DISS. ETH NO. -

THE IMPACT OF CRYPTOCURRENCIES ON THE INTERNET OF THINGS INSIGHTS FROM PROTOTYPES

A dissertation submitted to
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by
DOMINIC WÖRNER
Dipl.-Phys., Ruprecht-Karls-Universität Heidelberg
born on 06.02.1986
citizen of
Germany

accepted on the recommendation of
Prof. Dr. Elgar Fleisch, examiner
Prof. Dr. Frédéric Thiesse, co-examiner
Prof. Dr. Felix Wortmann, co-examiner

2017

ABSTRACT

The vision of an Internet of Things (IOT), in which every physical object can become part of the Internet is almost 25 years old. Only during the last years, due to the interplay of recent technologies like cloud computing, the unprecedented scale of the smart phone supply chain, and the global expansion of communication infrastructure, computing and networking has become so cheap and convenient that the number of connected objects is increasing rapidly. In 2016, there are more connected objects than humans on earth, and the number of connected devices is expected to reach 20 to 30 billion until 2020. Novel business and financing models for connected devices need a new a type of value exchange infrastructure that scales with the Internet itself. While information can travel bit by bit with the speed of light between humans, machines and across borders, the transfer of value has always been cumbersome. A new type of digital currencies may be able to change that. Instead of trusted third parties, cryptocurrencies are based on public peer-to-peer networks, cryptography and mechanism design. Within a few years Bitcoin has risen from funny Internet money to a global currency with a market capitalization of more than $10 billion. But Bitcoin is more than a currency. It is programmable money and a platform for permissionless innovation. Thus, a rich start-up ecosystem with a combined investment of more than $1 billion has emerged. A global community of developers is constantly improving Bitcoin itself, and is building new platforms based on the underlying technology, the blockchain. Hundreds of alternative cryptocurrencies have emerged and large corporations have formed consortia to utilize the technology in a permissioned setting.

Aside Bitcoin, the most prominent cryptocurrency is Ethereum which aims to build a global permissionless trusted computing platform with an integrated economy. On these platforms machines are first class citizens. This allows to rethink the capabilities of connected devices and their role in a global economy. One of the promising IOT business models is Sensing-as-a-Service (S^2aaS) in which a local sensing unit becomes a globally accessible resource. This thesis identifies and discusses a number of characteristics of cryptocurrencies that uniquely suit as a basis for a global S^2aaS infrastructure. This thesis further presents a first prototype based on Bitcoin to illustrate the concept. The major issues are scalability, latency and the impracticality of direct micropayments. The second prototype leverages Bitcoin's programmability to implement self-enforcing contracts, so called smart contracts, enabling mediated unidirectional micropayment channels – a means for low-latency low-trust micropayments based on a hub-and-spoke architecture. The sensing client in this prototype

is implemented as a smartphone application. Thus, providing the basis for a global mobile crowdsensing application.

Cryptocurrencies enable connected devices to autonomously handle money. In combination with smart contracts and the related concept of smart property, connected devices are able to participate in various low-trust economic interactions, both as subject and as object. The emerging concept of *economic devices*, is illustrated with a prototype of an Ethereum-enabled public display, which provides the service of showing user-selected content in exchange for cryptocurrency payments. Payments are handled by a smart contract, which automatically distributes the revenue to a fluid set of global investors. Thus, this example illustrates novel financing and ownership models for connected productive assets, which are of particular importance for emerging economies where traditional financial and judicial systems are underdeveloped.

