POLITECNICO DI MILANO

&

SOLVAY BRUSSELS SCHOOL OF ECONOMICS AND MANAGEMENT

Industrial and Information Engineering

Double Degree Master of Science in Management Engineering

The Blockchain as a Financial Market Infrastructure

Supervisor: Prof. Marco Giorgino

Student:

Alessandro Saglimbeni 835901

Academic Year 2015/2016

3.1 Data Selection and Collection

Abstract

This research provides a description of the blockchains and their inner workings, with the objective to understand whether or not they can be used as Financial Market Infrastructures (FMI) and what operational risks are incurred on permissionless networks. First the literature is analyzed with the objective to clearly define what is a blockchain, what are its purposes, which actors play what role, and what are the basic requirements that allow blockchain-based systems to work. This analysis helps to identify a blockchain’s emergent properties, its security model and the limits of the technology. With a good basic understanding of blockchain-based systems, we identify what types of blockchains can firstly be implemented as FMIs. The result of the literature review is that permissioned blockchains are still in their prototypal phase and only permissionless blockchains are currently deployed as FMIs. Nonetheless, open permissionless networks currently incur in several operational risks, most of which are still largely not understood. We therefore analyze such risks and identify potential mitigating actions. In order to do so we collected the data from different sources, on-the-field observations, testimonies, documents and an interview. We then proceeded to analyze this qualitative data through ethnographic analysis and grounded theory. The results emerging from such analysis identify a set of operational risks and mitigating actions, both of which are unprecedented for FMIs, and they exist because of the peculiar design of permissionless networks. Nonetheless, the mitigating actions have proven to be enough to guarantee the continued correct operations of such permissionless networks so far. The results can be partial and are potentially affected by a series of biases of the selected sample of data sources, nevertheless they provide the first systematic analysis of operational risks in permissionless blockchain-based networks. Ultimately this field of research is a novel one and it embraces a broad spectrum of disciplines ranging from game theory, cryptography, distributed computer science and economic and monetary theory. Further research is needed both for what concerns permissionless networks operational risks – with higher focus on attack vectors – but also regarding their inner workings more in general.

Keywords: Blockchain, Financial Market Infrastructure, Properties, Permissionless, Bitcoin, Operational Risk, Decentralized Governance, Hardfork, Softfork

Table of Contents

3.1 Data Selection and Collection

Table of Figures

3.1 Data Selection and Collection

Table of Sources

D

Document 1

I

Interview 1

O

Observation 1

Observation 2

Observation 3

Observation 4

Observation 5

T

Testimony 1

Testimony 2

Testimony 3

Testimony 4

Testimony 5

Testimony 7

INTRODUCTION AND THESIS STRUCTURE

In 1995 a man named McArthur Wheeler robbed a bank in Pittsburgh. He was caught, because his only disguise was lemon juice. He covered his face with it, because he knew that lemon juice could be used as an invisible ink when writing on paper. He knew so little about how that worked and he knew so little about how cameras worked, that he assumed, with extreme confidence, that lemon juice could make him invisible too.

Novices, people unskilled in particular disciplines will often overestimate their knowledge and abilities in said disciplines, because they do not know how little they know and how much more there is to learn. Experts on the other hand, underestimate their relative competence, because they do know how much is yet unexplored. This that we just described is a cognitive bias, called the Dunning-Kruger effect (Kruger and Dunning, 1999). What drives this bias is the fact that often the more you learn about something, the more you realize how rich, complex and full of yet unanswered questions that matter really is.

Paraphrasing the anecdote of Mr. McArthur, many professionals and journalists from all over the world are behaving exactly the same way. They believe the blockchain will solve humanity’s problems, just like the lemon juice should have made McArthur invisible to cameras. A deeper analysis of this emerging technology could hopefully help us to better understand what can and can’t be made with it, or at worst realize how little we know of it, yet.

This introductory section has three objectives:

1. Understanding what major incumbent companies are trying to use the blockchain for, as well as going through some examples of what we mean by background noise covering signal.
2. Clarifying the overall aim of this research and our contribution to the existing body of knowledge; and defining the scope of the thesis, by outlining the various sections we will go through.
3. Defining the research strategy that has been adopted by the author.

1 What is going on with the blockchain, hype & expectations

Why as days pass on, more and more people talk about blockchain? What are they even seeking for? Why are they trying to apply this *versatile* technology to each and every industry/ business/ process/ problem? Even though we will not analyze each and every single news that received press coverage, we want to go through some of the most salient ones. This exercise will prove to be useful in order to understand why we deem it necessary to demystify on what really is blockchain and what is not.

The blockchain is a technology that underlies Bitcoin. This is the first basic explanation given by Wikipedia (En.wikipedia.org, 2016a) as well as by most of the people when it comes to explain what a blockchain is. In reality, blockchain is a made up word. It was never called this way by Bitcoin's creator Satoshi Nakamoto in the original paper (Nakamoto, 2008). Indeed, all he talks about is a chain of blocks, i.e. hashes of transactions encapsulated into one another.

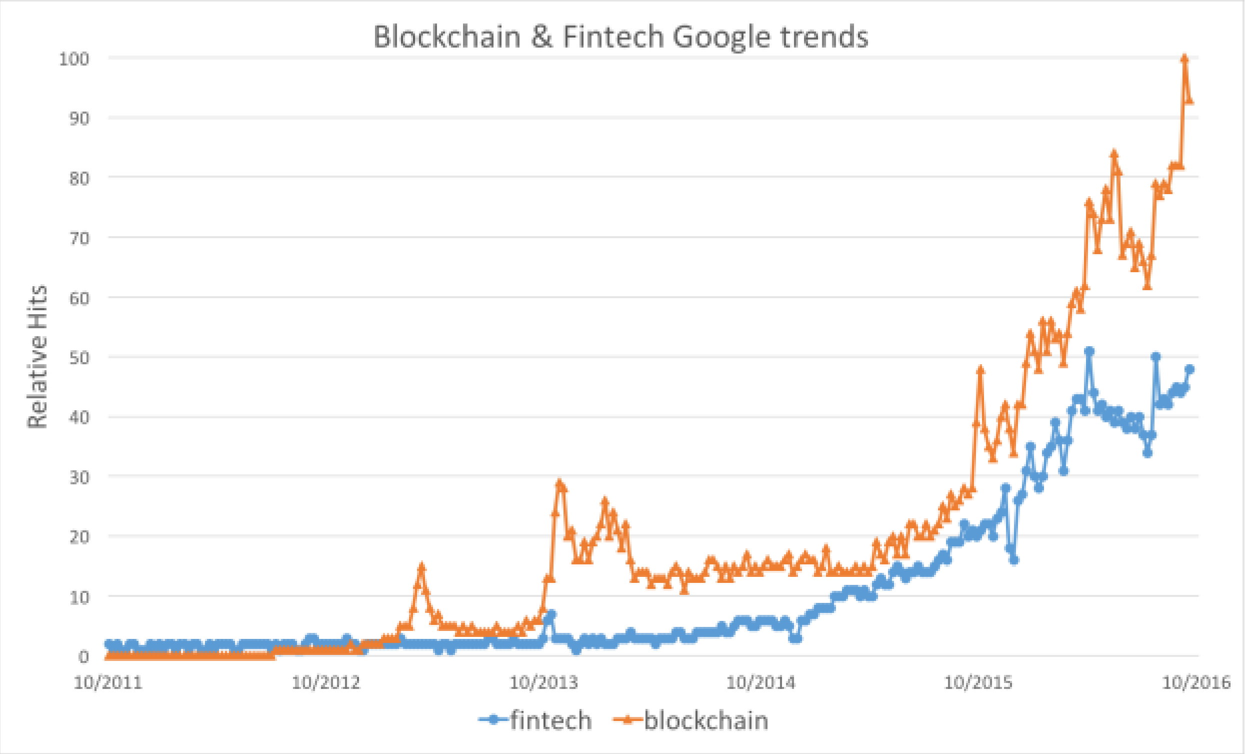


Figure 1 Google trends, relative research hits of the keywords Fintech and Blockchain.

The "blockchain" is so trendy that it made twice as many hits on Google as the keyword "fintech". Almost all banks globally have their own blockchain research team or laboratory, 70 financial institutions participate in a consortium (Reuters UK, 2016), 100 different companies publicly participated in about 70 Proof of Concepts on 25 different areas (Rizzo and Miles, 2016).

As identified also by Gartner's Hype Cycle for Emerging Technologies (Prwire.com.au, 2016), the blockchain made it all the way to the apex, but the expectations regarding what this technology could actually deliver, are inflated.

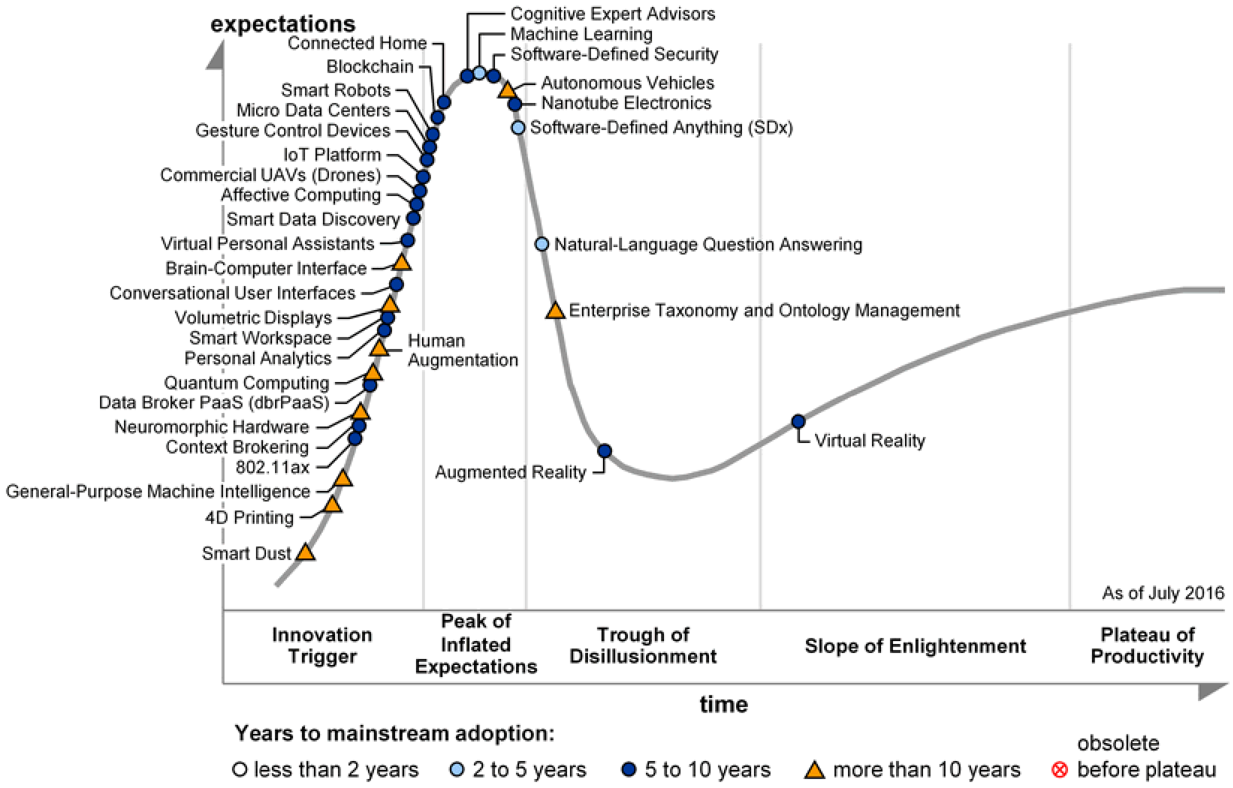


Figure 2 Gartner's 2016 Hype Cycle for Emerging Technologies

In a famous article that made The Economist cover (The Economist, 2015), this technology was described as a peer-to-peer transparent ledger with "a host of other uses [other than bitcoin]". The alleged uses that most news articles refer to, draw from a research made by Oliver Wyman for Santander InnoVentures (Belinky et al., 2015). There it was described how the "blockchain technology" (rather a distributed ledger, for being more accurate) could help to cut costs in banks' trade finance back-end processes thanks to IoT sensors and actuators; how it could enable smoother post-trade securities settlement; and how it could ease the process of issuing new mortgages. All of this without the need of intermediaries, hence leading banks to cut $15/20 billion/year of costs by 2022.

3.1 Data Selection and Collection

Therefore, the whole **point of this technology** would be that of **eliminating intermediaries, ease compliance and surveillance through the use of a shared ledger and inter-institution process automation thanks to smart contracts**. Now, we need to remember the above-mentioned figures and the expectations of Financial Institutions, because by the end of this research we will realize that it is not that obvious whether this technology will actually allow to cut on intermediaries' costs. We will see all of the different pieces making up this technology, thus understanding how early stage and experimental it is.

2 Research Focus

Given the expectations and the confusion around what improvements or disruptions "blockchain technologies" may bring to the financial markets, as well as to the society as a whole, the overall aim of this thesis is "understanding whether a blockchain can be used globally as a Financial Market Infrastructure". Given the generally low level of understanding over what this new technology really is, this is configured as an exploratory study. In order to provide an answer to our main research question, several inter-related objectives need to be assessed and analyzed first.

Clarify what are blockchain-based networks and **how do they work**:

- *understanding* the *purpose* and need for these networks
- *identify* the relevant *actors* and *assess their role* within such infrastructures,
- *identify* the basic *requirements* that a blockchain-based network should fulfill in order to serve its purpose,
- *classify* and *distinguish* the different *network architectures* along the relevant dimensions,
- *formulate* what are the *properties* and the *security model* of blockchain-based networks,

Assess the **relationship between Financial Market Infrastructures and blockchain-based systems,** with particular focus on:

- *understanding* how a blockchain-based network can be used as a Financial Market,
- *identify* the *current limitations and risks* inherent to these distributed consensus infrastructures, hindering the viability of the proposed applications.

Evaluate permissionless blockchain-based networks' risks identified throughout the literature review:

- Assess how the decentralized governace impacts on permissionless network security
- Assess to what extent they may hinder permissionless networks' viability as FMIs,
- Categorize the operational risks,
- Identify actions and processes for eliminating or mitigating those risks.

The researcher followed a largely non-linear process before getting a better understanding of all of the implications of the overall aim of this thesis. In fact, the relation between the various sub-objectives with the overall aim of the thesis are yet obscure and largely not researched in academic literature. This means that the sub-objectives were routinely refined in multiple, more detailed steps as the research evolved over time. This doesn't mean that the research followed a random path. On the contrary the researcher would continuously make assumptions with regards to its research questions and test them through a direct interaction with experts all around him. For instance, a reader may find it curious that sometimes (especially at the beginning) the researcher talked about blockchain, while later on he referred to blockchain based-networks. This is intentional. As a matter of fact, the writer himself did not realize what was the subtle difference between the two words, but with further research he came to understand that referring to blockchain was too general and prone to ambiguity, while actually we should be referring to blockchain-based networks, or even better blockchain-coordinated networks. The latter was never used, just because it seemed to be way too much of a verbose practice, but indeed the latter – blockchain-coordinated (distributed) network – would be even more correct. This gets into semantics, but chapter "4 What is a blockchain and what are its properties" explains all of these details and helps the reader in understanding why this matters.

When the researcher started its quest for discovering what the blockchain was exactly and how it could have impacted the financial sector, he wanted to run a cost-benefit analysis aimed at comparing the potential advantages and costs of a blockchain based infrastructure as compared to legacy centralized FMI. Not only the question is extremely vast, but the worst part of it was collecting data on this new technology. While for permissioned infrastructures no such data exist – simply because no permissioned network is up and running in production – for what concerns permissionless networks it simply does not make sense to compare its costs to a centralized infrastructure for two reasons. Firstly, the infrastructure is extremely young and its cost structure is evolving through a mix of separate layers, efficiency improvements within the protocol itself and the so called client-side validation methods. All of the above mentioned techniques will drastically change the costs of theses decentralized networks, making it pointless to compare the new ecosystem with the legacy FMIs. Secondly even when the innovation curve will stabilize as the technology becomes mature, the decentralized networks will be consistently more expensive, but this is because of two necessarily different objectives, trustlessness and censorship resistance. The target users of permissionless networks is starkly different to that of traditional FMIs, so different that comparing their costs would be as pointless as comparing the operating costs of horse-drawn carriages with those of a car. When you have a paradigm shift which is not aiming at efficiency gain, but rather at effectiveness gains you need to change perspective accordingly.

When the researcher realized this, his focus shifted towards the objective of explaining what really is a blockchain-based network. The reason for this shift, was that the word “blockchain” is still widely and wildly misused (JPMorgan, 2016, BigchainDB, 2016) – even right now almost as much as one year ago. Indeed, many companies pitch their services or claim that their innovation departments use “the blockchain”, even when those systems – if closely examined – are clearly not blockchain-based in any way.

The objective of the first iteration of this research – the one defended at Solvay Brussels School – therefore aimed at demonstrating the properties of a blockchain-based network while demystifying over the linguistic abuses of the word blockchain.

This objective also served to make another important clarification. The misconceptions around terminology lead many to believe that the “blockchains” are a disruptive innovation (Arnold, 2016). We can argue that basing a Financial Market Infrastructure on a public permissioned blockchain is indeed disruptive. Nonetheless, most of the times we are not actually talking about a blockchain-based system, but rather something akin to distributed databases, something that existed for decades and which represents incremental – rather than disruptive – innovation. Permissioned blockchains basically try to fit the old paradigm in the new clothes, but they are tight.

Time and further reflections, allowed the researcher to realize that the real matter needing further research both for the academia and for the general public, is not whether blockchain-based networks can become a Financial Market Infrastructure – be it a payment system or a distributed securities settlement system – but more specifically if permissionless blockchain-based networks can ever become any kind of global FMI. The first chapter of the literature review section will explain the different assumptions and structures of permissioned and permissionless blockchain networks. As a result, within the second chapter of the literature review we will get to appreciate how the different assumptions play a critical role when it comes to using a permissionless blockchain network as a FMI, involving a series of whole new operational risks which were not present in the legacy FMIs. From this understanding we realize that it is much more challenging, but at the same time disruptive, to use permissionless blockchain networks as Payment Systems, or as Securities Settlement Systems. This is why it is considerably more interesting to gather a further understanding on this specific issue through a series of interviews.

Contribution. Our main contribution consists of a thorough analysis of the risks affecting permissionless networks and a systematic review of the existing risk mitigation actions. Moreover, a contribution is given within the literature review section. In chapter 4 we systematized the available knowledge, in order to better define what a blockchain is, with a broad generic definition and an inductive approach. Our second main contribution is the identification of the properties of a blockchain-based system, using a deductive approach by leveraging on the previous definition.

3 Research Design

Usually the research design and methodology used to investigate on the overall aim are analyzed after the literature has been reviewed. Nonetheless, it is necessary for the reader to know ex-ante the rationale and the resources used by the author to develop the theoretical framework of the literature review, otherwise it may seem that it was created out of thin air.

As stated before, this study followed a non-linear path before getting to its current form. This is because the purpose of this research is ultimately an exploratory one. Clearly describing the extant knowledge regarding the topic was not an easy task. Since blockchain networks are a novel technology and nothing like them was taught in any course at the university, catching up with the basics required a long time. In order to speed up the process the author decided to approach the research through multiple means. The academic literature and the passive research were complemented with direct observation data collected throughout an internship (and later a job) at BLOCKCHAINLAB – a startup specialized in applied research in the field of distributed blockchain-based networks.

3.1 Data Selection and Collection

Depicting the state of the art of "blockchain protocols" (we will later understand why we are using quotation marks on those words) is a very hard task. The information asymmetry is huge and very often biased, or even biasing for the readers. As of December 2016 academic literature was widely lacking. Searching on *IEEE Xplore, JSTOR Business Collection, Scopus* and *Web of Science (ISI Web of Knowledge)*, we got 109 results searching the keyword *blockchain* – 159 as of April 2017 – of which only 9 are published on peer reviewed journals, none of which was relevant for our analysis. In fact, all of them focus on analyzing how Bitcoin works, or propose some brand new purpose-designed permissioned blockchain for supply chain management or IoT use cases, which is out of this thesis scope.

For this reason, we used a very heterogeneous set of information resources. The objective is to manage to reach a wide spectrum of different perspectives on the topic, so as to test for the validity and reliability of the various sources.

3.1 Data Selection and Collection

3.1.1 Secondary Data Sources

Textbooks. Those were mainly used when describing Bitcoin's inner workings. Indeed Antonopoulos (2014), Franco (2014) and Wattenhofer (2016) are the best available textbooks (together with Princeton University MOOC on Coursera) and are being used as reference books also by the first academic courses on Bitcoin and Blockchain technology (such as Politecnico di Milano's course). These books were used especially at the beginning in order to learn the very basics of this very large study domain which embraces:

- Game theory
- Cryptography
- Distributed computer networking and data transmission
- Economic and monetary theory.

Given the vastness of this research field, one easily realizes why is it that very few people are competent when talking about "blockchain technology".

Governmental Institutions' Reports and Research Papers. Those are another important source of information, not so much as for understanding the technology, but rather to get a sense of the various Government's and Regulators attitude towards this technology, whether they are permissive or if they plan to regulate this field.

These reports also played an important role when defining the scope of the thesis. In fact, what emerged from the analysis of these papers was that the regulators are – not surprisingly – more friendly towards permissioned blockchain companies than they are to permissionless networks. This is one of the reasons to further investigate into permissionless blockchain networks as FMIs, all the odds are against them. This is also why it is more interesting to examine whether the preconditions for using permissionless networks for FMIs exist or not.

Academic Literature. Unfortunately, the academic research worldwide is lagging behind when it comes to the study of blockchain technologies, both for what concerns its definition and the classification of various existing blockchain technologies. Way too often research papers offer little clarity over the terminology being used, due to the early stage development of this field of research. A few articles were worth citing during our analysis (because within the scope of our thesis), but none of them is peer reviewed. Again, this technology is extremely young. Bitcoin's whitepaper was released in late 2008, and the hype around the blockchain buzzword started its rise in 2014. Moreover, the expertise domains encompassed by these technologies is really vast, which makes it highly unlikely that a single researcher can possibly possess the necessary knowledge.

This is why any research that is limited at investigating on traditional secondary sources will yield little results at this stage. Interaction with many different domain experts is here – and now – necessary more than ever.

Private Companies' Papers. These are a very useful source of information. They are all practically useless when it comes to understand how the technology works, if it is secure, or even if it is viable for certain applications. They are too much biased and superficial for a scientific assessment of this technology. On the other hand, those reports are a critical piece of the puzzle as to understand the expectations of the relevant players and institutions, and by induction those of the markets alike. Probably these sources of information are the ones that are the most responsible for feeding the current hype level and creating irrational expectations.

Websites and Press Articles. These are another useful source of information. The articles in particular transparently reflect market expectations and opinions on the topic. Not only that, but since this ecosystem is rapidly evolving, following the news daily is not an option. Being constantly up-to-date on the latest developments is mandatory – both for what concerns new organizations entering into the market or startups pivoting, as well as for new technological developments or emerging protocols. On the other hand, online encyclopedias such as Wikipedia was used for some definitions of those concepts that are very much out of the Business Engineering field of studies.

3.1.2 Primary Data Sources

White Papers. These primary sources were extremely useful for understanding the architecture and the purpose of different existing blockchains. They usually describe a project's motivation, and specifications. For this reason, it is more useful to read those rather than secondary sources.

Direct observations and interaction with field experts. A big part of this research was helped by the guidance of experts with whom I got in touch through my internship at BLOCKCHAINLAB. Such contamination process allowed me to get beyond the "simpler" concepts of 2013 Bitcoin, allowing me to comprehend all the latest advancements in the Bitcoin Protocol and its surrounding protocols and blockchains. Of course referencing their contribution throughout the thesis is quite difficult (even though sometimes this has been done), since this was an ongoing process which started in late June 2016 and it is still ongoing.

A constant interaction with domain experts facilitated the learning process and helped the researcher in testing his own hypotheses regarding his understanding of many specific sub-objectives. It is worth explaining how these observations and interactions worked and how they were codified into knowledge within this thesis.

BLOCKCHAINLAB has a big network of experts collaborating with the company. Some of them are resident experts – R. Casatta, T. Bertani, L. Nahum, G. Zucco, F. Ametrano – the interaction with them was continuous and allowed to have a continuous feedback over unsolved issues which emerged over time. Others are visiting experts – P. Todd, J. Lopp, P. Sztork, A. Antonopolous, C. Allen, T. Dryja, J. Poon, M. Corallo, P. Wuille, R. Shea, N. Bacca, M. Erhardt, J. Grigg, C. Decker, R. Spagni – which the author had the chance to meet and talk to for around a week every time they would come to Milan and visit BLOCKCHAINLAB. Those interactions were less formal and more difficult to codify, as the interaction usually lasted for shorter amounts of time. Nonetheless each and every of those interactions was extremely valuable, as it allowed the researche to gain little by little a clearer big picture of what was the Bitcoin and blockchain technology all about. All of these interactions are eventually reflected into the overall structure of the thesis and explanation of the concepts, but some of them have been recorded and transformed into interviews for the purpose of this research.

Conferences. Thanks to the work experience at BLOCKCHAINLAB, the author of this research was also able to participate to three conferences, Scaling Bitcoin Milan, Blockchain Protocol Analysis and Security Engineering 2017 at Stanford and Construct 2017 at San Francisco. These conferences proved to be extremely useful also for the purpose of this study. The author had the chance to listen to and talk to the major experts in the field. At Stanford he even had the chance to confront himself with the only researcher investigating into whether permissionless blockchains shall be used as FMIs, Angela Welch.

3.2 Data Analysis

The objective of using multiple sources of data was that of comparing and cross-checking data collected through observations at different times, in different places, and from people with different perspectives.

The researcher's observations as direct participant in the blockchain ecosystem implies his immersion in the research setting, with the objective of sharing in peoples' view regarding the ecosystem, while attempting to learn their cryptic world:

- All of the observations were constantly recorded as they were generated. Little by little they were codified into categories and finally into hypotheses,
- Opportunistic conversations were used throughout the study to explore the perspective of knowledgeable individuals over specific issues,
- These conversations took place any time the single expert was free and willing to talk for a short period. Different questions would be asked depending on the specific subject's interests.
- The information gleaned from these was compared with the findings from observations and with secondary data sources such as the academic literature, governments reports or private companies reports.

The purpose of these comparisons was to identify differences between theory, high level concepts and practice, that could indicate a specific need for clarification over specific notions. This is why we constantly validated the data stemming out of the literature review, with the help of some expert which could assist us in directing the research towards a deeper understanding. This does not mean that the author simply trusted the experts making appeal to authority. On the contrary he relentlessly tried to understand as much as possible of those concepts on its own and then confronted his own understanding with that of a knowledgeable person, so as to identify all the fallacies of his own understanding.

3.2 Data Analysis

LITERATURE REVIEW

4 What is a blockchain and what are its properties

This chapter lays down the basic knowledge and concepts needed in order to understand what is the blockchain. In some passages the chapter will get a little bit technical. That is not pure academic exercise, but rather we try to introduce some notions that we deem critical for a more comprehensive understanding of the technology being explored, one that does not stop at the surface.

