[76] Bärenfänger, *Blockchain tokens: a review*.

[77] L. K. Bärenfänger and other contributors. *What are the flaws of this example contract?* 2020. URL: `https://ethereum.stackexchange.com/questions/83782/what-are-the-flaws-of-this-example-contract` (visited on 12/31/2020).

[78] Bärenfänger, *Blockchain tokens: a review*, p. 13.

through each of the three participants and transfers them their respective share (`beneficiaries[i].transfer(individualAmount)`). Note that this implementation has some serious drawbacks, which are accepted for the sake of simplicity.

As mentioned earlier, one of the most popular applications of smart contracts is to leverage those to implement alternative tokens on top of Ethereum or similar blockchain systems. In the Ethereum whitepaper, the following minimalist example of a "token system in Serpant" is given:[79]

```
1 def send(to, value):
2   if self.storage[msg.sender] >= value:
3     self.storage[msg.sender] =
4       self.storage[msg.sender] - value
5     self.storage[to] = self.storage[to] + value
```

A smart contract representing a token system

The basic idea is that the storage assigned to the contract account (`self.storage`) is used to keep track of each participant's balance of the new token. In line 1, the `send()` function is defined, which is called in order to transfer a given amount of the token (`value`) to another participant (`to`). In line 2, the storage assigned to the debtor (`self.storage[msg.sender]`) is accessed to verify that their funds are sufficient to cover the transfer of the amount in question (`>= value`). If this is the case, in lines 3–4, the debtors's balance is reduced by said amount (`self.storage[msg.sender] - value`) while the creditor's is increased by it (`self.storage[to] + value`).

2.2.4 Permissioned systems

Blockchain systems such as Bitcoin and Ethereum are, today, classified as permissionless blockchain systems. This is due to the fact that they enable consensus in open-access networks where nodes may join and leave freely at all times, as there is no gatekeeper.

In contrast to this, industry has introduced what are commonly referred to as permissioned blockchains. Typically, these are based on closed, i.e., gated, networks that consist of nodes that belong to a company or a conglomerate of companies and maintain information that oftentimes is not publicly accessible. Blockchain systems with the latter property are also referred to as private blockchain systems.

[79] Buterin, *Ethereum white paper (updated version by the Ethereum Foundation)*.

IBM defines permissioned blockchain systems as to "[e]stablish decentralized trust in a network of known participants rather than a public network with no identity."[80] This definition is, obviously, contradictory: A "network of known participants" is, by definition, centralized, as there has to be some form of gatekeeping mechanism in place that is controlled by a central party responsible for network admission. Claiming to establish "decentralized trust" in a centralized network makes no sense.

As mentioned before, it is the main contribution of blockchain technology to facilitate agreement between the nodes of a decentralized – in other words, unpermissioned – network. Thus, one might wonder what the contribution of permissioned blockchains is at all, since "[t]raditional Byzantine agreement solves the consensus problem in a closed system of N nodes"[81] – since 1980.[82] A similar line of reasoning is presented by former JPMorgan researcher Stuart Popejoy with regards to Hyperledger Fabric, a popular, IBM-backed permissioned blockchain system:[83]

> IBM's definition of blockchain captures the distributed and immutable elements of blockchain but conveniently leaves out decentralized consensus – that's because IBM Hyperledger Fabric doesn't require a true consensus mechanism at all. Instead, it suggests using an "ordering service" called Kafka, but without enforced, democratized, cryptographically-secure voting between participants, you can't really prove whether an agent tampers with the ledger. In effect, IBM's "blockchain" is nothing more than a glorified time-stamped list of entries.

All this is not to say that permissioned blockchain systems do not add value. Without question, they do, e.g., by transferring new paradigms that have emerged in the context of (public) blockchain technology into the already established domain of private networking – most notably, the aforementioned immutability, which results from the chaining of hashes of subsequent data sets. However, from a computer science point of view, permissioned blockchain systems do not bring anything new to the table.

[80] IBM. *Hyperledger Fabric: the flexible blockchain framework that's changing the business world*. 2021. URL: https://www.ibm.com/blockchain/hyperledger (visited on 01/01/2021).

[81] D. Mazières, L. Giuliano, and E. Gafni. *Simplified SCP*. 2019. URL: https://www.scs.stanford.edu/~dm/blog/simplified-scp.pdf (visited on 03/15/2021), p. 1.

[82] L. Lamport, M. Pease, and R. Shostak. "Reaching agreement in the presence of faults". In: *Journal of the Association for Computing Machinery* (1980). URL: https://lamport.azurewebsites.net/pubs/reaching.pdf.

[83] S. Popejoy. *IBM's Hyperledger isn't a real blockchain – here's why*. 2019. URL: https://thenextweb.com/podium/2019/05/05/ibms-hyperledger-isnt-a-real-blockchain-heres-why/ (visited on 01/01/2021).

Thus, as stated before, in the remainder of this thesis, blockchain technology is to be interpreted in its original and, arguably, true sense, i.e., in the sense of public blockchain systems based on decentralized networks. Analyzing the concept of finality in the context of permissioned blockchain systems would result in a different and much shorter discussion, as these have essentially the same properties as the centralized solutions that traditionally underpin payment systems.

Chapter 3

How is finality defined in the literature?

This chapter examines how the concept of settlement finality is defined in the literature. Since the most prominent incident regarding finality occurred within the domain of foreign exchange, finality will first be elaborated on in this context. Finality is then examined in the more general context of wholesale payment systems. Finally, a definition of finality is derived from the preceding accounts in the literature.

3.1 Foreign exchange systems

In the following, a first account of finality is derived from analyzing an incident that occurred in the realm of foreign exchange as well as its aftermath, since this illustrates well what finality is and why it is important in the first place.

3.1.1 Herstatt incident

The concept of settlement finality is commonly discussed in the context of settlement risks arising from the absence thereof. Such risks may have first come to the attention of a wider public in 1974, with the collapse of Herstatt bank, as the German bank's failure caused severe disruptions in international payments. The publication "Settlement risk in foreign exchange markets and CLS Bank,"[1] which the Bank of International Settlements (BIS) published in 2002, describes the incident as summarized in the following paragraph:[2]

"On 26 June 1974, at 15:30 CET," German regulators forced Herstatt bank, which was "very active in foreign exchange markets," into liquidation, when some of its

[1] G. Galati. "Settlement risk in foreign exchange markets and CLS bank". In: *BIS Quarterly Review* (2002). URL: https://www.bis.org/publ/qtrpdf/r_qt0212f.pdf.

[2] Ibid., pp. 55, 56.

counterparties had already paid "large amounts of Deutsche [M]arks" to it as the respective Deutsche Mark legs of foreign exchange transactions. As "all US dollar payments from the German bank's account" were "suspended" due to the insolvency proceedings, the corresponding US dollar legs of those transactions were never completed, though. As the Deutsche Marks had been paid "irrevocably," Herstatt's counterparties "became fully exposed to the value of those transactions." To make matters worse, "[o]ther banks [...] refused to make payments [...] until they received confirmation that their countervalue had been received," which caused a chain reaction, as "[t]hese disruptions were propagated further through the multilateral net settlement system [that was] used." The incident was the "first and most dramatic case [...] where incomplete settlement of foreign exchange transactions caused severe problems in payment and settlement systems."

Following the incident, "[t]he risk that one party in a foreign exchange trade pays out the currency it sold but does not receive the currency it bought," i.e., FX settlement risk, became known as Herstatt risk.[3]

Herstatt risk arises when "separate legs of a foreign exchange transaction are settled independently and in many cases at significantly different times."[4] Two decades after the Herstatt incident, in 1995, a lag of up to two business days between "the time when a party to a foreign exchange transaction can no longer cancel unilaterally a payment instruction for the currency it sells and the time when the currency purchased has been received" was still found to be common.[5] This can be attributed to payment systems not operating to a timetable that permits "simultaneous or near simultaneous settlement" and whose operating hours' overlap between time zones is "limited."[6]

From these remarks, a preliminary notion of finality can be derived. The Deutsche Mark legs of the FX trades from the Herstatt incident were described as "irrevocably" settled when they could no longer be rescinded or reversed. In that sense, they can be considered settled with finality. The FX transactions as a whole, however, were everything but settled with finality, which is why the completion of their respective other legs, i.e., the US dollar legs, could be prevented by German authorities, which, in turn, gave rise to the disruptive aftermath of Herstatt's failure.

3.1.2 Payment-vs-payment principle

To reduce Herstatt risk, several initiatives were undertaken. One the one hand, through the adoption of RTGS systems, "[i]ntraday final settlement was introduced

[3] Galati, "Settlement risk in foreign exchange markets and CLS bank", p. 57.
[4] Ibid., p. 56.
[5] Ibid., p. 56.
[6] Ibid., p. 56.

more widely" in order to "shorten the duration of settlement exposures."[7] On the other hand, "bilateral and multilateral arrangements for [...] netting" were introduced in order to reduce the "number and size of payments requiring settlement."[8] However, although these initiatives "reduced either the size or the duration of settlement exposures," "simultaneous finality of received payments" was not achieved and, hence, settlement risks in FX transactions were decreased but not eliminated.[9]

Thus, "[i]n the mid-1990s," "to tackle the problem of settlement risk," the G20 banks, "a group of major foreign exchange market participants," developed a solution based on the payment-versus-payment (PvP) principle, which aims to ensure the simultaneous settlement of "the two legs of a transaction" by stipulating that "one cannot occur without the other."[10] PvP is also described as a "settlement standard" where "funds of two counterparties are transferred simultaneously and one transfer is only considered final if the counter-transfer is final as well."[11] PvP thereby eliminates "the most important settlement risk, where one counterparty transfers the owed funds without receiving the counter-payment," i.e., Herstatt risk.[12]

CLS Bank International, an acronym for "continuous linked settlement," was set up to develop the G20 bank's PvP-based solution in 1997 and went into operation in 2002, "settling transactions involving seven currencies," including the euro and the US dollar.[13] The aforementioned BIS publication on settlement risks in FX describes the settlement phases of CLS as summarized in the following steps:[14]

1. Submission of transactions: "[M]embers submit [...] transactions to be settled by [...] 00:00 CET."
2. Calculation of settlement positions: "CLS Bank [...] calculates each [...] member's net total pay-in/pay-out position for each currency and at 06:30 CET issues a pay-in schedule."
3. Funding of settlement accounts: "Payments to CLS Bank are executed between 07:00 and 12:00 CET."
4. Settlement of transactions: "CLS Bank settles each trade over these accounts by simultaneously crediting the buyer's account in the currency that is bought and debiting the seller's account in the currency that is sold."

[7] Galati, "Settlement risk in foreign exchange markets and CLS bank", p. 59.
[8] Ibid., p. 59.
[9] Ibid., p. 59.
[10] Ibid., p. 60.
[11] A. Schaller. *Continuous linked settlement: history and implications*. 2007. URL: `https://www.zora.uzh.ch/id/eprint/163690/1/20080261.pdf` (visited on 03/16/2021), p. 6.
[12] Ibid., p. 6.
[13] Galati, "Settlement risk in foreign exchange markets and CLS bank", pp. 60, 61.
[14] Ibid., pp. 61, 62.

From these steps, it can be inferred that there is a "clear distinction" between the funding of settlement accounts and the settlement of transactions, as the former are funded on a net basis while the latter are settled on a gross basis.[15]

It can be said that "CLS eliminates credit risk [arising in FX trades] in all but very extreme circumstances" by leveraging the "payment-versus-payment principle and the positive account balance rule."[16] With regards to liquidity risk, the issue is, however, "more complex," since "CLS Bank is not automatically able to pay out to other members in the currencies due."[17]

It must be stressed, however, that PvP-based solutions, including CLS, do not prevent FX transaction legs from being reversed, since such solutions do not shield the latter from disruptive legal effects. They do, however, alleviate Herstatt risk, as they technically ensure that either both transaction legs settle with finality or neither. Thus, PvP-based solutions do not enable finality but solve a problem arising from the inability to do so – a problem specific to FX, though.

The reason PvP and, specifically, CLS is presented in this context is not only due to the fact that it is typically mentioned in publications on finality – it is because PvP represents an archetypal application of smart contracts, corresponding to the very kind of agreement whose digital formalization and execution was envisioned by Nick Szabo and is now enabled by general-purpose blockchain systems such as Ethereum.

It can be summarized that the PvP principle and solutions based on it effectively provide simultaneous settlement of the legs of an FX transaction, thereby greatly alleviating risks arising in the context of finality. Albeit being specific to FX, PvP-based solutions are relevant in the context of this thesis, as they are straightforward to implement atop of blockchain systems in the form of smart contracts.

3.2 Wholesale payment systems

In the following, the previously given preliminary account of finality is generalized with regards to wholesale payment systems. In the course of this, both DNS and RTGS are examined in this context. Furthermore, the EU's Settlement Finality Directive, which is taken as an example of legislation intended to enable finality in such systems, is analyzed to identify the notion of finality assumed therein.

[15] Galati, "Settlement risk in foreign exchange markets and CLS bank", p. 62.
[16] Ibid., p. 62.
[17] Ibid., p. 63.

3.2.1 Deferred net settlement

While the failure of Herstatt bank and its aftermath serve as an illustrative example of settlement risks arising from the inability to achieve finality, it is obvious that not only FX trades are exposed to such risks. In fact, settlement risks associated with finality or, rather, the lack thereof, are typically discussed in the broader context of payment systems in general, i.e., in the context of DNS systems employing net settlement and RTGS systems employing gross settlement.

One of the previously mentioned initiatives that were undertaken to limit settlement risks was the application of netting, as it reduces the "number and size of payments requiring settlement."[18] This advantage follows directly from the way netting works: If transactions are not settled individually but offset against each other until, at some point, each participant's net settlement position is transferred, there are, in total, fewer transactions to be settled ("number") and the respective net settlement positions are smaller than the sum of the individual transfer order values ("size"). Netting, therefore, reduces the "number of settlement cycles," the "value of the [t]ransfer [o]rders" – which, in turn, reduces "the size of the credit [and] liquidity risk exposures incurred by [...] participants" – as well as the overall "cost," as "less back-office capacity" and "[c]ollateral" are needed.[19]

It cannot be overstated, however, that the advantages of netting regarding settlement risks are predicated on the respective netting arrangement to have a strong legal footing, which is discussed in detail in the following paragraphs. If, however, said legal footing is lacking, netting does not only not have the aforementioned advantages but, in addition, in and of itself introduces the risk of not achieving final settlement. In other words, a DNS system without a strong legal basis does not reduce but exacerbate settlement risks and, consequently, systemic risk.

Such risks are present if the legal basis of a payment system employing netting is such that already submitted transfer orders and, therefore, the netting as a whole are not protected from legal challenge. The central question, therefore, is whether "transactions [are] hon[o]red as final" or if they, instead, "could [...] be considered void or voidable by liquidators and relevant authorities."[20]

Analogously to the Herstatt incident, this is commonly discussed in the context of a participant becoming insolvent. A competent liquidating authority might, "in some jurisdictions," legally challenge a netting "by performing only those forward contracts that are profitable to the [insolvency] estate while repudiating those [...] that

[18] Galati, "Settlement risk in foreign exchange markets and CLS bank", p. 59.
[19] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 40.
[20] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 16.

are unprofitable" – a practice referred to as cherry picking.[21] In other words, outgoing payments of the insolvent participant would be blocked and would effectively become unsecured claims against the assets of the insolvency estate, resulting in the failing participant's inability to pay their net settlement position at settlement time.

Obviously, cherry-picked transfer orders must be removed from the "netting calculation," i.e., "the entire netting process must be reversed," which is referred to as unwinding.[22] As the repudiated transfer orders, i.e., those that are "unprofitable to the [insolvency] estate," are "precisely those that are profitable to the [...] counterparty," that counterparty, be it an individual participant or a CCP, becomes exposed to the gross value of that transfer order.[23]

Since unwinding can drastically change the net settlement position of other participants, the aforementioned reduction of settlement risks attributed to netting does not apply in jurisdictions where cherry picking is legal. In this case, i.e., if a netting arrangement does not have a strong legal footing, netting "merely obscures the levels of exposure [to credit risk and liquidity risk]"[24] and thereby exacerbates these settlement risks.