ZUSAMMENFASSUNG

Die Vision eines Internets der Dinge, in dem jedes physische Objekt Teil des Internets werden kann, ist beinahe 25 Jahre alt. Allerdings hat erst in den letzten Jahren das Zusammenspiel aus der Reife neuer Technologien wie Cloud-Computing, der beispiellosen Grösse der Smartphone-Lieferkette und der globalen Ausbreitung von Kommunikationsinfrastruktur dazu geführt, dass Datenverarbeitung und Vernetzung so einfach und günstig wurde, dass die Anzahl vernetzter Objekte drastisch ansteigt. Schätzungen zu Folge gibt es seit dem Jahr 2016 mehr vernetzte Objekte als Menschen auf der Welt. Bis zum Jahr 2020 soll die Anzahl vernetzter Objekte gar auf etwa 25 bis 30 Milliarden ansteigen. Neuartige Geschäfts- und Finanzierungsmodelle für vernetzte Geräte benötigen eine Werttransferinfrastruktur, die mit dem Internet selbst skaliert. Während Information Bit für Bit mit Lichtgeschwindigkeit zwischen Menschen, Maschinen und über Grenzen hinweg reisen kann, war der Transfer von Wert immer mit Schwierigkeiten verbunden. Eine neue Art von digitalen Währungen könnte dies endgültig ändern. Kryptowährungen basieren auf offenen Peer-to-Peer Netzwerken, Kryptographie und Spieletheorie, und ersetzen damit die Notwendigkeit von Institutionen oder anderen vertrauenswürdigen Dritten. Bitcoin, die erste Kryptowährung, ist innerhalb weniger Jahre zu einer globalen Währung mit einer Marktkapitalisierung von mehr als 10 Milliarden USD gewachsen. Aber Bitcoin ist mehr als eine gewöhnliche Währung. Bitcoin ist programmierbares Geld und eine offene Plattform für Innovationen. Aus diesem Grund ist Bitcoin nicht nur die Basis für ein reichhaltiges Ökosystem an Start-up Unternehmen mit einem Gesamtinvestment von mehr als 1 Milliarde USD, sondern eine globale Gemeinschaft an Entwicklern arbeitet ständig daran Bitcoin zu verbessern und entwickelt neue Plattformen auf Basis der zugrundeliegenden Technologie, die *Blockchain* genannt wird. Dabei sind hunderte alternativer Kryptowährungen entstanden und Konzerne bilden Konsortien, um die Technologie in einem kontrollierten, nicht öffentlichen Umfeld einzusetzen.

Die weitverbreitetste Kryptowährung neben Bitcoin ist Ethereum. Ethereum stellt eine Generalisierung des Blockchainkonzepts dar und hat das Ziele eine globale, offene, Trusted-Computing-Plattform mit einer eingebauten Ökonomie zu erschaffen. Innerhalb dieser offenen Plattformen, die auf Kryptowährungen basieren, sind Maschinen Teilnehmer erster Klasse. Diese Grundlage erlaubt es das Potential vernetzter Entitäten und deren Rolle in der globalen Wirtschaft zu überdenken. Eines der vielversprechenden Geschäftsmodelle im Internet der Dinge ist Sensing-as-a Service (S^2aaS), bei dem eine lokale Sensoreinheit zu einer weltweit verfügbare Messeinrichtung wird. Diese Dissertation identifiziert und diskutiert Eigenschaften von Kryp-

towährungen, die als einzigartige Basis für eine globale S²aaS-Infrastruktur dienen. Dazu wird eine erste prototypische Implementierung dieses Konzepts vorgestellt. Die Hauptschwierigkeiten, die sich bei der Umsetzung dieses Konzepts zeigen, sind Skalierbarkeit, Latenz und die praktische Unmöglichkeit direkte Mikrozahlungen durchzuführen. Um diese Hauptschwierigkeiten anzugehen, wird die Programmierbarkeit von Bitcoin genutzt, um eine Art von selbst-vollstreckender Verträge, sogenannte *Smart Contracts*, zu entwickeln. Diese ermöglichen es vermittelbare Einwegzahlungskanäle zu erschaffen, die sichere Mikrozahlungen mit niedriger Latenz zwischen vielen Parteien mittels einer *Hub and Spoke*-Architektur erlauben. Der Sensor-Client, in der dazugehörigen prototypischen Implementierung, ist eine Smartphone Applikation und damit die Basis für eine globale *Crowdsensing*-Applikation.