The chapter itself is divided into three sections:

1. Blockchain Definition. In this part we inductively introduce a definition of what is a blockchain, what are its purposes, which actors play what role, and what are the basic requirements that allow blockchain-based systems to work. In order to do so we firstly present how Bitcoin – the world's most widespread blockchain-based network – works. Then we extrapolate the fundamental elements – so as to get rid of all the fringes – in order to get to a more general model.
2. Blockchain Taxonomy. Once we laid down the basic concepts and a framework of analysis, we try to understand how the existing (alleged) blockchain networks differentiate between one another.
3. Blockchain-based Networks Properties. Finally, based on the analysis of textbooks, academic literature, incumbents' expectations, government bodies and consulting businesses reports and press articles, we formulate the most appropriate properties associated with blockchain-based networks. We then put all the pieces together, in order to understand how the highlighted properties fit with how a blockchain-based system works.

First of all, one might wonder why most Financial Institutions (FIs) are looking at and investing on blockchain technologies, either within their own R&D teams or through specialized startups. The question is legitimate, hence let's have a look at what is being as advantages that can be gained by using a blockchain in the financial services industry:

- Using a blockchain as a global payments system infrastructure for inter-bank settlement (Mills et al., 2016). This would enable reduced clearing and settlement time with greater security against thefts and frauds, especially for certain operations such as international remittance (Williams-Grut, 2015), syndicated loans and collateral management (McWaters, 2016).
- The transparency and accessibility of a blockchain would facilitate Anti-Money Laundering and Know Your Customer processes, thus reducing compliance costs (Quinlan and Kwan, 2016, Trinder, 2015).
- Using a blockchain for transferring securities between their owners automatizing the full management cycle of private company securities and securing them just like in Bitcoin (2016c, Belinky et al., 2015, Mainelli and Milne, 2016, McWaters, 2016, Mills et al., 2016, Pinna and Ruttenberg, 2016, Symons et al., 2016, Van de Velde et al., 2016)
- The latest trending topic (Castillo, 2016b, De Meijer, 2017, Shin, 2016) has many thinking that a way for implementing all of this, may be a "sovereign" Blockchain (Ehsani, 2016), one run by a central bank.

Now, let's get back to our research question. Our purpose is to understand about the blockchain:

- What is the purpose of a blockchain;
- whether or not it *is* a Financial Market Infrastructure;
- if it is indeed a FMI, what type of FMI is it best fit for;

In order to get to understand such matters, we must start from a solid ground. We need a rigorous definition of what is a blockchain. We will see that this quest is quite a complex one and the following section 4.1 is going to be hard. In fact, defining the problem and abstracting the existing model requires some basics of distributed computer science that we assume the reader does not have. Why so? Because we will see that this overly complicated infrastructure was built this way solely for the purpose of being an extremely resilient *trustless* infrastructure (we will later understand what trustless really means).

4.1 Blockchain definition

There is no shared precise definition of what a blockchain is, nonetheless a formally correct definition exists. For instance, the New Oxford American Dictionary defines it as:

> *"a digital ledger in which transactions made in bitcoin or another cryptocurrency are recorded chronologically and publicly".*

This is a correct definition. The real problem is when it comes to semantics. Understanding what the various interlocutors one may come across actually are referring to is not straightforward. For instance Don and Alex Tapscott are two businessmen (and brothers) and technology enthusiasts, also famous for writing the suggestive (using an euphemism) book "Blockchain Revolution" (2016). In an article they recently stated:

> *"Blockchain technology has emerged globally as the second generation of the Internet." (Tapscott and Tapscott, 2016).*

In this context, the two brothers are actually referring to a *distributed computing systems*[1] based on strong permissionless *consensus mechanisms*, such as Bitcoin's, rather than to the virtual ledger of chronologically ordered blocks of transactions. This use became widely spread, but it is quite confusing. Indeed, the security properties and the peculiarities (which will be analyzed in section 4.3) of such distributed systems, widely vary mainly depending on the *consensus mechanism* they adopt in order to synchronize all of the nodes and to prevent – or rather tolerate – system failures[2].

At this point, it is very difficult to continue with further analysis of how to define a blockchain, without an explanation of how the first and most widespread one works.

4.1.1 Bitcoin

Thoroughly understanding Bitcoin is quite a challenge. This is a complex, delicate and new infrastructure, aimed at allowing any person around the globe, with an internet connection to be able transact value peer to peer. Nakamoto (2008) specifically refers to Bitcoin as a peer to peer

[1] A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal. Three significant characteristics of distributed systems are: concurrency of components, **lack of a global clock**, and independent failure of components. (En.wikipedia.org, 2016b).

[2] System failures are mainly of two kinds content or timing. A computing system may either deliver erroneous content, or it may deliver it with a wrong timing (too late, too soon, or never), or both at the same time. AVIZIENIS, A., LAPRIE, J.-C., RANDELL, B. & LANDWEHR, C. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing,* 1**,** 11-33.

electronic cash system. We will not investigate on whether or not the value of bitcoin tokens is justified, or if it could ever be considered a currency on its own. Rather we will take it as it is, a decentralized payment system infrastructure, because value (even if volatile) is effectively exchanged among its users through the Bitcoin Network infrastructure.

Four main resources were used in order to write the following section.

Understanding Bitcoin by Pedro Franco (2014) which is probably the best and most comprehensive textbook on Bitcoin as of 2016, especially suited for *non-techies*.

Mastering Bitcoin by Andreas Antonopoulos (2014) gives a little bit of a different perspective on the topic, delving into more specific technical details, while omitting some basics of computer science. One of the best characteristics of this book is that it is completely open-source because it is licensed under the *Creative Commons Attribution-SareAlike 4.0 International License*[3], this means that we are free to reproduce the book's images within this thesis.

The Science of the Blockchain by Roger Wattenhofer (2016) is a quite an advanced textbook that gives an overview of the existing algorithms and protocols for fault-tolerant distributed systems. This book is very technical, but it provides the readers with the basics for understanding distributed computing systems.

Finally, the website *Bitcoin.org* (2016) is an extraordinary source of detailed information.

First of all, some definitions:

Bitcoin Protocol. The set of rules which specify how to build the distributed database, how to construct the transactions, what are the necessary conditions for considering a transaction valid, and all of the rules and incentives which secure the entire system.

Bitcoin Core open source client implementation. This is the original software written in C++, which implements the rules as they are specified in the *Bitcoin protocol*.

Bitcoin Network. The peer-to-peer network to which nodes[4] running the *Bitcoin Core client implementation* connect. This is the layer enabling nodes to exchange messages containing mainly blocks and transactions.

[3] Under this license **You are free to: Share** — copy and redistribute the material in any medium or format; and **Adapt** — remix, transform, and build upon the material; for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Basically giving appropriate credit. (Creativecommons.org, 2016).

[4] A **node** is a basic unit used in computer science. Nodes are devices or data points on a larger network. (En.wikipedia.org, 2016c). We will see later on that in Bitcoin there are 3 main types of nodes.

bitcoin tokens. Notice that this is the only one with low-case b. In fact, Bitcoin's native currency (whose official ticker is XBT, not BTC) is differentiated from all the other terms. This token is divisible into 100,000,000 satoshis (the smallest unit of account as of today, but could be changed in the future), and there is a hard limit of 21 million bitcoins which will ever be issued.

The problem that Nakamoto was trying to solve was creating digital value, one such that it could be exchange between people. Many before him tried in this quest, the most famous and noticeable being Chaum et al. (1990), Dai (1998), Back (2002), Szabo (2008).

The naïve approach would be that of assigning value to a certain data pattern, some string of bits representing money. Unfortunately bits are way too easy to be copied, leading to an uncontrolled inflation of the available currency. Each user could reuse his/her funds an indefinite number of times, the *double-spend problem*. A way to make to make this data scarce is needed. Hence the next step would be that of creating a central database, containing a list of accounts in each row, representing the amount of funds held by the users of such system. This way you can easily solve the double-spend problem. The issue with such architecture is that this central database is a *centralized single point of failure*.

There are many different kinds of faults (Avizienis et al., 2004), which may lead to a system failure. A hardware fault for instance may lead the system to stop responding, hence reducing the *availability* of the system. Well no problem then, we could make this system redundant and replicate this database over multiple nodes. Nonetheless, when we take the path towards distribution of a computing system, we are exposing our system to other, much more complex kinds of faults. Indeed, we need to keep in mind that we are trying to distribute a database containing information regarding how much value each person holds. How can we be sure that none of the nodes is delivering wrong information to its users? Which means how can we ensure that the entire system is *consistent* with itself, at all times, with all users? Moreover, now that the system is a distributed one, how do we make sure that the system continues running even if some node fails? Hence, how do we ensure that the system is *partition tolerant*?

4.1.1.1 Bitcoin Protocol basics.

We now need to make a little step back and define some words, in order to understand what kind of problems blockchain-based distributed systems are trying to solve. Following Wattenhofer (2016) analysis, we want a distributed system to provide:

Availability. A system shall be *operational at all times* and instantly processing incoming requests.

Consistency. All system's nodes agree on the *current state of the system*.

Partition Tolerance. The system shall operate correctly even in the presence of a network partition. A network partition is a failure where the network splits into multiple parts that cannot communicate between each other.

CAP theorem. Fox and Brewer (1999) demonstrated that these three properties cannot be achieved *simultaneously* in a distributed system.

This is a critical step for us. We are trying to build an **infrastructure for holding and exchanging value**, but we need some mechanism in our protocol that allows to achieve the three of them. We want this system to be always *available* for its users. We need it to be *consistent*, otherwise a user may find out that he lost all of his money according to a node, while he still has his money according to some other node. Which node is giving the correct information? Finally, we want the system to be *partition tolerant*, otherwise we may end up with two different networks of nodes. As time passes the two partitions will create two alternative transaction histories, inconsistent between each other. How does Bitcoin solve this dilemma?

Bitcoin manages to achieve all of the three properties relaxing one condition. Indeed, Bitcoin achieves *Availability*, *Partition Tolerance*, and *Eventual Consistency*. The latter is a weaker form of *consistency* which allows nodes to disagree temporarily, but then a *conflict resolution* mechanism is necessary in order to solve for these inconsistencies. This mechanism is the **proof-of-work** and it is demonstrated by a node to the rest of the network by means of the **blockchain**. Let's now try to get to understand this.

4.1.1.2 Transactions

The best way to understand Bitcoin is going through the cycle of a simple transaction. Let's say we don't have any bitcoin and we ask a friend, Joe, to send us some. In order to do so he needs to know a Bitcoin address of ours, but we still do not have. In order to create one, we can use a software called *Wallet*. What the software does is basically creating a pair of private-public cryptographic keys. Asymmetric cryptography is very useful because it allows an individual to be authenticated in a Public Key Infrastructure (Paar and Pelzl, 2009). In Bitcoin it allows users to demonstrate that they really possess a certain amount of bitcoins.

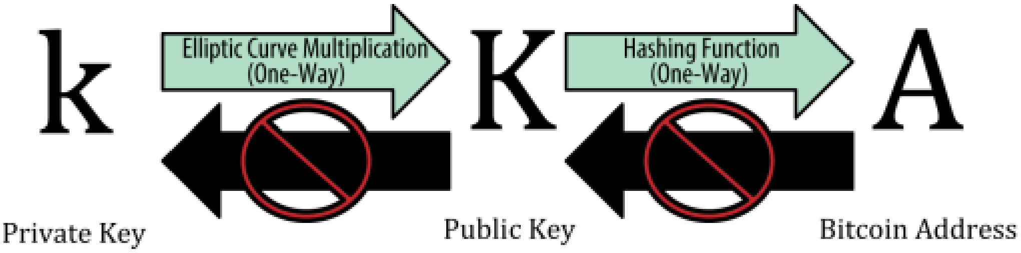


Figure 3 Creating a Bitcoin address (Antonopoulos, 2014).

Our wallet will therefore generate a private key, use very particular function (Elliptic Curve) in order to derivate a public key, which will then be (double-)hashed[5] in order to create our *Bitcoin Address*, which looks like a random alphanumeric sequence. We are now ready to receive the bitcoins.

[5] A “hashing algorithm” or simply “hash algorithm” is a one-way function that produces a fingerprint or “hash” of an arbitrary-sized input. Cryptographic hash functions are used extensively in bitcoin: in bitcoin addresses, in script addresses, and in the mining proof-of-work algorithm. (Antonopoulos, 2014).

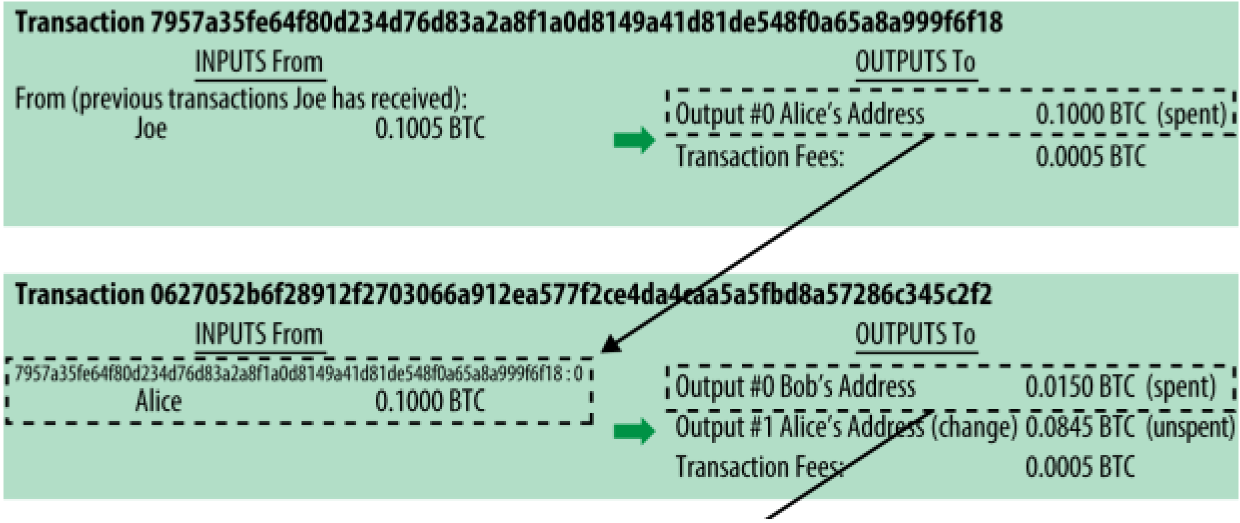

Transaction 7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18
INPUTS From
From (previous transactions Joe has received):
Joe 0.1005 BTC
OUTPUTS To
Output #0 Alice's Address 0.1000 BTC (spent)
Transaction Fees: 0.0005 BTC
Transaction 0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2
INPUTS From
7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18 : 0
Alice 0.1000 BTC
OUTPUTS To
Output #0 Bob's Address 0.0150 BTC (spent)
Output #1 Alice's Address (change) 0.0845 BTC (unspent)
Transaction Fees: 0.0005 BTC

Figure 4 Exchanging bitcoins (Antonopoulos, 2014).

We ask Joe to send us (Alice) 0.1 bitcoin. Joe will broadcast to the network a transaction (whose ID starts with 7957a35…) where he demonstrates that he owns the Inputs of the transaction (0.1005 XBT). In Bitcoin transactions have Inputs and Outputs. Notice how all Inputs have been Outputs at some point of the transaction history. Outputs are made of two parts: the number of bitcoins contained inside the Output, and the *locking script* which encumbers the bitcoins contained in the Output to a *Bitcoin Address*. This means that when Joe will send us 0.1 XBT, he will have to demonstrate to the entire network that he owns a secret private key that can mathematically generate his Bitcoin Address. But now Joe is sending this money to our newly created Bitcoin Address. This means that he is creating a transaction where in the Inputs he demonstrates that he owns the private corresponding to the address encumbering the bitcoins, while in the Outputs he tells the entire network to encumber 0.1 XBT to our Bitcoin Address. Effectively only us will be able to spend those Outputs in the future, because only us (if we are careful enough) possess the private key that generate that specific Bitcoin Address, in other words we are the only ones who can sign a transaction related to those Outputs. Until we do not spend these Outputs they will be called *Unspent Transaction Outputs – UTXOs.*

4.1.1.3 Blockchain and Proof-of-Work

We just understood how users can control their funds and transact between each other, but who decides whether a transaction is valid, and how are bitcoins initially distributed? Every time our wallet broadcasts a transaction, nodes running Bitcoin Core (the protocol's software implementation) will both verify the signature on the transaction, and then that we do not spend more than what we own. If both conditions are verified these nodes around the network will insert that transaction into a block.

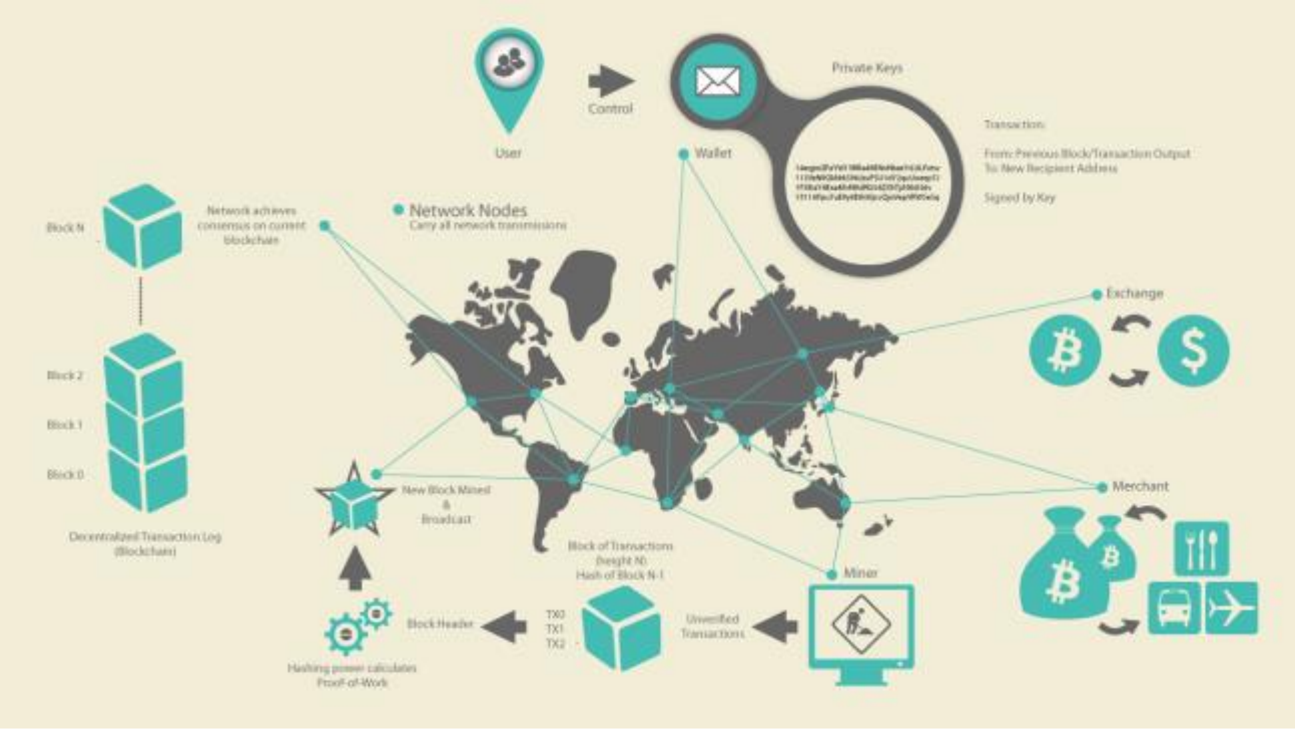

User
Control
Private Keys
Wallet
Transaction:
From: Previous Block/Transaction Output
To: New Recipient Address
Signed by Key
Network Nodes
Carry all network transmissions
Network achieves consensus on current blockchain
Block N
Block 2
Block 1
Block 0
Decentralized Transaction Log (Blockchain)
Exchange
Merchant
New Block Mined & Broadcast
Block of Transactions (height N) Hash of Block N-1
Miner
TX0
TX1
TX2
Block Header
Unverified Transactions
Hashing power calculates Proof-of-Work

Figure 5 Bitcoin's blockchain-building process (Antonopoulos, 2014).

The next step for the nodes around the network, is to take a *set of valid transactions*, aggregate them together, add the *hash of the header of the previous block*, add a random number, the *nonce*, and then hash them altogether. Hashing all of these information together will produce a number. This number is the new *hash of the block's header*. The objective for a node is to keep modifying the *nonce* until the result of the hash is smaller than a threshold. When this happens the node will broadcast this new block to the whole network. If the node was the first one to create such a block, then the entire network will recognize it as valid. This is the **Proof-of-Work**. The node is proving to the entire network that he tried many *nonces*, until he found the correct one.

Nodes will therefore use the *header* of this block in order to build a new block, following the above mentioned procedure. This is how the **blockchain** is built. In particular, each new block is found on average every 10 minutes. In fact, the software Bitcoin Core autonomously decides every two weeks what is the maximum threshold – below which the hash of the block's header should fall – so as to maintain the expected time between two blocks around 10 minutes.

Two things to notice. First every single node places a very special transaction called *coinbase,* among the set of valid transactions. This transaction encumbers a newly minted Output – containing the current *block reward* (today 12.5 XBT, halving every 210,000 blocks, 4 years circa) – to a Bitcoin Address owned by the node building the specific block. This is how Bitcoin's monetary expansion happens block after block. The second most important effect is that this new block updates the set of all Unspent Transaction Outputs, the *UTXO set*. Basically the *UTXO set* can be thought of as the photograph of the entire Bitcoin current status. With the entire blockchain, one can demonstrate that the current state of the *UTXO set* is correct[6]. In fact, each new block validates the delta between the previous state and the new state. Summing all of the deltas you can get to the current state, mathematically demonstrating that no one has ever cheated.

Now we need to ask the question that geniuses and children alike keep wondering, why? Why do Bitcoin nodes (miners) have to waste energy into this wasteful Proof of Work process? The answer is as simple as vital to understand the most centralissue of distributed networks: coordination. The most complex thing to achieve in a distributed environment is coordination. Especially when we are referring to dynamic distributed networks, such as Bitcoin. In fact, we have said that everyone can join this network at any time.

So, how does a new node understand whether the current resource distribution[7] it is presented with, is correct or not? It will have to compare the blockchain – which keeps track of all state changes, basically the evolution of the above mentioned UTXO set – with the current state of the system it is presented with. If the blockchain validates the current UTXO then it is to be considered correct.

What happens though if the node is also presented with a fake proof, a fake blockchain, one that would validate the current state, or even multiple contrasting blockchains? In this case the node will choose the one with the most proof of work.

Why is the proof of work a good conflict resolution method? This is because it cannot be faked anyhow – so long as the SHA256 hashing algorithm is not broken and its anti-collision property is verified – and if you are presented with a valid (verifiable through the hashing

[6] Notice that the UTXO set represent the *current state of the system*. This is a crucial step for a general understanding of what a blockchain-based network is.

[7] The researcher here urges the reader to start to realize that the UTXO set we were talking about before, is actually a sort of matrix keeping track of who owns what. This is effectively a data structure which keeps track of the distribution of resources among the agents participating in this network, and the resources specifically are bitcoins.

algorithm) PoW, then you can demonstrate that someone have spent resources, hence time actually passed. A node needs to have a reliable tamper-proof source of time keeping, which cannot be a simple shared clock, because a distributed network is asynchronous in the real world and it is easy to configure attacks where one node cheats on the time to the others. The network would never come to an agreement over what time is in this moment because of the lags and failures in the connections between the nodes.

> ***Key Takeaway 1.*** Bitcoin's Blockchain is an append-only log[8] that holds a history of modifications made to *state of the system*, which is represented by the UTXO set database. This history runs between the inception of Bitcoin in $\mathbf{t_0}$ and now, **T**.
>
> ***Key Takeaway 2.*** A blockchain is a peculiar data structure whose purpose is that of coordinating nodes over what is to be considered the current state of the system.
>
> ***Key Takeaway 3.*** If a node wants to change any transaction[9] at some point **t** of this history, it will have to prove to the rest of the network that it spent at least the same amount of work as committed on the current blockchain between **t** and **T**.

4.1.2 Abstracting & Modeling, a more rigorous blockchain definition

What we want to try to do now, is abstracting the system that we just explained, so as to identify the most important characterizing elements. First of all, we want to isolate all of the different actors operating in this system and explain their role. Secondly we need to identify the basic requirements of, and the embedded properties in a blockchain-based system.

In order to model our system, we based our analysis on several researches made by the BitFury Group (Garzik, 2015a, Garzik, 2015b, 2016b, 2016e), a Bitcoin mining company. They have arguably developed the most rigorous researches, when it comes to abstracting the inner workings of Bitcoin. This effort is very important because it will let us understand when a blockchain-based distributed infrastructure makes sense, and when it does not.

[8] Physically, a log is a file listing changes to the database, stored in a stable storage format (En.wikipedia.org, 2016).

[9] Watch out, a transaction in computer science is not exactly what an economist may have in mind. A bitcoin transaction is indeed a financial transaction, because value is being moved, most likely in exchange for something in the physical world. Nonetheless, even before than being a financial transaction, a Bitcoin transaction is a modification to the state of the system. The state of the system being who owns what, the UTXO set.

4.1.2.1 Actors

Based on the latest research run by the BitFury Group (2016e), “On Blockchain Auditability”, we can distinguish mainly three roles in a blockchain system:

Clients: these are the entities that initiate and authorize (directly or with the help of trusted intermediaries) changes to the system state. These changes are going to be called transactions[10]. They may be called transaction initiators, or simply *initiators* in other researches. We will call them clients throughout the thesis.

Auditors: these are the entities entitled to verify the transactions initiated by the clients. Moreover, they are also able to verify the whole history of transactions made in the system since its inception. For this reason, they are sometimes called *validators*.

Maintainers: finally – after the transactions initiated by the clients have been validated by one or more auditors – these entities write the data onto the blockchain, thus effectively updating the state of the system. The maintainers are also called *notaries* – or *miners* in Bitcoin – because they notarize the transactions. They build the blockchain.

Notice that this distinction of roles, will be quite useful, both for grasping how a blockchain works, but especially for understanding how the various blockchains differentiate between one another.

4.1.2.2 Basic requirements of a blockchain

Again, BitFury (2016e) went on with the description of what purposes are served by a blockchain. The first key takeaway we got from section “4.1.1 Bitcoin”, was that Bitcoin’s

[10] Note again that here we are using the computer science meaning of transaction, as explained in the previous footnote. In fact, we mean any set of operations that modify the state of the system, which in Bitcoin is partly equivalent to a financial transaction.

blockchain is a list of changes to the state of the system, an auditable log, which is replicated on several nodes on the network. Let's now try to highlight the main requirements of such log:

1) **Log Consistency**: all changes in the system are valid according to consistency rules. Therefore, *maintainers* shall always update the blockchain with valid transactions.
2) **State Consistency**: all of the history of changes could be replayed, achieving exactly the same state of the system as the current state. This means that if the blockchain *maintainers* and *validators* always abided the rules, the current state should be correct and coherent with the one implied by the history written in the blockchain itself.
3) **Transaction Finality**: correct[11] transactions modifying the state of the system are never removed, modified, or added retroactively into the blockchain. Hence, once a transaction is inserted in a block by a *maintainer* and enacted on the system, it may only be offset by a specular transaction, but the original transaction should never be removed.
4) **Reliable Timestamping**: each block should reliably be timestamped by the *maintainers,* with a degree of accuracy appropriate for the application of the blockchain system. This requirement is critical for the system as a whole. Indeed, a transaction may be valid at some point in time, but this does not imply that it will remain valid forever. Having a universal clock on which the system agrees is therefore crucial, otherwise some *auditors* may authorize a transaction while others will not.
5) **Log Uniqueness**: *clients, auditors* and *maintainers* shall never receive two (or more) different conflicting blockchains or system states. They should be confident on which one is the real history of the system, beyond doubts that they might be dealing with a tampered history of transactions.
6) **Blame Ascription**: it should be possible to be able to identify any actor failing to respect the above mentioned requirements. This produces the right incentives for all parties acting in the system to abide the prescribed rules, otherwise one could behave incorrectly and still go unpunished.
7) **Retrospective Auditability**: it should be possible to check all the requirements 1-6 for any moment in the past. If they have always been fulfilled, then the current state is

[11] Correct transactions are those respecting the consistency rules, which in turn determine whether a transaction initiated by a user was consistent with the current state of the system.

correct beyond any doubt. Moreover, it is critical for *auditors* to be able to check what happened in the past, if they need to investigate on some issues.