Moreover, "unexpected and sizeable change[s]" of other participants' obligations might, in fact, "result [...] in their inability to settle."[25] This can lead to a ripple effect, setting off a chain of defaults that could even go beyond the scope of the payment system. Thus, if not legally sound, a netting arrangement introduces systemic risk, too.

3.2.2 Real-time gross settlement

It must be pointed out that, in principle, the aforementioned settlement risks also apply to systems that settle on a gross basis – not only to those that settle on a net basis. If transfer orders are not protected from legal challenge, a transfer order submitted to an RTGS system is equally at risk to be cherry-picked as if it had been submitted to a DNS system.

There is, however, a crucial difference between the two types of systems, i.e., the period of time between a transfer order's acceptance by a system for settlement and final settlement. In DNS systems, which typically settle at the end of the day,

[21] Committee on Payments and Market Infrastructures. "Report of the committee on interbank netting schemes of the central banks of the Group of ten countries (Lamfalussy report)". In: *CMPI Papers* (1990). URL: https://www.bis.org/cpmi/publ/d04.htm, p. 8.

[22] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 41.

[23] Committee on Payments and Market Infrastructures, "Report of the committee on interbank netting schemes of the central banks of the Group of ten countries (Lamfalussy report)", p. 8.

[24] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 40.

[25] Ibid., p. 41.

that period of time is much longer than in RTGS systems, which may not always provide immediate settlement but do typically settle at multiple intervals during the day, thereby at least approximating real-time settlement. If said period of time is long enough, it is feasible for a participant to submit transfer orders, go insolvent and, as a consequence of this, have those very transfer orders cherry-picked – all before final settlement. The Herstatt incident is an example of this. Such scenario is, however, much less likely if there are only, say, minutes between transfer order submission and final settlement.

The main advantage of RTGS over DNS is, therefore, that the former "substantially reduces the duration of credit [and] liquidity [...] risk exposures," which, in turn "reduces systemic risk."[26] This is why the adoption of RTGS systems was one of the previously mentioned initiatives that were undertaken to limit settlement risks. Furthermore, as there is no netting when settling on a gross basis, there, consequently, is no netting calculation to be unwound in case of cherry picking. Thus, "RTGS precludes the possibility of unwinding payments."[27] Finally, with gross settlement, "[s]ettlement pressures are not concentrated at particular points in time."[28]

The reason why DNS is oftentimes preferred over RTGS comes down to cost – and has only little to do with settlement risks. The higher cost of RTGS not only follows from the fact that transfer orders are processed individually but are also the result of a higher demand in liquidity, "because participants need sufficient liquidity to cover [each of] their outgoing payments"[29] individually, as they are not offset against incoming payments. In DNS, to the contrary, "intraday liquidity is provided by participants in the system" but comes at the cost of "credit and liquidity risks."[30]

It can be summarized that DNS decreases settlement risks arising from the inability to achieve finality by reducing the number and size of transactions to be settled – if, and only if, transfer orders and netting calculations cannot be legally challenged. RTGS, on the other hand, decreases such risks by reducing the duration of exposure to them. The main advantage of DNS is that it is cheaper, as it leverages batch processing and requires less liquidity. The main advantage of RTGS is that – while "[i]n normal times, both DNS and RTGS networks can provide the same assurance of payment finality" – in times of "crises," RTGS "provides greater

[26] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 43.
[27] Ibid., p. 43.
[28] Ibid., p. 43.
[29] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 26.
[30] Ibid., p. 26.

assurance of payment finality and uninterrupted financial market operation in the event of multiple [...] participant failures."[31]

3.2.3 Settlement Finality Directive

The previously described settlement risks associated with the inability to achieve finality are, for the most part, consequences of the insufficient legal soundness of a payment system. Thus, the Settlement Finality Directive (SFD), formally, Directive 98/26/EC of the European Parliament and of the Council of 19 May 1998 on settlement finality in payment and securities settlement systems, was conceived "to reduce systemic risk [resulting from the settlement risks arising from inability to achieve finality] by removing various areas of uncertainty in payment and securities settlement systems."[32]

The parts of the SFD that are relevant in the context of this thesis are those that provide insight into what concept of finality is assumed and how said notion of finality is enabled by the SFD. To this end, the SFD's Articles 3, 4, 5, 7, 8 and 9(1) are discussed in more detail in the following paragraphs.

Article 3 provides the protection of netting from insolvency law, i.e., if a participant goes insolvent during the day, liquidating authorities cannot unwind the netting calculation to be settled at the end of that day.[33] The relevant excerpt from Article 3 reads as follows:[34]

> 1. Transfer orders and netting shall be legally enforceable and binding [...] even in the event of insolvency proceedings against a participant, provided that transfer orders were entered into the system before the moment of opening of such insolvency proceedings [...]. [...]
> 2. No law [...] on the setting aside of [...] transactions concluded before the moment of opening of insolvency proceedings [...] shall lead to the unwinding of a netting.
> 3. The moment of entry of a transfer order into a system shall be defined by the rules of that system. [...]

[31] H. Pagès and D. Humphrey. "Settlement finality as a public good in large-value payment systems". In: *ECB Working Paper Series* (2005). URL: `https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp506.pdf`, p. 5.

[32] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 19.

[33] Ibid., p. 19.

[34] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.

Article 3(1) stipulates that, "in principle,"[35] transfer orders entered into a system by an insolvent participant are enforceable if entered before the opening of the participant's insolvency proceedings. "In practice," however, transfer orders "entered after the opening" of such proceedings are protected as well, "insofar as they are 'carried out on the day of the insolvency proceedings.'"[36] Such payments "must [after settlement] be restituted by the beneficiary to the liquidator," though, as they "would otherwise fraudulently advantage the beneficiary to the detriment of the other unsecured creditors."[37]

This touches upon an important principle: the distinction between settlement finality and "obligation finality."[38] The SFD "aims at ensuring settlement finality," i.e., ensuring that "[t]ransfer [o]rders entered in a system are protected, so that the system can settle," while it does not aim at protecting the "legal validity and enforceability of the underlying transaction [or obligation]."[39] Accordingly, the SFD protects the settlement of transfer orders entered after the opening of insolvency proceedings if settlement is carried out on that day, while it does not, however, shield the underlying transaction from having to be restituted afterwards.

It can be concluded that Article 3(1) stipulates that "Member States must recogni[z]e the concept of netting," and therefore guarantees the "sound legal basis" that netting requires to "avert cherry picking."[40] Thus, Article 3(1) can be said to exclusively concern netting – "the finality of RTGS [t]ransfer [o]rders is guaranteed by other provisions of the Directive."[41]

Article 3(2) can be summarized to "confirm" the sentiment that the legislator's "overriding concern" is to avert "the unwinding of netting."[42]

Article 3(3) reflects the "specificities and complexity of [...] systems," and, consequently, leaves the definition of when exactly a transfer order is considered to have entered the system up to the participants.[43]

Article 4 provides the protection of settlement accounts from insolvency law, i.e., if a participant goes insolvent, liquidating authorities cannot prevent funds from that participant's settlement account from being used to settle. The relevant excerpt from Article 4 reads as follows:[44]

---

[35] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 46.
[36] Ibid., p. 47.
[37] Ibid., p. 49.
[38] Ibid., p. 48.
[39] Ibid., p. 48.
[40] Ibid., p. 45.
[41] Ibid., p. 45.
[42] Ibid., p. 48.
[43] Ibid., p. 50.
[44] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.

> Member States may provide that the opening of insolvency proceedings against a participant [...] shall not prevent funds or securities available on the settlement account of that participant from being used to fulfil that participant's obligations in the system [...] on the business day of the opening of the insolvency proceedings. [...]

As it is "typically" the "effect of an insolvency declaration" that the insolvent participant has their assets frozen, "settlement of the insolvent [...] participant's obligations would become impossible," even if their settlement account were funded.[45] To enable settlement, "Article 4 cancels this effect with regard to settlement accounts in a system."[46]

Article 5 provides the protection of transfer orders from insolvency law, i.e., from the moment a transfer order is processed in the system, it is ensured that it will be completed, "even if the inputting institution fails in the meantime."[47] The relevant excerpt from Article 5 reads as follows:[48]

> A transfer order may not be revoked by a participant in a system, nor by a third party, from the moment defined by the rules of that system. [...]

Revocation can be defined as the "voluntary act of rescinding a [t]ransfer [o]rder entered in a system," which can be effected by either the "sending [i]nstitution or a third party," e.g., the "instructing customer."[49]

From "the law of some Member States," another account of finality can be derived, which states that "a payment is not final until it reaches the beneficiary's account."[50] If such rules apply, the transfer orders instructing the "transfer [of] money or securities" are effectively opened up for revocation until "the money or securities are booked on the beneficiary's account."[51] This would, in turn, give rise to "systemic risk," e.g., because "system rules barring revocation would not be enforceable" since "mandatory rules from law" trump contractual agreements.[52]

Thus, Article 5 stipulates that "neither an [i]nstitution nor a third party can revoke a [t]ransfer [o]rder after a certain point in time [which is defined by the participants of a system]."[53]

---

[45] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 51.
[46] Ibid., p. 51.
[47] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 19.
[48] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.
[49] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 51.
[50] Ibid., p. 52.
[51] Ibid., p. 52.
[52] Ibid., p. 52.
[53] Ibid., p. 52.

Article 7 provides the prohibition of "retroactive effects of insolvency rules on rights and obligations in systems," i.e., such rights and obligations are shielded from "backdating [...] effects of an insolvency," e.g., zero-hour rules.[54] Article 7 reads as follows:[55]

> Insolvency proceedings shall not have retroactive effects on the rights and obligations of a participant arising from, or in connection with, its participation in a system earlier than the moment of opening of such proceedings.

Retroactive effects of an insolvency most prominently arise from zero-hour rules or suspect period rules. Zero-hour rules render "all transactions by a bankrupt participant void from the start ('zero hour') of the day of the bankruptcy," which, in DNS systems, "could cause the netting of all transactions to be unwound," or, in RTGS systems, "could [...] reverse payments that have apparently already been settled and were thought to be final."[56] Suspect period rules extend this concept – and its effects – by encompassing transactions that were entered even earlier, before the start of the day of the insolvency proceedings, i.e., during some "suspect period."[57]

Therefore, Article 7 "aims to disapply the rules that provide for automatic annulment [...] of certain transactions entered into [a system] shortly before the insolvency proceedings."[58] However, as the SFD is concerned with settlement finality and not obligation finality, Article 7 does not aim to disapply rules that "can lead to [such] a transaction being undone" after the underlying transfer order "is [...] long settled by this time."[59]

Article 8 provides the resolution of conflicts "between the system rules and the home country insolvency law of a foreign participant," as the "law governing a system" is stipulated to be the binding law in case of insolvency of that participant.[60] Article 8 reads as follows:[61]

> In the event of insolvency proceedings being opened against a participant in a system, the rights and obligations arising from, or in connec-

[54] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 19.
[55] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.
[56] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 17.
[57] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 57.
[58] Ibid., p. 57.
[59] Ibid., p. 57.
[60] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 19.
[61] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.

> tion with, the participation of that participant shall be determined by the law governing that system.

For each participant "from another Member State," it is obligatory to "determine [...] whether the system rules [...] will, in the event of that participant's insolvency, be enforceable" under or be in conflict with that Member State's insolvency law.[62] As such a beforehand analysis of the "potential impact of foreign insolvency legislation" is at least "complex," if not impossible,[63] Article 8 stipulates the law applicable in the event of insolvency proceedings is the law that governs the system – not the law of the insolvent participant's Member State.

Article 9(1) provides the insulation of collateral from insolvency law, so "it can be used to clear the debts to a system of a failed participant."[64] The relevant excerpt from Article 9(1) reads as follows:[65]

> The rights of a system operator or of a participant to collateral security provided to them in connection with a system [...] shall not be affected by insolvency proceedings [...]. Such collateral security may be reali[z]ed for the satisfaction of those rights.

In the event of a participant's default "it must be possible to quickly reali[z]e the [c]ollateral provided to the system" in order for the system to settle.[66] However, insolvency law of most Member States would prevent collateral securities from being realized within a system, just like it would prevent outgoing payments in order to not disadvantage other creditors and their unsecured claims. Furthermore, even if this were not the case, the realization of such collateral would "typically [require to] obtain a court order," which, since "the underlying assets [could] not be reali[z]ed immediately upon occurrence of the default," would prevent the system from (timely) settlement.[67] Therefore, Article 9(1) stipulates that "the [c]ollateral provider's insolvency should not have any effect on the [c]ollateral provided to a system."[68]

It can be summarized that the SFD contributes to the ability of payment settlement systems to achieve finality by shielding the components of such systems, e.g., netting calculations, transfer orders and settlement accounts, from the potentially disruptive impact of the Member States' insolvency laws and similar leg-

[62] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 58.
[63] Ibid., p. 58.
[64] Committee on Payments and Market Infrastructures, "Core principles for systemically important payment systems", p. 19.
[65] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.
[66] Vereecken and Nijenhuis, *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*, p. 60.
[67] Ibid., pp. 60, 61.
[68] Ibid., p. 61.

islation. It, furthermore, does so by harmonizing relevant legislation across the Member States and clearly stipulating whose laws apply when, so cross-border systems are legally sound as well.

3.3 Literature-derived finality definition

At this point, a more comprehensive account of finality can be derived. As mentioned before, finality is relevant in payment systems, as it is directly related to the settlement risks that participants incur. The previously given preliminary notion of finality characterizes final settlement as irrevocable in the sense that a transaction can no longer be rescinded or reversed. Conversely, the absence of final settlement can be characterized in the sense that such revocation is still possible, e.g., as an effect of insolvency law. Such notion of finality as an indicator of irrevocable settlement is in line with the first part of the finality definition by the BIS's Committee on Payments and Markets Infrastructures (CPMI), which considers final settlement "[t]he irrevocable and unconditional transfer of an asset or financial instrument."[69]

The second part of the CPMI's definition provides an alternative account of final settlement by equating it to "the discharge of an obligation by the FMI or its participants in accordance with the terms of the underlying contract," where "FMI" stands for "financial market infrastructure," which, in turn, is defined as "[a] multilateral system [...] used for the purposes of clearing, settling or recording payments, securities [...] or other financial transactions."[70] Note that said "obligation" does not refer to a liability arising from a contract outside the system's scope but to an obligation arising within the system, e.g., from submitting a transfer order. Thus, settlement finality is not equated to what was previously defined as obligation finality. The notion of finality as the discharge of an obligation within a payment system is in line with the previous remarks on finality, which overarchingly discussed protecting a transfer order from legal challenge until it is settled – in other words, until the obligation to or by the system is discharged.

The CPMI's definition ends with the crucial remark that final settlement "is a legally defined moment."[71] Both the observation that finality is a legal construct and that it refers to a moment in time are in line with the previous remarks on finality, which exclusively examined legal means through which a payment can be disrupted – and up to which point in time – as well as legal means through which said disruption can be prevented.

[69] Committee on Payments and Market Infrastructures, *A glossary of terms used in payments and settlement systems*, p. 8.
[70] Ibid., p. 8.
[71] Ibid., p. 8.

There is another definition of finality that is in accordance with the aforementioned insights but puts emphasis on the financial stability aspect of the issue. It was proposed in a European Central Bank (ECB) Working Paper from 2005:[72]

> Finality of settlement ensures that transactions made over payment networks will, at some point, be complete and not subject to reversal even if the parties to the transaction go bankrupt or fail. It is the assurance that even in times of financial system uncertainty, turmoil, or crisis the transaction being undertaken will go through.

For the sake of completeness, it is necessary to consider the widely cited definition of finality provided by Canadian law professor and expert in payment and settlement systems Benjamin Geva, which identifies three different accounts of finality:[73]

> In connection with a non-cash payment through the banking system, "finality of payment" has acquired diverse meanings. In one sense, it has come to denote the irreversibly of the payment process, particularly in connection with insolvency. Otherwise, it has also been taken to signify the loss of the right to recover a mistaken payment. Finally, it has been used to mark the accountability to the payee/beneficiary by a bank instructed to pay to that payee/beneficiary.