Es wird gezeigt, dass Kryptowährungen vernetzten Geräten den autonomen Umgang mit Geld über Distanz ermöglichen. Smart Contracts und das verwandte Konzept von *Smart Property* ermöglichen weitere ökonomische Interkationen mit vernetzten Geräten, bei denen das nötige Vertrauen in die Gegenpartie geringgehalten werden kann. Bei diesen Interkationen können vernetzte Geräte sowohl als Subjekt, als auch als Objekt, auftreten. Dieses Konzept, *ökonomischer Geräte*, wird mithilfe eines auf Ethereum basierenden Prototyps verdeutlicht. Der Prototyp stellt einen öffentlichen Bildschirm dar, der gegen Zahlung Inhalte anzeigt. Die Zahlungen werden von einem Smart Contract verwaltet und automatisch an Investoren als sofortige Mikrodividenden weltweit weiterverteilt. Dieses Beispiel illustriert neue Finanzierungs- und Besitzmodelle für produktive Vermögenswerte auf Basis vernetzter Geräte und ist insbesondere für Entwicklungsländer von Interesse, wo die traditionallen Finanz- und Justizsysteme unterentwickelt sind.

PUBLICATIONS

The following publications are included in parts or in an extended version in this thesis:

- Dominic Wörner and Thomas von Bomhard (2014). „When Your Sensor Earns Money: Exchanging Data for Cash with Bitcoin." In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. UbiComp '14 Adjunct. Seattle, Washington: ACM, pp. 295–298. URL: http://doi.acm.org/10.1145/2638728.2638786.

- Kay Noyen et al. (2014). „When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin." In: *CoRR* abs/1409.5841. URL: http://arxiv.org/abs/1409.5841.

- Dominic Wörner (2016). „Design of a Real-Time Data Market Based on the 21 Bitcoin Computer." In: *Tackling Society's Grand Challenges with Design Science: 11th International Conference, DESRIST 2016, St. John's, NL, Canada, May 23-25, 2016, Proceedings*, pp. 228–232. URL: https://www.springer.com/us/book/9783319392936.

- Dominic Wörner, Thomas Von Bomhard, et al. (2016). „The Bitcoin Ecosystem: Disruption Beyond Financial Services?" In: *European Conference on Information Systems (ECIS)*. Istanbul, Turkey.

The following publications were part of my PhD research, but are outside the scope of this thesis:

- Marcus Köhler, Dominic Wörner, and Felix Wortmann (2013). „Platforms for the Internet of Things – An Analysis of Existing Solutions." In: *5th Bosch Conf. Syst. Softw. Eng.* Technical Report.

- Dominic Wörner, Thomas von Bomhard, and Felix Wortmann (2013). „Occupancy-based Heating Control for Residential Buildings Using Environmental Sensors." In: *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. BuildSys'13. Roma, Italy: ACM, 28:1–28:2. URL: http://doi.acm.org/10.1145/2528282.2528313.

- Dominic Wörner, Thomas von Bomhard, Marc Röschlin, et al. (2014). „Look Twice: Uncover Hidden Information in Room Climate Sensor Data." In: *International Conference on the Internet of Things (IOT)*. Cambridge, MA, pp. 25–30.

- Thomas von Bomhard, Dominic Wörner, and Marc Röschlin (2014). „Towards Smart Individual-room Heating for Residential Buildings.“ In: *Computer Science-Research and Development*, pp. 1–8.
- Thomas Von Bomhard and Dominic Wörner (2016). „Design and Evaluation of a “Smart Individual-room Heating IS” to Improve Comfort And Save Energy in Residential Buildings.“ In: *European Conference on Information Systems (ECIS)*. Istanbul, Turkey.
- Remo M. Frey, Dominic Wörner, and Alexander Ilic (2016). „Collaborative Filtering on the Blockchain: A Secure Recommender System for e-Commerce.“ In: *22nd Americas Conference on Information Systems (AMCIS)*. San Diego, CA.