At this point we want to understand why we have these seven requirements. Why do we even want this log, the blockchain, to be auditable? For whom should those requirements be beneficial?

First of all, we need to remember that the blockchain was introduced in Bitcoin as a mean to resolve any dispute between its users. Basically the intention was to create a *payment infrastructure* where anyone could start using it at any point in time, and still be sure of how this infrastructure has operated in the past, and be confident of how it will operate in the future. Therefore, these requirements where intended to benefit Bitcoin users, the *clients*, which are also *auditors,* and used to be *maintainers*[12]. In fact, Bitcoin aims to be a Payment System Infrastructure where its users do not need to trust any other entity. Instead, they need to trust that the incentives in the Bitcoin protocol are well specified, and that they produce the intended outcome.

What about non-Bitcoin applications of the blockchain? Would it be useful to design any other blockchain in such a way as to respect those requirements? *Maintainers* are the entities building up the blockchain, while *auditors* are the watchdogs who make sure that the process runs smoothly and without any misbehavior by part of any other actor in the system. Hence we may assert that we do want all the 7 requirements to be fulfilled, so as to have an audit trail that is set on stone and points out to any party not conforming to the rules, especially if this audit trail was to be used for a Financial Market Infrastructure application.

For this reason, we now need to investigate a little more on these ideal requirements. In fact, we will realize that depending on the macro-architecture of a blockchain-based system, not all of the 7 requirements are achievable. This might be a problem, depending on the application of such a system. Nonetheless, we need to keep in mind that this technology is quite young. It has

[12] Bitcoin maintainers are commonly called miners. At the beginning (2009-2011) every user used to run the Bitcoin software implementation on its computer, mining bitcoins with its machine. As the economic incentives increased, some users started investing more capital on their machines, raising the bar and making it unprofitable to mine on home computers. TAYLOR, M. B. Bitcoin and the age of bespoke silicon. Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, 2013. IEEE Press, 16.

been in existence for a little more than 8 years now, and it is quite unpredictable where further scientific developments might head towards.

4.2 Blockchain Taxonomy

Categorizing the blockchains helps in understanding what purposes do the different technologies serve. In order to succeed in such task, we need a dimension along which we can classify them in an intuitive and correct fashion. At the same time, we also need to get to the core of the problem and verify whether or not the 7 requirements are fulfilled or not. Indeed, as we will illustrate in section “4.3 Blockchain Properties”, if the requirements are not fulfilled, then the “innate properties”—pointed at by many analysts, when claiming the disruptive potential of the technology – cannot even exist.

When we say that we need to get to the core of the problem, we mean that high level classifications do not really work, because they are too generalist and imprecise. For instance, Tsai et al. (2016) proposed two different kinds of blockchains. The first one is the Trading Blockchain (TBC) – one such that it would be used only for transactions and settlement – whilst the second kind is the Account Blockchain (ABC) – a blockchain that would hold information such as cash balance stock portfolios and derivatives contracts for its users. Such differentiation indeed exists, nonetheless the two authors do not describe in any way how the TBC or the ABC should work at the consensus level, which is critical to verify whether or not the 7 requirements are fulfilled. This sort of taxonomy that the authors propose is done at a very high level instead, it is rather at the application level. Our concern is that if we proceed in this fashion we fall in the same trap as McArthur Wheeler did, it is important to be conscious of why we would need chains of blocks. This is why we will rather turn to the most well-known differentiation. Let’s say the one where there is the most scientific consensus on.

4.2.1 Public-ness & Permissionless-ness

The most common distinction that can be drawn upon blockchains is made around two dimensions, “public-ness” and “permissionless-ness”. Garzik (2015a) and Swanson (2015) are the two reference authors on this matter, since those were the first accurate descriptions of blockchains’ taxonomy. Without the need to get to a technical level, the major differences among blockchains are detectable at the actors’ level.

Figure 6 Actors of the blockchain.

Public vs. Private Blockchains. The term public refers to what actors are allowed to *read the data written on the blockchain*. Bitcoin has a fully public blockchain for instance, which means that any actor – be it *clients, auditors* or *maintainers* – can read the entire history of transactions. In some private blockchain architectures instead, only *auditors* and *maintainers*, but not *clients*, can read the content of the blockchain. Some even more complex architectures may allow some parties to read a portion of the blockchain, but not all of it. In general, the level of private-ness of a blockchain can vary in a continuum, according to the specifications needed for a certain application. For instance, if a blockchain was to be used for exchanging assets between banks, the various institutions (*clients* here) would not want their competitors to know exactly how many units they hold of a specific security. On the other hand, it would still be necessary that some auditing firm may be granted permission to read all of the transactions executed on the distributed system (Brown, 2016).

Permissionless vs. Permissioned Blockchains. In the blockchain jargon, the permission refers to whom is allowed to be a *maintainer* typically. Nonetheless this permission can be extended to the other roles as well. Bitcoin, as well as Ethereum (Buterin, 2013), or ZCash (Hopwood et al., 2016) are permissionless blockchains, meaning that any person or entity can be a *client, auditor* or *maintainer*, at will. This is why the above mentioned blockchain-based are also public, it would be a nonsense to allow everyone to be any kind of actor, but then restrict who can see what data on the blockchain. Hence permissionless blockchains are deemed to be public[13]. As soon as a blockchain protocol specifies that only a set of predefined subjects with known identities can perform the tasks of *maintainers, auditors,* or *clients*, then such system would be a permissioned blockchain. There are several reasons as for why financial institutions and regulators alike prefer permissioned blockchains over permissionless ones. One for all is that they would not want normal clients (consumers) to be able to play the role of the *maintainers*, the transaction processors, which is today domain of the banks. Moreover, permissioned blockchains can prove to be much more scalable then permissionless ones, as we will see.

4.2.2 Permission & log uniqueness

Up to now we omitted a very relevant fact, a replicated system – such as a blockchain-based one – needs a *consensus algorithm.* Distributed systems have been into existence for quite a while, therefore many different kinds of such algorithms exist. Nonetheless, the one that we need, also needs to be tolerant to a very special kind of faults, Byzantine faults. These type of behaviors are the worst and most unpredictable ones that can happen in a distributed system (Avizienis et al., 2004). While ordinary, non-byzantine faults, can be network faults or hardware faults, byzantine faults can be an active malign behavior of a node that is inconsistent with the prescribed rules (Lamport et al., 1982, Pease et al., 1980). Basically, we need to take into account the possibility that some node may be acting in a completely different way, compared to what we would expect him to do. For this reason several Byzantine-fault tolerant (BFT) consensus algorithms have been developed so far, such as PBFT (Castro and Liskov, 1999), QU (Abd-El-Malek et al., 2005), Zyzzyva (Kotla et al., 2007), RBFT (Aublin et al., 2013), or Aardvark (Clement et al., 2009). Also Bitcoin is Byzantine-fault tolerant, but in a fundamentally different way. While the above mentioned BFT algorithms can tolerate at most 1/3 of all nodes to behave in a byzantine way, for Bitcoin such limit is not defined. In fact, Bitcoin is completely permissionless, anyone can set up

[13] An interesting fact, well out of the scope of this thesis, is that even if the data on the blockchain is public, it may be obfuscated through cryptography. In Bitcoin for instance everyone knows to what public Bitcoin Address are the tokens being exchanged. Nonetheless, in principle those addresses are pseudonyms, and unless they are traced back (through what is called taint analysis) to a real world identity (i.e. asking bitcoin online exchanges for the log of their users), this system should grant user privacy. In ZCash instead the cryptography is much more complex and allows its users to hide both the Address and the amount of tokens protected under that address. The ZCash *auditor*s can check the correctness of the transactions through Zero-Knowledge Proofs. Hence, even if the data is public on a blockchain, it does not mean that it is intelligible by everyone else.

its own node. Instead, in all the BFT algorithms that have been studied up to now, the number and the identity of all nodes is known at the set-up of the system.

A permissioned blockchain-based system fundamentally works like this (Garzik, 2015a):

- There is a fixed number *N* of *maintainers*. Each of them possesses a public/private key pair which identifies them and which they use to digitally sign the blocks.
- Blocks are authorized by the system *maintainers* at a constant rate (e.g. 15 seconds). This time interval should be enough in order to grant enough time to be propagated to the whole network and verified by the remaining *maintainers*. The orders with which maintainers create a block may be random, or sequential, following a predetermined order.
- This cycle is constantly repeated and if a *maintainer* cannot sign a block at a certain time, it will miss the round.

Now there is a fundamental problem with this kind of algorithms, it can be easy to change the blockchain – colluding *maintainers* can present different *validators* with different blockchains. Indeed, the digital signature put by *maintainers* on each block is not a costly operation. If all *maintainers* collude in order to hide some transactions, they can easily remove that transaction from its block, and resign all blocks until the very last one. In Bitcoin instead, each *maintainer* – the miner – needs to demonstrate the *proof of work*, which is an expensive task. The deeper a transaction is into Bitcoin's blockchain, the more time it is required in order to rebuild the whole blockchain, and in the meantime the network advances and keeps building new blocks. Bitcoin makes it economically disadvantageous for its *maintainers* to try and revert history anyhow, thus granting *log uniqueness (*cf. 4.1.2.2 Basic requirements of a blockchain).

Permissioned blockchains instead, do not grant the above mentioned property. Because they cannot grant any immutability whatsoever, it is difficult to call them blockchains. In general, whenever the consensus algorithm does not require its *maintainers* to irreversibly spend some form of energy or resource, we are actually trusting on their *bona fide* not to collude, rather than on a predictable algorithm. Poelstra (2015) wrote a very neat paper on this regard, it may seem cryptic, but its argumentations are extremely clear.

4.2.3 Anchoring

We just argued that permissioned blockchains cannot be really considered blockchains, because they do not fulfill the 5th requirements, log uniqueness. Now this is true if we take standard BFT algorithms per se. Nonetheless, there are ways to make it much more expensive for *maintainers* of a permissioned system to change the history that they build. The basic idea is to commit the history they create block after block to as a large an audience as possible. Recalling from subsection "4.1.1.3 Blockchain and Proof of Work", a block header is very basically the hash digest of the latest block. This represents the digital fingerprint of that specific block. Anyone that possesses the data that generated the hash under examination, is able to recreate exactly the same "fingerprint". The *maintainers* of the permissioned blockchain-based system could for instance send such hash to every individual on earth. If even say 1% of them is storing those hashes, then we could count on around 70 million individuals to be able to check back in the history of the blocks and spot any discrepancy between the history they are presented with, and the one they had been presented up to that moment. This that we just described is called anchoring and more details can be found on many different papers. The best practices are described in the whitepaper "On Blockchain Auditability" (2016e), as well as by Garzik (2015a) and Todd (2016).

Of course, such procedure is pretty expensive and generates a huge amount of spam data on internet. The whole point is to make history public to the interested parties, straight away, while it is being created, so that they cannot change it ex post. Therefore, if we talk about permissioned blockchains for banks, the interested parties could be the auditors and the regulators. Nonetheless, there may be cases where the final users of the blockchain may not trust even these few *verifiers,* because it fears that they may falsify this history through social engineering. Again even this can be solved by publishing the headers on a permissionless blockchain. Basically public permissionless blockchains such as Bitcoin's can be used as a widespread and easily accessible source of truth, by simply committing (i.e. publishing block header hash digests) the state of the system on them. Citing Todd (2016):

> *Bitcoin isn't the only possible way to do this, but regardless it and systems like it all share something in common: [...] we can use them as consensus amplifiers, allowing us to leverage a bit of consensus on one system to efficiently provide consensus for a whole world of other systems.*

Food for thought, future developments. This last passage is very interesting. One way of possible development for the whole blockchain ecosystem may closely resemble the evolution of the Internet. In fact, internet is a layered stack of protocols, at the very basis we find the IP – the Internet Protocol, which routes the data packets, thus enabling internetworking – and above that there is the TCP – the Transmission Control Protocol, which ensures that data packets are delivered without being lost in the network. Now what many experts observe, is that the blockchain ecosystem may follow a similar path. This means that it may be that the *consensus layer* of a large, solid and tested blockchain, becomes the basic layer for new *application layers* built on top of that fundamental level. This is all very interesting, it is going to be matter of discussion in the following years, and most importantly it explains a little bit better what is meant by "the blockchain is today what the internet was in the 90s" or "Bitcoin is the internet of value".

4.3 Blockchain-based Networks Emergent Properties and Peculiarities

Now that we have analyzed what are blockchains and how do they work, it is important to sum up on the properties that emerge from the design of those distributed systems. Beware that these properties do exist if, and only if, the basic requirements of section 4.1.2.2 are fulfilled. Moreover, just as a proviso, the above mentioned requirements come out of a research that have not gone through peer review yet, but they are recognized among experts in the blockchain ecosystem to be necessary, but maybe not sufficient conditions. Finally, these emergent properties do not come from a specific research paper on the topic – which is currently lacking actually – but rather they are extrapolated from all of the resources and articles cited throughout the thesis.

4.3.1 Immutability & durability

From a philosophical perspective immutability is not really something within human reach, everything changes in our lives, so why shouldn't a blockchain[14]. Indeed, it changes constantly, what is immutable is the past, the old blocks. But are they really? Well indeed we argued in multiple passages that actually even Bitcoin's past blocks can be changed. The real problem is how many resources the attacker is willing to sacrifice. If now is T and the attacker wants to revert a

[14] Please notice the use of the word blockchain here, not blockchain. We are effectively talking about blockchain as the data structure of the chain of blocks cryptographically committed to one another. We are not referring to the network coordinated through a chain of blocks. The researcher hopes that making explicit reference to the multifaceted aspects of this technology shall help the reader in having a more precise idea of what it is really that we are talking about.

transaction that he made – or for which he possesses the private key – in *T-t*, then the attacker will need to produce strictly more[15] *proof of work* then the one produced during the time interval *(T+*Δ*)-t*. Notice that is *T*+Δ because some time will pass between the moment in which you start the attack and the moment in which you manage to reproduce a blockchain that is acceptable for the network. We will stop here and do not get into further details on how to compute the amount of energy that needs to be spent in order to perform this kind of attack. Nonetheless, many analysis have been performed, in the original whitepaper (Nakamoto, 2008), but also in other interesting papers such as the one by Poelstra (2015).

So the real point with a permissionless blockchain whose consensus algorithm is based on proof of work, is that it is measurable what level of commitment an attacker should put at stake. If we contrast other algorithms, such as the BFT (discussed in sub-section 4.2.2), or the famous Casper (Ethereum) proof of stake[16], the proof of work needs physical work to be performed. Physiscal work cannot be faked[17], so real – and not virtual – scarce resources need to be provably "wasted" – or even better, sacrificed . Therefore, Bitcoin's mining activity is a form of securing the network. The most power is spent into mining, the more secure Bitcoin is, as well as any other blockchain anchoring to it.

This being said this short explanation of how blockchain immutability is achieved through a costly **alteration process** is just part of the concept of immutability itself. There is another important nuance actually and it is related to the cost required to **maintain the data** on the blockchain. Once one manages to make it costly to alter some piece of data, it still needs to be stored for the eternity.

This is costly, who is going to bear this cost? Every single node participating in the consensus process (full nodes) will have to maintain its own up-to-date copy of the blockchain.

[15] Remember that measuring the proof of work is banal, it is just matter of summing all of the block headers of the blockchain between the *genenesis block* and the latest block received from the network. Then be careful, because the chain with the *most* proof of work, is actually the one with the lowest sum of block headers. Indeed, the real work that is performed by miners, is to try to find a block with a block header whose value is lower then the current threshold, as defined by the protocol for the current retargeting period.

[16] Very basically proof of stake is a form of consensus. The *maintainers* put some of their tokens at stake, they freeze them. The more tokens a *maintainer* puts at stake, the highest the probability that he will create the block. This form of consensus seems attractive, because the network doesn't need to "waste" electric energy. Nonetheless, it suffers from a severe problem currently without solution – the nothing at stake issue. Again, Poelstra (2015) did the best job in formalizing this issue, which makes it unusable as a form of consensus algorithm right now.

[17] If someone manages to do so, then the 2nd law of thermodynamics would be violated. When that will happen, then we will be assisting to one of the most, if not the most, spectacular scientific revolution of our lifetime.

Storing data whose content is *unrelated* to the state of the system being maintained by a blockchain-based network produces an externality on the entire network. Why would such infrastructure (and the actors participating in it) be willing to bear this cost? This is a very hard question, as committing ordinary data onto the blockchain modifies the incentives originally designed for the system. In order to solve for this a specific operator was introduced in Bitcoin, the OP_RETURN code. Such condition is written in the script attached to each transaction. A transaction containing this OP code can be pruned away from the blockchain. This means that a node willing to maintain the state of the system and just the state of the system, will be able to get rid of the data committed through the OP_RETURN, so as to remove the externality cost. All of the nodes willing to maintain also a copy of the commitments of the external data will be able to do so and will have to pay to maintain their own copy of the data. A more detailed analysis of this issue can be found in an article by DeRose (2016).

4.3.2 Security

We need to make two critical points about a permissionless blockchain security model, so as to show how fundamentally different it is when compared to our current FMIs. There are only two ways to illicitly get into possession of someone else's funds on a permissionless virtual currency network, either the attacker manages to retrieve a person's private keys, or he reverts one of his old transactions. These two kind of attacks are respectively called *local* and *system-wide* threats. The former only affects a minor part of the network – the owner of the stolen keys – while the latter affects the system as a whole.

Local attacks. The strongest defense of cryptocurrency networks against *local* attacks, comes from its *decentralized* nature. Every single individual is responsible for the security of its own funds. This makes the life of hackers much more difficult. Instead of having big single entities to be attacked providing large bounties, the hackers are presented with a very widespread (pseudonymous) cohort of different preys. The users on their side, have different security schemes and best practices which are already widespread among medium-advanced users, such as multi-signature wallets, hardware wallets, plausible deniability tactics, or a mix of them all. This is the key difference of the magnitude in the impact of a *local* attack on decentralized blockchain-based-systems, versus distributed, but centralized infrastructures. SWIFT suffered an $81 million heist at the beginning of 2016, and the head of threat intelligence at BAE system commented on this (Finkle, 2016):

> *I can't think of a case where we have seen a criminal go to the level of effort to customize it for the environment they were operating in, I guess it was the realization that the potential payoff made that effort worthwhile.*

With this perspective, bitcoin exchange markets for instance, play exactly the same role of big single points of failure of traditional systems. Mt.Gox, or Bitfinex, got attacked (Hayase, 2016) because they are easily identifiable, they offer a big bounty, and hackers can make their way through their security practices thanks to social engineering. On the other hand, very few individuals on earth are really ready and fully capable of taking full care of their funds. This is especially true in the moment they are exposed to cyber-threats, which can require very advanced and specialized skills in order to be handled.

System-wide attacks. This is the second typology of attack that a blockchain-based system may suffer. We have already analyzed it, when we talked about immutability in the previous sub-section. The most interesting fact that we want to stress here is the difference between permissionless systems, and permissioned ones together with traditional FMIs. In fact, permissionless based systems provide a *secret-free* security model against system-wide attacks, which consist in changing the history of the changes in state of the system. Permissioned blockchain-based systems along with traditional FMIs instead, can be attacked at system level if the hacker manages to access the relevant private keys of system gateways (the *maintainers*).

4.3.3 Transaction Programmability, or Smart Contracts

The property that makes blockchain-based systems attractive to Financial Institutions, is the fact that the transaction logic is embedded into the protocol implementation (2015). This means that what transactions shall be accepted, because valid, and what shall not, is defined at the software level. Hence the *auditing* and *maintaining* nodes participating to the system can automatically approve and refuse transactions, making the clearing and settlement operations smoother. A transaction that abides the predefined rules is automatically cleared by *auditors* and enforced by *maintainers*. All of this without the risk of fraudulent transactions being approved, thanks of the architecture(s) of blockchain-based systems that we described up to now.

This is a baby step towards what Nick Szabo (1997) called *smart contracts*. These were envisioned as cryptographic contracts whose verification of contractual obligations were to be executed and enforced by computer code. This is exactly what happens in Bitcoin (which we described in sub-section 4.1.1.2), as well as in other public blockchain-based systems such as Ethereum (Buterin, 2013). The difference lies in the level and complexity of clauses that can be expressed, verified and enforced by the network. For instance, as of now in Bitcoin examples of smart contracts include (2015) escrow accounts, multi-signature addresses whereas the keys can be distributed to multiple parties to authorize any transaction, micro-payments channels (which also represent a scaling solution), or temporized transactions. Ethereum in turn, allows to express much more complicated contractual clauses, thanks to the fact that transactions can even have loops expressed within their scripts and they can even refer to other transactions (smart contracts). Nonetheless, this level of complexity brought into the network is proving to be difficult to manage (Hertig, 2016), since it brings about new forces that unbalance the strength of the consensus, both at a software level and at a social level.

The most complex feature of a “real” smart contract, is its self-eforcing property, or better the automatized enforcement enacted through the consensus, by the *maintainers* of the network. This is based on the assumption that the outcomes of the code, was exactly what was intended by the parties. Therefore, there should be no recourse to dispute resolution through institutional legal systems, nor any reference to normal legacy legal prose contract terms.

The famous bank’s consortium led by the company R3 is chasing this phantom of smart contracts with its Distributed Ledger, Corda (Brown, 2016). In order to achieve this goal, right now they are trying to build the full stack protocol, from the consensus layer, passing through the networking layer and up to the transaction logic and applications. A similar effort is being brought forward by Digital Asset Holdings (2016a), under the lead of Blythe Masters. Right now is too early to judge them. Indeed, all of the available information is mainly produced for marketing purposes, rather than scientific advancements. The only thing that we are certain of concerning those protocols design, is that up to now they do not provide any form of anchoring of their ledgers. As we discussed earlier, it is critical for permissioned systems to be anchored to a widespread medium of communication, possibly retrospectively immutable, having otherwise a severe flaw regarding the reliability of the entire system.

Ultimately, we could really see in the long term some forms of self-executed and enforced contracts, where the parties do not need to trust any intermediary. On the contrary they will only need to ensure that the code fully expresses both parties intentions, and they shall have peace of mind, based on the fact that they ultimately trust that the protocol is flawless. Of course this vision of the future seems both a little bit utopic as of now, and lies very much ahead in time.

4.3.4 Auditability

Auditing is to prove that a system is meeting some predefined rules. As we saw in the three previous sub-sections, a blockchain shall provide all of the necessary tools that allow for a complete audit of the state of the system. The main difference between permissionless and permissioned blockchains, is about which actors are entitled to perform such audits. So that permissionless blockchains represent sort of a neutral democratic system, where no single specific entity needs to be trusted upon.

Now this problem of no trust in third parties raises another question, who is a blockchain-based system trusting ultimately?

4.3.5 Trustlessness, or Disintermediation of Third Parties

This one, as well as the next property, are distinctive of permissionless blockchains.

In most of the private companies reports that can be found in the bibliography, the various authors state that the permissionless blockchain are trustless. Does this mean that the users of such systems do not necessitate to trust anyone outside of themselves? Trust, or a shared belief, is one of the most common ways for solving the epistemological paradox of the skeptic regression. In fact, when trying to prove that a conclusion is correct, a justification is required. Nonetheless, also the justification itself needs to be sustained with another argumentation. This process can form an endless chain of justifications, or it could form a circular path, a logical fallacy. Another possible solution is to provide a shared belief as a justification, an ultimate argumentation that we *trust* to be correct. Whilst the ultimate source of trust in our current FMIs are the intermediaries, the banks that we trust upon, in permissionless systems this trust is spread across a cohort of *maintainers*, but ultimately we can be sure they will abide to the rules if we assume that the game theoretic incentives are enough to ensure that they will rationally follow the protocol.

Therefore, it is not properly true that permissionless systems are *trustless*, in the end trust in a well-designed protocol is required, together with the basic assumption of participants' rationality. Nonetheless, agents rationality is also required in current FMI, in fact we trust that they will abide to the institutional rules so as not to risk their reputation, nor to incur in fines. The actual difference concerns the level of concentration of this trust, in one case it is focused on few parties, while on the other it is spread among several dispersed competing parties.

4.3.6 Borderless & Censorship Resistance

Finally, the core value proposition of permissionless blockchain-based systems is that they should be censorship proof. This basically descends from all of the previous properties. In fact, for any *maintainer* that tries to censor some specific transaction, there will be many others that do not have any interest in blocking any specific funds movement, wherever and whenever. This characteristic makes permissionless systems very neutral, and this must be the reason why the "crypto-valley" is flourishing in Zug, Switzerland (Castillo, 2016a). This is exactly why many times Bitcoin is associated with illicit activities, drugs, terrorism, or simply tax evasion and a means for capital flee. Now even though from an ethical point of view this characteristic may seem immoral, and even dangerous in our times, we could also take into consideration that these criminals have effectively used the traditional channels for their purposes, with HSBC and Standard Chartered and others, involved in money laundering scandals in 2012 (Whitehead, 2016). Moreover, a censorship resistant form of money together with privacy tools such as TOR, may prove very helpful for citizens living under a regime.

4.4 Conclusions on Technology

At the beginning of this first literature review chapter the focus was on:

- understanding how the technology works and
- identifying what the limits of each type of blockchain-based network are.

What we understood by the end of this analysis is that permissioned and permissionless technologies are so different that it does not even make sense to compare them. Their purposes are really different because the assumptions on top of which they are built are opposed and often incompatible with one another:

- the basic purpose of permissionless networks is that of forgoing any trusted third party in any form.
- Permissioned networks instead are built on the assumption that it is desirable to have a trusted third party instead, which is exactly the same as today. The third parties operating such systems are only obscured by complexity.

Based on such assumptions it makes little sense – if it does at all – to compare the two technologies over the same applications, just like it is completely wrong and misleading to either talk about trustless exchanges, or trustless smart contract execution and enforcement, or about immutability.

Another relevant unsolved issue concerns the need for a chain of blocks in permissioned environments. What we want to argue is that a permissioned distributed network does not need a chain of blocks in order to work, if both the *auditors* and the *maintainers* are static and pre-specified. In permissionless systems we see the use of blocks of authorized transactions each committed to the previous one, producing a chain of cryptographic proofs of the history. This system for tracking the history of state transitions is crucial in a dynamic system, one where any new user needs to check whether the current state he is presented with is reliable. Nevertheless, in a system where both *maintainers*, as well as regulators and authorities playing the role of external *auditors*, are fixed and known since the setup, then they don't need a proof of the history, because they can directly see it and store it on the making. Blocks may still be useful, because it is a way of efficiently batching groups of transactions to be authorized, but then chaining them together – as a way to prove their history – is irrelevant in an environment where those blocks are constantly forwarded to the relevant external auditors. As overly meticulous this observation may seem, we deem it necessary to for anyone observing the blockchain ecosystem to ask the right questions, one and for all, why do we need chains of blocks?