Obviously, the previous remarks assume the first-listed meaning of finality – and so does the remainder of this thesis. The second-listed meaning goes beyond the scope of a transaction within a payment system, as it refers to the possibility of recovering a payment that is settled with finality. This account of finality is, therefore, rather to be interpreted in the sense of what has previously been called obligation finality. The third-listed meaning is very specific to traditional banking systems and their architectures, and, therefore, is less relevant in the context of blockchain technology.

It can be summarized that the preceding survey of the literature has unilaterally shown that settlement finality is regarded a legal construct – and the inability to achieve it a legal problem. Therefore, this traditional notion of finality is referred to as "legal finality" in the remainder of this thesis.

[72] Pagès and Humphrey, "Settlement finality as a public good in large-value payment systems", p. 6.

[73] B. Geva. "Payment finality and discharge in funds transfers". In: *Chicago-Kent Law Review* (2008). URL: `https://scholarship.kentlaw.iit.edu/cklawreview/vol83/iss2/7`, pp. 633, 634.

Chapter 4

Can blockchain systems provide legal finality?

As finality is traditionally seen as a legal concept, this chapter examines whether blockchain-based payment systems can provide said notion of legal finality. To this end, a general overview of regulatory approaches to blockchain technology is given. Then, it is assessed whether systems based on it are covered by the SFD, which once again is used as an example of similar such laws .

4.1 Regulatory approaches

In the following, two approaches to regulating blockchain technology are characterized, namely what I consider "indirect" and "direct regulation."

4.1.1 Indirect regulation

At the time of this writing, blockchain systems have existed for well over a decade and, if the current Bitcoin bull run[1] is any indication, are as popular as ever – all the more reason for regulators to address the technology if they have not already. In practice, regulators must choose between "three paths [...] regarding regulation in this area," i.e., banning and/or restricting, ignoring or attempting to "provide clarity and a regulatory framework regarding how such activities can operate within the jurisdiction."[2]

While the first option is, due to the decentralized nature of blockchain systems, "extremely hard, if not impossible, to enforce" and the second option may lead to

[1] N. Popper. *Bitcoin hits new record, this time with less talk of a bubble*. 2021. URL: https://www.nytimes.com/2020/11/30/technology/bitcoin-record-price.html (visited on 01/14/2021).

[2] J. Ellul et al. "Regulating blockchain, DLT and smart contracts: a technology regulator's perspective". In: *ERA Forum* (2020). URL: https://link.springer.com/article/10.1007/s12027-020-00617-7, p. 210.

the emergence of "legal uncertainty," driving away "stakeholders," it is the third option – "legal certainty [...] provided through a regulatory regime" – that fosters innovation.[3] Note that albeit this assessment was originally made in the specific context of cryptocurrencies, it can, arguably, be applied to blockchain technology as a whole.

As to how regulators can and do go about providing said legal certainty with regards to blockchain technology, it is to be pointed out that "[t]raditionally, the technology which enables a regulated financial product or service is considered to be outside the purview of the law – it is the actions of the parties involved in providing or using the services that are to be regulated."[4] Consequently, it is possible for legislators to introduce legal certainty to the blockchain domain without having to explicitly address the blockchain technology that underpins the systems in question – a practice that, in the remainder of this thesis, is referred to as "indirect regulation of blockchain technology" – for lack of better terminology.

This appears to be the go-to approach in many jurisdictions. One example of this has already been presented, namely the Liechtenstein Blockchain Act, which does not directly target blockchain technology but defines roles for those that interact with systems based on it, e.g., the aforementioned physical validator role, and regulates those who conform to these roles. Another example of this is what became known as the BitLicense, which any "person (whether an individual or a company) that engages in Virtual Currency Business Activity" in New York State, US, must apply for.[5]

The role that seems to be under most regulatory scrutiny is that of the exchanges where blockchain tokens and legal tender are traded, as "for regulators Bitcoin exchanges are the most logical institutional choke point in the Bitcoin ecosystem."[6] One of the main reasons for this is that the participants of many blockchain systems remain pseudonymous[7] or anonymous,[8] which poses challenges, e.g., in the context of anti money laundering (AML). Thus, exchanges – which, by contrast, are legally tangible institutions – are commonly required to identify their customers. Generally speaking, exchanges are targeted by regulatory efforts because, in many respects, they are seen as the link between the oftentimes (pseudo-)anonymous

[3] Ellul et al., "Regulating blockchain, DLT and smart contracts: a technology regulator's perspective", p. 210.

[4] Ibid., p. 211.

[5] Department of Financial Services, New York State, US. *Virtual currency businesses: BitLicense FAQs*. 2021. URL: https://www.dfs.ny.gov/apps_and_licensing/virtual_currency_businesses/bitlicense_faqs (visited on 01/10/2021).

[6] M. Tsukerman. "The block is hot: a survey of the state of Bitcoin regulation and suggestions for the future". In: *Berkeley Technology Law Journal* (2015). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2587421, p. 1153.

[7] R. Zhang, R. Xue, and L. Liu. "Security and privacy on blockchain". In: *ACM Computing Surveys* (2019). URL: https://arxiv.org/pdf/1903.07602.pdf, p. 6.

[8] V. van Saberhagen. *CryptoNote v2.0*. 2013. URL: https://github.com/monero-project/research-lab/blob/master/whitepaper/whitepaper.pdf (visited on 03/16/2021).

on-chain world and traditional, well regulated financial infrastructures. One example of this is the Foreign Accounts Tax Compliance Act (FATCA) in the US, as exchanges correspond to foreign financial institutions (FFI) as defined therein, which "are required to identify their US account holders to the Internal Revenue Service (IRS), or face a 30 percent gross tax on payments received from US sources."[9]

It can be summarized that in order to not stifle but foster innovation, regulators have no choice but to provide legal certainty in the blockchain domain. It appears as if the go-to approach for doing so is what can be referred to as indirect regulation of blockchain technology, i.e., the practice of regulating those who interact with blockchain systems in a given jurisdiction instead of the technology that underpins such systems. One of many examples of this is the Lichtenstein Blockchain Act.

4.1.2 Direct regulation

As emphasized previously, it is the facilitation of consensus in a decentralized setting that is the main contribution of blockchain technology. However, "the very same enabling features that bring decentrali[z]ation also pose challenges" in the context of regulation.[10] While these challenges are circumvented by the aforementioned approach of indirect regulation – where the technology itself, and with it its decentralized nature, are outside the purview of the law – they become evident when examining the extent to "which [...] government action can influence the blockchain payment systems in the first place," as "the question whether state regulation can, if desired so, effectively restrain blockchain payment systems is a valid and a non-trivial topic to investigate."[11] Such government-initiated influence or restraint directed at blockchain systems directly is, in the remainder of this thesis, referred to as "direct regulation of blockchain technology" – again, for lack of better terminology.

One of the central regulatory challenges arising from the decentralized structure of blockchain systems is the fact that such systems do "not exist in a central location but rather through a peer-to-peer [...] network composed of all [...] users."[12] To complicate matters further, the networks that make up such systems do not only span virtually all jurisdictions, they do so in a fluctuating way, as nodes enter and

[9] Tsukerman, "The block is hot: a survey of the state of Bitcoin regulation and suggestions for the future", p. 1151.

[10] Ellul et al., "Regulating blockchain, DLT and smart contracts: a technology regulator's perspective", p. 209.

[11] S. Shanaev et al. "Regulatory implications for the cryptocurrency market". In: *Research in International Business and Finance* (2019). URL: https://www.sciencedirect.com/science/article/abs/pii/S0275531919305963, p. 2.

[12] Tsukerman, "The block is hot: a survey of the state of Bitcoin regulation and suggestions for the future", p. 1128.

leave the system continuously. Consequently, it can be said that blockchain systems transcend jurisdictions in the traditional, i.e., territorial, sense of the word and "are necessarily exterritorial."[13] They, thus, fall outside any national legal framework or regulatory regime.

Another central regulatory challenge is a direct consequence of decentralization as well: "No particular party can be said to 'control' the blockchain."[14] In other words, there exists no one institution responsible for the operation of a blockchain system, as, again, said operation is done collectively by a transnational and fluctuating set of participants. This, after all, was the main motivation for creating the original blockchain system, Bitcoin, in the first place, as it "was launched on an ethos of anti-institutionalism."[15] Consequently, it can be said that no legally liable operator exists for blockchain systems.

The absence of a tangible physical location as well as the absence of a legally liable operator are only two examples of how the decentralized structure of blockchain systems de-facto shields such systems from being impacted by means of law enforcement – which, in turn, makes imposing legislation on them near pointless. One could, thus, argue that direct regulation of blockchain systems, as opposed to regulating those who interact with them in a given jurisdiction, seems infeasible, at least as it stands today.

One of many examples of how blockchain technology is de-facto beyond the reach of law enforcement concerns data protection and privacy legislation as "the immutability of [blockchain] records [...] could be in contradiction with" Article 17 of the EU's General Data Protection Regulation (GDPR), i.e., the right to be forgotten.[16] Yet, there appears to be no way to enforce the GDPR and actually have those blockchain records removed that are in breach of it. Generally speaking, "[t]he structure of blockchain records could [...] generate legal issues where regulators or laws could demand that erroneous or illegal transactions be unwound,"[17] while the reality seems to be that such demands are, in fact, unenforceable.

It must be pointed out, however, that these remarks assume regulatory efforts to be confined to individual or groups of nation states and are not to deny that "regulation, at least theoretically, can have a significant impact on [...] blockchain payment systems" or, rather, blockchain systems in general. One could, in fact, argue

[13] Shanaev et al., "Regulatory implications for the cryptocurrency market", p. 2.

[14] Tsukerman, "The block is hot: a survey of the state of Bitcoin regulation and suggestions for the future", p. 1129.

[15] P. Yeoh. "Regulatory issues in blockchain technology". In: *Journal of Financial Regulation and Compliance* (2017). URL: https://www.emerald.com/insight/content/doi/10.1108/JFRC-08-2016-0068/full/html, p. 6.

[16] European Securities and Markets Authority. "The distributed ledger technology applied to securities markets". In: *ESMA Report* (2017). URL: https://www.esma.europa.eu/sites/default/files/library/dlt_report_-_esma50-1121423017-285.pdf, p. 35.

[17] Yeoh, "Regulatory issues in blockchain technology", p. 7.

that "regulation can be effective, especially if it is enforced internationally."[18] As of today, however, this is not the case.

It can be summarized that the decentralized nature of blockchain systems is, among other things, the reason why such systems fall outside any legal framework and lack legally liable operators, which, in turn, makes them impossible to come by through means of law enforcement and, therefore, de-facto unregulatable, at least in the sense of direct regulation. This assessment is, however, predicated on the absence of internationally enforced regulation, as is the case today.

As a side note – for lack of a better place to do so – it must be pointed out that these anarchic characteristics did not come about accidentally. The Bitcoin paper can only be understood as to propose a way to deliberately avoid "mediating disputes" – legal disputes, one must assume – as traditional "financial institutions" are unable to do so.[19] Among other factors, it is this open disdain for government regulation why David Golumbia, professor of humanities at the Virginia Commonwealth University, attributes Bitcoin and similar projects to the realm of "cyberlibertarianism" that is consistent with "a holistic worldview that has been deliberately developed and promulgated by right-wing ideologues":[20]

> By far the majority of interest in Bitcoin came from technologists [...]. To those of us who were watching Bitcoin with an eye toward politics and economics, though, something far more striking than Bitcoin's explosive rise in value became apparent: in the name of this new technology, extremist ideas were gaining far more traction than they previously had outside of the extremist literature to which they had largely been confined. [...]
>
> To anyone aware of the history of right-wing thought in the United States and Europe, [these ideas] are shockingly familiar: that central banking [...] is a deliberate plot to 'steal value' from the people to whom it actually belongs; that the world monetary system is on the verge of imminent collapse due to central banking policies [...]; that 'hard' currencies such as gold provide meaningful protection against that purported collapse; that inflation is a plot to steal money from the masses and hand it over to a shadowy cabal of 'elites' who operate behind the scenes; and more generally that the governmental and corporate leaders and wealthy individuals we all know are 'controlled' by those same 'elites.'

On a personal note, I must emphasize that I am appalled that blockchain technology is likely rooted in extreme right-wing ideology. Although I did notice some

[18] Shanaev et al., "Regulatory implications for the cryptocurrency market", p. 2.
[19] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 1.
[20] D. Golumbia. *The politics of Bitcoin: software as right-wing extremism*. The University of Minnesota Press, 2016. ISBN: 9781452953816, pp. 5, 6.

clues along the way, e.g., the aforementioned sentiment of the Bitcoin paper and the support of Nick Szabo – someone in Satoshi Nakamoto's inner circle, after all – for Donald Trump and his extreme right-wing conspiracy theories,[21] I was blinded by the elegance of the technology and, regrettably, needed to stumble upon the research quoted above to make sense of its political origins.

While I believe that blockchain technology has been and will continue to be used for good, I, too, believe that everyone dealing with it needs to be made aware of its likely origins. In this thesis, this appeared to be the most fitting place to do so.

4.2 Legal finality in blockchain systems

Whether a payment system is able to provide legal finality depends on the regulatory framework it is embedded in. Thus, in assessing whether blockchain-based payment systems can achieve final settlement, it is crucial to establish whether they fall under the protection of legislation aiming to enable payment systems to do so, e.g., the SFD, which is analyzed in the following.

Previously, it was pointed out that blockchain technology is commonly regulated indirectly, i.e., it is those who interact with it that are regulated rather than the technology itself, as is traditionally the case with technology underlying a financial product or service. Furthermore, it was pointed out that regulating a blockchain system directly is likely infeasible, as its decentralized nature seems to place it beyond reach of law enforcement. Thus, the question arises whether the SFD regulates payment systems in an indirect or direct fashion, since the latter case would likely render blockchain technology incompatible with it and, in turn, unable to achieve legal finality.

At a first glimpse, one could get the impression that the SFD continues the aforementioned tradition in that it does not impose regulation on a payment system directly, since its Articles enable a system's ability to achieve final settlement predominantly by limiting the effects of other legislation, e.g., insolvency law.

A closer look, however, reveals that the SFD is actually not technology-agnostic and, thus, does address payment systems directly, as it defines criteria and requirements for such systems that cannot be considered in isolation from their underlying technology – at least not when it comes to blockchain-based payment systems. In fact, the SFD only applies to a payment system in the first place if that system meets the Directive's definition of such[22] – which may well have been con-

[21] N. Szabo. *Tweet echoing Donald Trump's unsubstantiated claims of wide-spread voter fraud in the US presidential election of 2020*. 2020. URL: https://twitter.com/NickSzabo4/status/1334392015304368128 (visited on 01/16/2021).

[22] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*, Article 2(a).

ceived without technology in mind but is, nonetheless, very relevant as to whether or not a payment system based on blockchain technology qualifies.

It is this very definition based on which one could argue that a blockchain-based system cannot constitute a payment system according to SFD, since the Directive defines such a system as a "formal arrangement" between "participants" that is "governed by the law of a Member State" and is "designated [...] as a system and notified to the Commission by the Member State whose law is applicable."[23] However, a blockchain system such as Bitcoin is an informal arrangement between a fluctuating set of participants whose decentralized nature precludes it from being governed by the law of a Member State – or the law of any other territorial jurisdiction, for that matter. From this also follows that there is no Member State to designate such a system and notify said designation to the Commission. It can, thus, be said that the SFD's payment system definition excludes blockchain systems.

As stated before, for a blockchain-based payment system to not constitute such a system pursuant to the SFD means that the transactions it processes are not protected by the Directive, e.g., they are not shielded from the effects of applicable insolvency law(s) and are, effectively, opened up to the potentially disruptive impact of the latter. Consequently, a blockchain-based payment system's transactions are not guaranteed to achieve finality in a legal sense.

As a side note: If, in fact, the aforementioned laws, e.g., insolvency law, do apply to a blockchain-based payment system, then its decentralized structure inhibits their enforcement. Could one infer from this that a blockchain-based payment system is, by default, in conflict with certain laws?