ACKNOWLEDGMENTS

This dissertation is the result of my research at the Institute of Information Management at ETH Zurich, the Bosch Internet of Things Services Lab, the Human Dynamics Group and the Digital Currency Initiative at the MIT Media Lab. I would like to express my sincere gratitude to all those who so greatly supported me in these projects and made this work possible. In particular I would like to thank my supervisor Prof. Dr. Elgar Fleisch for giving me the opportunity to benefit from the unique and stimulating research environment he has created. Furthermore, I'm very thankful for his constant support to explore my ideas. I would like to thank Prof. Dr. Frédéric Thiesse for his willingness to co-supervise my research, and his interest in my work. I would like to thank Ass.-Prof. Dr. Felix Wortmann for his guidance throughout my research, and I am grateful for the many things I learned from him. I would like to thank Prof. Dr. Markus Weinberger and Timo Gessmann for supporting me and my projects from the industry perspective. I would like to thank Prof. Alex *Sandy* Pentland for inviting me to spend 8 months at his research group at the MIT Media Lab. I also would like to thank the Digital Currency Initiative, in particular Joi Ito, Dr. Neha Narula, and Michael Casey for a stimulating environment and a welcoming and supportive atmosphere.

I would like to thank the Bosch Group and Bosch Software Innovations in particular, whose commitment to research in the Internet of Things provided the foundation of this work. Furthermore, I would like to thank the Swiss National Science Foundation for supporting my research stay at the MIT Media Lab.

I would like to thank my colleagues at and around the Institute of Information Management at ETH Zurich and the Institute of Technology Management at the University of St. Gallen for the excellent working atmosphere and the many memorable hours we spent together. In particular, I would like to thank Thomas von Bomhard, Paul Rigger, Kristina Flüchter, Marcus Köhler, Stefanie Turber, André Dahlinger, Benjamin Ryder, Dominik Bilgeri, Bernhard Gahr, Matthieu Chanson, Andreas Bogner, Arne Meeuw, Dirk Volland and Kay Noyen. Furthermore, I would like to thank Christian Decker, Francisc Bungiu, and Andrew Koh for insightful collaboration.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

— Mark Weiser

1.1 CONTEXT AND MOTIVATION

25 years ago Mark Weiser envisioned the computer of the 21st Century. Rather than interacting with a personal computer, computing technology is seamlessly integrated into the fabric of everyday life. Computing is to become ubiquitous (Weiser 1991), embedded in every object, and equipped with communication and sensing technologies to form an IOT. Besides all the hype (Gartner 2015b; Manyika et al. 2015), today, we still experience only glimpses of true ubiquitous computing. Start ups and established hardware companies are embedding computing chips and network stacks in ever more products. From light bulbs to cars, everything is getting infused with parts from the smartphone supply chain (Evans 2015; Evans 2016). But instead of seamless interconnectivity and interoperability, i.e. objects playing in concert, we are getting ever more soloists, remotely controlled by proprietary smartphone apps – or the manufacturer's backend. In exchange for this convenience we pay by giving up ever more privacy. Data about our daily habits, our health and our finances are constantly collected, and stored in large corporate databases. Data is the new oil (Schwab et al. 2011), but only few know how to refine it (Manyika et al. 2015). Thus, a lot of the collected data lies dormant, awaiting the next security breach[1]. Exchange of consumer data between companies is opaque and bears liability risks for the companies (Black 2013), instead of being available to create societal value.

This is not the only inefficient allocation of resources. Most of the time your computer uses a few percent of its computing power. Your disc is only filled by 30%, and your Internet connectivity is only exhausted when streaming a movie. The cloud computing paradigm (Hayes 2008; Armbrust et al. 2010) with the technology of virtualization (Barham et al. 2003) has brought efficient resource allocation to data centers, but with the emergence of ubiquitous computing and the Internet of Things, it can be assumed that most of the computing power will soon be outside of corporate data centers.

1 See https://www.privacyrights.org/data-breach for public data breaches.

New paradigms like Edge or Fog computing (Bonomi et al. 2012; Yi, C. Li, and Q. Li 2015) attempt to extend the cloud paradigm to end devices, but centralized control is always a bottleneck. Grid computing has only been moderately successful in scientific contexts (David P. Anderson et al. 2002; Beberg et al. 2009), and peer-to-peer systems (Rodrigues and Druschel 2010) suffer from free-riders (Sanford J. Grossman 1980) and sybil attacks (Douceur 2002). Nevertheless, there are examples of successful peer-to-peer systems such as BitTorrent (Cohen 2003), that have been resisting malicious attacks and governmental interference.