Our focus now is to understand if a blockchain can serve as a financial market infrastructure. In order to do so, should we consider both permissioned and permissionless blockchains or should we focus on one of the two? The logical conclusion is this one, permissionless value networks without trusted third parties are something that humanity has never seen throughout its history. This means economical exchanges without the need for an authority to enforce them, or an institution to appeal to. Permissioned blockchains instead, are networks which eventually rely on a trusted third party, or a set of trusted third parties, which is not very much different from what we have today. These closed trusted blockchains are definitely a whole new concept which is being introduced within the spectrum of Financial Market Infrastructures. Nonetheless, they are a subset of the existing paradigms.

Permissionless blockchains instead, truly represent a paradigm shift, because they are a whole new type of network which is trying to be used as a FMI. Nonetheless, they are being discarded because they pose multiple risks. Our goal is now, is to investigate with further literature review what Financial Market Infrastructures could be implemented with these distributed networks and what risks emerge when using these networks to transfer value.

5 Blockchain Networks and Financial Market Infrastructures

Now that we have an understanding of how the blockchains work, we can better understand whether or not they can be used in order to set up a Financial Market Infrastructure (FMI). In order to do so, we divide the chapter into three logically consequent parts:

1. Financial Market Infrastructures. At first we define what a FMI is and what different typologies do exist, then we examine the key risks faced by these actors.
2. Permissionless blockchains. Here we examine the main issues stemming out of the peculiar structure of permissionless blockchains, with a particular focus on the risks associated with being unregulated systems lacking any form of structured governance. These limitations do not make permissionless blockchains viable FMI any time soon.
3. Permissioned blockchains. Finally, we analyze how these systems – developed by private companies – can be given a well-defined legal basis and governance & risk-governance structure, so as to allow them to be FMIs authorized by the relevant regulators. Ultimately, we examine how the technology is forecasted to be applied to the existing processes, and what impact it may have from a cost perspective.

Up to now we talked about these complex distributed systems, with their idiosyncratic characteristics, with Bitcoin in particular which allows their users to effectively update the state of the global system, so as to transfer the property of a valuable token, with transaction finality within 1 hour circa[18]. Bitcoin, Litecoin, Dodgecoin, or even Ethereum users, effectively manage

[18] Recall from section 4.1.1.3. We talked about the Proof of Work consensus protocol. The peculiarity of such protocol is that it poses a hard problem to the network and it needs to be solved in a limited amount of time since every other node is trying to solve it. As soon as one miner finds a solution, it propagates the block to the network and, if valid, a new problem is built on top of this new block. This means that a malicious miner that wants to either *censor* or *double spend* a specific transaction, needs to find a solution for its own fraudulent block before the rest of the network does and broadcast it to the network. If it doesn't manage to do so it will have to find a solution for the first block and then to another one on top of it, and so on and so forth until the malicious node manages to keep up with the network. The problem here is that a single actor is trying to fraud the rest of the system for its own benefit, while the entire network is incentivized to follow the protocol rules. The only way for a malicious miner to succeed (in the long run, not just temporarily) in this plan is being faster than the rest of the network, which can be achieved only with superior computing power (measured in hashes per unit time, the hashing power), i.e. a 51% attack. In any other case where the attacker controls less than 50% of the hashing power, any fraud is "detected" (more correctly *resolved*, as explained in section 4.1.1.1 when talking about eventual consistency) with large confidence (>99.9%) within 6 blocks from the first confirmation of our transaction. This is mathematically proven by Nakamoto in the last section of the white paper.

to transfer valuable items from one person to another. Is this enough to make those permissionless blockchains a Financial Market Infrastructure? There is a famous quotation of a riddle posed by Abraham Lincoln to a group of religious leaders who wanted him to sign an Emancipation Proclamation for slaves (Quoteinvestigator, 2015):

> *"If I should call a sheep's tail a leg, how many legs would it have?"*
>
> *"Five." 'No, only four; for my calling the tail a leg would not make it so." "Now, gentlemen, if I say to the slaves, 'you are free,' they will be no more free than at present."*

Now with the same spirit we want to try and understand what is it that makes up a Financial Market Infrastructure.

5.1 Financial Market Infrastructures

In order to run this analysis, we mainly drew from the Bank for International Settlements' (BIS) Principles for Financial Market Infrastructures (Russo and Mooney, 2012). Consistently with the standards set by the G20 and Financial Stability Board (Ferrarini and Saguato, 2014), BIS defines "financial market infrastructures" (or FMIs) as "**multilateral systems** among participating financial institutions, including the system operator**, used for the purposes of clearing, settling, or recording payments, securities, derivatives, or other financial transactions**," that "include payment systems, central securities depositories, securities settlement systems, central counterparties, and trade repositories" (Russo and Mooney, 2012).

The standards proposed by BIS aim to harmonize and strengthen the existing international standards for:

- **payment systems** (PS) that are systemically important, and transfers funds between or among its participants. A payment system is generally categorized as either a retail payment system or a large-value payment system (LVPS). Retail payment systems may be operated either by the private sector or the public sector, using a multilateral deferred net settlement (DNS) or a real-time gross settlement (RTGS) mechanism, while LVSP are normally operated by central banks using RTGS mechanisms;

- **central securities depositories** (CSDs), which provides securities accounts, central safekeeping services, and asset services, which may include the administration of corporate actions and redemptions;
- **securities settlement systems** (SSSs), that allow to transfer and settle securities. Such systems allow transfers of securities either free of payment or against payment (Delivery versus Payment, DvP);
- **central counterparties** (CCPs), an entity which interposes itself between counterparties to contracts traded in one or more financial markets, becoming the buyer to every seller and the seller to every buyer and thereby ensuring the performance of open contracts;
- over-the-counter (OTC) derivatives CCPs and **trade repositories** (TRs), an entity that maintains a centralised electronic record (database) of transaction data. TRs have emerged as a new type of FMI and have recently grown in importance, particularly in the OTC derivatives market.

The reasons why such regulation is much needed – especially after the events that took place during 2008 global financial crisis – are, and we cite Russo and Mooney:

> *While safe and efficient FMIs contribute to maintaining and promoting financial stability and economic growth, FMIs also concentrate risk. If not properly managed, FMIs can be sources of financial shocks, such as liquidity dislocations and credit losses, or a major channel through which these shocks are transmitted across domestic and international financial markets.*

For this reason, the principles are intended to be applied to all systemically important FMI that perform any of the above mentioned functions. In fact, such infrastructures could potentially concentrate and transmit risks to other FMIs with whom they are interconnected.

The BIS research also identifies the key risks faced by such FMIs:

Systemic risk. Efficient FMIs are supposed to mitigate systemic risk. Nonetheless, they may directly be affected by disrupting defaults of institutions participating in their infrastructure. Such events would have adverse effects on their normal operation flow, making it necessary to reverse payments or deliveries; delay the settlement of guaranteed transactions; or forcing them to immediately liquidate collateral, and assets at fire sale prices. Moreover, the interconnectedness of systemically important FMIs can transmit such disruptions to other participants of the FMI, thus quickly spreading the contagion.

Legal risk. The unexpected enforcement of a law resulting in a loss, may also represent a source of risk, especially in cross-border transactions. Examples of such risk include the application of a regulation that renders a contract unenforceable, or a freezing of position as a consequence of a legal procedure.

Credit risk. This arises when a counterparty becomes unable to fully meet its financial obligations in the future. Other than caused by the failure of a market participant, BIS also identifies two other forms of credit risk, replacement-cost risk – when a party faces a loss on unrealized gains on unsettled transactions, thus becoming exposed to the cost of replacing that transaction at current market prices – and principal risk – when a seller of a financial asset irrevocably delivers its asset, but the payment leg of the transaction is unsettled.

Liquidity risk. It arises when a counterparty has insufficient funds to meet its financial obligations when due (credit risk involves a permanent inability of the entity to fulfill its obligations).

General business risk. This is the general risks associated with running an enterprise, a FMI in this case.

Custody and investment risk. Custody risk consists in the loss of the assets held in custody, due to insolvency, negligence or fraud by part of the custodian. Investment risk instead, arises when an FMI invests and loses a participant's resources.

Operational risk. Such risks are a very important class of risks for us. They stem from inadequacies in information systems, internal processes, human errors, or disruptions from external events. The results of such failures can be the reduction, or discontinuation of the services provided by the FMI. Operational failures may also impair the ability of the affected FMI to complete settlement, or to monitor and manage its credit exposures.

Ultimately the BIS analyzes the principles for FMIs, and states (Russo and Mooney, 2012):

> *The **foundation of an FMI's risk-management framework** includes its authority, structure, rights, and responsibilities. The […] principles provide guidance on:*
>
> *(a) the legal basis for the FMI's activities,*

(b) the governance structure of the FMI, and

(c) the framework for the comprehensive management of risks, to help establish a strong foundation for the risk management of an FMI.

Now that we outlined the main elements of FMIs and the risks to which they are exposed, we try to analyze how blockchain-based networks fit into such framework. In order to do so, we will proceed as follows:

- as always we draw a first layer of distinction over permissionless and permissioned blockchain-based networks. They are two incompatible environments, based on different technologies and ultimately they give rise to different considerations over the question whether such networks can or cannot be used as Financial Market Infrastructures;
- both for permissioned and permissionless networks we will shortly exemplify how those networks can be used, extended or designed in order to implement them as a specific FMIs;
- finally – both for permissioned and permissionless systems – based on the literature review and on a critical analysis, we will highlight the most relevant issues of the systems being proposed.

This last step will turn to be useful for our analysis, as our interviews to the selected sample, will mostly be based on the results emerging from this very last analysis.

5.2 Permissionless Blockchains

The US Federal Reserve's Policy on Payment System Risks (2016d) applies to "payment systems that expect to settle a daily aggregate gross value of U.S. dollar-denominated transactions exceeding $5 billion on any day during the next 12 months". Bitcoin is the most widespread cryptocurrency with its $14 billion (circa) market cap (Acheson, 2017) and still it is not close to the $5 billion threshold of the FED. In fact, making some very rough estimation based on blockchain.info data (Blockchain.info, 2017), 300,000 bitcoins are transacted every day, using an upper-bound bitcoin price of $1000 we get to $300 million transacted daily. Notice that both the price and the number of bitcoin transacted per day are way lower than those figures (statoshi.com for instance, visualizes the data on the blockchain and quotes bitcoin daily transactions to be 3 orders of magnitude lower than the ones quoted by blockchain.info), and still the daily volumes are one order of magnitude lower than FED's threshold.

It is already quite obvious that no public permissionless blockchain can qualify as a Financial Market Infrastructure nowadays, not only because their market capitalizations and daily volumes transacted are drops in a sea, but especially because of their nearly non-existing governance structures. Even if one day one of these cryptocurrency networks becomes large enough to be considered systemically important FMIs, which type of FMI would they be? A global *payment system* for settling peer to peer electronic cash, as described by Nakamoto in its original paper, or would it be a *de-centralized securities depository* with an embedded *security settlement system* for what many call the digital gold, a synthetic commodity (Selgin, 2015)? The debate on whether bitcoin is money or something more akin to a digital commodity, a new asset class, is still open, only time will give us an answer. On the other hand, we can already check on the risks that this kind of FMI may face in the future.

Of all the key risks identified by the BIS, between systemic, legal, liquidity, general business, custody and investment, and operational risks, the latter is probably the one where the biggest issues lie. Walch (2015) did a remarkable job in outlining the most relevant operational issues faced by public permissionless blockchains such as Bitcoin's. She also agrees that these virtual infrastructures could have a potential as a FMI at some point in the future, for this reason she highlights some open points that remain unsolved with permissionless blockchains.

In particular, she points at the following criticalities for permissionless blockchains:

Blockchain protocol software implementation. We already explained at the beginning of sub-section 4.1.1 how the protocol that allows these blockchain-based networks needs to be implemented into a piece of software. This arises at least four different problems:

1. *Bugs*. It may seem basic, but it is normal just like for any other software, there are bugs that need to be solved. Now bugs may simply impact on the seamlessness of users experience, or worse there may be bugs that affect the network as a whole, possibly disrupting its operations.
2. *Disruptive updates.* For the above mentioned reasons Bitcoin Core developers, voluntarily but relentlessly look out for bugs, trying to solve them with a new software release. Updates on a FMI software are extremely critical though, even in a little blockchain such as Bitcoins, $15+ billion are at stake. For this reason Timon and Drake (two Bitcoin Core developers) are currently at work, trying to separate the various layers of Bitcoin Core. The objective

is to isolate the *consensus layer*. This is the most delicate part of the software, the one that if touched inappropriately may lead to severe losses and global system failures.

3. *Vulnerability to attacks.* Walch correctly differentiates between Bitcoin's vulnerabilities, versus the vulnerability of external companies operating in this space, such as currency exchanges or payment processors. In fact, an attack to those entities does not pose a menace to the network as a whole. Nonetheless, a permissionless blockchain is exposed to two main kind of attacks, threats that exploit glitches or bugs of the protocol software implementation, and threats that exploit a fundamental issue at the protocol level, such as the infamous 51% attack.
4. *Users full consciousness of the risks when using these software.* Technologies can be hard to understand and many Bitcoin users are highly specialized computer scientists. For the rest of the global population, Bitcoin or ZCash even worse, are not so easy to use. The security practices that would need to be put in place by users are out of the reach of an average Joe.

Open-source software. Operating an open-source software is a double-edged sword. On one hand it grants huge indiscriminate accessibility to anyone in the world, fostering open-innovation and participation by a potentially infinite pool of brilliant minds. On the other hand nobody pays Bitcoin Core developers, their only incentive is both the idealism behind permissionless blockchain, joint with their best interest in keeping the network up and running so as to allow a wider adoption and higher value of their own tokens.

Decentralized structure. Finally, also the decentralized nature of blockchains is a double-edged blade. On one hand it provides resilience to the entire network, but on the other hand there is no one truly and legally responsible for a permissionless blockchain continued operations. At a governance level, well there is no true governance structure, besides from Bitcoin Core project maintainers. Right now they are Van der Laan, Schnelli and Falke, they are basically the ones in charge of accepting and reviewing the software updates. Moreover, they are also the ones holding the keys that allow to send a critical warning messages to all nodes on the network, but again, they do not get paid for this job, neither they are legally responsible.

5.2.1 Payment system

Bitcoin was ideated since the beginning as a pure peer to peer electronic cash system, which means a payment system, following a more traditional naming system. We have already seen that Bitcoin's users operating on this system are covered by pseudo-anonimity, which means that their addresses are not necessarily linked to a real world identity – though they can be – this is why this payment method is akin to what in the physical world to cash.

Nonetheless, the view of what bitcoin is in 2017 are widely different. So different from its inception, that the different visions led to a fracture within Bitcoin's community. As a matter of fact, Bitcoin is not recognized as money by most governments and they treat it as a (synthetic) commodity, just like gold, while others treat it as a currency. Some Bitcoin users keep considering bitcoin as an inclusive, low cost form of money which can be *cheaply* transferred anywhere around the world – provided that the sender has an internet connection. Other users are attracted by bitcoin because they consider it a store of value, synthetic gold, hence they tend to move large sums of bitcoins and their demand to use Bitcoin's network is inelastic to transaction fees.

Using the same global system for these two very different types of transactions raises questions over what kind of payment system Bitcoin really is. Indeed, it is not written anywhere whether Bitcoin shall be a Large Value Payment System (LVPS), or something more akin to a Retail Payment System, it is left largely up to users whether they want to use it in one way or another, but this issue is what ultimately is causing most of the frictions regarding the debate over how to scale Bitcoin.

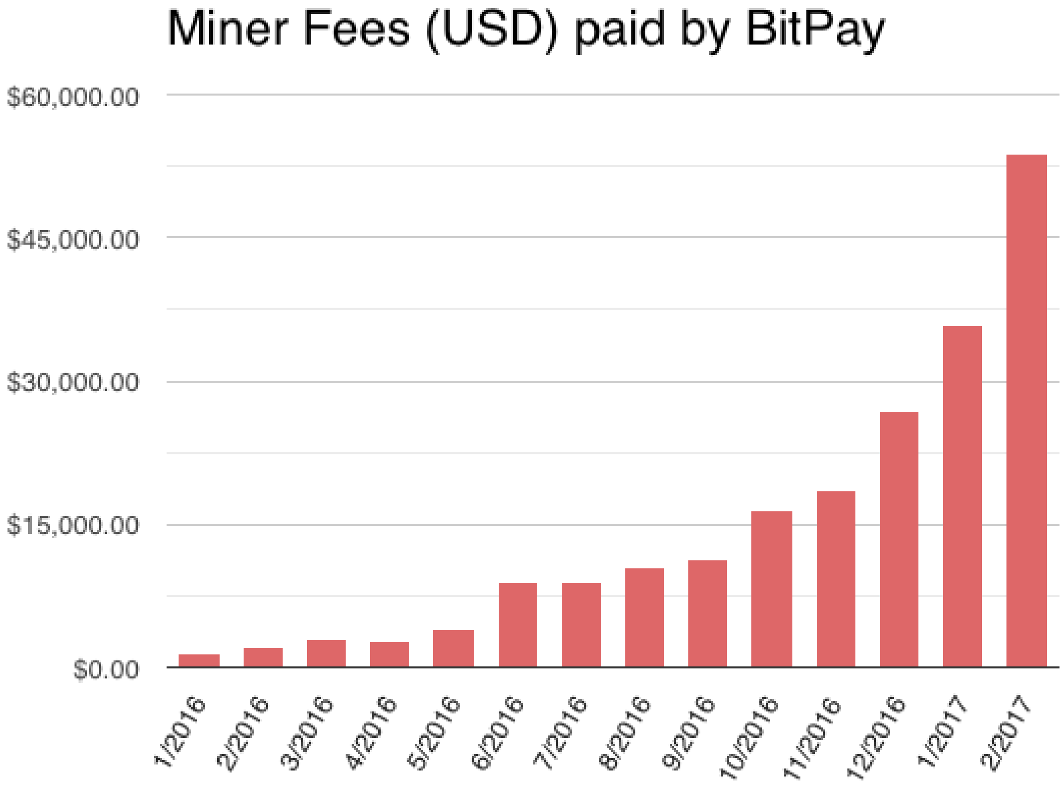


Figure 7 Bitcoin transaction fees paid by Bitpay (Pair, 2017)

What is currently happening within Bitcoin's ecosystem is that we are assisting to a sort of political debate. Early users want transaction fees as low as possible so as to facilitate widespread adoption. On the other hand, speculators and synthetic-gold users do not care about 0.5$ fees or eve 10$ fees, because it would still be much cheaper and convenient than moving physical gold across the world.

Why is it getting so costly to move bitcoins from one address to another? And how does this relate to the thesis? We have already seen how complex is to reach consensus over a distributed network, the network maintainers, the miners, shall keep updating the state of the system, or how resources are distributed. But as the network grows more and more clients (bitcoin users) demand transaction services to the system maintainers and the number of transactions that can be recorded on the blockchain per unit time is limited. This is why sending bitcoins is getting more and more expensive by the day. This mechanism is very relevant for our thesis, because the demand for low cost transactions persists. This demand is spurring the development of a whole lot of new systems that help to satisfy this need. A new infrastructural layer is emerging, that of the Retail Payment System targeted to high velocity and low value transfers.

Second Layer Infrastructure as Retail Payment System. The Lightning Network was proposed in late 2015 (Poon and Dryja, 2015) as a second layer scaling solution. Very briefly, what this means is that clients, instead of operating all of their transactions directly on-chain, they will make most of their transactions off-chain and they will finally settle the net of all transactions through an on-chain transaction. How does that work?

Payment Channel. As we said Bitcoin contains an advanced scripting system which dictates the conditions for which transactions are considered valid, thus allowing to program transactions. Lightning Network is based on a multi-party smart contract. The two parties involved in recurring economical relationship, will deposit part of their funds on a shared Bitcoin address, which is called a payment channel. In order to spend money from this address both parties will have to agree over the latest state of the balance. The latest balance is stored on the second layer – the lightning network layer – as the most recent transaction signed by both parties which spends from the shared Bitcoin address. When one party wants to pay the other, both the sender and the recipient will sign a transaction updating the balance of their shared address, thus moving funds from one party to the other. Be aware that both parties will be willing to do so, because the sender needs to sign the transaction in order to receive the good/service s/he is paying for, and the receiver will sign it in order to actually receive the money. Moreover, these transactions will never be recorded on the blockchain unless one of the two parties will decide to withdraw its own funds from the shared address, the payment channel. Finally, in order to make it trustless and ensure that no party can possibly misbehave by trying to close the balance of the payment channel with a favorable (to him/her) older balance state, a cryptographic proof is committed to older transactions. This means that the injured party will be able to broadcast a "punishment transactions" which will automatically refund to him/her the entire balance of the payment channel, so as to disincentivize any fraudulent behavior.

Lightning Network. The payment channel is the basic building block of the lightning network. In fact, this second layer retail payment system is designed as a network of payment channels. Each individual will open multiple channels with multiple counterparties with which s/he conduces frequent low value transactions. This way s/he will have many open balances. One can imagine this network as a graph. The nodes are the people, while the arcs are the payment channels. It is easy to realize that each and every node is – at least indirectly – interconnected to every other node through one or many hops over the various payment channels separating them.

The goal here is to allow money to be sent over the various payment channels without being stolen by the nodes separating the two parties involved in the real-world transaction. Theoretically this is easy, if one node in the middle decreases its balance in one of its payment channels by x and at the same time increases by x its balance over another of its channels, the money was effectively transferred. The tricky part is doing so in a way such that the intermediary will not be able to get possession of the funds by, for instance, enhancing by x one payment channel, but failing to decrease by x the next payment channel. Fortunately, this is pretty easy, the receiver will generate a secret which s/he is the only one who knows it and thells a hash of this secret to the payment sender. The sender will thus sign a transaction to be passed over the network of payment channels which conditions tells: this payment is unlockable by whom possesses the secret whose hash is exactly the one incorporated in this transaction.

This way Poon and Dryja – careful that a rudimental version of this system was already envisioned by Nakamoto back in 2010 – managed to create a Retail Payment System where money is transferred instantaneously – at the same speed that it takes for the information to move along the second layer lightning network nodes – without needing to be constantly settled on the Bitcoin blockchain, but mst importantly again without a central counterparty taking care that no client operates maliciously.

Lightning Network represents a decentralized instantaneous retail payment system. It is not implemented yet in production, as it is still in its beta phase, but it is ready to be tested on Bitcoin’s main-net as soon as the transaction malleability bug is fixed. This is very relevant because this design for a payment network goes beyond any specific blockchain implementation. This layer can be made interoperable with any other blockchain-based network – one where the state of the system being represented concerns some sort of scarce resource, and one such that payment channels can be replicated – which means that it represents a very widespread interoperable retail payment system, which crosses the barriers of a specific network. This is the kind of paradigm shift allowed by permissionless networks, an unprecedented kind of innovation that was not possible in the legacy world of financial market infrastructure where the IT infrastructures are completely closed and permission must be asked to the operator to work or use their infrastructure.

5.2.2 De-Centralized Securities Depository and Settlement System

As stated earlier, Bitcoin itself could be considered as a decentralized security depository, since bitcoin can be viewed as a synthetic commodity. Nonetheless, the fiat currency leg of the transaction cannot be straightforwardly represented on Bitcoin's blockchain. What could be done instead is issuing securities either on an alt-chain or through colored coins, and allow users to exchange them and settle them using bitcoin, or any other cryptocurrency. To be noted though that all of the existing methods to exchange securities on permissionless networks are still affected by a serious problem, which is price discovery. When exchanges happen on a centralized platform, investors can post their orders, which will be put into the order book and filled as soon as another party willing to execute the trade at your conditions is found. Settling securities trades in a decentralized fashion poses the question of how to determine a market price. This being said let's see what are the main methods for trading securities over permissionless networks, keeping in mind that neither of these approaches is being effectively used, both because of the price discovery issue, but also due to specific current limitations. Such limitations can be so severe that they make the entire use case worthless. This is the case of colored coins, as trading them is highly inefficient and poorly scalable. Atomic cross-chain swaps on the other hand could prove to be effective once the issue of decentralized order matching and price discovery is solved.

Atomic Cross-chain Swaps (Nolan, 2014). This method requires that both parties involved in the trade have a wallet on both chains. Let's say there are user A and B, respectively owning assets x and y on two separate blockchains. At first A generates a secret, call it z, which only s/he knows, which will be used to grant that A (B) will not be able to take possession of y (x), without B (A) taking possession of x (y). Now the exchange happens in two phases.

First both A and B create two transactions each. One transaction that transfers the asset (x or y) to the other party and one transaction that nullifies the asset transfer transaction. The first set of transactions is not broadcasted to the network yet, but when they will be broadcasted they will effectively transfer the funds. Both these transactions are conditioned to the knowledge of the secret z in order to be valid. The second set of transactions instead, is needed in case something goes wrong as a backup plan and it is immediately signed by the counterparty. Now both parties have their backup plan and can confidently broadcast the first set of transations.

The second phase sees both parties broadcasting the transaction transferring x from A to B and y from B to A. Both transactions need the secret z to be revealed in order to be considered valid and be included in a block. So as soon as A will try to take possession of the asset y it will reveal the secret z to the network, so that also B sees it, thus finally taking possession of the asset x.

5.3 Permissioned Blockchains, or Distributed Ledger Technologies

Permissioned blockchains are a whole different thing. First of all, they are not in production yetas they did not get beyond the experimental stage as of March 2016. Nonetheless, their embedded characteristics may allow such blockchain-based systems to be recognized and used as Financial Market Infrastructures in a much shorter amount of time, if compared to permissionless networks. In fact, we already saw how the developers of a permissioned blockchain-based software can decide pretty much everything about who has permission to do what. They are built in a bottom-up fashion, by startups and companies with a well-defined governance structure and a legal basis, as required by the BIS basic principles for Financial Market Infrastructures. The applications of such systems may be on *payment systems* as a rail for inter-bank settlement; or can be *"distributed" securities depositories* for recording the ownership of securities; or even *trade repositories* with automatic settlements based on the contractual obligations expressed in smart contracts, built for the purpose.

5.3.1 Payment Systems

The idea envisioned for permissioned distributed networks used as payment systems is that of using the cryptocurrencies (likely tokens generated within a specific permissioned environment) as the backend for peer to peer transactions, where the user does not know that what is being transacted is a synthetic asset on a blockchain-based network. No such system currently exists as a production-grade infrastructure, except for Ripple, which is not a blockchain-based system, but rather pertains to a superclass, that of the Distributed Ledger Technologies.

What happens nowadays in within traditional Payment System FMIs is that (non-cash) funds are moved from the sender to the recipient across multiple level of intermediaries (Kokkola, 2010). Large Value inter-bank payments are settled mostly in real time without netting through various infrastructures depending on the country(ies), UK uses the CHAPS, US uses CHIPS or

Fedwire and TARGET2 in Europe. All of these systems are centralized as this yields higher efficiency[19].