These and similar considerations are likely the reason why some central banks have concluded that if blockchain technology "were ever to form the core of large-value payment systems, that technology would be deployed in a permissioned manner."[24] Indeed – since permissioned blockchain technology is, by definition, centralized, it does not pose any of the previously identified hurdles to regulation and can, therefore, very well underpin a payment system that is designated as such pursuant to the SFD. At this point, it is, however, worth repeating that the contribution of permissioned blockchain technology over traditional approaches to networking and databases is not clear, which is why the technology is not considered in this thesis.

However, one could dismiss the previously presented reasons as to why blockchain systems do not conform to the SFD as mere formal arguments. After all, the SFD

[23] European Parliament and Council of the European Union, *Directive 98/26/EC on settlement finality in payment and securities settlement systems*.

[24] N. Liao. "On settlement finality and distributed ledger technology". In: *Yale Journal on Regulation* (2017). URL: https://www.yalejreg.com/nc/on-settlement-finality-and-distributed-ledger-technology-by-nancy-liao/.

is more than two decades old and, accordingly, was not conceived with blockchain technology in mind. What if the SFD's payment system definition were to be revised to encompass blockchain systems? And, more generally speaking, what if the main hurdles to regulating blockchain systems were to no longer exist, e.g., because a way to internationally enforce laws were to emerge?

The central question that this line of thought boils down to is: Given blockchain-friendly finality legislation and/or international law enforcement, would blockchain-based payment systems be able to achieve legal finality or would the application of the SFD and similar legislation still fail – due to intrinsic properties of blockchain technology?

In this thesis, I argue that blockchain systems in the style of Bitcoin, i.e., PoW-based blockchain systems, would, in fact, still be incompatible with the SFD, because the concept of finality that the Directive implicitly assumes is in conflict with intrinsic properties of blockchain technology.

As previously established, "settlement finality is generally defined in reference to a [discrete] point in time."[25] Thus, technology that underpins a payment system that achieves final settlement must be able to reflect this point in time on a technical level. Traditional, centralized database systems are, without a doubt, able to do so: After submission to such a system, from a discrete point in time onward, a transaction is recorded – in fact, permanently so, as it is guaranteed that the system will never autonomously roll back said record. Thus, traditional, centralized database systems can be said to provide what, in the remainder of this thesis, is referred to as "technical finality."

However, the same is not true for blockchain systems, which cannot provide this novel notion of technical finality – at least so I argue. Since the latter is implicitly assumed in the SFD, though, I argue that blockchain-based payment systems are incompatible with it and, hence, neither provide finality in a technical nor in a legal sense. After all, what good does it do to insulate a payment system's transactions from legal effects that may cause their reversal, if the very database that records such transactions may roll them back? In other words, who benefits from a payment that is de-jure final but is de-facto not recorded, i.e., does not exist?

It can be summarized that the SFD regulates payment systems in a direct fashion, as it only applies to those that conform to its payment system definition, which cannot be considered in isolation from the underlying technology – at least not when discussing blockchain technology. Expectedly, as blockchain systems cannot be regulated directly, payment systems based on blockchain technology do not fall under the auspices of the SFD and, hence, cannot guarantee legal finality.

More importantly, though, it is argued that such systems would be unable to achieve legal finality even if the SFD were to formally apply to them, because the

[25] Liao, "On settlement finality and distributed ledger technology".

concept of finality that the SFD is based on implicitly assumes that the technology underpinning payment systems is able to provide final settlement in a technical sense, i.e., as a discrete point in time after which a transaction is permanently recorded. I argue blockchain technology is unable to do so – a claim I will justify in the following – and, therefore, state that blockchain-based payment systems can neither guarantee technical nor legal finality.

Chapter 5

Can blockchain systems provide technical finality?

As the SFD and similar legislation implicitly assume payment systems to provide technical finality, i.e., to preclude transaction reversal on a technical level from a discrete point in time onward, it was hypothesized that blockchain systems are unable to achieve said technical finality. This chapter examines the theoretical and practical properties of PoW-based blockchain systems with regards to this, introduces alternative approaches and concludes with an assessment on whether blockchain systems can provide technical finality.

Note that up to this point, the term "blockchain system" referred to Bitcoin-style, PoW-based blockchain systems, as was declared in the prerequisites. In this chapter, however, the term refers to the category of systems in general, as different approaches to blockchain technology are discussed.

5.1 Theoretical background

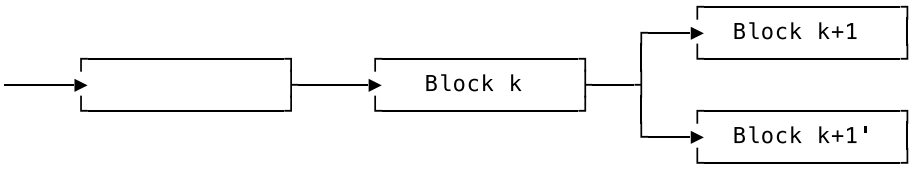
In the following, the theoretical background of PoW-based blockchain systems with regards to technical finality is laid out. In the course of this, blockchain reorganization, i.e., the relevant part of PoW in this context, as well as probabilistic finality and the CAP theorem are described.

5.1.1 Blockchain reorganization

For a better understanding of blockchain reorganization, recap the purpose of blockchain technology: At their core, blockchain systems enable the nodes of a decentralized network to reach consensus on a certain value. In the particular case of Bitcoin, it is the transaction history, referred to as ledger, on which agreement is reached. The consensus algorithm proposed in the Bitcoin paper and employed in the live Bitcoin network is PoW.

According to it, certain nodes referred to as miners bundle transactions they learn of into blocks. Miners expend computational power referred to as work to guess a so-called PoW for a block, i.e., a number that when hashed along with the block's data and the preceding block's hash results in a hash that starts with a number of zero bits. The fact that a block hash references the preceding block's hash shows that blocks form a conceptual chain, i.e., a blockchain. The number of preceding zero bits that is required reflects the network's current difficulty, which is adjusted continuously to ensure that on average every ten minutes a PoW is found on the network.

Once a miner finds a PoW for a block, it announces the block and its hash to the network. Given the transactions in it are valid, other miners will accept the block and start building their new block on top of it. However, given a block k, two or more subsequent blocks, e.g., $k + 1$ and $k + 1'$, may happen to be announced roughly at the same time, splitting or "forking"[1] the blockchain into two or more alternative chains.



Fork due to two blocks at block height $k + 1$

With some nodes being aware of one chain and others being aware of the other, this is emblematic of the very problem blockchain systems set out to remedy. After all, with regards to Bitcoin specifically, block $k + 1$ could contain a transaction that transfers ownership of a bitcoin b to a participant A while block $k+1'$ could contain a conflicting transaction that transfers b to a participant B. In other words, it is imperative to reach consensus on which is the "correct" chain. To this end, the following is stipulated in the Bitcoin paper:[2]

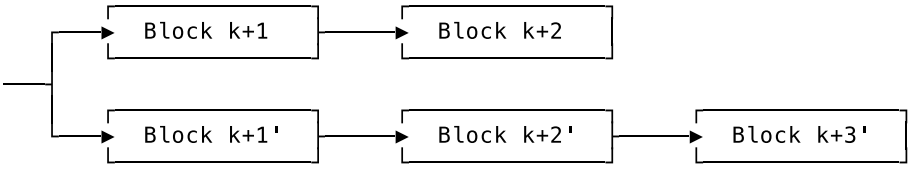
> Nodes always consider the longest chain to be the correct one and [mining nodes] will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

[1] bitcoin.it Bitcoin wiki. *Fork (disambiguation)*. 2015. URL: `https://en.bitcoin.it/wiki/Fork_(disambiguation)` (visited on 02/09/2021).

[2] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 3.

For the sake of correctness, note that the Bitcoin paper implicitly describes the "longest chain" as the chain that consists of most blocks. In practice, this turned out to constitute a vulnerability, as this notion of the longest chain does not necessarily represent the chain that was created with most computational power. Thus, the longest chain is rather to be interpreted as the chain that has most work invested in it. This realization was reflected in a change to the Bitcoin source code in 2010.[3] However, for the sake of simplicity, this paper illustrates the longest chain as the one consisting of the most blocks.

As an example, take again the occurrence of two blocks at block height $k + 1$. Assume the branch starting with block $k + 1$ will have one block appended to it, while the branch starting with block $k + 1'$ will be extended by two blocks. Nodes learning of the latter branch, i.e., the longer branch, will accept it as the correct chain and discard the former. Thereby, all nodes will eventually agree on one chain.



Two alternative chains, one being the longer, "correct" one

The process in which a node discards a previous chain and accepts a longer one that it just learned of is called chain reorganization (or "reorg"). The blocks of the discarded branch are no longer part of the blockchain, the only evidence of their existence is in the log entries of those nodes that once accepted and later discarded them. Such blocks are considered "stale."[4] Although technically it means something different, the term "orphaned" is commonly used synonymously.[5]

It can be summarized that blockchain reorganization causes blocks to become stale, which, in turn, causes the transactions contained in these blocks to be reversed, as they cease to be included in the ledger. Furthermore, from the fact that reorgs are integral to how PoW facilitates consensus follows that transaction reversal is, in turn, integral to PoW-based blockchain systems such as Bitcoin.

[3] S. Nakamoto and G. Andresen. *Commit 40cd036 to the GitHub repository of the Bitcoin Core software*. 2010. URL: https://github.com/bitcoin/bitcoin/commit/40cd0369419323f8d7385950e20342e998c994e1#diff-608d8de3fba954c50110b6d7386988f27295de845e9d7174e40095ba5efcf1bbL1217 (visited on 02/09/2021).

[4] bitcoin.it Bitcoin wiki. *Vocabulary: stale block*. 2018. URL: https://en.bitcoin.it/wiki/Vocabulary#Stale_Block (visited on 02/10/2021).

[5] bitcoin.it Bitcoin wiki. *Orphan block*. 2019. URL: https://en.bitcoin.it/wiki/Orphan_Block (visited on 02/10/2021).

5.1.2 Probabilistic finality

The fact that nodes always take the longest chain as the correct blockchain is the reason why PoW-based blockchain systems do not achieve deterministic or "absolute" finality. For any given transaction, it can never be ruled out that the nodes of such a system will converge on a longer blockchain whose blocks do not contain it. In other words, the reversal of a transaction constitutes a matter of likelihood, which is why PoW-based blockchain systems provide "probabilistic" finality.

The probability of a transaction's reversal depends on a variety of variables, e.g., the speed at which information travels through the peer-to-peer network, the level of interconnectedness of nodes and so forth. One key metric is the amount of processing power that is in the hands of "honest" miners versus that in the hands of an "attacker."

The more processing power an attacker has in relation to all other miners in the network, which are assumed to be honest, i.e., the larger an attacker's share of the total hash rate is, the faster they can mine their own chain in relation to the speed at which all other miners work on the main chain. And the faster they can mine their own chain, the more likely it is that it eventually overtakes the main chain, enabling the attacker to void any transaction contained in the latter.

The Bitcoin paper presents some calculations on the probability of an attacker catching up with the "honest chain," whereby p is the "probability an honest node finds the next block," q is the probability an attacker does so and z is the number of "blocks [that] have been linked after [the block containing the transaction in question]":[6]

> Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. [...]
>
> Solving for P less than 0.1%...
>
> ```
> P < 0.001
> q=0.10   z=5
> q=0.15   z=8
> q=0.20   z=11
> q=0.25   z=15
> q=0.30   z=24
> q=0.35   z=41
> q=0.40   z=89
> q=0.45   z=340
> ```

[6] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, pp. 6, 7, 8.

These calculations have given rise to the conventional wisdom to wait for a transaction to be "confirmed"[7] six times, i.e., to wait for it to be accepted into 1 block and then wait for that block to have another $z = 5$ blocks appended to it. It appears to be assumed that it is unlikely for an attacker to gain more than $q = 10\%$ of the total hash rate. It cannot be overstated, however, that if the assumption $p > q$ does not hold true, the probability of an attacker to eventually catch up is 1.

It can be summarized that in PoW-based blockchain systems, a party that controls the majority of the network's computational power can reverse any transaction, which is known as a majority attack or, colloquially and incorrectly, "51% attack."

5.1.3 CAP theorem

Distributed web services are often categorized according to the CAP theorem, which states that such a service can only simultaneously provide two of the following three properties:[8]

Consistency (C) A "consistent service" must establish "a total order on all operations such that each operation looks as if it were completed at a single instant." In other words, "requests of the distributed shared memory [must be required] to act as if they were executing on a single node, responding to operations one at a time."

Availability (A) An available service must ensure that "every request received by a non-failing node in the system must result in a response." In other words, "any algorithm used by the service must eventually terminate." As there is "no bound on how long the algorithm may run before terminating," this can be considered a "weak definition of availability."

Partition tolerance (P) A partition-tolerant service must withstand partitioning of the underlying network, i.e., "the network will be allowed to lose arbitrarily many messages sent from one node to another." In other words, network partitioning is present when "all messages sent from nodes in one component of the partition to nodes in another component are lost."

It must be pointed out that Eric Brewer, the author of the original conjecture, later pointed out that the common interpretation of "'2 of 3' is misleading," essentially

[7] bitcoin.it Bitcoin wiki. *Confirmation*. 2018. URL: `https://en.bitcoin.it/wiki/Confirmation` (visited on 02/10/2021).

[8] S. Gilbert and N. Lynch. "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services". In: *ACM SIGACT News* (2002). URL: `https://dl.acm.org/doi/10.1145/564585.564601`.

stating that since distributed systems must deal with network partitioning in any case, they really can only choose between consistency and availability:[9]

> The easiest way to understand CAP is to think of two nodes on opposite sides of a partition. Allowing at least one node to update state will cause the nodes to become inconsistent, thus forfeiting C. Likewise, if the choice is to preserve consistency, one side of the partition must act as if it is unavailable, thus forfeiting A. Only when nodes communicate is it possible to preserve both consistency and availability, thereby forfeiting P. The general belief is that for wide-area systems, designers cannot forfeit P and therefore have a difficult choice between C and A.

Brewer also added that the "2 out of 3" notion is "misleading because it tend[s] to oversimplify the tensions among properties," stressing that the theorem only prohibits "perfect availability and consistency in the presence of partitions."[10] In reality, systems can be designed to compromise and provide both properties to some degree:[11]

> [A]ll three properties are more continuous than binary. Availability is obviously continuous from 0 to 100 percent, but there are also many levels of consistency, and even partitions have nuances, including disagreement within the system about whether a partition exists.

With regards to blockchain systems, consistency means that all nodes assume the same chain to be the correct one at the same time, i.e., they maintain identical copies of the ledger continuously. Availability means that non-failing nodes maintain a chain at all times, albeit not necessarily the correct one. Partition tolerance means that the system continues to operate when individual or clusters of nodes are unable to communicate over the peer-to-peer network.

Since PoW-based blockchain technology appears to provide all three properties, "[i]t seems that the CAP theorem is violated," but it really is not, as PoW-based blockchain systems only achieve a weak form of consistency, i.e., "eventual consistency," where "there can be a temporary disagreement between nodes on the final state, but it is eventually agreed upon."[12] This is because nodes only ever consider the longest chain they are aware of to be the correct one, while it is assumed that each node eventually learns of the truly longest chain.

As a side note, later in this thesis, the notion that PoW-based blockchain systems are partition-tolerant is challenged as well. They are technically, but for the practical purposes of payment systems, they are not, at least so I argue.

---

[9] E. Brewer. *CAP twelve years later: how the "rules" have changed*. 2012. URL: https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/ (visited on 03/04/2021).

[10] Ibid.

[11] Ibid.

[12] Bashir, *Mastering Blockchain*, p. 35.

It can be summarized that since the nodes of a PoW-based blockchain system reach consensus on the correct chain over a unbounded period of time, i.e., reach consistency eventually, there is no discrete point in time at which a transaction can be said to be part of the ledger. This comes in addition to the fact that there is no guarantee it will not be removed from it later anyway, as has already been established previously.