If transaction costs are low, the market and price mechanisms allow for division of labor and efficient resource allocation (A. Smith and Nicholson 1887). With increasing transaction costs, however, hierarchical structures and centralized control become more efficient (Coase 1937).

Information technology is decreasing transaction costs. The Internet, based on an open protocol stack, diminished global communication costs. Databases can store and query huge amounts of data. The IOT captures real-time information of the physical world, and advances in machine learning and artificial intelligence begin to replace costly human decision making, or at least narrow down choices and consequences such that humans with bounded rationality (Simon 1982) are able to make a quick informed decision.

In a truly interconnected world automation has to bridge trust boundaries. Smart objects have to work in unison to unlock the greatest and economic and societal benefit (Manyika et al. 2015). Many smart objects are mobile and thus the topology of the network is changing constantly. While at one point in time an object is surrounded by trusted objects, i.e. the neighbors provide credentials that are attested by a trusted source, e.g. the manufacturer. At some later time the situation might be different and only untrusted devices are around.

Let us make an analogy to our daily lives. If you transact with friends and family you trust in reciprocity, and therefore you have little hesitation to cooperate. Transactions are implicit – stored in the shared mind of the participants – and will eventually cancel each other out. However, if you transact with a stranger, the transaction is made explicit by exchanging money. Thus, reducing the need for trust. However, if you do not pay by cash, but by credit card, the direct participants in the transactions have to place trust in intermediaries. The payee has to trust that she eventually receives the money, and the payer has to trust that the payee does not charge more than agreed upon. The eventual enforcement of this implicit contract is executed by the trusted intermediary. This manifests in literal transaction costs. Trusted third parties are costly (Szabo 2005).

Until 2009 there was no digital equivalent of physical cash allowing for low-trust, pseudonymous payments over the Internet. Prior schemes always involved a trusted third party, and all implemented solutions have failed. Bitcoin is peer-to-peer electronic cash (Nakamoto 2008) that solves the trusted

third party issue by cleverly combining a variety of cryptographic techniques and mechanism design, i.e. aligning economic incentives to create a completely open self-sustaining system without requiring any kind of identification. The underlying technology is usually termed *blockchain technology*. Although the system is not perfect in theory (Eyal and Sirer 2014), it is surprisingly successful in practice. The market capitalization of Bitcoin the currency oscillates around $10 billion USD. More than 210000 transactions are recorded daily with a volume of more than $150 million USD, neglecting the *off-chain* trades on the numerous online exchanges. Bitcoin has often been described as digital gold because of the implemented monetary policy. The maximal number of bitcoins[2] that will be ever created is limited to 21 million, and the resource-intensive process of *minting new coins* that ensures security is termed *mining*. In effect, the number of bitcoins and their rate of creation is transparent, and independent of any governmental monetary policy.

As will be clear later on, Bitcoin is far more than an ordinary currency. Bitcoin is the first programmable money and the first digital bearer asset. Ownership and control of bitcoins is non-custodial and augmented by possession of cryptographic keys. As Richard Gendal Brown, former IBM executive, put it: "On the blockchain, nobody knows you're a fridge" (Brown 2013). Machines are fist-class citizens. Giving machines control over money and enabling true micropayments orders of magnitude below the traditional interpretation of micropayments, i.e. a few dollars, will stimulate new types of autonomous transactions without human involvement. The beginning of an economy of things, and an additional driver towards ubiquitous computing.

Bitcoin itself is a role model of how a trusted third party can be replaced and automated by a network of untrusted computers by defining rules through code and the market mechanism (Lessig 2009). Indeed, there are already hundreds or thousands of alternative cryptocurrencies and blockchains exploring the space of possible implementations. However, most of them do not add much value. The creation and maintenance of such a decentralized manifestation of a trusted third party, however, does not come without cost either. Currencies are prime examples of network effects in action: A currency is only useful if a trading party is accepting it, and the party will only accept it if she is convinced that other parties, who she might want to trade with at a later time, accept it as well. Transaction networks are comparable to communication networks, and the derived value from a growing user base should thus be superlinear (c.f. Metcalfe's law (Metcalfe 2013) and Reed's law (David P Reed 1999)). Since a cryptocurrency ecosystem involves more parties than merchants and customers, namely, miners, developers and speculators, network effects are multi-sided and feedback loops can be particularly strong (Giaglis and Kypriotaki 2014). Only time will tell if Bitcoin's dominance will be disputed, and if there will be a consolidation or a rich ecosystem of cryptocurrencies and blockchain-based networks.