Payment cards (through VISA, MasterCard, AmEx, etc.), credit transfers and direct debits (through SEPA or Fedwire), E-money and cheques are used instead for retail payments. The problem with all of these different infrastructures is that since they are very fragmented and the payment process involves different entities, the payments are actually settled – with settlement finality – with some delay (especially when it comes to retail payment systems. This situation should dramatically improve as the SCT instant payment scheme shall be implemented in Europe by November 2017. The Euro Retail Payments Board defined instant payments as (European Central Bank, 2017):

> *Electronic retail payment solutions available 24/7/365 and resulting in the immediate or close-to-immediate interbank clearing of the transaction and crediting of the payee's account with confirmation to the payer (within seconds of payment initiation). This is irrespective of the underlying payment instrument used (credit transfer, direct debit or payment card) and of the underlying arrangements for clearing (whether bilateral interbank clearing or clearing via infrastructures) and settlement (e.g. with guarantees or in real time) that make this possible.*

Instant payments could help business customers to improve their cash-flow management, and reduce their need to external financing. The availability of funds 24/7 would help to optimize their liquidity management.

Therefore, a centralized solution for slow payments has already been developed, thus the need for a distributed technology to be used solely as Payment System is not stringent. On the other hand, it is worth noting that the SCT instant would work solely in Europe, while cross-border payments would still require long settlement times due to the number of intermediaries, the correspondent banks, playing a role in the process.

[19] TARGET, the first generation of TARGET2 was actually decentralized at its inception back in 1999, but it was later centralized over the Single Shared Platform in order to provide higher service level, faster payments and lower overall costs. This is a clear example of the basic principle that centralization allows to reach higher efficiency levels. It would be very interesting to run a study case on TARGET vs. TARGET2, digging into the data in order to unveil all of the relevant efficiency factors and then compare that transition with the opposite one being pursued by banks nowadays with DLTs.

Thus, a Distributed Ledger Technology-based network may still prove to be useful, since funds within such network may move at near-instant speed. Even further a Payment System is crucial for any other FMI. Securities Settlement Systems need to move funds together with the assets being transferred. Any FMI based on Distributed Ledger Technology will therefore need a Payment System embedded within – or at worst interoperable with – the network itself, otherwise it would still suffer the same frictions experienced today in legacy FMIs.

5.3.2 Distributed Securities Depository and Settlement System

In the case of an application of permissioned blockchain networks to securities trading, it is still unclear how the trade lifecycle will be configured, which actors will leave the scene and which will remain. In fact, as of now today's actors participating in the settlement cycle, carry out a set of functions that are bundled together (verification of ownership, preparation for exchange including associated borrowing of securities or cash, delivery of value against payment) using long standing arrangements (tiered accounts, fungibility of securities ownership). Thus benefitting from the use of a blockchain-based permissioned system for securities trade clearing and settlement will require a substantial reengineering of these arrangements.

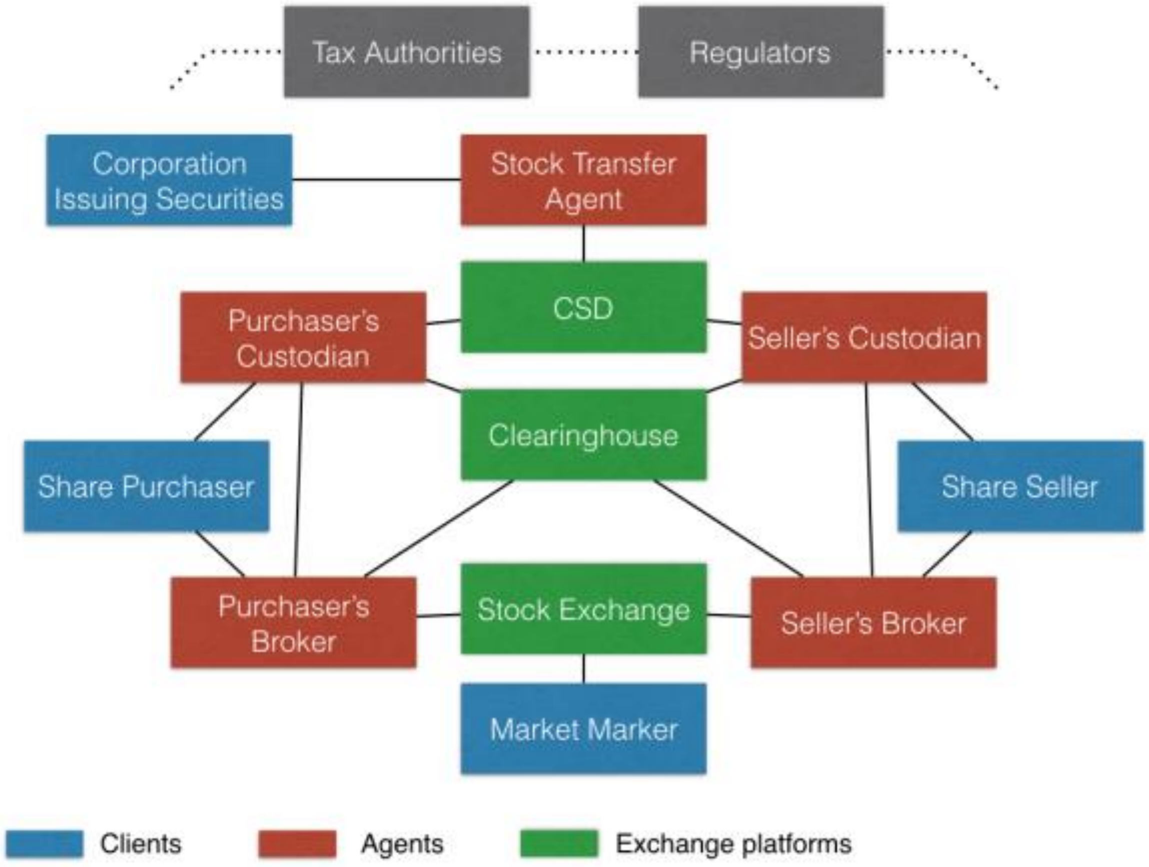


Figure 8 Post-trade securities clearing as-is. Freely inspired from Evans et al. (2016).

As an alternative to the crowded as-is state, market observers (Evans et al., 2016) propose two different scenarios depending on how quickly the technology becomes adopted, but most prominently on which actors will be given certain responsibilities.

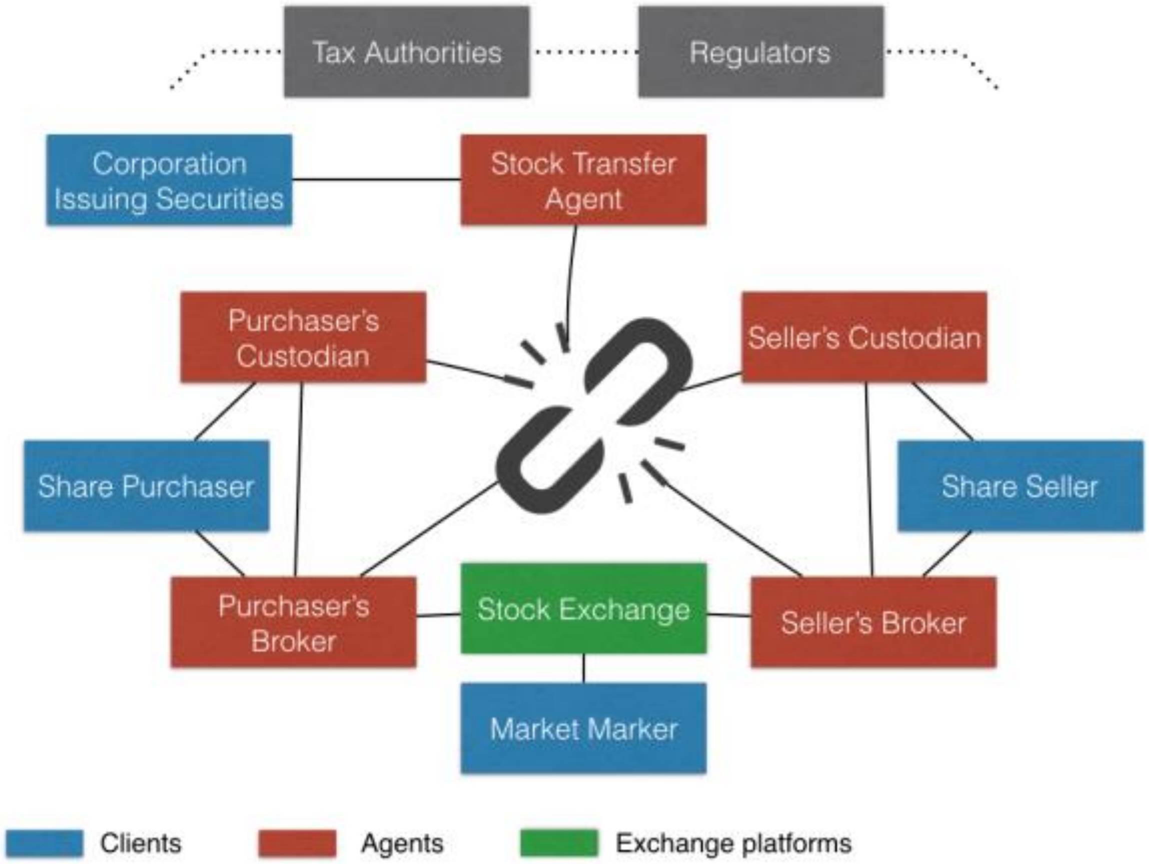


Figure 9 Permissioned blockchains among custodians and transfer agents. Freely inspired from Evans et al. (2016).

For instance, in the first scenario security custodians remain in existence, continuing their role as the *auditors* authorizing the transactions to be posted on the blockchain. The *client* role, those initiating the transaction, will not be given directly to the end clients, but it will a role operated by professional brokers. In this scenario CSD and clearinghouses alone are substituted by a distributed blockchain-based system.

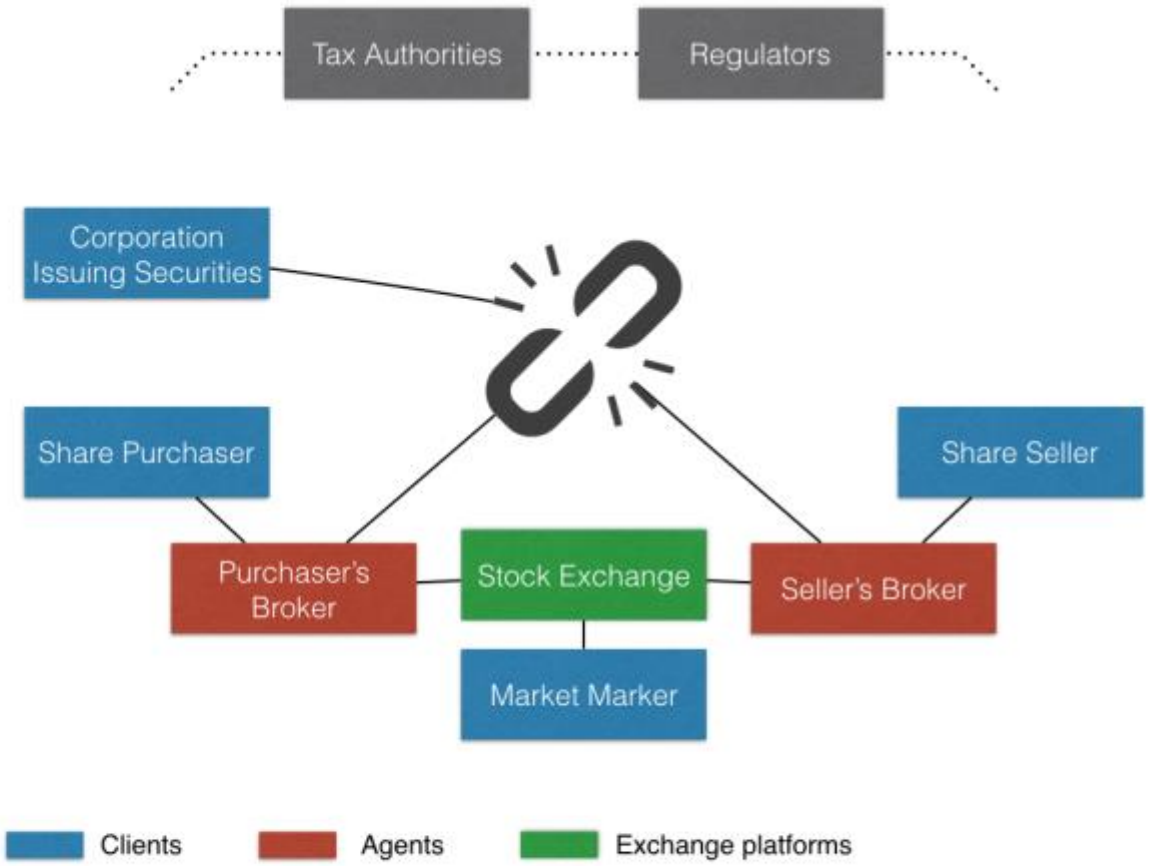


Figure 10 Permissioned blockchain among brokers and issuers. Freely inspired from Evans et al. (2016).

Finally in a more advanced scenario even custodians and stock transfer agents may be left out and completely substituted through a blockchain. According to Mainelli and Milne (2016) using blockchain-based systems for managing securities trade lifecycle, may allow market participants to save up to $40 billion per year. Moreover, for what concerns the permissioned nature of blockchains to be used as FMIs, what stems out of their research is that the interviewed focus groups wanted to be able to accurately control the access and updating rights to the various ledger participants.

Up to March 2017 neither DLT nor blockchain-based permissioned networks are being used in production as FMIs. Only PoCs and pilot projects are being designed and tested, but no tangible result was released to the public, hence it is impossible to drive a cost benefit analysis of such IT infrastructures. Regardless of what will be the results of such tests on permissioned infrastructures, we can already confidently infer that their cost structure will be very different from that of permissionless systems. In fact, permissionless decentralized networks pay for both fixed and variable costs through two sources of revenue: a network fee (transaction fee) and through monetary inflation (the coinbase, or miner reward). Permissioned systems instead, will not have their own currency, neither they will get payed through monetary expansion. The issue is that permissionless systems have initially been envisioned as low cost FMIs, but this was due to an "optical illusion". In fact, ceteris paribus decentralized systems are much more expensive than their centralized equivalent. Bear in mind that information, communication, computation and hardware resources are largely duplicated, and make no mistake, this is not done for redundancy purposes, but rather to solve for the issue of social trust and solving for this is extremely expensive.

Finally, a last word concerning a very recent news (Castillo, 2017). The DTCC has just announced that it will clear derivatives for a value of $11 trillion starting at the beginning of 2018. This decision comes after the assessment of the cost-benefits gained by using this technology for post-trade processes. This is a very important news, because it declares a very serious commitment by part of one of the most important clearing and settlement companies around the world. All previous statements and news had always been vague and concerned proof of concept stage projects. Nonetheless, we find a pretty disturbing quote from that very same article:

> *"Since the distributed ledger's record is immutable, a regulatory node has the potential to give government observers access to real-time data about transactions, instead of having to wait for reports from market participants."*

As we argued previously in sub-section 4.3.1, permissioned blockchains are not immutable per se. At the contrary they are very easily mutable at almost no cost, depending on what role (within the network) will be played by the relevant regulators. For this reason, once again we invite the readers to critically assess what promises are being made regarding this technology.

5.4 Conclusions on Blockchain Networks and Financial Market Infrastructures

We analyzed what are FMIs, what kind of FMIs could be designed through the use of blockchain based networks in permissionless environments, or more generally with distributed ledgers in the context of permissioned networks. Few relevant facts emerged through the literature review:

- Permissionless payment systems currently are the only FMI which is currently operational and open to public access. All of the other applications of blockchain-based networks to FMIs are still in alpha or beta version. Some of them are publicly testable (permissionless ones) while others are being tested behind closed doors.
- Permissionless networks suffer from at least five increasingly dangerous operational risks:
    - Users inability to use the system can easily cause them to lose their funds, with no ability to make recourse,
    - Bugs and disruptive updates may hinder the network's continued operations,
    - The decentralized governance and open source nature of the system, may hinder both decision making and assurance of continued operation of the infrastructure,
    - Majority attacks are intrinsically inevitable,
    - Hard forks may cause a significant loss of confidence in the system.

Starting from these observations we now want to investigate on those operational risks, so as to understand to what extent they can be eliminated or mitigated, or even if they actually pose a significant risk to the proper functioning of permissionless distributed networks as FMIs, and specifically as payment systems.

As a side note the author wants to stress over the fact that being able to provide an answer to the above mentioned questions is relevant also for future developments of the ecosystem. In fact, as we studied earlier in this chapter, FMIs other than payment systems can be built on top of the existing permissionless networks. Risks at the base layer would cause systemic risk to the rest of the ecosystem through a bottom-up contagion process. This is why it is extremely relevant both for the Bitcoin community and ultimately for all of the existing FMIs, to understand whether the above mentioned operational risks can actually be a show stopper.

RESEARCH METHODOLOGY

The overall aim of this thesis is to understand and explore if and how blockchain-based networks could be used as a Financial Market Infrastructure. The first chapter of the literature review question illustrated what is a blockchain-based network, what is its purpose and how new emerging networks – specifically permissioned and permissionless – differ between one another at a high level. Subsequently, in the second chapter of the literature review we assessed what types of FMIs these networks are being (tentatively) applied to. As a result, we understood that most of the FMIs that are trying to be deployed through blockchain-based networks are still in their concept phase.

In particular, permissioned networks are being tested privately, but they are not production ready yet, as of end-March 2017. Currently we cannot conduct much further research on this topic, given the privacy covering the results and their inaccessibility. This being said, studies in the form of Case Studies would be ideal as future researches investigating on the advantages and disadvantages of specific applications of permissioned networks to certain Financial Market infrastructures.

Permissionless networks on the other hand are deployed and functioning since a longer time. Their development is completely open source and most of the discussion takes place openly on internet, thus making it easier to access information. The previous chapter identified Payment Systems and De-Centralized Securities Depositories and Settlement System as potential FMIs. While the second kind are still a concept, the first kind is deployed into production – with Bitcoin, but many other cryptocurrencies as well – but does this make these networks good Financial Market Infrastructures? As we analysed in the literature review we identified a series of operational risks which may seriously affect the viability of permissionless infrastructures as payment systems. Here we identified a research gap. In fact, research focusing on the relevance as well as the possible actions which may mitigate those risks is widely lacking.

Ultimately the question that we want to find an answer to is: are the operational risks faced by permissionless blockchain-based networks insurmountable and they are doomed, or can they still continue to play a role in the future?

Specifically, five operational risks emerged as latent dangers which may jeopardize the stability and the viability of permissionless blockchain-based networks:

- Users inability to use the system can easily cause them to lose their funds, with no ability to make recourse,
- Bugs and disruptive updates may hinder the network's continued operations,
- The decentralized governance and open source nature of the system, may hinder both decision making and assurance of continued operation of the infrastructure,
- Majority attacks are intrinsically inevitable,
- Hard forks may cause a significant loss of confidence in the system.

Our research will focus on assessing the magnitude of their negative effect on the network stability and in identifying what actions may eliminate or mitigate those risks, if any.

6 Research Strategy

The question we want to provide an answer to is related to an extremely specific case. The risks being assessed are, as a matter of fact, so peculiar, that they never materialized in reality, or they did in very few occurrences and in different conditions. Any answer that could be given to this question is bound to be speculative, and no quantitative data exists that could support it. For this reason, the explorative research we are going to conduct is a qualitative one. Many different strategies could be deployed in order to analyze the qualitative data that we are going to collect. Let's go through them and understand which one is the one that best suits our research.

Case Study. A case study is an intensive, holistic description and analysis of a single, bounded phenomenon. The boundedness of the case is of paramount importance. If the phenomenon of interest is not intrinsically bounded, then it is not a case. In other words, if there is no end, actually or theoretically, to the number of people who could be interviewed or to observations that could be conducted, then the phenomenon is not bounded enough to qualify as a case. Our question falls exactly in this field, since we are trying to explore an area of research where no one really has an answer. Actually a definitive answer may be provided when permissionless networks will be completely dismissed because insecure, for instance. Until that day these infrastructures may indefinitely continue to work.

Action Research Studies. This aims at solving a problem in practice, or developing an intervention and to research not only its overall effects, but also how the process itself unfolds. As such, the data analysis in qualitative action research studies is going to focus not only on what happens but also *how* it happens over the course of the ongoing action research cycle of plan, act, observe, reflect. This would be a perfect approach for addressing specifically one of the operational risks we have identified. Unfortunately, the problem of this research strategy is that it requires really long research timeframes and is not applicable for this research. Nonetheless, this type of research would be very warmly advisable to study the matter of decentralized governance across permissionless ecosystems.

Phenomenological Analysis. Phenomenologists are interested in our "lived experience", rather than categorizing, simplifying, or reducing phenomena to abstract laws. To get at the essence of the meaning of an experience, the phenomenological interview is the primary method of data collection. The interviewee shall be one that experienced the phenomenon, but this is the fact with risks, they have never been experienced, because they happen in the future. Nonetheless, some risks have been experienced by some in the past, therefore for these the interviews will be a useful source of data to be analysed through the phenomenological analysis. In fact, the phenomenon of permissionless networks stems from a very peculiar ideology, with its own values and culture. For this reason, any answer provided to our questions will need to be contextualized within the environment we are investigating in.

Ethnographic Analysis. With this kind of analysis, immersion in the site as a participant observer is the primary method of data collection. Interviews, formal and informal, and the analysis of documents and records constitute the data set, along with a fieldworker's testimony of personal feelings, ideas, impressions, or insights with regard to those events. This type of analysis is extremely useful in our case since the author had the chance to cover the role of a complete participant in the permissionless blockchain ecosystem. Therefore, the data collected mainly through on the field observations will help us to cut through the sub-questions related to the main question, as we will see in the next section.

Grounded Theory. The objective of this type of qualitative study is to build a theory that emerges from, or is "grounded" in, the data. Grounded theory is particularly useful for addressing questions about how something changes over time, which is exactly our case. Will the operational risks faced by permissionless networks be a showstopper? The data analysed this way can come from interviews, observations, and a wide variety of documentary materials, also online documents.

Each piece of data is constantly compared with previous results. Initially each segment of collected data which may provide an answer to our questions is analytically coded, which means that the data will be interpreted reflecting on its meaning and how it contributes to provide an answer for our inquiry. Constantly comparing the data involves determining similarities and differences and then grouping the codes together on a similar dimension, so as to create a category. The overall object of this analysis is to identify patterns in the data. These patterns are arranged in relationships to each other in the building of a grounded theory.

Building a substantive theory involves the identification of a core category. The core category is the main conceptual element through which all other categories and properties are connected by the hypoteses. The core category shall be central, i.e. related to as many other categories and their properties as is possible, must appear frequently in the data and must develop the theory.

The theory we will try to build, inductively develops the categories and hypotheses based on which permissionless networks may find their raison d'être.

Ultimately this study is going to be a multi-method qualitative research. Phenomenological and ethnographic analyses together with grounded theory are our strategies of choice, because of their fit to our research question.

7 Data Collection Methods

In order to find relevant answers for our questions regarding the operational risks faced by permissionless ecosystems, we will make use of a broad variety of sources. The data that will be collected and analysed is composed of on-the-field observations from the author (complete participant in the ecosystem), documents and interviews.

7.1 Complete Participant Observations

Being a complete participant in a certain environment means that the researcher will be actively participating in the context under analysis, while conducting its own research. This is exactly the situation in which the author operated. The observations were formally and informally taken over time since end-June 2016 and they are still ongoing. The observations include reconstructions and interpretations of the interactions that the author had with technical experts, colleagues, clients, but even basic users, but also the observations stemming out of the conferences.

These observations will prove to be very useful to start addressing the issue of assessing the magnitude of the effects of risk materialization as well as to understand the culture, value and existing incentives driving those informally responsible to help in mitigating the operational risks.

7.2 Documents and Online Data

Reddit discussions in the r/bitcoin subreddit, tweets (and tweet storms) and private mailing lists area very important source of information. Even though this source of primary data is a non-conventional one for a research, most of the discussions in this ecosystem happen on these channels.

Questions like, what happens in the event of a hard fork or even defining what a hard fork versus a soft fork is, have all already been largely discussed publicly and online. After all we need to realize that we are analyzing a distributed network, not a centralized private organization. It is natural that the information is informally spread on the internet, because blockchain-based networks are the first example of internet organizations. Since their inception, they are meant to serve as a payment system that leverages on the internet distributed infrastructure to work. There is no CEO, no single decision-maker, nonetheless there are agents acting as employees, they are both developers and miners, they all work to maintain the system up and running for its users to operate on it.

7.3 Interviews

Finally, semi-structured interviews are conducted with experts and developers with the objective of identifying methods and actions that could eliminate or at least mitigate the impact of the operational risks faced by permissionless networks.

These interviews have been conducted in person and recorded when this was possible. Being semi-structured interviews they were focused on discussing few relevant points:

- What is the impact of bugs and disruptive updates on the network’s continued operations? How could the negative effects be mitigated?
- Could the decentralized governance and open source nature of the system hinder both decision making and assurance of continued operation of the infrastructure? What actions could mitigate the negative effects of this risk?

- Majority attacks are intrinsically inevitable, what is the magnitude of their effects and how could they be mitigated?
- Hard forks may cause a significant loss of confidence in the system and its overall stability, what do you think is the magnitude of their effects? Also, could this menace be mitigated or eliminated?

At the end of every interview the data was analysed immediately so as to codify it and find relevant patterns in the interviewee's answers which could lead to a generalization through the grounded theory approach. Also it is worth noting that a sort of pilot interviews were conducted before the final interviews. In fact, these questions are a common topic of discussion within the permissionless ecosystem, therefore the researcher was exposed to such topics since the very beginning of its participatory experience within such environment.

8 Sample Selection

The data necessary for providing an answer cannot be possibly collected from all the population, even defining what the population is can be tricky. For this reason, we need to select a sample.

We will not be able to draw any conclusion based on statistical inference on the data that will be collected, since this is highly qualitative and it concerns future unrealized events.

We would like the sample to be representative of the entire population, nonetheless at this early stage no quota segmentation is available, so we don't know how is the entire population segmented. Any sample that we would select would be anyhow dubiously representative of the entire population.

Because of the above reasons, but also because the purpose of the study – which is still a highly exploratory one concerning a new phenomenon – we will choose a self-selection sampling method. We need to keep in mind that we are trying to assess the impact of the operational risks and identify possible mitigating actions. In order to do so we need to refer to the portion of people knowledgeable about how permissionless networks work and those that possibly directly operate within these systems. The ideal sample from which to collect our data – observations, documents and interviews – shall include the most relevant stakeholders of the permissioned systems, i.e. users, relevant holders (retaining a large stake in cryptocurrency), developers, miners, businesses operating in this ecosystem.

RESEARCH FINDINGS

9 Data Collection and Analysis

This section elaborates on the findings emerging from observational, documentary, testimonial and interview data.

The first relevant finding concerns the convoluted nature of permissionless networks decision making processes. They happen on separate layers, a basic automatized layer enforced by full validating nodes, and the architects' layer on top, where development decisions are taken regarding how the trustware shall evolve, or be maintained. Any decision taken by the second layer of developers cannot simply be enforced on the entire network, since it is the first layer which ultimately takes decisions. These upgrades can eventually be deployed both as hardforks or softforks, but both the security and effectiveness trade-offs are yet to be researched upon.