5.2 Practical implications

In the following, the practical implications of the theoretical properties of PoW-based blockchain systems with regards to technical finality are analyzed. In the course of this, incidents of forking in such systems are described as well as vulnerabilities of those that are incentivized both intrinsically and extrinsically.

5.2.1 Forking incidents

It was previously described that with the PoW consensus algorithm, the blockchain may fork due to the occurrence of two or more blocks at the same block height and that in the process of reconvergence, i.e., reorganization, blocks or chains of blocks become stale, reversing the transactions they contain. The obvious question, then, is to what extent this actually occurs in Bitcoin, the largest PoW-based blockchain system.

However, since the Bitcoin network is decentralized, there is no "bird's eye view" on it, which would be necessary to become aware of all forks. To use the words of a user in the Bitcoin community on Stack Exchange, "[t]he problem with blockchain forks is that once they are resolved the only trace they leave is a log entry,"[13] referring to the logs of the individual node(s) that carried out the reorg in question. Thus, an answer to the above question cannot be given comprehensively but can only be approximated.

It can be said that assuming normal operation, the occurrence of stale blocks is quite rare, as the Bitcoin block interval is 10 minutes while the median block propagation time in 2014/15 was 8.7 seconds.[14] In 2016, ETH Zürich researchers reported Bitcoin's stale block rate at 0.41%, measured over 10,000 blocks.[15] This corresponds to roughly 18 stale blocks per month. The nodes of BitMEX's Fork-

[13] C. Decker and other contributors. *What is the longest blockchain fork that has been orphaned to date?* 2017. URL: https://bitcoin.stackexchange.com/a/4638 (visited on 02/12/2021).

[14] K. Croman et al. "On scaling decentralized blockchains: a position paper". In: *Lecture Notes in Computer Science* (2016). URL: https://link.springer.com/chapter/10.1007/978-3-662-53357-4_8, p. 111.

[15] A. Gervais et al. "On the security and performance of proof of work blockchains". In: *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016). URL: https://dl.acm.org/doi/10.1145/2976749.2978341, p. 3.

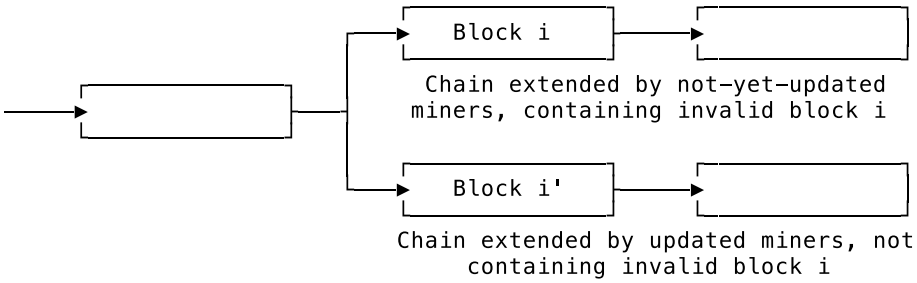
Monitor pick up fewer than that, e.g., creating only four “stale block candidates” alerts in 12/2020 and 01/2021, respectively.[16]

Most incidents where two or more blocks exist on the same block height are less relevant in the context of finality, because most of these forks are resolved with the next block or so. Since it is customary to wait for six confirmations until a transaction is deemed final, the reversal of a transaction due to its containing block becoming stale will, in practice, only be perceived as a reversal if said block had at least five blocks appended to it.

More relevant in the context of finality is the latter scenario, i.e., the orphaning not of an individual block but of an entire chain consisting of six or more blocks, as this constitutes the reversal of transactions that are considered final as per conventional wisdom. It cannot be overstated that in the Bitcoin system, this happened – and more than once.

Before listing such incidents, the concepts of a “softfork”[17] and a “hardfork”[18] must be introduced. Both represent an update to the software run by the nodes of a blockchain system, e.g., Bitcoin Core.[19] Updated nodes follow a rule that is not followed by not-updated nodes. Most often, as a result, certain blocks produced by not-updated nodes are no longer accepted as valid blocks by updated nodes.

Consequently, not-updated miners keep extending the blockchain by appending blocks to a block i, which is, however, considered invalid according to updated miners, who, therefore, start and extend an alternative chain, beginning with a block i'. In other words, such an update causes the blockchain to fork.



Two alternative chains, caused by either a softfork or a hardfork

[16] BitMEX. *ForkMonitor: stale block candidates*. 2021. URL: https://forkmonitor.info/notifications (visited on 02/12/2021).

[17] bitcoin.it Bitcoin wiki. *Softfork*. 2018. URL: https://en.bitcoin.it/wiki/Softfork (visited on 02/13/2021).

[18] bitcoin.it Bitcoin wiki. *Hardfork*. 2019. URL: https://en.bitcoin.it/wiki/Hardfork (visited on 02/13/2021).

[19] Contributors to the Bitcoin Core software. *GitHub repository of the Bitcoin Core software*. 2021. URL: https://github.com/bitcoin/bitcoin (visited on 02/13/2021).

In the case of a softfork, the not-updated nodes accept the blocks created by the updated nodes, i.e., a softfork is backwards-compatible. Therefore, a chain split created by a softfork will eventually resolve, since either branch i or branch i' will eventually become the longest and universally accepted chain. However, the "losing chain" will become stale, reversing all its transactions.

Hardforks, by contrast, are not backwards compatible, i.e., the not-updated nodes reject the blocks created by the updated nodes. Consequently, a chain split created by a hardfork will never resolve, both chains will simply grow independently of one another. Note that softforks can be utilized to undo accidental hardforks.

With softforks and hardforks defined, find below a compilation of incidents where reorgs of the Bitcoin blockchain caused chains consisting of six or more blocks to become stale, i.e., incidents where transactions that were viewed as final were reversed. For the same reasons elaborated on before, this compilation cannot be considered complete.

Value overflow incident On August 15, 2010, a bug in the Bitcoin Core software was exploited to create a transaction that transferred roughly 184 billion bitcoins to three different addresses, although Bitcoin imposes a limit of roughly 21 million bitcoins to ever come into existence. The underlying vulnerability was that integer overflows were not checked for. After five hours, an update, i.e., a softfork, was rolled out to invalidate the respective block 74,638. Its containing chain was overtaken at block height 74,691, thus, 53 blocks became stale.[20] This is likely the longest chain to ever be orphaned in Bitcoin.

Database switch incident On March 11, 2013, miners running version 0.8 of the Bitcoin software created blocks that were rejected by nodes running earlier versions. Version 0.8 switched the underlying database from BerkeleyDB to LevelDB, inadvertently introducing an incompatibility, thus, constituting an accidental hardfork. The blockchain forked at block height 225,430. After miners were asked to switch back to version 0.7, the majority hash rate backed the "version 0.7 chain," and the blockchain reconverged at block height 225,454, causing 24 blocks to become stale.[21]

Invalid blocks incident On July 4, 2015, invalid blocks were created. Miners failed to validate these and built on top of them, as "[a]lmost all" instances of the software "besides Bitcoin Core 0.9.5 and later" were unable to detect

[20] bitcoin.it Bitcoin wiki. *Value overflow incident*. 2016. URL: `https://en.bitcoin.it/wiki/Value_overflow_incident` (visited on 02/13/2021).

[21] V. Buterin. *Bitcoin network shaken by blockchain fork*. 2013. URL: `https://bitcoinmagazine.com/articles/bitcoin-network-shaken-by-blockchain-fork-1363144448` (visited on 03/15/2021).

their invalidity.[22] The latter versions implemented BIP66[23] to no longer depend on OpenSSL's signature parsing, constituting a softfork. The blockchain forked at block height 363,731 and reconverged at block height 363,737, causing 6 blocks to become stale.[24]

For a more comprehensive compilation of such incidents, refer to BitMex's "[more] complete history of Bitcoin's consensus forks."[25]

Obviously, transaction reversal due to forks affects all PoW-based blockchain systems, not just Bitcoin. In 2016, Ethereum, the second largest blockchain system, suffered the most prominent chain split of all. Attackers exploited a bug in the DAO ("decentralized autonomous organization"), a venture capital fund based on smart contracts, stealing from it about 3.6 million ether, i.e., Ethereum's cryptocurrency – roughly $50 million.[26]

In the aftermath of the incident, "over 90 percent of the hashrate signaled its support" to issue a hardfork and subsequently back the chain on which "funds were returned to investors as though the [DAO] organization had never existed," which is now Ethereum's main chain.[27] As some view this intervention as contradictory to the "value propositions of the Ethereum platform," the original chain is still maintained, now known as "Ethereum Classic."[28]

What all these incidents have in common is that a small group of developers release an update and then a similarly small group of mining pool operators decide to throw their weight, i.e., the majority hash rate, behind it, ultimately causing the reversal of certain transactions previously believed to be final. Firstly, this shows that these systems are, in fact, rather centralized and, secondly, that these incidents amount to what technically are 51% attacks – despite all good intentions.

What sounds like an exaggeration is actually documented plainly on the respective projects' web sites. E.g., in the wake of the database switch incident, influential Bitcoin Core contributor Pieter Wuille publicly instructs miners on which chain to

[22] Bitcoin Project. *Some miners generating invalid blocks*. 2015. URL: `https://bitcoin.org/en/alert/2015-07-04-spv-mining` (visited on 02/14/2021).
[23] Contributors to the Bitcoin Core software. *BIP66 in the GitHub repository of the Bitcoin Core software*. 2015. URL: `https://github.com/bitcoin/bips/blob/master/bip-0066.mediawiki` (visited on 02/14/2021).
[24] The `r/Bitcoin` community on Reddit. *F2Pool is not properly validating blocks*. 2015. URL: `https://www.reddit.com/r/Bitcoin/comments/3c2cnj/f2pool_is_not_properly_validating_blocks_their/` (visited on 02/14/2021).
[25] BitMEX. *A complete history of Bitcoin's consensus forks*. 2017. URL: `https://blog.bitmex.com/bitcoins-consensus-forks/` (visited on 02/15/2021).
[26] V. Dhillon, D. Metcalf, and M. Hooper. *Blockchain enabled applications*. Apress, 2017. ISBN: 9781484230800, p. 75.
[27] Ibid., p. 76.
[28] Ibid., p. 76.

extend, confidently adding that since mining pool "BTCGuild is switching" to it, the desired chain "will get a majority hash rate soon."[29]

Given that individual mining pools have at times held the majority of the computational power in the entire Bitcoin system, e.g., mining pool "GHash.io" in 2014,[30] it should come as no surprise that all it takes to alter the blockchain, i.e., to reverse transactions, is the cooperation between few influential individuals.

It can be summarized that transaction reversal is not only theoretically possible but happened in practice, affecting virtually all PoW-based blockchain systems, including Bitcoin and Ethereum. These incidents were the result of well-intended initiatives that nonetheless amount to majority attacks, demonstrating the de-facto centralization of such systems.

5.2.2 Incentive incompatibility

The preceding remarks show that the property of PoW-based blockchain systems to discard any block as part of a reorg has already caused transaction reversal in practice. However, these incidents were benign – a majority hash rate only came together to reverse the effects of bugs or attacks, preventing the respective system from breaking. Thus, the question arises whether and how a majority hash rate can be attracted by malignant motives.

Relevant with regards to this is a landmark paper in the field of blockchain security, "Majority is not enough: Bitcoin mining is vulnerable,"[31] in which Cornell researchers show that mining honestly by following the protocol proposed in the Bitcoin paper rather than gaming a PoW-based blockchain system is not economically rational and that, therefore, such systems are not "incentive compatible."[32]

In said paper, a strategy called "selfish mining" is proposed, which assumes that there are two mining pools, one consisting of selfish miners and one consisting of honest ones. The basic idea is – roughly speaking – that it is more profitable for selfish miners to not broadcast blocks they create to the network but keep them private, tricking honest miners into wasting computational power, i.e., money, as those keep working on building on top of previous blocks.

The proposed algorithm can be summarized as follows. The selfish miners maintain a private chain, while the honest miners work on the public chain. As long as

[29] P. Wuille and other contributors. *Alert: chain fork caused by pre-0.8 clients dealing badly with large blocks*. 2013. URL: `https://bitcointalk.org/index.php?topic=152030` (visited on 02/15/2021).

[30] R. Böhme et al. "Bitcoin: economics, technology, and governance". In: *The Journal of Economic Perspectives* (2015). URL: `https://www.jstor.org/stable/24292130`, p. 222.

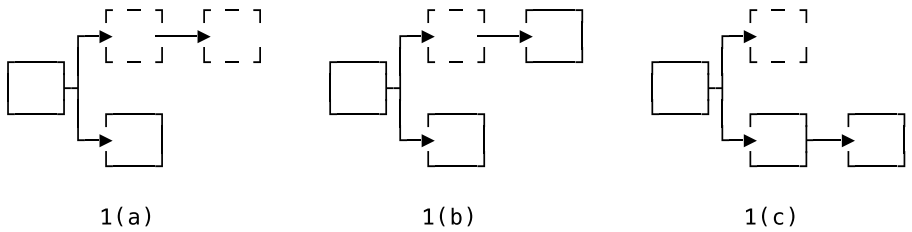
[31] I. Eyal and E. G. Sirer. "Majority is not enough: Bitcoin mining is vulnerable". In: *Lecture Notes in Computer Science* (2014). URL: `https://link.springer.com/chapter/10.1007/978-3-662-45472-5_28`.

[32] Ibid., p. 1.

the selfish miners do not find a block themselves, their private chain is equivalent to the honest miners' public chain.

Once the selfish miners do find a block, they add it to their private chain instead of announcing it to the network. Their private chain is then ahead of the public chain by one block. This leads to two alternative scenarios:

1. The honest miners find a block as well, making the two chains equally long. This causes the selfish miners to announce to the network the block they kept private immediately, as the honest miners will try to create a block on top of whichever block they learn of first, i.e., they will either work on extending the honest miners' or the selfish miners' chain. Obviously, the selfish miners will try to add another block on top of their block. This, in turn, leads to three alternative scenarios:
   (a) The selfish miners find another block on top of their block, which they publish as well and, thus, gain the revenue of both their blocks, as these now represent the longest chain, while the honest miners' block has become stale.
   (b) The honest miners find a block on top of the selfish miners' block, causing both the honest miners and the selfish miners to gain the revenue for their one block, respectively.
   (c) The honest miners find a block on top of their block, gaining the revenue of both their blocks, while causing the selfish miners to gain no revenue at all, since their block has become stale.
2. The selfish miners find a second block, causing their private chain to be ahead of the public chain by two blocks. The selfish miners will try to find more blocks to add on top of their private chain. Whenever the honest miners add a new block to the public chain, the selfish miners announce to the network their oldest private block. Once the honest miners' public chain has caught up so that it is only one block behind the selfish miners' private chain, the selfish miners publish their entire private chain, gaining the revenue for all its blocks.

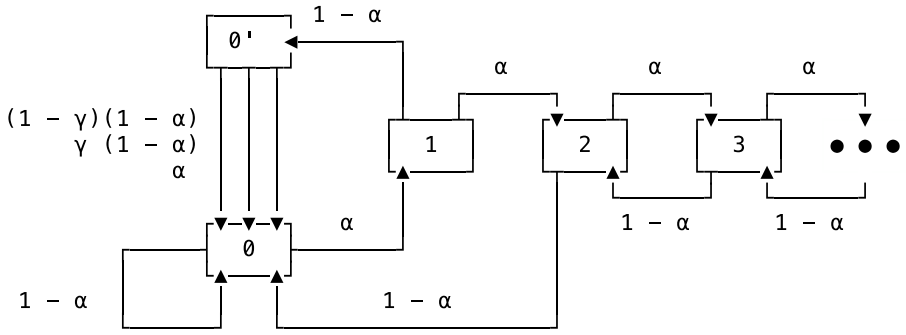


Scenarios 1(a), 1(b) and 1(c), with the selfish miners' blocks dashed

The selfish mining paper illustrates probabilities with the state machine depicted below, where the states $0, 1, ..., n$ correspond to the number of blocks that the selfish miners' private chain is ahead of the honest miners' public chain. The state $0'$ corresponds to scenario 1 of the algorithm as outlined above. The three edges leading from state $0'$ to state 0 correspond to the three scenarios 1(a), 1(b) and 1(c), respectively.