2 Bitcoin with capital *B* refers to the system, whereas bitcoin refers to the unit of currency.

In terms of market capitalization the next cryptocurrency after Bitcoin is Ethereum (Buterin 2014), which is valued collectively at approximately a tenth of Bitcoin. Ethereum has a different mission statement than Bitcoin. It aims to be *the world computer*, a decentralized always-on trusted backend for applications - a multi-purpose trusted third party. In this vision, the concept of programmability becomes the focus, and the currency becomes a means to this end.

Besides coordination between smart objects and their resources in an economy of things, there is another economy that is fueled by the IOT: The *sharing economy* (Sundararajan 2016).

The most prominent representatives of this economy are Uber and AirBnB. These are platform businesses that coordinate underutilized, mostly private, assets such as cars and apartments to increase their economic value. Only a few years ago it was inconceivable for most people to rent their homes to some stranger over the Internet, or to enter a private car of some stranger. But these norms have been updated. Apartments and cars can be seen as the tip of an iceberg. Shared usage of assets starts with high value assets, but may continue to lower value assets if transaction costs are decreasing. Furthermore, the value of an asset is relative. A $250 USD solar home system that provides electricity to charge a few phones and batteries is not a high value asset to our standards, but in east Africa or southeast Asia the situation is quite different. In these regions products are already equipped with IOT-technology in combination with mobile payment technologies to enable innovative business models like *pay-as-you-go* and *lease-to-own* in countries with low-levels of legal enforcement to make products affordable for the people at the bottom of the pyramid (Alstone, Gershenson, and Kammen 2015). As software is eating the word (Andreessen 2011), the *as-a-Service paradigm* originating from cloud computing, takes hold in the physical world.

Cryptocurrencies in unison with the concept of *smart contracts* and *smart property* (Szabo 1997; Hearn 2011b) can support these business models by decreasing the need of trust in counterparties, and attaching financial services to the product itself.

1.2 OBJECTIVE AND APPROACH

This thesis aims to provide initial insights into the impact of cryptocurrencies on the IOT. Due to the novelty of cryptocurrencies, the research approach is based on design and engineering of prototypical applications. Thus, the research is explorative and iterative. One promising application of cryptocurrencies in IOT is investigated - the trading of digital goods and services between connected devices exemplified by the Sensing-as-a-Service scheme. Characteristics of cryptocurrencies in general, and Bitcoin in particular, to aid this scheme are identified, and two prototypes are developed. The first prototype illustrates the concept of trading data for electronic cash based entirely on the

Bitcoin network and protocol. The evaluation of the concept and prototype reveals issues concerning confidentiality, latency, and the inappropriateness to perform micropayments. By leveraging the programmability of Bitcoin transactions, smart contracts are introduced to enable low-latency micropayments between a large number of data requesters and data providers mediated by a trust-minimized hub. The concept is implemented as a mobile crowdsensing application allowing users of a smartphone application to offer data on a global market. Further investigation of the concept of smart contracts and the related concept of smart property leads to the notion of economic devices. The concept of economic devices is explained and illustrated with a prototype of an Ethereum-enabled public display. The display offers the service to show user-selected content in exchange for cryptocurrency payments and issues tokens that entitle the bearer to receive a share of revenue in real-time.

As a foundation to understand cryptocurrencies and their significance, this thesis provides a technological introduction to the two most important cryptocurrency platform, Bitcoin and Ethereum, and an investigation of the broader cryptocurrency and blockchain ecosystem.