The second relevant series of findings regards the incoherence of how operational risks have been presented up to now in extant literature. For this reason, we propose a better organized framework for risk categorization, built through a continuous process of knowledge coding, as advised by the grounded theory approach. The two relevant dimensions for risk categorization concern the risks severity and their nature. On one hand there are partial or critical service failure, depending on the impact that each risk has if it materializes. On the other hand these risks have a double nature, either they are faults and failures internal to the existing codebase or rules and processes, or they can arise as attacks coming from external or rogue actors.

Finally, the last series of findings come about with the analysis of risk mitigation actions. What has emerged is that Internal faults and failures can be mitigated through processes and practices which are becoming quite standard. External attacks on the other hand, are not conclusively mitigated through any standard procedure, rather a case by case mitigation strategy shall be adopted.

A foreword on how this chapter will be organized. It was extremely difficult to segregate the different data collection sources and yet maintain a coherent text structure which followed a logic flux of information. Observations, documents, testimonies and interviews help all together and at the same time to expose the different concepts needed to get to the root of our question. It is important for the validity of the results to explicitly state whenever we are making use of an observation, a testimony or an interview, for this reason we will apply the following formatting:

> ***Observation #.*** *The data being analyzed here is not aligned with what happens in reality. We observed throughout the last year that there is a great amount of confusion regarding this or that, hence we need to explain how does that mechanism actually work.*

Usual formatting will be used for further explanations of the matter under scrutiny. Observations may or may not be sustained by further testimonies or interviews:

> ***Testimony / Document / Interview*** *#. The author of the testimony is here cited with the link to the relevant information being used. A brief introduction to the individual may be useful, so as to perceive the value of his/her analysis.*

Testimonies and interviews will be followed by the body of the relevant parts of those interventions:

Any further explanation by the experts will be copied and cited using this formatting. These quotations will be alternate by plain text with the normal formatting, this will be done so as to provide further analysis by the researcher and to link the concepts being explained to the objectives of the research.

9.1 Understanding the operational risks

Our objective is to understand whether the operational risks to which open permissionless networks are exposed can be a showstopper or not. The major risks identified are:

- Bugs and disruptive updates may hinder the network's continued operations,
- Users inability to use the system can easily cause them to lose their funds, with no ability to make recourse,
- The decentralized governance and open source nature of the system, may hinder both decision making and assurance of continued operation of the infrastructure,
- Majority attacks are intrinsically inevitable,
- Hard forks may cause a significant loss of confidence in the system.

> ***Observation 1***. *The operational risks identified in previous literature are quite messy and incoherent between one another. Many different concepts are called in for scrutiny and they are neither well known nor understood broadly. The two most important concepts here are the Decentralized governance and the hardforks (together with the softforks). Both concepts are strictly interrelated and they ultimately intertwine with the very fundamental mechanisms of open source permissionless networks.*

9.1.1 Decentralized governance

A system's governance, relates to "the processes of interaction and decision-making among the actors involved in a collective problem that lead to the creation, reinforcement, or reproduction of social norms and institutions." (Hufty, 2011). When it comes to permissionless networks, many talk about their decentralized governance, meaning that the decision-making process is not controlled by one single entity, but this is very vague. De Filippi and Loveluck (2016) got straight to the point when they identified two separate layers of governance within the overall Bitcoin system, they are the *infrastructural layer* and the *architects layer*.

9.1.1.1 The infrastructural layer, or the protocol consensus governance

The infrastructural layer is the one implemented into Bitcoin's (or Ethereum, or Monero's, or ZCash's) consensus rules through the *full nodes* software. Basically this is the automatized governance layer and it is highly decentralized. We saw in section 4.1.1.3 and 4.1.2.1 how transactions over permissionless networks are automatically validated and enforced by the auditors, i.e. the full nodes, and then they are chronologically ordered into blocks – so as to avoid double spends – by the system maintainers, the miners.

At this level, the network – through the protocol implementation software – automatically takes the most important decisions regarding what constitutes a valid transaction, but also most importantly, what constitutes the valid chain of blocks, hence the valid coins.

> ***Observation 2***. *Here we observed a very great deal of misunderstanding over which actors enforce the software rules. Way too many observers misunderstand the power of the economic incentives embedded within the Bitcoin (as well as most of the other permissionless blockchain networks) protocol driving each and every stakeholder's choice. Miners solely chronologically order transactions, but it is full nodes which enforce the protocol consensus rules as by their software version. This*

is a very critical passage both for understanding the most nuanced details of how permissionless networks work, as well as to understand how full nodes role play out in guaranteeing the system's stability against operational risks.

***Testimony 1**. Emin Gün Sirer (Sirer, 2016) has a very well explained take on this issue. He is professor at Cornell university, his research spans operating systems, networking and distributed systems. He is Co-Director of the Initiative for Cryptocurrencies and Smart Contracts (IC3) one of the most prominent academic partnerships with the objective of studying blockchain technologies.*

It was painful to read Coinbase CEO Brian Armstrong's piece [Armstrong, 2016] on the state of Bitcoin. There are two reasons for feeling depressed, one technical, one social.

Armstrong states:

If a majority of bitcoin miners "vote" for a particular upgrade then by definition this is the new version of bitcoin.

It is a common misperception that Bitcoin miners determine the shape of the blockchain. So common, in fact, that I had an academic colleague fall into the same trap, where he thought miners could arbitrarily decide to mint 1 million coins for themselves. [...]

If miners can indeed unilaterally decide the shape of the block chain, if they can unilaterally change the rules, why don't they? If you could mint 1 extra coin per block, why would you not? If your answer to this is "because 51% of the miners would not go along with that," then you need to ask yourself why the heck not. There are already suspicions that the Chinese pools are actually just one entity that exceeds 51%. We had three previous occasions when a miner exceeded 51%, and they did not unilaterally change the rules. [...] Why doesn't this happen?

*Suppose some miner did create a block whose coinbase transaction awarded him far more than the 25BTC that is allowed these days. And suppose this miner had majority of the hashing power. He would be able to make a really long chain, longer than the chain that the rest of the miners are able to build. And yet **not a single person would honor his misshapen blocks**. He can have all the bitcoins on his own blockchain, but he won't be able to convert them to actual wealth. No one would honor them.*

I tried to get this point across to Armstrong, who responded with "maybe the right way to think about it is miners have the votes, wallets and exchanges have a veto." That's not right, either. [...] suppose you live in a town with a large restaurant, with a fat chef, and a small restaurant, with a skinny one. The large restaurant has 51% of the food-making power, while you have all the cash in town (population: you). In this town, does the chef have the vote, and do you, as the consumer, have a veto? Are you compelled to eat everything that the fat chef puts in front of you, and pay him for it, just because he has a large restaurant? If you still said yes to all those, what happens when he starts serving haggis, spleen, and mountain oysters? If you still think you're a helpless consumer controlled by the fat chef, think about it from his perspective: what would he do if you just dropped his dish on the floor, walked out without paying, and checked out the wholesome food prepared by the skinny chef next door? How is he going to convert his food to dollars? Remember, you're the only game in town, the only person with actual dollars; he needs your cash, and all he has is some offal that no one in their right mind wants to touch.

The bottom line is that the ***consumers in Bitcoin, wallets and exchanges, have immense power****. They wield that power, by proxy, through large companies like Coinbase. Where, does anyone think, will the miners convert their bitcoin into cash, if not at exchanges like Coinbase? How will they pay themselves, let alone for their next generation of equipment? And I will be polite and leave alone the touchy subject of their electricity costs.*

So it was shocking to me that the CEO of the largest Bitcoin exchange does not realize either his power or his responsibilities.

The bottom line is that it is full nodes enforcing consensus rules, not miners. Even more there is a fundamental nuance to that. Large economical nodes have the most power. Currently there are somewhere around 7000 full nodes, but only a small portion of those are operated by economic full nodes. Eventually it all turns down to who gives value to bitcoin, or any other token.

Let’s explain that with a thought experiment. One may start to run a full node with its very own “Bitcoin” rules, which are different from the rules that the rest of Bitcoin’s full nodes agree upon. The implication is that this guy’s full node (which scans the chain of blocks in order to find valid transactions included into valid blocks) may not accept a transaction coming from the original network. The reverse could be true as well, so that this guy may not be able to send out valid transactions to the rest of the network, because those transactions would not be considered valid by the rest of the full nodes.

The bottom line is that running a node with different consensus rules (from the rest of the network) would completely isolate the nodes rendering their network valueless. Ultimately the value of a cryptocurrency comes from the network effects, this is why it is critical for the network not to split. As we will see this concept is strictly related to most of the concerns regarding permissionless networks.

But if it is true that every full node takes decisions individually and automatically, and if it is true that all of the nodes participating in the network should comply to exactly the same rules as written in the exact same software, then is this governance layer actually decentralized at all?

The answer is yes. Each single node enforces the rules that it is programmed to enforce. Who programs it to do so? The owner of the node, but since the entities controlling the nodes can have conflicting interests, then the infrastructural governance layer is a very decentralized one. The economic incentives generated by the network effects normally cause this layer to look very uniform and cohesive, but in principle this layer could be fractured leading to a chain division, which in turn splits the economy[20].

This said, the automatized infrastructural governance layer is the most important one. This is the ideal governance layer and all of the disputes and decisions for the networks should be ideally solved at this layer. No human intervention shall ever be required, so as to guarantee that all of the rules are clear for everyone and no discretional decision is ever taken.

> ***Observation 3****. Most of the complexity and delicacy of permissionless networks comes from their ideological foundations. In fact, the most basic assumption is that every entity can freely join the network and it shall not be coerced to agree to new rules different and in contrast to the ones s/he initially agreed to. This means that it shall ever be unilaterally introduced a contentious change of the rules of the game, such that early adopters would be excluded from the network. Some sort of XXI century libertarian social contract. This is true for Bitcoin, but it proved to be false in Ethereum's case with the DAO case (Torpey, 2016).*

> ***Testimony 2****. Alex Morcos is one of the Bitcoin core developers which contributed the most to its codebase. In an article which analyzed the DAO case (Torpey, 2016), Morcos stated:*

The question on whether any cryptocurrency is eventually going to be truly successful depends in large part on the model for social consensus that we develop for this novel concept. What this model should look like is hard to answer, and we see Bitcoin struggling with that question as well. But I think many in the Bitcoin ecosystem at least share the belief that the proposed Ethereum hard fork is a bad precedent to set in trying to build the right social model.

It's important to build a community that wouldn't demand it, because that's not their expectation about the way things should work.

This is the reason why decentralized governance is so hard to be dealt with. Emergency situations arising from unaccounted exceptions require the intervention of the software architects, who act as maintainers, but will their proposal be accepted – opted-in – by the network?

[20] Of the specific permissionless network that is splitting.

9.1.1.2 The architects layer, or the development team governance

Of course having all of the rules automatically enforced through the infrastructural layer is as utopic as it sounds. In fact, the infrastructural governance layer is inherently as fallacious as the humans who wrote the code. For this reason, there is another governance layer, as correctly identified by De Filippi and Loveluck (2016), which is the development team layer. The developers contribute with the code for the protocol software implementation with the objective to maintain and improve its functionalities.

This governance layer is decentralized as well, because permissionless blockchain-based networks are run with open source software. This means that ultimately there is no owner of this software and it can be simply copied or modified at will by anyone. Nonetheless, we have come to understand that the actual value of these permissionless networks come from their network effects. In particular, the network effects stem out of this codebase if and only if the node uses a software which enforces exactly the same consensus rules as everyone else on the network. Currently Bitcoin's software implementation is available on Github, a web based repository where programmers host their codebases. This platform is extremely versatile and particularly useful for distributed development. It provides access control so as to restrict updating of a codebase only to the few with the commit access. Bitcoin's reference software implementation is called Bitcoin core and its latest version is 0.14.

> ***Testimony 3***. *Satoshi Nakamoto himself is our next testimony (Nakamoto, 2010). In this post on Bitcointalk.org – Bitcoin's original forum – Nakamoto analyzes the convoluted nature of permissionless blockchain network's software.*

*The nature of Bitcoin is such that once version 0.1 was released, **the core design was set in stone for the rest of its lifetime**. Because of that, I wanted to design it to support every possible transaction type I could think of. The problem was, each thing required special support code and data fields whether it was used or not, and only covered one special case at a time. It would have been an explosion of special cases. The solution was script, which generalizes the problem so transacting parties can describe their transaction as a predicate that the node network evaluates. The nodes only need to understand the transaction to the extent of evaluating whether the sender's conditions are met.*

The script is actually a predicate. It's just an equation that evaluates to true or false. Predicate is a long and unfamiliar word so I called it script.

The receiver of a payment does a template match on the script. Currently, receivers only accept two templates: direct payment and bitcoin address. Future versions can add templates for more transaction types and nodes running that version or higher will be able to receive them. All versions of nodes in the network can verify and process any new transactions into blocks, even though they may not know how to read them.

This very last passage is very relevant and it will be analyzed again later on when we will explain what hardforks and softforks are, so let's just keep this in mind.

The design supports a tremendous variety of possible transaction types that I designed years ago. Escrow transactions, bonded contracts, third party arbitration, multi-party signature, etc. If Bitcoin catches on in a big way, these are things we'll want to explore in the future, but they all had to be designed at the beginning to make sure they would be possible later.

I don't believe a second, compatible implementation of Bitcoin will ever be a good idea. ***So much of the design depends on all nodes getting exactly identical results in lockstep that a second implementation would be a menace to the network****. The MIT license is compatible with all other licenses and commercial uses, so there is no need to rewrite it from a licensing standpoint.*

This is one of the most critical points of blockchain-based networks. They are distributed computing networks which run on consensus. Every single node of these network must achieve exactly the same result after running the scripts of the transactions, if they don't the network will get diverging and inconsistent states, what is called a network split. This represents a system's failure for any distributed computing network, but even more so when the system is a mission critical one and people's funds depend on this system's stability and well-functioning. This is the reason why few developers of the core development team have commit access to Bitcoin's codebase (the same is true for every other permissionless network), only the project maintainers. Project maintainers have commit access and are responsible for merging patches from contributors. They perform a janitorial role merging patches that the team agrees should be merged. They also act as a final check to ensure that patches are safe and in line with the project goals. The maintainers' role is by agreement of project contributors, usually called core developers.

How does a programmer actually contribute to Bitcoin's codebase?

Document 1*. On Bitcoin's Github repo we can find some documentation explaining the contribution process (btcdrak, 2015) and the decision making process made at this level.*

The Bitcoin Core project operates an open contributor model where anyone is welcome to contribute towards development in the form of peer review, testing and patches.

[...] Firstly in terms of structure, there is no particular concept of "Core developers" in the sense of privileged people. Open source often naturally revolves around meritocracy where longer term contributors gain more trust from the developer community. However, some hierarchy is necessary for practical purposes. As such there are repository "maintainers" who are responsible for merging pull requests as well as a "lead maintainer" who is responsible for the release cycle, overall merging, moderation and appointment of maintainers.

[...] The codebase is maintained using the "contributor workflow" where everyone without exception contributes patch proposals using "pull requests". This facilitates social contribution, easy testing and peer review.

*[...]**"Decision Making" Process***

*[...] **Whether a pull request is merged into Bitcoin Core rests with the project merge maintainers and ultimately the project lead**.*

Maintainers will take into consideration if a patch is in line with the general principles of the project; meets the minimum standards for inclusion; and will judge the general consensus of contributors.

In general, all pull requests must:

- *have a clear use case, fix a demonstrable bug or serve the greater good of the project (for example refactoring for modularization);*
- *be well peer reviewed;*
- *have unit tests and functional tests where appropriate;*
- *follow code style guidelines;*
- *not break the existing test suite;*
- *where bugs are fixed, where possible, there should be unit tests demonstrating the bug and also proving the fix. This helps prevent regression.*

***Patches that change Bitcoin consensus rules are considerably more involved than normal because they affect the entire ecosystem and so must be preceded by extensive mailing list discussions and have a numbered BIP**. While each case will be different, one should be prepared to expend more time and effort than for other kinds of patches because of increased peer review and consensus building requirements.*

Here we can see what is the responsibility of both core developers, but especially maintainers. They have to review and eventually accept any changes to the codebase. This is a long process which needs to undergo thorough inspections by competent developers, because at stake is the entire market capitalization of the underlying tokens. With this in mind we should also consider that, as for any other open source software, the project maintainers are not paid for this lengthy and demanding process. We will later see what the implications of this decision making process are for the overall network security and dependability when assessing the operational risks.

The concepts illustrated in this subsection 9.1.1 are specific to Bitcoin. Nonetheless, the same high level concepts are true for any other permissionless network, but in a softer or more tolerant way. In fact, all of the permissionless blockchain based networks have at least two layers of governance. Most of the differences between the various networks lie in the base governance layer, the consensus layer, where the rules regarding the validity of transactions are written in code.

In fact, different crytpocurrencies have different validation rules, most of the time. The second governance layer on the other hand is much more centralized for permissionless networks other than Bitcoin, as they are younger and smaller projects.

Regardless whether good or bad decisions are taken, as a rule of thumb centralized governance processes are much more efficient than decentralized ones. This is due to the widespread and conflicting interests of the different stakeholders participating in a dispersed governance structure.

The mechanisms of consensus over decentralized networks are at least as delicate in permissionless networks (other than Bitcoin) in general, as they are in Bitcoin. This is caused by relationship existing between the actual level of decentralization of such networks and their inertia. The more a network's infrastructural governance layer is dispersed[21], the more it will tend to maintain status quo regarding the consensus rules.

Each single individual, or an entity, owning a full node is responsible for maintaining and updating his/her own node(s). By extension this individual will only trust his/her very own node(s), and that alone, to "tell" him what is the valid chain, hence what is the current state of the system (resource distribution for cryptocurrency networks).

Here comes into play the second layer governance, that of the architects. Anyone can propose a change in the rules dictate by the software, the issue is who will opt-in these rules? The answer is whoever agrees that the proposed change is actually beneficial to himself and the rest of the network. This is very reasonable and it holds, as long as the majority of entities controlling full nodes (thus making decisions) in the first governance layer will actually opt-in for this rule change. Nonetheless, it may well happen that 50% of the clients and validators (recall section 4.1.2.1) will agree to the change, while the remaining does not want to update their nodes to the new version. This causes a split in the network.

The actors in the second layer proposing changes for the first layer, usually also participate in the first layer through their full nodes. If the second layer exerts significant power – through their own full nodes but also through dialectical methods – over the first layer, then the decision

[21] Be careful that measuring the actual level of decentralization is practically impossible by design. In fact, it is not the number of existing full nodes over the P2P network (also a complex estimate per se) that determines how decentralized a network is, but rather it is the number of entities actually controlling all of the existing nodes.

making process would practically happen only at the second layer. This is true in semi-decentralized networks such as Ethereum as we saw with the second testimony at the end of section 9.1.1.1. In Bitcoin instead, the community grew so much and in such a dispersed way, that first and second layer have little in common, thus making the whole decision making process much slower.

9.1.2 Hardforks and softforks

Now that we have understood how Bitcoin governance works, we need to understand what exactly means to change the rules embedded within the first governance layer.

Let's imagine consensus rules as that set of conditions that determines whether a block of transactions can be considered valid, therefore other system maintainers (miners) can build on top of that or not.

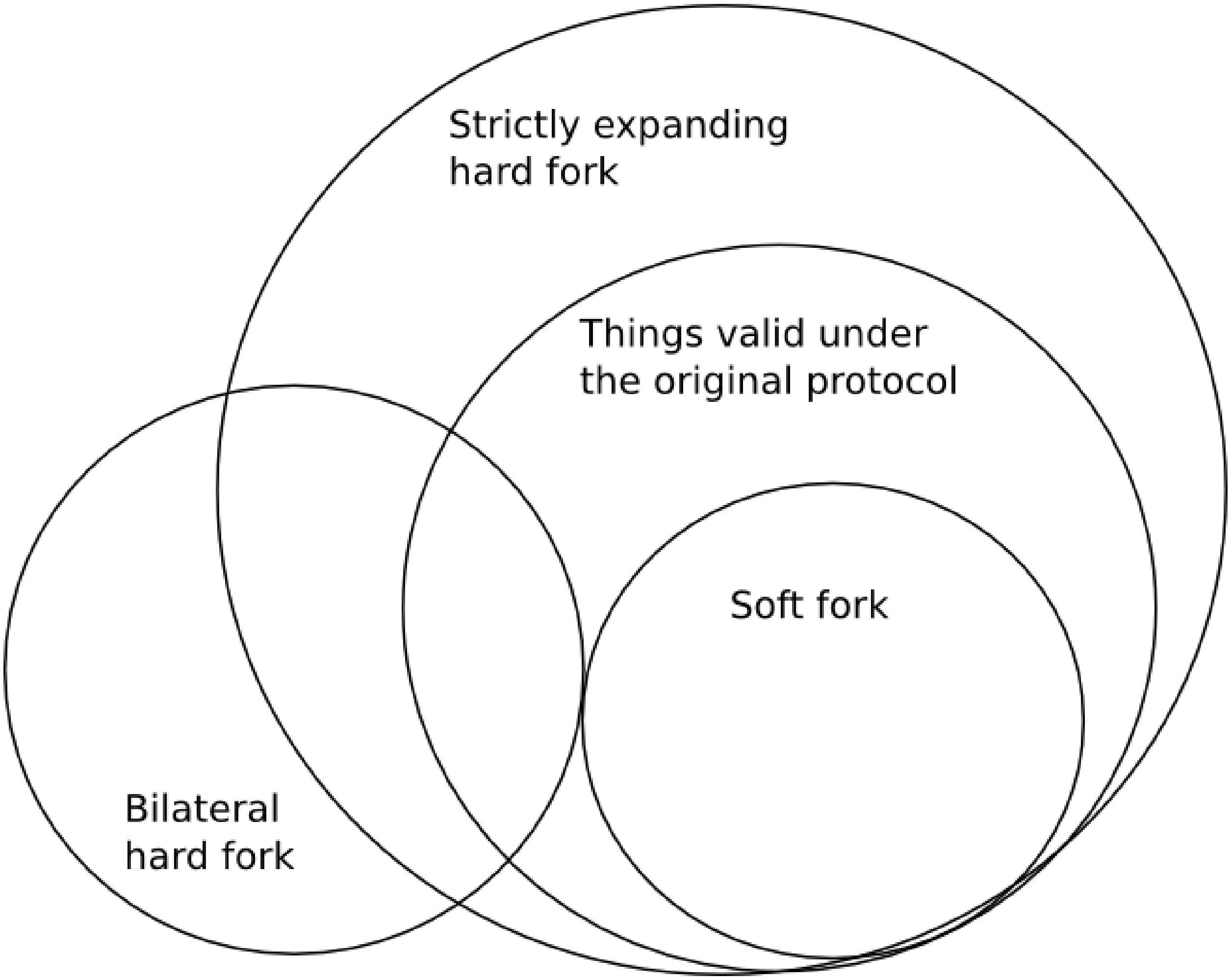

Strictly expanding hard fork
Things valid under the original protocol
Soft fork
Bilateral hard fork

Figure 11 Consensus protocol rules, hardforks and softforks (Buterin, 2017)

As we can see from the figure above we can understand that:

- **Softforks *add* new rules to the protocol, making previously *valid* blocks *invalid*.** Adding new rules, means making the rule set even stricter, making it a subset of the original set. The first implication of such a rule change is that any validator, a *full node*, that accepted the original protocol rules, will accept also blocks mined with the new rules, because they will simply represent a particular case of previous blocks. The second implication is that maintainers/miners shall start mining only with the new rules. In fact, if the economic majority of full nodes switched to the new rule set, then they would not accept blocks mined with old rules outside of the new softfork rules, thus making miners work worthless.
- **Hardforks *remove* rules from the protocol, making previous *invalid* blocks *valid*.** This type of rule change requires a *quick* and *coordinated* major migration of full nodes to the new upgraded ruleset. If they don't they risk losing their funds, because there is no first layer automated conflict resolution mechanism anymore, just two different rulesets a chain & network split. The full nodes should hence decide on which chain they should be operating (sending and receiving transactions), hoping that their chain of choice will eventually be the one with the strongest network effects. Nonetheless this is not as easy as it sounds. In fact a strictly expanding fork causes a great deal of confusion for full nodes, the line between the old rules and the new rules may not be as straightforward as it looks from the outside, so that many attack vectors can be introduced by such hardforks.

Bilateral hardforks are not going to be discussed here, since they only bring additional complexity to this thesis. For the interested reader Pace (2017), Todd (2016) and Buterin (2017) provide further high level intelligible discussions over the issue of what types of forks exist.

> ***Testimony 4**. Andreas Antonopoulos, author of Mastering Bitcoin (one of the books we used as reference for this thesis), had a very clear take on the issue of decision making and update processes in permissionless networks (Antonopoulos, 2017). Here he introduces what he calls trustware:*

How do you coordinate an international network, so that everybody changes the rules at the same time and they don't accidentally invalidate old transactions? *This is the essence of trustware, we are now writing software that is backed by hardware, deployed on a network and establishes a set of global consensus rules. If you make a mistake on those consensus rules and go out of consensus, you can lose millions and can get cheated out of transactions, you can get replay attacks or malleation attacks. This is not a game, it is a new software frontier, only it is not software, but trustware. Trustware is way more complicated than software, or hardware or the two combined together, because there is also the global network component that is controlled by independent actors. [...]*

Coordinating the entire dispersed network, so as to ensure that everybody is onboard for a ruleset change is the most complex thing to be operated on permissionless networks. Nonetheless, finding a safe way to implement rule changes is critical, because as we will see in the subsequent section 9.2, practically all of the most dangerous operational risks can be solved for with an emergency patch, through a software (trustware, as Antonopoulos calls it) upgrade.

What is the safest way to implement those rule changes then? Well as always it is a matter of trade-offs.

Testimony 5*. Vitalik Buterin (2017), Ethereum's creator, discusses why hardforks are preferable to softforks in his opinion:*

The benefits commonly cited for the two are as follows.

- *Hard forks allow the developers much more flexibility in making the protocol upgrade, as they do not have to take care to make sure that the new rules "fit into" the old rules*
- *Soft forks are more convenient for users, as users do not need to upgrade to stay on the chain*
- *Soft forks are less likely to lead to a chain split*
- *Soft forks only really require consent from miners/validators (as even if users still use the old rules, if the nodes making the chain use the new rules then only things valid under the new rules will get into the chain in any case); hard forks require opt-in consent from users*

Aside from this, one major criticism often given for hard forks is that hard forks are "coercive". The kind of coercion implied here is not physical force; rather, it's coercion through network effect. That is, if the network changes rules from A to B, then even if you personally like A, if most other users like B and switch to B then you have to switch to B despite your personal disapproval of the change in order to be on the same network as everyone else.

[...] Soft forks are a dangerous game, and they become even more dangerous if they are contentious and miners start fighting back. Strictly expanding hard forks are also a dangerous game. Miner-activated soft forks are coercive; user-activated soft forks are less coercive, though still quite coercive because of the economic pressure, and they also have their dangers. If you really want to make a contentious change, and have decided that the high social costs of doing so are worth it, just do a clean bilateral hard fork, spend some time to add some proper replay protection, and let the market sort it out.