α denotes the relative mining power of the selfish miners and $1-\alpha$ that of the honest ones. E.g., if $\alpha = 0.3$, the selfish miners chances of leading by one block is 30%, by two blocks 9%, diminishing exponentially. γ denotes the ratio of honest miners that accept and, thus, build on top of a block announced by the selfish miners and $1-\gamma$ the ratio of honest miners that accept the competing honest miners' block. E.g., if $\gamma = 0$, honest miners will never mine on top of a block published by the selfish miners, if $\gamma = 1$, they always will.

The state machine given in the selfish mining paper corresponds to a Markov chain, constituting a stochastic model:[33]



Markov chain modeling states and probabilities of selfish mining

The profitability of selfish mining is presented in contrast to the profitability of mining as prescribed by the PoW algorithm, where a miner's relative computational power corresponds to the miner's relative revenue, e.g., a miner with 10% of the total hash rate is expected to earn, on average, 10% of the total revenue. Selfish mining compares to this as follows:

- If $\gamma = 0$, i.e., if the selfish miners lose every block race, which is the key factor in scenario 1, selfish mining is more profitable than mining according to protocol whenever the selfish miners' relative hash rate exceeds 33%.
- If $\gamma = 0.5$, i.e., if the selfish miners win every other block race, selfish mining is more profitable than mining according to protocol whenever the selfish miners' relative hash rate exceeds 25%.

[33] Eyal and Sirer, "Majority is not enough: Bitcoin mining is vulnerable", p. 8.

- If $\gamma = 1$, i.e., if the selfish miners win every block race, selfish mining is more profitable than mining according to protocol at all times, independently of the selfish miners' relative hash rate.

In any case, the selfish miners' relative revenue increases asymptotically with their relative hash rate, reaching 100% of total revenue with only 50% of the total hash rate. Furthermore, as one of the authors points out in a talk at the Israel Institute for Advanced Studies,[34] "you can get γ to be fairly high," as "you can fracture the network," referring to the 2013-paper "Information Propagation in the Bitcoin Network."[35]

Most scary, however, is the observation that it is rational for miners to "join the selfish pool to increase their revenues."[36] In the same talk, "pool formation" is described as follows: "Suppose I am a 10% [of the total hash rate] miner, and I could go selfish and make, say, 11% [of the total revenue]. Suppose you are a 10% miner, and you could go selfish. [...] If you compete with me, we're going to start knocking each other out. But [...] if we join forces [...] we get a bonus, we go from 11% for 10% to 23% for 20%. [...] This is absolutely horrible."[37]

While this constitutes an incentive to join selfish mining pools, pool formation can also be explained by the disincentive to remain honest, I may add. After all, the additional income of the selfish miners is to the detriment of the honest miners, who are tricked into wasting work as they solve stale problems. One way or another, ultimately, "[t]he selfish pool would [...] increase in size, unopposed by any mechanism, until it becomes a majority."[38]

With selfish mining incentivizing the formation of ever-growing pools of selfish miners, the question arises whether selfish mining has actually occurred in practice and whether it has lead to the formation of pools holding a majority hash rate. The authors of the selfish mining paper respond to this[39] by stating that the two "distinct network signatures of selfish mining" are the "[n]umber of abandoned (orphaned) blocks," which, as mentioned before, is very hard to establish, as "abandoned blocks are pruned inside the Bitcoin network," and the "[t]iming of successive blocks," which bears difficulties as well, as "timing gap analysis [...] is a statistical test, and it may take a fair bit of selfish mining activity before it detects that something is amiss." While the authors say that it probably did not take place yet, they imply that if it had, it would have been difficult to detect.

[34] E. G. Sirer. *Alternatives to Nakamoto consensus*. 2018. URL: `https://www.youtube.com/watch?v=1gUX9ikG1FI` (visited on 02/20/2021).
[35] C. Decker and R. Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P 2013 Proceedings* (2013). URL: `https://ieeexplore.ieee.org/document/6688704`.
[36] Sirer, *Alternatives to Nakamoto consensus*.
[37] Ibid.
[38] Ibid.
[39] I. Eyal and E. G. Sirer. *How to detect selfish miners*. 2014. URL: `https://hackingdistributed.com/2014/01/15/detecting-selfish-mining` (visited on 02/20/2021).

Be that as it may, what can be summarized is that, in addition to the fact that PoW not only allows for transaction reversal in theory but has enabled such incidents in practice, PoW is also incentive incompatible in that the rational strategy for miners is to game PoW-based blockchain systems – and to coalesce in order to do so. In other words, such systems incentivize the formation of an attacker pool that eventually holds a majority hash rate, at which point it can reverse any transaction.

5.2.3 Extrinsic incentives

Many of the proposed attacks on PoW-based blockchain systems, including selfish mining, reward those who employ them with tokens of the system's native cryptocurrency, so it is in the attackers' best interest to not break such systems by reversing transactions, as this would devalue their own revenue. This incentive is already expressed in the Bitcoin paper:[40]

> If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments [i.e., reversing transactions], or using it to generate new coins [i.e., engaging in mining as prescribed]. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

This incentive may apply within the implied game-theoretic model, which is, however, limited in scope to the Bitcoin ecosystem. Consequently, it does not take effect in the case of attackers whose incentives are extrinsic to said model and, therefore, irrespective of the cryptocurrency token's value.

This aspect is all the more important in the context of the discussion on PoW-based blockchain technology to underpin mission-critical payment systems in major economies. Rival nation-states or enterprises may have an incentive, e.g., geopolitical and/or financial, to destroy those systems, knowing full well and accepting that the attack in itself may represent a net loss.

This begs the question: How much does it cost to attack Bitcoin and other PoW-based blockchain systems? Two metrics are usually referenced to answer this question, namely the cost of a "rental attack" and that of a "building attack." The former corresponds to the cost of temporarily renting computational resources equivalent to a majority hash rate, while the latter corresponds to the cost of permanently acquiring those resources.

[40] Nakamoto, *Bitcoin: a peer-to-peer electronic cash system*, p. 4.

See below an estimate of the costs of a rental attack on Bitcoin for two different days. The total hash rate was taken from whattomine,[41] the rental prices for computational power from NiceHash[42] and the Bitcoin exchange rates from CoinMarketCap.[43] As can be seen from the table, my results roughly correspond to the data published by Messari[44] and Crypto51.[45]

	2021-02-22	**2020-12-02**
Total hash rate in PH/s	143,324.81	114,915.00
Rental prices in BTC/PH/s/d	0.0075	0.008
Exchange rate in USD/BTC	52,056.36	19,201.09
Rental attack cost in USD/h	2,331,552.47 Messari: 2,754,271.39	735,497.75 Crypto51: 716,072.00

Cost estimates of rental attacks on Bitcoin

What is striking about these numbers is that the hourly price of a rental attack increases from hundreds of thousands of dollars to a single-digit million dollar figure in a matter of months. This is because Bitcoin-specific computational power available for rent is priced in bitcoin, resulting in the price of rental attacks being pegged to that of bitcoin, which increased 2.5 fold over this period.

This dynamic could have been offset by taking into account mining revenues, which increased in value at the same rate. However, in order to obtain a conservative estimate, this is not commonly done, as mining revenues can "reduce the attack cost by up to 80%" – but only "in some cases."[46] Also with the intention of obtaining an upper bound, it is assumed that an attacker does not take over any existing hash power but matches all of it with additional capacities.

Nonetheless, these figures are theoretical. As Bitcoin mining requires specialized hardware that cannot be used for much else, it is unrealistic to assume that there

[41] whattomine. *Bitcoin (BTC)*. 2021. URL: https://whattomine.com/coins/1-btc-sha-256 (visited on 02/22/2021).
[42] NiceHash. *SHA-256 marketplace*. 2021. URL: https://www.nicehash.com/my/marketplace/SHA256 (visited on 02/22/2021).
[43] CoinMarketCap. *Historical data for Bitcoin*. 2021. URL: https://coinmarketcap.com/currencies/bitcoin/historical-data/ (visited on 02/22/2021).
[44] Messari. *Bitcoin metrics*. 2021. URL: https://messari.io/asset/bitcoin/metrics (visited on 02/22/2021).
[45] Crypto51. *Bitcoin (BTC): cost for a 51% attack*. 2021. URL: https://www.crypto51.app/coins/BTC.html (visited on 02/22/2021).
[46] Crypto51. *About*. 2021. URL: https://www.crypto51.app/about.html (visited on 02/22/2021).

are enough capacities available for rent to match the system's total hash rate. E.g., on 2021-02-22, the hash rate available on NiceHash, by far the largest hash power marketplace, was 557.6056 PH/s, corresponding to 0.3% of Bitcoin's total hash rate that day.

For smaller PoW-based blockchain systems, especially if their hash algorithm does not require specialized hardware, rental attacks are feasible, though. E.g., in a 2017-paper, NYU researchers propose a rental attack on Ethereum,[47] estimating the cost of renting Nvidia K80 GPUs, each performing 50–100 MH/s, from AWS EC2 at "$1 million/h to perform a temporary takeover," adding that "few hours of disruptive attacks could be sufficient to cause a major loss in value to the system, which has a market cap of almost $30 billion."

Within the same publication, the costs of building attacks against Bitcoin and Ethereum are estimated as well.[48] For Bitcoin, the acquisition of AntMiner S9 ASIC miners, each performing 14 TH/s at a purchase price of $2,000 and 1 kW energy consumption, is proposed. Assuming the then-current total hash rate of 1,000 PH/s, "roughly $1.5 billion would build enough capacity to take over the Bitcoin blockchain."

For Ethereum, the acquisition of Radeon Rx Vega 56 GPUs, each performing 36 MH/s at a purchase price of $550, is proposed. Assuming the then-current total hash rate of 100 GH/s, "roughly $1.5 billion [...] build enough capacity to take over the Ethereum blockchain." The fact that both building attacks are valued at $1.5 billion – despite Bitcoin's much higher total market value – is assumed to be due to more investment in Ethereum mining equipment, which, in turn, is assumed to be due to higher relative revenues and the ability to mine using non-specialized hardware.[49]

In the context of extrinsically incentivized attacks on PoW-based blockchain systems, it must be mentioned that these can also be negatively affected by measures that are not even directed at them in the first place. E.g., a nation state could close off its domestic internet from the rest of the world, thereby not only partitioning the Internet but also, as a consequence, partitioning any PoW-based blockchain network based on it.

This scenario is commonly discussed in the context of Bitcoin and China, as Chinese-operated mining pools have at times managed up to 74% of Bitcoin's total hash rate, while, at the same time, China operates "a variety of Internet control measures that can affect Bitcoin traffic," most notably "the Great Chinese Fire-

[47] J. Bonneau. "Hostile blockchain takeovers". In: *Financial Cryptography and Data Security* (2018). URL: `https://link.springer.com/chapter/10.1007/978-3-662-58820-8_7`, p. 95.
[48] Ibid., pp. 95, 96.
[49] Ibid., p. 96.

wall."[50] What would be the effect on Bitcoin with regards to transaction reversal if China were to close off its internet?

The Bitcoin blockchain would fork into one chain maintained by Chinese miners, which would continue to process Chinese transactions, and one chain maintained by the rest of the world's miners, which would continue to process the remaining transactions. However, as soon as the partitioning would resolve, the system would reconverge, whereby the shorter chain would be discarded, and with it all its transactions. Therefore, one could state that PoW-based blockchain systems are partition-tolerant technically but not practically, as the transactions in the partition with less computational power are indeed recorded yet reversed later.

It can be summarized that extrinsically incentivized rental attacks and building attacks on the largest PoW-based blockchain systems are estimated to cost within the single-digit million USD/h and single-digit billion USD, respectively, albeit the former figure is theoretical in the case of Bitcoin. Assuming an attacker is able to afford such an attack for an unbounded period of time, they are able to reverse any transaction, no matter how many times it has been confirmed. Furthermore, measures such as Internet censorship can have side effects on PoW-based blockchain systems, causing transaction reversal as well.

5.3 Possible solutions

In the following, Stellar, a blockchain system that does provide technical finality is presented. In the course of this, the Stellar Consensus Protocol is introduced, which, to the best of my knowledge, is the only system that does so without sacrificing decentralization.

5.3.1 Stellar Consensus Protocol

With Stellar, there exists one blockchain system that puts special emphasis on finality. Its consensus algorithm, the Stellar Consensus Protocol (SCP), guarantees so-called "issuer-enforced finality" without sacrificing "[o]pen membership,"[51] a hallmark of decentralization. However, unlike PoW, which assumes a trustless environment, SCP is based on "flexible trust,"[52] thereby solving a slightly different problem than PoW.

[50] B. Kaiser, M. Jurado, and A. Ledger. "The looming threat of China: an analysis of chinese influence on Bitcoin". In: *Computer Science, Cryptography and Security (cs.CR)* (2018). URL: https://arxiv.org/pdf/1810.02466.pdf, p. 5.

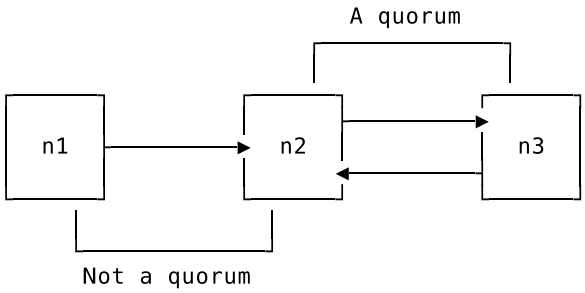
[51] M. Lokhava et al. "Fast and secure global payments with Stellar". In: *SOSP '19: Proceedings of the 27th ACM Symposium on Operating Systems Principles* (2019). URL: https://dl.acm.org/doi/10.1145/3341301.3359636, p. 1.

[52] D. Mazières. *The Stellar Consensus Protocol: a federated model for Internet-level consensus*. 2016. URL: https://www.stellar.org/papers/stellar-consensus-protocol (visited on 02/25/2021), p. 2.

SCP is "a construction for," i.e., type of, federated Byzantine agreement (FBA),[53] which was introduced along with SCP and is, in turn, a generalization of traditional Byzantine agreement (BA).[54] Traditional BA enables a set of n nodes to reach consensus on one "value of information" by voting based on "two-party messages," assuming $n \geq 3m + 1$, where m is the number of "faulty" or malicious nodes.[55]

In traditional BA, the nodes have to be known beforehand, though, which contradicts decentralization, "where two nodes may not even know of each other's existence."[56] FBA generalizes the principles of BA to overcome this limitation. To this end, FBA introduces the concept of a "quorum slice," defined as a set of nodes for which a node expresses that it requires agreement with.[57] Furthermore, a "quorum" is defined as "non-empty set [...] of nodes encompassing at least one quorum slice of each non-faulty member."[58]

Consider the following example, which is based on a more complex one from the SCP Internet Draft:[59] Node n_1 expresses that it requires consensus with node n_2, i.e., n_1 has one quorum slice $\{n_1, n_2\}$. Nodes n_2 and n_3 require consensus with each other, thereby they both have one quorum slice $\{n_2, n_3\}$. Then, $\{n_2, n_3\}$ is a quorum, while $\{n_1, n_2\}$ is not, because it lacks a quorum slice of n_2.