***Testimony 6**. Pieter Wuille (2015), Bitcoin Core contributor and former Core Project Maintainer, has a different take on softforks. One thing shall be made clear as of now already, his perspective does not entail coercion issues at all, but only the security concerns of implementing upgrades into "trustware":*

> *[...] there are also security advantages that softforks offer:*
>
> *A) Softforks do not require the pervasive consensus that hardforks need. Soft forks can be deployed without knowing when all full nodes will adopt the rule, or even whether they will ever adopt it at all.*
>
> *B) Keeping up with hard forking changes puts load on full node operators, who may choose to instead switch to delegating full validation to third parties, which is worse than just validating the old rules.*
>
> *C) Hardfork coordination has a centralizing effect on development. As hardforks can only be deployed with sufficient node deployment, they can't just be triggered by miner votes. This requires central coordination to determine flag times, which is incompatible with having multiple independent consensus changes being proposed. For softforks, something like BIP9 supports having multiple independent softforks in flight, that nodes can individually chose to accept or not, only requiring coordination to not choose clashing bit numbers. For hardforks, there is effectively no choice but having every codebase deployed at a particular point in time to support every possible hard forks (there can still be an additional hashpower based trigger conditions for hardforks, but all nodes need to support the fork at the earliest time it can happen, or risk being forked off).*
>
> *D) If you are concerned about the security degradation a soft fork might bring, you can always configure your node to treat a (signaled) softfork as a hardfork, and stop processing blocks if a sortfork condition is detected. The other direction is not possible.*

The point of choosing hardforks over softforks for generic trustware upgrade or ruleset changes deployment, needs to be evaluated on a case by case basis. Unfortunately, there is no general rule yet, and in-depth analytical studies should be run by expert computer scientists, both practitioners and academics. Nevertheless, softforks represent an optimal strategy for emergency situations resolutions as we will see later on.

9.2 Operational risks: severity assessment & classification

We now have a more informed view of what are the underlying complexities of permissionless networks. Let's get back to the operational risks identified in previous literature:

- Bugs and disruptive updates may hinder the network's continued operations,
- Users inability to use the system can easily cause them to lose their funds, with no ability to make recourse,
- The decentralized governance and open source nature of the system, may hinder both decision making and assurance of continued operation of the infrastructure,

- Majority attacks are intrinsically inevitable,
- Hard forks may cause a significant loss of confidence in the system.

***Observation 4**. Again continuing from observation 1, we observe that these operational risks are still largely mixed together and uncoherent between each other. For this reason, we want to propose two dimensions that can guide us in better framing these risks: their severity and their category.*

9.2.1 Risks severity

Not all risks are equal, some have a sensibly larger impact on the overall network compared to others. In order to guide our analysis we need to get back to our overall question. Can these risks prevent permissionless networks from being a Financial Market Infrastructure? This one generates another question at once. Why could a risk – any risk – prevent a system from being used as a FMI?

Ideally a *perfect* FMI is a system such that:

- When queried, it provides a correct state of the distribution of resources (be it money, securities, or any form of valuable asset) among the parties using such system *at all times*. The correctness of the data provided depends on a set of clear pre-defined rules, the so called consistency rules.
- When instructed by its users, the system allows to *instantly* transfer the ownership of the resources to other parties participating in the system, if the consistency rules are satisfied.

Now notice that such a perfect system would be a real-time FMI, because it entails a consistent view of the current state of the resource distribution status at any time. Such a system is ideal and is not feasible even with a centralized system[22]. Now what we do as participants in such systems is to tolerate them to not be real-time, we accept some lag in our everyday transactions. The smaller the time gap between the real-time system and the latest state we are presented with by the asynchronous system we operate in, the more the system is a good FMI. Therefore, the **timeliness** of a FMI merely determines the service level which will be provided to its users. This dimension is a driver for market adoption towards one FMI rather than another one, if they can freely choose.

[22] Simply because of the time it takes for information to be conveyed (through light) across the distance separating the actors participating in the system. Synchronicity is not achievable in reality, though we can get really close to it.

On the other hand, there is another, much more critical dimension which is the **internal consistency** of the system being used as an FMI. A lack of consistency within a FMI determines a critical failure of the system. This means that a user could tolerate a FMI to be extremely slow as long as it remains consistent. Worst case the user will simply switch to a timelier system, if available. For instance, the user could start paying through (near-)instant payment systems such as Paypal, rather than using wire transfers. On the other hand, a lack of internal consistency within a payment system may likely cause users to lose their funds because of a system error, which is not tolerable.

There are therefore two broad classes of risks:

- The ones causing a service level degradation, they represent a **partial service failure**,
- The ones causing a state consistency failure, they represent **critical service failures**.

With this in mind, we can get back to the identified operational risks and assess their severity:

Users inability to use the system can easily cause them to lose their funds, with no ability to make recourse. Such risk is caused by a user inability to correctly use the system. The user experience provided by most of the permissionless networks is pretty poor. Users are not identified with their real names, but rather with a long string of random numbers, and they are not static, but rather change every time a new transaction is made (see section 4.1.1.2). Nevertheless, this is not a system's failure. Permissionless networks are purposefully designed to work like that, because they aim to protect users' privacy as much as possible. A user that is not able to use the system as it is – and still wants to use it – will be able to turn to companies specialized in providing a user friendly experience through their services built on top of the basic settlement layer. Therefore, this is is an operational risk, but it is not proper of the FMI under scrutiny. In fact, it is responsibility of the entrepreneurs working at the application layers – providing wallet services to the users – to ensure that their software is easy to use and does not causes people to lose their funds

Bugs and disruptive updates may hinder the network's continued operations. Here we need to make use of the distinction we have drawn before. The bugs within the software implementation could disrupt the network on two different levels. The disruption may result in a temporary – however lengthy – service unavailability, or in a critical failure of the system to comply with the consensus rules, thus resulting in users losing their funds. Both risks are considered, but the severity of the second event could cause serious drawbacks such as a drastic loss of network participants' confidence in the FMI.

Hardforks may cause a significant loss of confidence in the system. A proviso needs to be made on this issue, why is a fork being made? As we saw forks can be used as a means for software upgrades, or protocol modifications more in general. If the change being implemented (through a hardfork) is highly controversial among the network users, then it may cause severe disruptions. Full nodes decide which chain shall be followed, but in the event of a contentious, nebulous hardfork it may be non-straightforward for full nodes to correctly select the correct chain. Hardforks are dangerous and difficult to manage. A great deal of off-chain coordination is needed and still, results are not guaranteed and new attack vectors for both the resulting chains may cause people to lose their funds. Ultimately the most dangerous hardforks are the contentious ones, not the uncontentious ones used solely as a means to upgrade the trustware features. We will delve deeper into this distinction when we will review and classify the risks in a slightly more coherent way in section 9.2.2.

The **decentralized governance** and open source nature of the system, may hinder both decision making and assurance of continued operation of the infrastructure. Recalling the distinction made in section 9.1.1, we know that the decision making process happens on at least two distinct layers. On one hand there is the decentralized decision making enabled and enforced by the nodes running the protocol implementation software, this process is highly automated and predictable. On the other hand, there is no specific single person or authority *legally* vested with the power to take decisions concerning changes to the software run by the network. Bitcoin core project leads, can propose a software upgrade, but then it is left entirely to the network whether they want to adopt it or not. Lead developers' intervention could also be necessary to solve for some crisis caused by a bug, or by an attack causing critical service failure. Such an emergency requires a prompt effective intervention by part of the project maintainers. Most of the concerns though revolves around the fact that these developers are not legally accountable neither they are payed to intervene. This may be a cause of major concerns, but it can be mitigated as we will see in section 9.3.

Majority attacks are intrinsically inevitable. As always we need to ask a question, does a 51% attack causes loss of funds, or service availability disruption? There's mainly two things an entity with 51% of the network hashing power could do. They could prevent transactions of their choice from gaining any confirmations, thus making them invalid, potentially preventing people from sending funds between addresses. They could also reverse transactions they send during the time they are in control (allowing double spend transactions), and they could potentially prevent other miners from finding any blocks for a short period of time. Nevertheless, they couldn't reverse transactions from long ago, create new coins out of thin air (besides through regular coibase), or steal coins from other people's wallets. This kind of attack is a serious concern since it could both deteriorate the FMI continued operations, but also it could cause people transacting with this malign entity to lose their funds. Any attack of this kind would be detectable off-chain – meaning that recognizing this requires coordination through a medium other than the peer to peer network of the software protocol implementation – and contrasted with drastic measures such as a change in Proof of Work, so as to render the attacker's equipment completely ineffective for sustaining its attack.

9.2.2 Risks classification

First of all, we can identify two major sources of risks which are currently mixed together: internal faults and failures and external attacks. While users inability is a risk that lies in the application layer, rather than the protocol's trustware, all of the remaining identified operational risks lie within these two broad categories.

> ***Observation 5****. Based on our observations it is largely more coherent to distinguish the two main sources of risks, internal faults and external attacks. Further categorization can be executed for both of these broad categories. Internal faults include software and incentives faults, and process failures. External attaks can be both "covert" or deceitful somehow, or they can be "overt" or evident.*

BROAD CATEGORIES	SUB-CATEGORIES	EXAMPLES
INTERNAL FAULTS AND FAILURES	Software faults	Bugs, both dormant or newly introduced through updates
	Incentives faults	Unforeseen broken incentives
	Process failures	Broken decentralized decision making process

EXTERNAL ATTACKS	Deceitful attacks	Contentious hardforks
	Evident attacks	51% attack

Now that we have a clearer view of what risks exist, how can they be mitigated?

9.3 Mitigating actions and processes

When analyzing how operational risks hinder permissionless networks viability as FMIs, the major concerns are caused by critical service failures. As seen previously, these failures cause users to lose their funds. Depending on how extensively the failure is widespread, the user base may quickly lose its confidence in the system, causing a panic sell, hence negatively affecting the entire ecosystem, both of the specific blockchain-based network, but even of the broad permissionless networks ecosystem. This is why we mainly focused in analyzing mitigating actions mainly concerning critical, disruptive failures. We will start from our interview with Luke Dashjr.

> ***Interview 1****. Luke Dashjr is one of the most important Bitcoin core contributors. He currently covers the role of Bitcoin Improvement Proposals (BIP) maintainer. The BIP is the process through which any relevant patch changing Bitcoin's codebase at its consensus level shall go through. He is therefore a very recommended candidate for our interview. In fact, most of the mitigating actions involve changing Bitcoin's codebase with the objective to solve for the problem. The objective of this interview was mostly aimed at identifying the best mitigating actions, as well as identifying any other potential operational risk, previously unaccounted for. At first the researcher started with a brief introduction to the scope of the research and to the objective of the interview, then we started with the relevant questions.*

Alessandro: How do you mitigate the effect of a software fault, such as a fault which may cause people to lose their funds?

Luke: The resolution would almost always be a patch correcting the bug, introduced through a softfork. The bug has to have certain conditions under which it is triggered, otherwise it would have been latent and solved through the usual process. So you would implement a softfork that prevents those conditions from ever occurring again.

A: The problem in that case would be that the core developers (especially the project maintainers) should quickly intervene in order to promptly solve for that problem.

L: It would depend a lot on the miners to intervene in that case. Because the miners can enforce the softfork for all nodes that have not updated yet. All that we, as developers, could do is write code.

A: How would you (developers) coordinate with the network and especially miners in order to enforce this change? The reason for this question is that the -systemalert was recently deprecated with Bitcoin Core 0.13.

L: Well in terms of reaching all the miners, we have a list of phone numbers. We call people up and have them enforce the changes in a prompt way.

A: there is no formalized protocol for such emergency situations?

L: There is the announcement mailing list which was created exactly for this purpose. Anyone can subscribe to this mailing list, and only critical alerts will be send through this channel. Therefore, not just miners but also all of the most important economical nodes which have their systems relying on Bitcoin's continued operations, can subscribe and receive emergency alerts. This was the official replacement for the alertsystem.

A: Well then in this case, this practically solves for any other situation of emergency! Any dormant/triggered bug, any update introducing fresh new critical bugs, could be solved in this fashion.

L: No! There is always the risk that the developers could be compromised and try to deply an update that is not good for the network, it doesn't have consensus or it introduces disrupting bugs. In that case the announcement mailing list would work against the purpose it was built for.

A: so that it becomes the attack vector itself! So what mitigation actions do you foresee that could be put in place to solve for that?

L: I think the only mitigation is to educate the community to make its own decisions when it comes to updates.

A: So recapping, miners are the ones who can introduce rule changes to solve for emergency situations, and you need them to be very well educated..

L: ideally. Not all fix changes would need miners to intervene of course. If it's not a real bug fix change the one introduced with an update, but it is rather a malicious attack introducing critical bugs, then probably it would depend on the users to reject the changes.

A: So if jt's really urgent you want the miners on your side, if it's not then it shall be a broad user based decision.

L: Not quite the same situation. It all depends on what the attack really and it would probably be a malicious softfork or hardfork, or just a backdoor even. And in that case you really don't want anyone updating. Because even if just the users update, then they might have the backdoor and can be compromised that way, no matter what the miners do.

A: so you need stronger security guarantees for developer driven attacks. You need to know that close to nobody updated their software.

L: right, you don't want the users to blindly trust the developers.

A: do you think that an off-band coordination like the one that solved for the 0.8 Core update in 2013 would be feasible today with the current level of decentralizations? That time a bug was introduced with 0.8 such *that it disrupted the database of all updated nodes. This required a hardfork and especially the coordination with miners which had to renounce to 6 full hours of their revenues.*

L: I hope so, but with the contention that exists today between miners and developers, it might be hard to coordinate.

A: But in that case it would be in their best interest to revert the transactions and downgrade their software for instance. In fact, people would be losing funds, and if they lose funds they are going to quit the network and dump their stake causing a panic sell. Therefore there is a huge downside on miners stemming out of a non-collaborative behavior.

L: You would hope they would be behaving differently than they are right now. It's hard to say for sure.

A: A best practice was identified to introduce ruleset changes, is that of using softforks, rather than hardforks. What is your take on that?

L: That is pretty much accurate especially to solve for any potential bug, since in emergency situations you want the deployment to be as safe as possible. Though there is a handful of situations when a hardfork might be preferable. It really depends on the kind of update nonetheless. Some updates can only be deployed through hard forks, like confidential transactions, miblewimble.

A: There is also a take that suggests to introduce major rule changes such as SegWit through a softfork, and then when every full node complies to the new rules a hardfork could be performed so as to clean up the codebase thus eliminating the technical debt. What is your take on this?

L: So because of the nature of the blockchain and how its history goes back to its inception, I don't think it would clean up any technical debt, it would probably add even more technical debt.

A: This is because the nodes should keep remembering all the dirty part and maintain exeption rules just to validate that part.

L: Right.

A: Regarding deceitful covert attacks such as the hardfork proposed by the Bitcoin Unlimited group. These kind of attacks might come over and over again, how can their impact on stability be mitigated?

L: the only problem there is the political side where they try to convince everyone, other than that it's just an alt-coin. Again higher education would be necessary for the user base so that they can conscoiuously take their decisions.

A: Regarding overt attacks instead, do you feel that mitigating actions like the PoW change would be a viable path?

L: With a 51% attack, PoW change is probably the only realistic option. This could only be implemented as a hardfork.

A: Would a 51% attack only impact double spends on their own spends plus censoring, or would there be some other impacts on the overall networks?

L: Well such an attack would likely impact also second layers such as Lightning Networks, or decentralized sidechains, like drivechains. This is by design. And again this could be solved through a PoW change.

The result of this interview is pretty interesting as we understood some interesting mechanism behind how emergency situations are dealt with. In particular, for what concerns internal faults and failures:

- Bugs should be solved for through a new patch to be deployed as a softfork,
- Such softforks should then be enforced by miners in a quick and effective way, but miners should proficiently inspect the code.

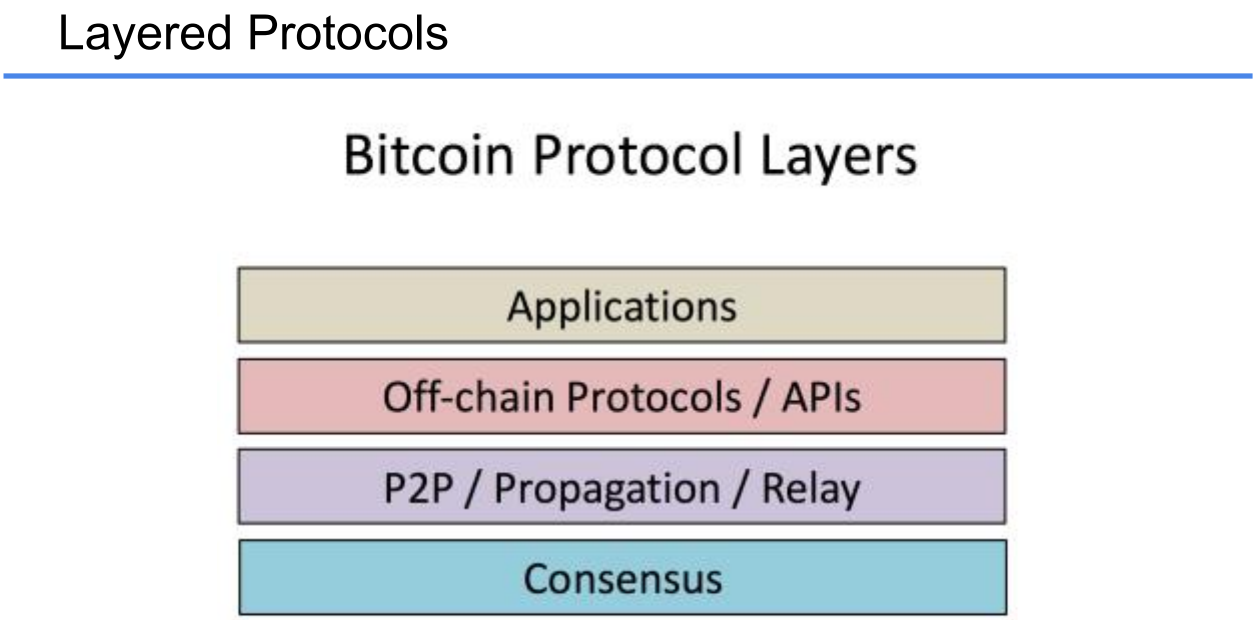
External attack vectors are much more difficult to deal with. They can come as contentious hardforks, 51% attacks, but even as urgent updates proposed by rogue core developers. The attacking scenarios are near infinite and unpredictable. 51% attacks are the most evident attacks, since they can be broadly detected, nonetheless covert deceitful attacks are tricky. In those cases the user base owning full nodes should be more educated and aware of what it is doing.

> ***Testimony*** *7. In order to provide different takes on these same issues we also introduce another testimony by Eric Lombrozo.Eric Lombrozo is a major core developer. The following excerpt was taken during his intervention at Scaling Bitcoin Milan (Lombrozo, 2016).*

[...] When I first started contributing to Bitcoin 2011, I started writing my first bitcoin applications. I started the same way a lot of other people probably do. I started with the RPC build into Bitcoin Core. [...] So this is the first pull request that I tried to submit and as you can see, I was very excited about getting this great multiple wallet support complete. [...] After the obligatory bikeshedding and nitpicking and the rebases and doing that over and over again for several months. Someone said "This is awesome work and I hope it makes it to btc soon". Unfortunately it didn't work that way; Wladimir closed it, which I don't blame him for, it got too out of sync for the project. I'm sure that other people have had this frustration as well. I think people get upset-- I thikn they are right to be upset thhat it's frustrating to contribute to bitcoin. The process does have some serious bottlenecks. Not many people can do the code review, and there's a lot of risk inherent in changing the code.

bottlenecks. ***Not many people can do the code review, and there's a lot of risk inherent in changing the code****.*

I think something that could help fix this to a large extent and ***reduce the problem is using layered architecture****. In the case of the internet, you have a base layer like IPv4/IPv6, which just gets a packet from point A to point B, and then an application layer that interprets data and shows it to the user and has some interaction or whatever. Bitcoin has a similar structure that we can think of. At the base layer, we have consensus, and then as you go further up, the compatibility requirements become less strict. Applications might want to share data, but it's not critical. At the consensus layer, if two nodes disagree, then you end up with mutually incompatible histories and that's something we're trying to avoid.*



Layered Protocols

Bitcoin Protocol Layers

Applications

Off-chain Protocols / APIs

P2P / Propagation / Relay

Consensus

Figure 12 Bitcoin protocol layers, BIP-123

One thing I've been trying to work on is bip123, which tries to make it so that BIP authors when they submit a BIP they categorize their BIPs according to ***layers*** *and* ***makes it easier to sort through them and to prioritize them and be able to see which ones are more critical for review****.*

For the application layer, you might not need any Core developers to review it. Just build an app, great. At the consensus layer, it requires the entire community to agree. In this particular BIP, I separated the BIPs into 4 different categories. You have wallet- like address formats, bip32 you have password-encrypted wallet stuff, tihngs that are necessary to have compatibility between different applications but not necessary for the entire ecosystem to function.

At the RPC level you have interfaces that applications might depend on to interface with the bitcoin network. You can have several different APIs and several different models .

At the peer services layer, you need some level of -- they need to agree on the data format and you can add new messages and you can deprecate old messages.

In the consensus layer, you need everyone to agree, and you must try to keep from making incompatible changes. Consensus rule changes are one of the most difficult challenges. This requires everyone agree. It's really hard to get everyone to agree on something. Engineering always has tradeoffs. There might be some reason why someone doesn't want something. It makes it very difficult. The fact that nobody can change the rules from under you means that you have certain guarantees that your coins have value and that the rules are encoded there so that nobody can change it from under you. This is frustrating to people who want to innovate on the protocol because it makes it difficult to change things. There are mistakes that we can see now, that in hindsight we would have changed the protocol in the first place, but it's difficult to coordinate this change now.

Satoshi made this post in 2010 where he says and I quote, once version 0.01 is released, the core designed is set in stone for the rest of its lifetime. This is basically Satoshi saying that changing things is difficult. He also said that a second incompatible version would be a massive headache for him. Any incompatibilities would cause serious problems for the network. Fortunately, we have made some progress since those days, or start to do those things without those risks. Hardforks are one of the most naive ways to change the consensus rules. If you think transaction formats are more compact or whatever, well guess what, not all clients are going to accept that. Basically you're going to have nodes that don't agree, and you're going to get a hardfork. ***Softforks are the best mechanism we have so far for deploying consensus changes*** *because not everyone needs to upgrade at the same time****. You just need a supermajority of miners to upgrade first, and then the rest of the network can upgrade at their own leisure****. I don't think this is something that Satoshi considered, but right now it's basically the only thing used to deploy consensus changes.*

We need to think about bootstrapping networks. ***We're trying to upgrade the system while it's moving and there's billions of dollars of other people's money on this, and if you screw up then people can lose a lot of money****. This is not something we take lightly. It's very hard to do this. This has been a big problem for being able to innovate at that level. We have the development bottlenecks which have to do a lot with the fact that we have a limited number of code reviewers and they all have to look at the same stuff. No matter how risky the code is, it usually has to change consensus code or something, so that makes it a problem.* ***We need a different modularization where we can have different modules working in loose coordination rather than hard coordination****."*

Lombrozo here confirmed those that are considered to be the best practices for updating Bitcoin's trustware. Nonetheless, Lombrozo' constribution is not limited to that. Indeed, he also proposes a very relevant practice through his BIP 123. Dividing the current codebse into several layers and having contributors making explicit to what layer of the codebase does their update have an impact on. This way the process can become a more specialized, efficient and effective one:

- the software's (trustware) codebase is clearly separated into 4 layers,
- the most fundamental layer, the consensus layer, being the most critical one for the network's continued operations, shall remain largely untouched so as to minimize operational risks,
- this development processes additionally optimizes project maintainers efforts and the result should be a much more effective code review and maintenance, thus largely benefiting the entire network well-being.

CONCLUSIONS AND FUTURE RESEARCH

10 Findings discussion

The operational risks identified up to now in literature are inhomogeneous and largely unclear. We observed on-the-field that this is probably caused by a poor understanding of how trustware (as Antonopoulos called permissionless protocols software implementations) ultimately works. This is why we start All of the operational risks are intertwined with two relevant **idiosyncratic dynamics**, proper of permissionless blockchain-based networks: **decentralized governance** and **ruleset changes through hardforks and softforks**.

The decentralized governance processes of trustware happens on at least two distinct layers, a basic infrastructural layer – where any decision is automatized and enforced by the network of full nodes – and the architects layer – the informal, uncoordinated level of open source software (trustware) development. Any ruleset changes at the infrastructural layer entail the strict cooperation of the actors participating in both layers. If this does not happen, major network disruptions become a highly likely result.

What we subsequently noted was the incoherence of how operational risks have been presented up to now in extant literature. For this reason, we propose a better organized framework for risk categorization, built through a continuous process of knowledge coding, as advised by the grounded theory approach. Through a careful observation of the permissionless ecosystem, our analysis identified **two broad categories of risks**:

BROAD CATEGORIES	SUB-CATEGORIES	EXAMPLES
INTERNAL FAULTS AND FAILURES	Software faults	Bugs, both dormant or newly introduced through updates
	Incentives faults	Unforeseen broken incentives
	Process failures	Broken decentralized decision making process

EXTERNAL ATTACKS	Deceitful attacks	Contentious hardforks
	Evident attacks	51% attack

Table 1 Operational Risks in permissionless networks

The extent to which these risks may hinder a permissionless blockchain-based network viability as a financial market infrastructure largely depends on **how severely** would **these risks impact on the network's continued operations**. In fact, we observe that two types of failures exist:

- The ones causing a service level degradation, they represent a **partial service failure**. Any fault or failure causing the network not to process transactions in a timely manner pertains to this category. Transactions are thus eventually executed correctly but not when they were asked for. Examples of such failures could be misaligned incentives causing miners not to include valid transactions within their blocks for long time extents, or 51% attacks where miners selectively censor certain transactions.
- The ones causing a state consistency failure, they represent **critical service failures**. Any fault or attack that causes users to lose their funds represents a critical failure. In fact, depending on how extensively the failure is widespread, the user base may quickly lose its confidence in the system, causing a panic sell, hence negatively affecting the entire ecosystem, both of the specific blockchain-based network, but even of the broad permissionless networks ecosystem. Examples of these failures includes practically any external attack, but even disruptive bugs in the protocol software implementation.

Ultimately we analyzed what **actions and processes** may be deployed so as **to mitigate the negative effects of operational risks materialization**. The focus was on how to solve for critical service failures, since their impact may cause the entire network to eventually fade into irrelevance.

What emerged from this analysis is that **internal faults and failures** can be addressed and solved for thanks to:

- a set of previously tested practices to be used when critical failures materialize
    - Bugs should be solved for through a new patch to be deployed as softforks,
    - Such softforks should then be enforced by miners in a quick and effective way, but miners should proficiently inspect the code;

- the formalization of a new development processes to contribute to permissionless networks codebase
    - the software's (trustware) codebase is clearly separated into layers,
    - the most fundamental layer, the consensus layer, being the most critical one for the network's continued operations, shall remain largely untouched so as to minimize operational risks,
    - this development processes additionally optimizes project maintainers efforts and the result should be a much more effective code review and maintenance, thus largely benefiting the entire network well-being.

Operational risks pertaining to the internal faults and failures category, can be effectively mitigated thanks to the alignment of incentives between first and second layer governance. In fact, when these risks materialize, the architects generate uncontentious solutions while the miners are likely to cooperate, since it is in their best interest to do so.