Quorum slices and a quorum

Furthermore, two nodes n_1 and n_2 are considered "intertwined" "when every quorum of $[n_1]$ intersects every quorum of $[n_2]$ in at least one non-faulty node."[60] As any FBA protocol, including SCP, "can ensure agreement only between intertwined

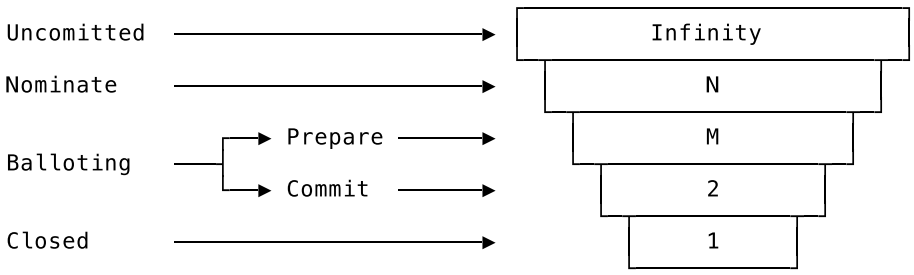
[53] Mazières, *The Stellar Consensus Protocol: a federated model for Internet-level consensus*, p. 2.
[54] Ibid., p. 5.
[55] Lamport, Pease, and Shostak, "Reaching agreement in the presence of faults".
[56] Lokhava et al., "Fast and secure global payments with Stellar", p. 4.
[57] Ibid., p. 4.
[58] Ibid., p. 4.
[59] N. Barry et al. *Network Working Group Internet-Draft: The Stellar Consensus Protocol (SCP)*. 2018. URL: `https://tools.ietf.org/html/draft-mazieres-dinrg-scp-05#section-2.1` (visited on 02/25/2021).
[60] Lokhava et al., "Fast and secure global payments with Stellar", p. 4.

nodes," SCP guarantees exactly this, claiming "that the nodes people care about will be intertwined."[61]

While this claim may sound flippant, the Internet is actually living proof that it applies, as the topology of intertwined nodes is strikingly reminiscent of that of the Internet. Accordingly, Stellar provides its "goals" "under a novel but empirically valid 'Internet hypothesis.'"[62]

As is the case in traditional BA, SCP achieves agreement using voting. The proposed voting protocol is called "federated voting." Federated voting is a protocol consisting of three stages, i.e., "vote," "accept" and "confirm."[63] However, federated voting "might not terminate."[64] In each consensus round, SCP goes through two stages, i.e., "nominate" and "balloting," both of which make use of federated voting.[65]

Since the intricacies of SCP are complex and go far beyond the scope of this paper, it is, at this point, best to leave it at the following illustration, which is based on one published on the Stellar Developers blog – it simply conceptualizes SCP as a "funnel" that narrows down an infinite set of values to a single one:[66]



SCP as a funnel

It is, however, crucial to take a closer look at SCP's properties and how they compare to PoW. A "multiprocess program" typically defines "two essentially different types of properties about it," i.e., "safety" and "liveness," where the former states "something [that] will not happen" and the latter "something [that] must happen."[67] SCP defines these as follows:

[61] Lokhava et al., "Fast and secure global payments with Stellar", p. 4.
[62] Ibid., p. 1.
[63] Ibid., p. 6.
[64] Ibid., p. 6.
[65] Ibid., pp. 5, 6.
[66] M. Lokhava. *Intuitive Stellar Consensus Protocol*. 2019. URL: https://www.stellar.org/developers-blog/intuitive-stellar-consensus-protocol (visited on 02/25/2021).
[67] L. Lamport. "Proving the correctness of multiprocess programs". In: *IEEE Transactions on Software Engineering* (1977). URL: http://www.cis.umassd.edu/~hxu/courses/cis481/references/Lamport-1977.pdf, p. 125.

Safety "A Byzantine agreement protocol is safe when no two well-behaved nodes output different decisions and the unique decision was a valid input."[68] The former property is referred to as "agreement" and the latter as "validity."[69]

Liveness "A protocol is live when it guarantees that every honest node eventually outputs a decision."[70]

"Unfortunately, achieving all three of safety, guaranteed termination [i.e., liveness], and fault tolerance turns out to be impossible."[71] In SCP, this dilemma manifests itself in the fact that the increasing size of quorums is, on the one hand, conducive to safety, but, on the other hand, detrimental to liveness.[72]

SCP chooses to prioritize safety over liveness, as "the consequences of a safety failure (namely double-spent digital money) are far worse than those of a liveness failure (namely delays in payments that anyway took days before Stellar)."[73] Thus, people who run nodes are actively encouraged to "select large quorum slices," which increases intertwinedness, which, in turn, is the aforementioned condition for SCP to guarantee agreement.[74]

Speaking in terms of the CAP theorem, "Stellar prefers consistency and partition resilience [i.e., partition tolerance] over liveness [i.e., availability]."[75] Thus, in practical terms, "when faced with consensus uncertainty," SCP "prefers to halt," which, as a side note, has happened in practice and generally requires nodes to reconfigure their quorum slices as a fix.[76] With PoW, on the other hand, "'the chain must go on' even at the price of soft forks."[77]

"As [nodes] add new blocks to their ledger history, they will never go back and change them,"[78] which is why SCP precludes the previously discussed transaction reversal scenarios. However, as Stellar allows for the issuance of alternative tokens referred to as "stablecoins" next to its native cryptocurrency token "lumens" (XLM), another finality-related vulnerability opens up, which Stellar tackles with its concept of "issuer-enforced finality."[79]

---

[68] Lokhava et al., "Fast and secure global payments with Stellar", p. 3.
[69] D. Mazières. *Safety vs. liveness in the Stellar network*. 2019. URL: `http://www.scs.stanford.edu/~dm/blog/safety-vs-liveness.html` (visited on 02/26/2021).
[70] Lokhava et al., "Fast and secure global payments with Stellar", p. 4.
[71] Mazières, *Safety vs. liveness in the Stellar network*.
[72] Ibid.
[73] Lokhava et al., "Fast and secure global payments with Stellar", p. 4.
[74] Ibid., p. 4.
[75] Stellar Development Foundation. *May 15th network halt*. 2019. URL: `https://stellar.org/developers-blog/may-15th-network-halt` (visited on 02/26/2021).
[76] Ibid.
[77] Ibid.
[78] J. Chlipala. *Issuer-enforced finality explained*. 2020. URL: `https://www.stellar.org/blog/issuer-enforced-finality-explained` (visited on 02/27/2021).
[79] Ibid.

Assume the following scenario, taken again from the Stellar Developers blog,[80] with slight modifications. The ECB has issued 100 tokens of their stablecoin "e-EUR" to Alice, which Alice attempts to double-spend with the help of the nodes in the ECB's, i.e., the issuer's, quorum set, which is quite an unrealistic assumption to begin with:[81]

1. Alice spends her 100 e-EUR by transferring them to some Bob, receiving some goods in return.
2. The "[i]ssuer's [i.e., the ECB's] quorum set purposefully omits that transaction from a new block," so the ECB is not aware that these 100 e-EUR are now V's.
3. Alice transfers the 100 e-EUR to the ECB for redemption, which still believes the 100 e-EUR are Alice's.
4. The ECB pays Alice 100€.

Stellar introduces a very simple solution to this issue: The "[i]ssuer makes clear to the world that nobody should believe they have e-EUR unless [the] [i]ssuer's [node] says they do."[82] This way, Bob would have never accepted Alice's payment to begin with, as the ECB had never confirmed it. "That is what [i]ssuer-[e]nforced [f]inality is."[83]

On the one hand, issuer-enforced finality certainly is a centralized approach, as the operationality and integrity of stablecoins directly depend on that of their respective issuers. On the other hand, such stablecoins are accessible within the decentralized network that is Stellar. This duality is reflected in Stellar's design goal "to let entities (safely) issue digital representations of currencies with the certainty of a permissioned [i.e., centralized] network, but the interoperability of a permissionless [i.e., decentralized] network."[84]

This willingness to compromise and take the best of both worlds, centralized and decentralized, is not limited to Stellar's take on stablecoins but is emblematic of Stellar's philosophy in general. It is reflected in SCP as a whole, as SCP, like PoW, facilitates agreement between the nodes of a decentralized network, but, unlike PoW, discards the assumption of trustlessness and, instead, is based on the principle of flexible trust.

While this means that Stellar is technically "solving a different, weaker problem than Bitcoin,"[85] it cannot be overstated that in doing so, it manages to shed Bit-

[80] Chlipala, *Issuer-enforced finality explained*.
[81] Ibid.
[82] Ibid.
[83] Ibid.
[84] Ibid.
[85] A. Grigorean. *Latency and finality in different cryptocurrencies*. 2018. URL: `https://hackernoon.com/latency-and-finality-in-different-cryptocurrencies-a7182a06d07a` (visited on 02/27/2021).

coin's greatest shortcoming, i.e., the fact that transaction reversal is embodied in its consensus algorithm PoW. This is not to mention that by discarding Bitcoin's assumption of trustlessness as well as its disdain for centralized institutions, Stellar also discards properties of Bitcoin that have been traced to extreme right-wing ideology.

It can be summarized that Stellar supports absolute finality in the sense that its consensus algorithm, SCP, has no notion of transaction reversal, as it, unlike Bitcoin, favors consistency over availability, of which one has to be sacrificed in a distributed system according to the CAP theorem. This is the right choice with regards to finality in particular and payment systems in general, as "it's much better for a financial network to go offline temporarily than to produce permanent false or disputed results."[86]

5.4 Technical finality in blockchain systems

In the theoretical analysis of the PoW consensus algorithm, it became clear that transaction reversal is an integral part of it. It is the consequence of blockchain reorganization, which, in turn, is what enables the convergence of nodes on one chain after a chain split in the first place. Furthermore, it became clear that an attacker controlling a majority hash rate can exploit this mechanism to reverse any transaction by creating a new longest chain that does not contain it. The fact that individual nodes only eventually reorganize their blockchain according to the longest chain they are aware of causes PoW to be classified a system that favors availability over consistency according to the CAP theorem. This also implies that there is no discrete point in time at which a transaction begins to be considered part of the ledger – this comes in addition to the fact that it can be removed from it later anyway.

As to the practical implications of these findings, it must be noted that the reversal of transactions previously believed to be final in PoW-based blockchain systems is not only a theoretical possibility but a historical fact in most such systems, including Bitcoin and Ethereum. While the efforts that caused such reversals were, in fact, well intended in that they aimed at reversing actual attacks or software bugs, they technically amount to successfully carried out majority attacks. Furthermore, one must add that PoW is incentive incompatible, which is not least problematic because it incentivizes the formation of ever-growing, eventually majority-representing mining pools that game the system. PoW is also vulnerable to extrinsically motivated attacks, e.g., building attacks, which may come at $1.5 billion for both Bitcoin and Ethereum, respectively, which is remarkable, given that the two systems, at the time of this writing, maintain tokens of a combined value just shy of $1 trillion between them.

[86] Stellar Development Foundation, *May 15th network halt*.

The Stellar system poses a possible solution for these problems, since its consensus algorithm SCP precludes transaction reversal. SCP is fundamentally different from PoW, as it is a type of FBA, which, in turn, generalizes traditional BA, a voting-based consensus algorithm originally conceived for closed systems. However, like PoW, SCP enables agreement between the nodes of a decentralized network. It does so without the assumption of trustlessness, as SCP is based on nodes explicitly expressing trust in one another. Therefore, SCP can be said to solve a slightly weaker problem than PoW. The fact that SCP precludes transaction reversal can be traced to the design decision to favor consistency over availability – as the Stellar network rather halts than have two nodes publish different ledgers, there is no need for a reorg, which, in turn, is why there are no transaction reversals.

To conclude, it must clearly be stated that PoW-based blockchain systems cannot theoretically and have not practically provided technical finality, since a transaction maintained by such a system is only ever probabilistically final, never absolutely. However, with SCP, there exists a consensus algorithm for blockchain systems that provides technical finality. Thus, when considering blockchain systems as a whole, technical finality is possible. The guiding principle is a system's classification according to the CAP theorem – those that sacrifice consistency cannot provide technical finality, at least not from a discrete point in time onward, and, thus, are likely a poor choice as a basis for payment systems.

Chapter 6

Conclusion

As blockchain systems are increasingly considered to underpin wholesale payment systems, this thesis set out to clarify whether and how blockchain-based payment systems achieve final settlement. This question is of particular importance as the ability to achieve settlement finality has a direct impact on the exposure to settlement risks incurred by those system's participants.

In the first part of this thesis, a coherent definition of finality was derived from the literature. Based on an analysis of the Herstatt incident in the realm of FX, a preliminary account of finality was given, characterizing finality as irrevocable settlement in the sense that a transaction can no longer be rescinded or reversed. Said rescinding or reversal is assumed to be due to legal reasons, since the Herstatt incident was a direct effect of the enforcement of German insolvency law.

Said notion of finality turned out to not only apply to FX systems but to payment systems as a whole. It was analyzed under which circumstances competent authorities may challenge and reverse transactions or transfer orders in DNS and RTGS systems. Due to longer settlement cycles with DNS compared to RTGS, transfer orders are more likely to be cherry picked from a DNS system's netting, causing the unwinding of the latter. Thus, RTGS systems are favorable in terms of settlement risks. However, as they process transfer orders individually, they also come at a higher cost.

The SFD was examined as an example of how legislation may enable payment systems to achieve finality. It does so mainly by limiting the effects of other legislation that might interfere with final settlement, e.g., by shielding netting calculations, transfer orders and settlement accounts from legal impact.

Ultimately, finality was as defined as a legal construct, constituting the point in time after which a transaction can no longer be rescinded or reversed. Such notion was compared to other finality definitions from the literature, in the course of which it was clearly differentiated from obligation finality and interpretations specific to traditional banking infrastructures.

In the second part of this thesis, it was examined whether blockchain systems can provide finality as it was defined before, i.e., in the sense of legal finality. Since whether or not a payment system provides legal finality largely depends on its regulatory context, two complimentary approaches to regulating blockchain technology were introduced, i.e., indirect regulation, where it is those who interact with a technology that are regulated, as opposed to direct regulation, where governments exert influence on a system directly.

It was then concluded that a payment system, including a blockchain-based payment system, can only provide legal finality if it – and with it the transactions it processes – falls under the protections of legislation such as the SFD. It was analyzed whether such a system qualifies as a payment system under the SFD's definition, which it does not, which, in turn, is why it was concluded that blockchain-based payment systems cannot guarantee legal finality.

However, the reason why blockchain systems do not conform to said definition is not only due to the fact that their decentralized structure makes them intangible in the sense that they have no legally liable operator or physically tangible location. It was also established that the Articles of the SFD implicitly assume a notion of technical finality, defined as a discrete point in time after which a transaction is permanently recorded in the sense that it is not rolled back by the underlying database, which can be taken for granted in traditional payment systems based on centralized technology but, so the claim, cannot be achieved with blockchain technology.

In the third part of this thesis, this claim was validated, i.e., it was examined whether blockchain systems actually cannot provide this novel notion of technical finality. With regards to Bitcoin-like blockchain systems, it was shown that since transaction reversal is an integral part of the PoW consensus algorithm, such systems can only ever provide probabilistic finality. However, it was pointed out that with Stellar, a blockchain system exists that can provide absolute finality and therefore satisfy the requirement of technical finality.

The critical difference between PoW and Stellar's consensus algorithm SCP was identified in their classification according to the CAP theorem, which states that a distributed system has to sacrifice either consistency or availability. For payment systems, prioritizing consistency, which is the choice in SCP, was identified to be the right choice.

In conclusion, it can be stipulated that settlement finality, defined as the irreversibility of a transaction, is traditionally seen as a legal concept, whereas with the emergence of blockchain-based payment systems, a technical interpretation of the concept becomes necessary as well. Due to their decentralized nature, such systems do, at the time of this writing, not fall under the auspices of legislation such as the SFD, which is why they are unlikely to provide legal finality. PoW-based blockchain systems, i.e., blockchain systems in their most common form, definitely

do not provide technical finality, as they only achieve probabilistic finality. However, blockchain systems employing alternative consensus algorithms, e.g., Stellar's SCP, do provide absolute finality.

The key take-away point is that once regulators adapt legislation such as the SFD to encompass blockchain-based payment systems, and if such systems provide technical finality, e.g., as Stellar does – both of which is in the realm of possible – blockchain-based payment systems can provide settlement finality and are, in this sense, perfectly suitable to underpin large-value payment systems.

References

Bärenfänger, L. K. *Blockchain tokens: a review*. 2020. URL: https://github.com/lkbaerenfaenger/blockchain-tokens-paper (visited on 03/15/2021).

Bärenfänger, L. K. and other contributors. *What are the flaws of this example contract?* 2020. URL: https://ethereum.stackexchange.com/questions/83782/what-are-the-flaws-of-this-example-contract (visited on 12/31/2020).

Barry, N. et al. *Network Working Group Internet-Draft: The Stellar Consensus Protocol (SCP)*. 2018. URL: https://tools.ietf.org/html/draft-mazieres-dinrg-scp-05#section-2.1 (visited on 02/25/2021).

Bashir, I. *Mastering Blockchain*. Packt Publishing, 2020. ISBN: 9781839213199.

Bech, M. and J. Hancock. "Innovations in payments". In: *BIS Quarterly Review* (2020). URL: https://www.bis.org/publ/qtrpdf/r_qt2003f.pdf.

Bitcoin Project. *Some miners generating invalid blocks*. 2015. URL: https://bitcoin.org/en/alert/2015-07-04-spv-mining (visited on 02/14/2021).

bitcoin.it Bitcoin wiki. *Confirmation*. 2018. URL: https://en.bitcoin.it/wiki/Confirmation (visited on 02/10/2021).

– *Controlled supply*. 2020. URL: https://en.bitcoin.it/wiki/Controlled_supply (visited on 01/03/2021).

– *Fork (disambiguation)*. 2015. URL: https://en.bitcoin.it/wiki/Fork_(disambiguation) (visited on 02/09/2021).

– *Hardfork*. 2019. URL: https://en.bitcoin.it/wiki/Hardfork (visited on 02/13/2021).

– *Orphan block*. 2019. URL: https://en.bitcoin.it/wiki/Orphan_Block (visited on 02/10/2021).

– *Proof of work*. 2020. URL: https://en.bitcoin.it/wiki/Proof_of_work (visited on 12/18/2020).

– *Softfork*. 2018. URL: https://en.bitcoin.it/wiki/Softfork (visited on 02/13/2021).

bitcoin.it Bitcoin wiki. *Value overflow incident*. 2016. URL: https://en.bitcoin.it/wiki/Value_overflow_incident (visited on 02/13/2021).

– *Vocabulary: stale block*. 2018. URL: https://en.bitcoin.it/wiki/Vocabulary#Stale_Block (visited on 02/10/2021).

BitMEX. *A complete history of Bitcoin's consensus forks*. 2017. URL: https://blog.bitmex.com/bitcoins-consensus-forks/ (visited on 02/15/2021).

– *ForkMonitor: stale block candidates*. 2021. URL: https://forkmonitor.info/notifications (visited on 02/12/2021).

Blockchain.com Bitcoin Explorer. *Block: 0*. 2009. URL: https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f (visited on 12/17/2020).

Böhme, R. et al. "Bitcoin: economics, technology, and governance". In: *The Journal of Economic Perspectives* (2015). URL: https://www.jstor.org/stable/24292130.

Bonneau, J. "Hostile blockchain takeovers". In: *Financial Cryptography and Data Security* (2018). URL: https://link.springer.com/chapter/10.1007/978-3-662-58820-8_7.

Brewer, E. *CAP twelve years later: how the "rules" have changed*. 2012. URL: https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/ (visited on 03/04/2021).

Buterin, V. *Bitcoin network shaken by blockchain fork*. 2013. URL: https://bitcoinmagazine.com/articles/bitcoin-network-shaken-by-blockchain-fork-1363144448 (visited on 03/15/2021).

– *Ethereum white paper (original version by Vitalik Buterin)*. 2013. URL: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (visited on 03/15/2021).

– *Ethereum white paper (updated version by the Ethereum Foundation)*. 2021. URL: https://ethereum.org/en/whitepaper (visited on 03/15/2021).

– *Tweet declaring Serpent deprecated*. 2017. URL: https://twitter.com/VitalikButerin/status/886400133667201024 (visited on 12/29/2020).

Buterin, V. and other contributors. *Readme file of the GitHub repository of the Serpent programming language*. 2017. URL: https://github.com/ethereum/serpent/blob/develop/README.md (visited on 12/29/2020).

Chlipala, J. *Issuer-enforced finality explained*. 2020. URL: https://www.stellar.org/blog/issuer-enforced-finality-explained (visited on 02/27/2021).

CoinMarketCap. *Historical data for Bitcoin*. 2021. URL: https://coinmarketcap.com/currencies/bitcoin/historical-data/ (visited on 02/22/2021).

Committee on Payments and Market Infrastructures. *A glossary of terms used in payments and settlement systems*. 2006. URL: https://www.bis.org/dcms/glossary/glossary.pdf?scope=CPMI&base=term (visited on 03/16/2021).

– "Core principles for systemically important payment systems". In: *CMPI Papers* (2001). URL: https://www.bis.org/cpmi/publ/d43.htm.

– "Report of the committee on interbank netting schemes of the central banks of the Group of ten countries (Lamfalussy report)". In: *CMPI Papers* (1990). URL: https://www.bis.org/cpmi/publ/d04.htm.

– "Settlement risk in foreign exchange transactions". In: *CMPI Papers* (1996). URL: https://www.bis.org/cpmi/publ/d17.htm.

Contributors to the Bitcoin Core software. *BIP66 in the GitHub repository of the Bitcoin Core software*. 2015. URL: https://github.com/bitcoin/bips/blob/master/bip-0066.mediawiki (visited on 02/14/2021).

– *GitHub repository of the Bitcoin Core software*. 2021. URL: https://github.com/bitcoin/bitcoin (visited on 02/13/2021).

Croman, K. et al. "On scaling decentralized blockchains: a position paper". In: *Lecture Notes in Computer Science* (2016). URL: https://link.springer.com/chapter/10.1007/978-3-662-53357-4_8.

Crypto51. *About*. 2021. URL: https://www.crypto51.app/about.html (visited on 02/22/2021).

– *Bitcoin (BTC): cost for a 51% attack*. 2021. URL: https://www.crypto51.app/coins/BTC.html (visited on 02/22/2021).

Decker, C. and other contributors. *What is the longest blockchain fork that has been orphaned to date?* 2017. URL: https://bitcoin.stackexchange.com/a/4638 (visited on 02/12/2021).

Decker, C. and R. Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P 2013 Proceedings* (2013). URL: https://ieeexplore.ieee.org/document/6688704.

Department of Financial Services, New York State, US. *Virtual currency businesses: BitLicense FAQs*. 2021. URL: https://www.dfs.ny.gov/apps_and_licensing/virtual_currency_businesses/bitlicense_faqs (visited on 01/10/2021).

Dhillon, V., D. Metcalf, and M. Hooper. *Blockchain enabled applications*. Apress, 2017. ISBN: 9781484230800.

Ellul, J. et al. “Regulating blockchain, DLT and smart contracts: a technology regulator’s perspective”. In: *ERA Forum* (2020). URL: https://link.springer.com/article/10.1007/s12027-020-00617-7.

Ethereum Foundation. *Proof-of-stake (PoS)*. 2020. URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/ (visited on 12/29/2020).

– *The Eth2 upgrades: upgrading Ethereum to radical new heights*. 2021. URL: https://ethereum.org/en/eth2/ (visited on 03/15/2021).

European Parliament and Council of the European Union. *Directive 98/26/EC on settlement finality in payment and securities settlement systems*. 1998. URL: https://eur-lex.europa.eu/eli/dir/1998/26/oj (visited on 03/15/2021).

European Securities and Markets Authority. “The distributed ledger technology applied to securities markets”. In: *ESMA Report* (2017). URL: https://www.esma.europa.eu/sites/default/files/library/dlt_report_-_esma50-1121423017-285.pdf.

Eyal, I. and E. G. Sirer. *How to detect selfish miners*. 2014. URL: https://hackingdistributed.com/2014/01/15/detecting-selfish-mining (visited on 02/20/2021).

– “Majority is not enough: Bitcoin mining is vulnerable”. In: *Lecture Notes in Computer Science* (2014). URL: https://link.springer.com/chapter/10.1007/978-3-662-45472-5_28.

Frankel, J. A. *Foreign exchange*. 2020. URL: https://www.econlib.org/library/Enc/ForeignExchange.html (visited on 12/09/2020).

Galati, G. “Settlement risk in foreign exchange markets and CLS bank”. In: *BIS Quarterly Review* (2002). URL: https://www.bis.org/publ/qtrpdf/r_qt0212f.pdf.

Gervais, A. et al. “On the security and performance of proof of work blockchains”. In: *CCS ’16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016). URL: https://dl.acm.org/doi/10.1145/2976749.2978341.

Geva, B. “Payment finality and discharge in funds transfers”. In: *Chicago-Kent Law Review* (2008). URL: https://scholarship.kentlaw.iit.edu/cklawreview/vol83/iss2/7.

Gilbert, S. and N. Lynch. “Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services”. In: *ACM SIGACT News* (2002). URL: https://dl.acm.org/doi/10.1145/564585.564601.

Golumbia, D. *The politics of Bitcoin: software as right-wing extremism*. The University of Minnesota Press, 2016. ISBN: 9781452953816.

Grigorean, A. *Latency and finality in different cryptocurrencies*. 2018. URL: https://hackernoon.com/latency-and-finality-in-different-cryptocurrencies-a7182a06d07a (visited on 02/27/2021).

IBM. *Hyperledger Fabric: the flexible blockchain framework that's changing the business world*. 2021. URL: https://www.ibm.com/blockchain/hyperledger (visited on 01/01/2021).

Kaiser, B., M. Jurado, and A. Ledger. "The looming threat of China: an analysis of chinese influence on Bitcoin". In: *Computer Science, Cryptography and Security (cs.CR)* (2018). URL: https://arxiv.org/pdf/1810.02466.pdf.

Lamport, L. "Proving the correctness of multiprocess programs". In: *IEEE Transactions on Software Engineering* (1977). URL: http://www.cis.umassd.edu/~hxu/courses/cis481/references/Lamport-1977.pdf.

Lamport, L., M. Pease, and R. Shostak. "Reaching agreement in the presence of faults". In: *Journal of the Association for Computing Machinery* (1980). URL: https://lamport.azurewebsites.net/pubs/reaching.pdf.

Liao, N. "On settlement finality and distributed ledger technology". In: *Yale Journal on Regulation* (2017). URL: https://www.yalejreg.com/nc/on-settlement-finality-and-distributed-ledger-technology-by-nancy-liao/.

Liechtenstein Government. *Report and application of the government to the parliament of the principality of Liechtenstein concerning the creation of a law on tokens and TT service providers (Tokens and TT Service Provider Act; TVTG) and the amendment of other laws*. 2019. URL: https://www.naegele.law/files/Downloads/2019-07-12_BuA_TVTG_en_full_report.pdf (visited on 03/15/2021).

Löber, K. M. "The developing EU legal framework for clearing and settlement of financial instruments". In: *ECB Legal Working Paper Series* (2006). URL: https://www.ecb.europa.eu/pub/pdf/scplps/ecblwp1.pdf.

Lokhava, M. *Intuitive Stellar Consensus Protocol*. 2019. URL: https://www.stellar.org/developers-blog/intuitive-stellar-consensus-protocol (visited on 02/25/2021).

Lokhava, M. et al. "Fast and secure global payments with Stellar". In: *SOSP '19: Proceedings of the 27th ACM Symposium on Operating Systems Principles* (2019). URL: https://dl.acm.org/doi/10.1145/3341301.3359636.

Mazières, D. *Safety vs. liveness in the Stellar network*. 2019. URL: http://www.scs.stanford.edu/~dm/blog/safety-vs-liveness.html (visited on 02/26/2021).

– *The Stellar Consensus Protocol: a federated model for Internet-level consensus*. 2016. URL: https://www.stellar.org/papers/stellar-consensus-protocol (visited on 02/25/2021).

Mazières, D., L. Giuliano, and E. Gafni. *Simplified SCP*. 2019. URL: https://www.scs.stanford.edu/~dm/blog/simplified-scp.pdf (visited on 03/15/2021).

Megue, J. P. *SEPA credit transfer: how to understand and add value to your SCT payment project*. Paiementor, 2018. ISBN: 9791094710012.

Messari. *Bitcoin metrics*. 2021. URL: https://messari.io/asset/bitcoin/metrics (visited on 02/22/2021).

Nakamoto, S. *Bitcoin: a peer-to-peer electronic cash system*. 2008. URL: https://bitcoin.org/bitcoin.pdf (visited on 03/15/2021).

Nakamoto, S. and G. Andresen. *Commit 40cd036 to the GitHub repository of the Bitcoin Core software*. 2010. URL: https://github.com/bitcoin/bitcoin/commit/40cd0369419323f8d7385950e20342e998c994e1#diff-608d8de3fba954c50110b6d7386988f27295de845e9d7174e40095ba5efcf1bbL1217 (visited on 02/09/2021).

NiceHash. *SHA-256 marketplace*. 2021. URL: https://www.nicehash.com/my/marketplace/SHA256 (visited on 02/22/2021).

Pagès, H. and D. Humphrey. “Settlement finality as a public good in large-value payment systems”. In: *ECB Working Paper Series* (2005). URL: https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp506.pdf.

Popejoy, S. *IBM’s Hyperledger isn’t a real blockchain – here’s why*. 2019. URL: https://thenextweb.com/podium/2019/05/05/ibms-hyperledger-isnt-a-real-blockchain-heres-why/ (visited on 01/01/2021).

Popper, N. *Bitcoin hits new record, this time with less talk of a bubble*. 2021. URL: https://www.nytimes.com/2020/11/30/technology/bitcoin-record-price.html (visited on 01/14/2021).

Saberhagen, V. van. *CryptoNote v2.0*. 2013. URL: https://github.com/monero-project/research-lab/blob/master/whitepaper/whitepaper.pdf (visited on 03/16/2021).

Schaller, A. *Continuous linked settlement: history and implications*. 2007. URL: https://www.zora.uzh.ch/id/eprint/163690/1/20080261.pdf (visited on 03/16/2021).

Shanaev, S. et al. “Regulatory implications for the cryptocurrency market”. In: *Research in International Business and Finance* (2019). URL: https://www.sciencedirect.com/science/article/abs/pii/S0275531919305963.

Sirer, E. G. *Alternatives to Nakamoto consensus*. 2018. URL: https://www.youtube.com/watch?v=1gUX9ikG1FI (visited on 02/20/2021).

Stellar Development Foundation. *May 15th network halt*. 2019. URL: https://stellar.org/developers-blog/may-15th-network-halt (visited on 02/26/2021).

Szabo, N. *Smart contracts*. 1994. URL: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (visited on 03/16/2021).

– *The idea of smart contracts*. 1997. URL: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html (visited on 03/16/2021).

– *Tweet echoing Donald Trump’s unsubstantiated claims of wide-spread voter fraud in the US presidential election of 2020*. 2020. URL: https://twitter.com/NickSzabo4/status/1334392015304368128 (visited on 01/16/2021).

The r/Bitcoin community on Reddit. *F2Pool is not properly validating blocks*. 2015. URL: https://www.reddit.com/r/Bitcoin/comments/3c2cnj/f2pool_is_not_properly_validating_blocks_their/ (visited on 02/14/2021).

Tsukerman, M. “The block is hot: a survey of the state of Bitcoin regulation and suggestions for the future”. In: *Berkeley Technology Law Journal* (2015). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2587421.

Vereecken, M. and A. Nijenhuis. *Settlement finality in the European Union: the EU directive and its implementation in selected jurisdictions*. Kluwer Legal Publishers, 2003. ISBN: 9789013004878.

whattomine. *Bitcoin (BTC)*. 2021. URL: https://whattomine.com/coins/1-btc-sha-256 (visited on 02/22/2021).

Wuille, P. and other contributors. *Alert: chain fork caused by pre-0.8 clients dealing badly with large blocks*. 2013. URL: https://bitcointalk.org/index.php?topic=152030 (visited on 02/15/2021).

Yeoh, P. “Regulatory issues in blockchain technology”. In: *Journal of Financial Regulation and Compliance* (2017). URL: https://www.emerald.com/insight/content/doi/10.1108/JFRC-08-2016-0068/full/html.

Zhang, R., R. Xue, and L. Liu. “Security and privacy on blockchain”. In: *ACM Computing Surveys* (2019). URL: https://arxiv.org/pdf/1903.07602.pdf.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig abgefasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich bin mit einer Plagiatsprüfung einverstanden.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum	Unterschrift