External attacks risk mitigation actions and processes are somewhat less comforting instead. Besides from the overt attack of a 51% hashing-power takeover – which can be easily solved for through a change in the Proof of Work algorithm – any other attack should be analyzed in a case by case fashion. Unfortunately attack vectors directed at disrupting permissionless networks at a system-wide level can get overly complicated. Currently a good strategy to prevent attacks to permissionless networks, suggest that every user runs his/her own full node and maintains status quo over any proposed rule change, which is exactly what is happening in Bitcoin, thus confirming the analysis.

10.1 Validity and Reliability

Up to now major risks already materialized in permissionless networks. The best practices for risk mitigation have proven to be solid so far. As a result permissionless networks may possibly be viable Financial Market Infrastructures.

In Bitcoin's case the network experienced and survived both internal failures – such as disruptive bugs the latest one being in 2013 when a 6 hours' reorg took place – and external attacks – deceitful attacks like Bitcoin Unlimited, but also miners hashing power concentration in Bitcoin's early days.

Of course, just as for any other system, past performances are not indicative of future results. Same is true for permissionless systems, the boundary conditions that allowed permissionless networks to flourish up to this point are not a guarantee that they will survive future failures and attacks under different conditions. Nonetheless decentralized networks could provide even stronger demonstration of their intrinsic anti-fragility nature in the future, when the value at stake will (likely) be even higher, and failures and attacks may either cause these networks to become stronger or to perish.

This being said, a number of biases need to be made explicit regarding both the sample we selected and why they may be overly optimistic regarding the permissionless ecosystem:

- The mere ownership effect, our sample is biased because practically all of the testimonies, the interviewee and most of the people the researcher interacted with, possess Bitcoins or other cryptocurrencies. This cognitive bias is different from the irrational escalation of commitment (caused by sunk costs), because this latter assumes that clear evidences demonstrating higher costs from sticking with a certain choice have already emerged – which is not the case yet for permissionless ecosystems.
- The pseudocertainty effect united with the strong ideology pervading permissionless ecosystems may be a mental barrier for those operating within, as they may turn their backs to other more rational choices.
- The pro-innovation bias is also likely to be present in our sample. The innovation's "champion" has such strong bias in favor of the innovation, that he/she may not see its limitations or weaknesses and continues to promote it nonetheless.
- Similar to the previous there is also the appeal to novelty fallacy. The fallacy may take two forms: overestimating the new and modern, prematurely and without investigation assuming it to be best-case, or underestimating status quo, prematurely and without investigation assuming it to be worst-case. This may actually be a recurring attitude also for incumbents, which may be jumping on the bandwagon just like with the dotcom bubble.

On the other hand, it is worth noticing that biases also strongly affect the detractors of permissionless systems:

- Those refusing the whole idea behind Bitcoin and permissionless networks, may be affected by dallo zero-risk bias and loss aversion. Those who are highly invested in a particular system might be so averted to losses, that they wouldn't risk even a fraction of their assets in a new, unknown unstable system.
- Skepticals may also be affected by the status quo bias. In fact, external observers may easily perceive permissionless networks are a threat to everything they have ever worked for.

Finally I had a series of biases as a researcher:

- I myself may have been affected by the blind spot bias. In fact, I may not realize all of my biases,
- I was affected definitely cognitive dissonance. In fact, I needed to either justify what I previously believed, or at least understand if it was really worth looking at something I wouldn't have related to before.
- For the above reason, I consciously refused to be fooled by the cognitive dissonance bias, which in turn may have led me to the effort justification bias, because I'm skeptic by nature,
- The echo chamber may have been a serious bias. Nonetheless, I knew was conscious since the very beginning that this may have happened, hence I have always been skeptic and I continue to be skeptic, which is sort of a curse,

11 Conclusions

Through this research we investigated many different concepts. Firstly, we understood what a blockchain is. Basically in a distributed state-system, all the parties accessing the system and operating changes on the system, need a secure and reliable tool that allows them:

- to verify that the current state, and all the steps that brought to it, is correct and compliant with a set of predefined consistency rules;
- to be sure that they are accessing the most recent system state,
- to be sure that every other actor operating on the network is accessing exactly the same history of state changes, and that they are synchronized on the most recent state (log-uniqueness),

- to be sure that their, as well as other parties, changes on the system will not be revertible in a subsequent moment in time.

Thus, a blockchain is an auditable, chronologically ordered set of authorized state changes. Fundamentally it is supposed to be an incorruptible source of truth, that allows competing and non-mutually trusting parties to agree on the history of a system.

Depending on whether the system is dynamical – a permissionless systems, where any entity can start to participate and act on the system state – or static – a permissioned system with a set of pre-authorized parties with known identity – both the log-uniqueness assumption and the security characteristics of the system as a whole are affected.

Permissioned systems are developed by private companies. Thus, they can be given a well-defined legal basis and governance & risk-governance structure, allowing them to be FMIs authorized by the relevant regulators. The way they will be applied on financial markets and what type of infrastructure they could be is still unclear. They are currently proposed to be used both as a new efficient payment system, an extra rail for inter-bank settlement, but also as decentral securities depositories, which should ease the post-trade security lifecycle, by automating the processes and making them more transparent to the regulators. The problem of permissioned networks is that they are still in their prototypal phase, therefore they cannot be investigated much further, yet.

Permissionless systems are undergoing under stress testing in strongly adversarial conditions. Slowly they are trying to prove that FMIs could function even without the need for trusted intermediaries. Nonetheless – being unregulated systems and lacking any form of structured governance – they present some serious operational risks which shall not be underestimated by whomever operates with cryptocurrencies.

From the literature review what emerged was that permissionless networks are still largely misunderstood and we don't quite yet fully grasp the operational risks they are subject to. This is why we collected qualitative data through observations, documents, testimonies and an interview, and analyzed through entographic analysis and grounded theory.

What emerged is that most of the misunderstandings when it comes to permissionless networks revolve around the convoluted nature of their decision making processes. They happen on separate layers, a basic automatized layer enforced by full validating nodes, and the architects' layer on top, where development decisions are taken regarding how the trustware shall evolve, or be maintained. Any decision taken by the second layer of developers cannot simply be enforced on the entire network, since it is the first layer which ultimately takes decisions. These upgrades can eventually be deployed both as hardforks or softforks, but both the security and effectiveness trade-offs are yet to be researched upon.

The second relevant series of findings regards the incoherence of how operational risks have been presented up to now in extant literature. For this reason, we propose a better organized framework for risk categorization, built through a continuous process of knowledge coding, as advised by the grounded theory approach. The two relevant dimensions for risk categorization concern the risks severity and their nature. On one hand there are partial or critical service failure, depending on the impact that each risk has if it materializes. On the other hand these risks have a double nature, either they are faults and failures internal to the existing codebase or rules and processes, or they can arise as attacks coming from external or rogue actors.

Finally, we identified the most relevant risk mitigation actions. What emerged is that Internal faults and failures can be mitigated through processes and practices which are becoming quite standard. External attacks on the other hand, are not conclusively mitigated through any standard procedure, rather a case by case mitigation strategy shall be adopted.

Our conclusion is that permissionless networks are still largely unexplored and delicate. Using them as Financial Market Infrastructures is already a viable choice. Of course it is a risky infrastructure, therefore each individual shall risk a portion of assets that s/he can afford to lose. Ultimately permissionless networks truly represent a paradigm shift under many perspectives and they are revolutionary in every single field of knowledge they embrace: game theory, cryptography, distributed computer science and economic and monetary theory.

12 Future Research

The blockchain ecosystem is very rapidly evolving and many open questions remain. First of all, the **benefits of using a blockchain-based system in permissioned environments** still need to be demonstrated. In this regard, a DTCC announcement (Castillo, 2017) from January may indicate that indeed they allow to increase operational efficiency. Nonetheless, this application will not start to be used before than one year, only then we will be able to see tangible results.

Concerning **permissionless blockchains** instead further research shall be conduced on the **external attack vectors** and their mitigation methods. Also we realized how complex and delicate are the **updates to the ruleset**, therefore much more research will have to be done on **how to deploy them through hardforks and softforks**. Finally, a **comparison of permissionless networks** from **functional** perspective but also **security perspective**, because this is trustware managing money, the two aspects are strictly related

Ultimately, a last word on permissionless networks. Even though they are not going to be broadly used as FMIs any time soon, they still hold a huge potential. The fact that they are open-source and open-innovation systems, joint with the fact that their developers can directly benefit (through cryptocurrencies) from all improvements and advancements of such networks, makes them fascinating object with great potential for disruptive innovation. An example of future innovations that might be brought to permissionless blockchains are the tree-chains, ideated by Peter Todd. The idea is to have many different blockchains, each operating on a different state-system machine, but all interoperable with, and secured on, a large central blockchain (a sort of tree log) providing strength and security to all attached blockchain branches. Permissionless blockchains could therefore serve as a public infrastructure, just like a public good, thus allowing for a wider set of applications that go beyond simple cryptocurrencies. Nonetheless, this day still lies far ahead.

Bibliography

Textbooks & Books

ANTONOPOULOS, A. M. 2014. *Mastering Bitcoin: unlocking digital cryptocurrencies*, " O'Reilly Media, Inc.".

FERRARINI, G. A. & SAGUATO, P. 2014. Regulating financial market infrastructures.

FRANCO, P. 2014. *Understanding Bitcoin: Cryptography, engineering and economics*, John Wiley & Sons.

PAAR, C. & PELZL, J. 2009. *Understanding cryptography: a textbook for students and practitioners*, Springer Science & Business Media.

TAPSCOTT, D. & TAPSCOTT, A. 2016. *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*, Penguin.

WATTENHOFER, R. 2016. *The Science of the Blockchain*, CreateSpace Independent Publishing Platform by Amazon, January.

White Papers

2015. Smart Contracts on Bitcoin Blockchain. BitFury Group.

2016a. The Digital Asset Platform. Digital Asset Holding.

2016b. Digital Assets on Public Blockchains. BitFury Group.

2016e. On Blockchain Auditability. BitFury Group.

BACK, A. 2002. Hashcash-a denial of service counter-measure.

BROWN, R. 2016. Introducing R3 Corda™: A Distributed Ledger Designed for Financial Services. *R3 Cev*.

BUTERIN, V. 2013. Ethereum white paper.

CHAUM, D., FIAT, A. & NAOR, M. Untraceable electronic cash. Proceedings on Advances in cryptology, 1990. Springer-Verlag New York, Inc., 319-327.

DAI, W. 1998. B-Money.

GARZIK, J. 2015a. Public versus Private Blockchains. *Part 1: Permissioned Blockchains.* BitFury Group.

HOPWOOD, D., BOWE, S., HORNBY, T. & WILCOX, N. 2016. Zcash Protocol Specification.

NAKAMOTO, S. 2008. Bitcoin: A peer-to-peer electronic cash system.

POON, J. & DRYJA, T. 2015. The bitcoin lightning network: Scalable off-chain instant payments.

SWANSON, T. 2015. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. R3, Tech. Rep., 6 Apr. 2015.[Online].

SZABO, N. 1997. Formalizing and securing relationships on public networks. *First Monday,* 2.

SZABO, N. 2008. Bit gold. *Website/Blog*.

Government and International Organizations' Reports & Papers

2016d. Federal Reserve Policy on Payment System Risk. *In:* RESERVE, F. (ed.). U.S. Federal Reserve.

KOKKOLA, T. 2010. The Payment System: payments, securities and derivatives, and the role of the eurosystem. Frankfurt: European Central Bank.

MCWATERS, R. J. 2016. The future of financial infrastructure - An ambitious look at how blockchain can reshape financial services. WEF.

MILLS, D., WANG, K., MALONE, B., RAVI, A., MARQUARDT, J., CHEN, C., BADEV, A., BREZINSKI, T., FAHY, L., LIAO, K., KARGENIAN, V., ELLITHORPE, M., NG, W. & BAIRD, M. 2016. Distributed Ledger Technology in Payments, Clearing, and Settlement. *Finance and Economics Discussion Series,* 2016.

PINNA, A. & RUTTENBERG, W. 2016. Distributed ledger technologies in securities post-trading Revolution or evolution? *In:* ECB (ed.).

RUSSO, D. & MOONEY, J. 2012. Principles for financial market infrastructures. *In:* COMMITTEE ON PAYMENT AND SETTLEMENT SYSTEMS, T. C. O. T. I. O. O. S. C. (ed.). Bank for International Settlements, IOSCO.

Academic Literature, Published & Unpublished

ABD-EL-MALEK, M., GANGER, G. R., GOODSON, G. R., REITER, M. K. & WYLIE, J. J. Fault-scalable Byzantine fault-tolerant services. ACM SIGOPS Operating Systems Review, 2005. ACM, 59-74.

AUBLIN, P.-L., MOKHTAR, S. B. & QUÉMA, V. Rbft: Redundant byzantine fault tolerance. Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on, 2013. IEEE, 297-306.

AVIZIENIS, A., LAPRIE, J.-C., RANDELL, B. & LANDWEHR, C. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing,* 1**,** 11-33.

CASTRO, M. & LISKOV, B. Practical Byzantine fault tolerance. OSDI, 1999. 173-186.

CLEMENT, A., WONG, E. L., ALVISI, L., DAHLIN, M. & MARCHETTI, M. Making Byzantine Fault Tolerant Systems Tolerate Byzantine Faults. NSDI, 2009. 153-168.

DE FILIPPI, P. & LOVELUCK, B. 2016. The Invisible Politics of Bitcoin: Governance Crisis of a Decentralized Infrastructure.

FOX, A. & BREWER, E. A. Harvest, yield, and scalable tolerant systems. Hot Topics in Operating Systems, 1999. Proceedings of the Seventh Workshop on, 1999. IEEE, 174-178.

HUFTY, M. 2011. Investigating policy processes: the governance analytical framework (GAF).

KOTLA, R., ALVISI, L., DAHLIN, M., CLEMENT, A. & WONG, E. Zyzzyva: speculative byzantine fault tolerance. ACM SIGOPS Operating Systems Review, 2007. ACM, 45-58.

KRUGER, J. & DUNNING, D. 1999. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology,* 77**,** 1121.

LAMPORT, L., SHOSTAK, R. & PEASE, M. 1982. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS),* 4**,** 382-401.

MAINELLI, M. & MILNE, A. 2016. The impact and potential of blockchainon the securities transaction life-cycle. SWIFT Institute.

PEASE, M., SHOSTAK, R. & LAMPORT, L. 1980. Reaching agreement in the presence of faults. *Journal of the ACM (JACM),* 27**,** 228-234.

POELSTRA, A. 2015. On Stake and Consensus. Blockstream.

SELGIN, G. 2015. Synthetic commodity money. *Journal of Financial Stability,* 17**,** 92-99.

TAYLOR, M. B. Bitcoin and the age of bespoke silicon. Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, 2013. IEEE Press, 16.

TSAI, W.-T., BLOWER, R., ZHU, Y. & YU, L. 2016. A System View of Financial Blockchains. *2016 IEEE Symposium on Service-Oriented System Engineering.*

WALCH, A. 2015. The Bitcoin blockchain as Financial Market Infrastructure: a consideration of Operational Risk. *NYU Journal of Legislation & Public Policy,* 18**,** 837-None.

Private Companies Reports

2015. Smart Contracts on Bitcoin Blockchain. BitFury Group.

2016c. Embracing Disruption: tapping the potential of distributed ledgersto improve the post-trade landscape. DTCC.

2016e. On Blockchain Auditability. BitFury Group.

BELINKY, M., RENNICK, E. & VEITCH, A. 2015. The FinTech 2.0 report: Rebooting Financial Services. *In:* ANTHEMIS (ed.). Santander Innoventures & Oliver Wyman.

BROWN, R. 2016. Introducing R3 Corda™: A Distributed Ledger Designed for Financial Services. *R3 Cev*.

EHSANI, F. 2016. The Advent of Crypto Banking - A New Paradigm for Central and Commercial Banking. *In:* FOUNDERY (ed.). FirstRand Bank Limited.

GARZIK, J. 2015a. Public versus Private Blockchains. *Part 1: Permissioned Blockchains.* BitFury Group.

MAINELLI, M. & MILNE, A. 2016. The impact and potential of blockchainon the securities transaction life-cycle. SWIFT Institute.

MCWATERS, R. J. 2016. The future of financial infrastructure - An ambitious look at how blockchain can reshape financial services. WEF.

QUINLAN, B. & KWAN, Y. 2016. From KYC to KYT

blockchain's emerging role in payments system. Quinlan & Associates.

RIZZO, P. & MILES, B. 2016. State of Blockchain 2016 Q3. *In:* COINDESK (ed.) *State of Blockchain.* New York.

SYMONS, P., PEETERS, I., MEAD, J. & KINGSLEY, B. 2016. Blockchain settlement - Regulation, innovation and application. Euroclear, Slaughter and May.

TRINDER, D. 2015. Deutsche Bank's response to ESMA's call for evidence on Virtual currencies and

Distributed ledgers. Deutsche Bank AG.

VAN DE VELDE, J., SCOTT, A., SHEPHERD, B. & ALLCHIN, C. 2016. Blockchain in Capital Markets. *The prize and the Journey.* Euroclear, Oliver Wyman.

Uncategorized References

GARZIK, J. 2015b. Public versus Private Blockchains. *Part 2: Permissionless Blockchains.* BitFury Group.

Websites & Articles

Acheson, N. (2017). *Bitcoin Price Index - Real-time Bitcoin Price Charts*. [online] CoinDesk. Available at: http://www.coindesk.com/price/ [Accessed 9 Jan. 2017].

Antonopoulos, A. (2017). *Hardware, Software, Trustware*. [online] YouTube. Available at: https://www.youtube.com/watch?v=Etyjc1JdmFU [Accessed 6 Apr. 2017].

Armstrong, B. (2016). *Bitcoin's Elegant Upgrade Mechanism: Miner Voting.* [online] The Coinbase Blog. Available at: https://medium.com/@barmstrong/bitcoin-s-elegant-upgrade-mechanism-miner-voting-66faa35d27af#.59a0e5kj4 [Accessed 4 Apr. 2017].

Arnold, M. (2016). *Financial industry faces extreme disruption in payments*. [online] Ft.com. Available at: https://www.ft.com/content/1b82a0e6-4f67-11e6-8172-e39ecd3b86fc [Accessed 26 Nov. 2016].

BigchainDB. (2016). *BigchainDB, The scalable blockchain database.*. [online] Available at: https://www.bigchaindb.com/ [Accessed 26 Dec. 2016].

Bitcoin.org. (2016). *Developer Guide - Bitcoin*. [online] Available at: https://bitcoin.org/en/developer-guide#block-chain [Accessed 17 Dec. 2016].

Blockchain.info. (2017). *Estimated Transaction Value*. [online] Available at: https://blockchain.info/charts/estimated-transaction-volume?daysAverageString=7 [Accessed 9 Jan. 2017].

btcdrak, (2015). *Contributing to bitcoin*. [online] GitHub. Available at: https://github.com/bitcoin/bitcoin/blob/master/CONTRIBUTING.md [Accessed 4 Apr. 2017].

Buterin, V. (2017). *Hard Forks, Soft Forks, Defaults and Coercion*. [online] Vitalik.ca. Available at: http://vitalik.ca/general/2017/03/14/forks_and_markets.html [Accessed 5 Apr. 2017].

Castillo, M. (2016a). *For Blockchain Startups, Switzerland's 'Crypto Valley' is No New York - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/blockchain-innovation-switzerland-crypto-valley-new-york/ [Accessed 3 Jan. 2017].

Castillo, M. (2016b). *Korea's Central Bank Considers 'Supernode' for Blockchain Oversight - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/koreas-central-bank-considers-supernode-for-blockchain-oversight/ [Accessed 3 Jan. 2017].

Castillo, M. (2017). *$11 Trillion Bet: DTCC to Clear Derivatives With Blockchain Tech - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/11-trillion-bet-dtcc-clear-derivatives-blockchain-tech/ [Accessed 9 Jan. 2017].

Creativecommons.org. (2016). *Creative Commons — Attribution-ShareAlike 4.0 International— CC BY-SA 4.0*. [online] Available at: https://creativecommons.org/licenses/by-sa/4.0/ [Accessed 18 Nov. 2016].

DeRose, C. (2016). *Why Blockchain Immutability is a Perpetual Motion Claim*. [online] CoinDesk. Available at: http://www.coindesk.com/immutability-extraordinary-goals-blockchain-industry/ [Accessed 10 Dec. 2016].

De Meijer, C. (2017). *Blockchain and Central banks: a Tour de Table Part 1*. [online] Finextra. Available at: https://www.finextra.com/blogposting/13507/blockchain-and-central-banks-a-tour-de-table-part-1 [Accessed 3 Jan. 2017].

En.wikipedia.org. (2016a). *Blockchain (database)*. [online] Available at: https://en.wikipedia.org/wiki/Blockchain_(database) [Accessed 26 Nov. 2016].

En.wikipedia.org. (2016b). *Distributed computing*. [online] Available at: https://en.wikipedia.org/wiki/Distributed_computing [Accessed 26 Nov. 2016].

En.wikipedia.org. (2016c). *Node (computer science)*. [online] Available at: https://en.wikipedia.org/wiki/Node_(computer_science) [Accessed 27 Nov. 2016].

En.wikipedia.org. (2016d). *Transaction log*. [online] Available at: https://en.wikipedia.org/wiki/Transaction_log [Accessed 30 Nov. 2016].

Evans, P., Aré, L., Forth, P. and Harlé, N. (2016). *BCG - Thinking Outside the Blocks*. [online] http://blockchain.bcg.com. Available at: https://www.bcg.com/blockchain/thinking-outside-the-blocks.html [Accessed 3 Jan. 2017].

European Central Bank. (2017). *Instant payments*. [online] Available at: https://www.ecb.europa.eu/paym/retpaym/instant/html/index.en.html [Accessed 25 Mar. 2017].

Finkle, J. (2016). *Bangladesh Bank hackers compromised SWIFT software, warning issued*. [online] Reuters. Available at: http://www.reuters.com/article/us-usa-nyfed-bangladesh-malware-exclusiv-idUSKCN0XM0DR [Accessed 3 Jan. 2017].

Hayase, N. (2016). *Bitfinex Heist Rings the Alarm of Bitcoin Centralization - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/bitfinex-bitcoin-alarm-centralization/ [Accessed 3 Jan. 2017].

Hertig, A. (2016). *Ethereum Forks But Blockchain Attacks Keep On Coming - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/ethereum-forks-blockchain-attacks-keep-coming/ [Accessed 3 Jan. 2017].

Lombrozo, E. (2016). *Build Scale Operate*. Available at: https://scalingbitcoin.org/transcript/milan2016/build-scale-operate [Accessed 10 Apr. 2017].

JPMorgan. (2016). *Quorum, J.P. Morgan*. [online] Available at: https://www.jpmorgan.com/country/US/EN/Quorum [Accessed 26 Dec. 2016].

Nolan, T. (2014). *Atomic Cross Chain Transfers*. [online] GitHub. Available at: https://github.com/TierNolan/bips/blob/bip4x/bip-atom.mediawiki [Accessed 27 Mar. 2017].

Ou, E., Hunt, A., Winkler, M., El-Erian, M. and Lake, E. (2016). *What Happened to the Financial Blockchain Revolution*. [online] Bloomberg View. Available at: https://www.bloomberg.com/view/articles/2016-12-08/what-happened-to-the-financial-blockchain-revolution [Accessed 12 Dec. 2016].

Quoteinvestigator.com. (2015). *Suppose You Call a Sheep's Tail a Leg, How Many Legs Will the Sheep Have?*. [online] Available at: http://quoteinvestigator.com/2015/11/15/legs/ [Accessed 7 Jan. 2017].

Pace, A. (2017). *Guest Post: Chain Splits and Resolutions*. [online] Bitcoin Magazine. Available at: https://bitcoinmagazine.com/articles/guest-post-chain-splits-and-resolutions/ [Accessed 6 Apr. 2017].

Pair, S. (2017). *The Bitcoin Fee Market*. [online] Medium. Available at: https://medium.com/@spair/the-bitcoin-fee-market-4df1857d12b7#.lp41u6b9y [Accessed 26 Mar. 2017].

Prwire.com.au. (2016). *Press Release: Gartner: Blockchain and Connected Home are almost at the peak of the Hype Cycle*. [online] Available at: http://prwire.com.au/pr/62010/gartner-blockchain-and-connected-home-are-almost-at-the-peak-of-the-hype-cycle [Accessed 26 Nov. 2016].

Reuters UK. (2016). *Exclusive: Blockchain platform developed by banks to be open-source.* [online] Available at: http://uk.reuters.com/article/us-banks-blockchain-r3-exclusive-idUKKCN12K17E [Accessed 26 Nov. 2016].

Nakamoto, S. (2010). *Satoshi Nakamoto Institute*. [online] Satoshi.nakamotoinstitute.org. Available at: http://satoshi.nakamotoinstitute.org/posts/bitcointalk/126/ [Accessed 4 Apr. 2017].

Shin, L. (2016). *Central Banks Explore Blockchains: Why Digital Dollars, Pounds Or Yuan Could Be A Reality In 5 Years*. [online] Forbes.com. Available at: http://www.forbes.com/sites/laurashin/2016/10/12/central-banks-explore-blockchains-why-digital-dollars-pounds-or-yuan-could-be-a-reality-in-5-years/#54ff6d6576d8 [Accessed 3 Jan. 2017].

Sirer, E. (2016). *Time for Bitcoin Users to Reclaim Their Voice*. [online] Hacking Distributed. Available at: http://hackingdistributed.com/2016/01/03/time-for-bitcoin-user-voice/ [Accessed 4 Apr. 2017].

Tapscott, D. and Tapscott, A. (2016). *The Tapscotts on Blockchain in 2016 and What's Next - CoinDesk*. [online] CoinDesk. Available at: http://www.coindesk.com/the-tapscotts-on-blockchain-2016-and-whats-next/ [Accessed 15 Dec. 2016].

The Economist. (2015). *The trust machine*. [online] Available at: http://www.economist.com/news/leaders/21677198-technology-behind-bitcoin-could-transform-how-economy-works-trust-machine [Accessed 7 Jan. 2017].

Todd, p. (2016). *Preventing Consensus Fraud with Commitments and Single-Use-Seals*. [online] Petertodd.org. Available at: https://petertodd.org/2016/commitments-and-single-use-seals [Accessed 29 Dec. 2016].

Todd, P. (2016). *Soft Forks Are Safer Than Hard Forks*. [online] Petertodd.org. Available at: https://petertodd.org/2016/soft-forks-are-safer-than-hard-forks [Accessed 6 Apr. 2017].

Torpey, K. (2016). *The DAO Disaster Illustrates Differing Philosophies in Bitcoin and Ethereum*. [online] CoinGecko. Available at: https://www.coingecko.com/buzz/dao-disaster-differing-philosophies-bitcoin-ethereum?locale=en [Accessed 4 Apr. 2017].

Wild, J. (2016). *Central banks explore blockchain to create digital currencies*. [online] Ft.com. Available at: https://www.ft.com/content/f15d3ab6-750d-11e6-bf48-b372cdb1043a [Accessed 15 Dec. 2016].

Williams-Grut, O. (2015). *Santander is experimenting with bitcoin and close to investing in a blockchain startup*. [online] Business Insider. Available at: http://uk.businessinsider.com/santander-has-20-25-use-cases-for-bitcoins-blockchain-technology-everyday-banking-2015-6 [Accessed 3 Jan. 2017].

Wuille, P. (2015). *[bitcoin-dev] On the security of softforks*. [online] Lists.linuxfoundation.org. Available at: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-December/012014.html [Accessed 6 Apr. 2017].