1.3 OUTLINE

Chapter 2 provides a brief introduction to IOT. The historical evolution from Radio-Frequency-Identification (RFID) to the consumer IOT is provided. Furthermore, opportunities and challenges for individuals, the industry, and the society as a whole are discussed. Many of these challenges can be addressed by creating a more decentralized IOT with increasing autonomy of devices and greater user control. Thus, the chapter concludes with an overview of developments towards decentralized architectures, and a motivation for the need of a digital equivalent to physical cash. Chapter 3 introduces the technologies behind digital cash. The chapter starts with an indepth presentation of Bitcoin. Focus is on the programmability of Bitcoin transactions, which provides the basis for smart contracts used in later chapters. In addition, the current challenges of Bitcoin, and decentralized cryptocurrencies in general, are discussed based on a literature review. The challenges are complemented with current approaches to tackle them. Thereafter, ways to extend Bitcoin's functionality are discussed. This leads to another, more general cryptocurrency design with the focus on a decentralized application and smart contract platform embodied in Ethereum. After presenting Ethereum, a brief comparison with Bitcoin is given. The chapter ends with the introduction of permissioned and private blockchain platforms, which have become increasingly popular throughout industries in recent months. Chapter 4 provides a complementary perspective by reviewing the Bitcoin and blockchain ecosystem. After reviewing the Bitcoin start-up ecosystem, the broader ecosystem entailing altcoins, metacoins, and permissioned blockchains is investigated. The chapter concludes with a discussion of the economic relevance of cryptocurrencies.

Having laid out the foundations, the first application of cryptocurrencies in IOT is discussed. Chapter 5 investigates the concept of S²aaS with Bitcoin. The most important example of the exchange of a digital good or service against payment in the IOT. Thus, relevant characteristics of Bitcoin are presented, and a first prototype is discussed. Chapter 6 extends the initial prototype in order to enable low-trust instant micropayments between a large number of data requesters and data providers. Therefore, unidirectional hub and spoke payment channels based on Bitcoin smart contracts are developed and illustrated by a mobile crowdsourcing application allowing anybody to sell smartphone sensor data. Chapter 7 introduces the concept of economic devices which is enabled by cryptocurrencies. The concept is illustrated with a prototype of an Ethereum-enabled public display, and its significance for developing countries with underdeveloped financial services and legal systems is discussed. Chapter 8 concludes the thesis by recapitulation the key findings and its implications for research and practice. Furthermore, an outlook and ideas for future work is provided.

1.4 CREDITS

The work presented in this thesis is based on collaboration with colleagues at the Chairs of Prof. Elgar Fleisch at ETH Zurich and the University of St. Gallen, the Chair of Distributed Computing at ETH Zurich, and the Digital Currency Initiative at the MIT Media Lab.

The discussion of the Bitcoin start-up ecosystem in Chapter 4 and the case studies in Section A is based on joint work with Thomas von Bomhard, Yan-Peter Schreier, and Dominik Bilgeri (Wörner, Von Bomhard, et al. 2016, c.f.).

The concept of S²aaS with Bitcoin and the extraction and discussion of relevant characteristics as presented in Chapter 5 is based on joint work with Kay Noyen, Dirk Volland, and Elgar Fleisch (Noyen et al. 2014, c.f.). The initial prototypical implementation was developed together with Thomas von Bomhard (Wörner and Bomhard 2014, c.f.). The construction of mediated payment channels to enable a scalable solution to incentivize mobile crowdsensing is joint work with Christian Decker. The prototypical implementation of the system was mainly done by Francisc Bungiu[3].

The idea to use the concept of economic devices to finance and capitalize productive assets in developing countries emerged in discussions with Michael Casey and Anders Brownworth at the MIT Media Lab. The Ethereum-enabled display was developed with the help of Andrew Koh.

3 See https://github.com/domwoe/datamarket

2

THE INTERNET OF THINGS

We're building a world-sized robot, and we don't even realize it.

— Bruce Schneier

In this chapter, we introduce the IOT. The main take away will be that although IOT promises great opportunities for businesses, consumers, and the society as a whole, the prevailing architecture leads to severe issues ranging from the sustainability of business models to privacy and security. We argue that a more decentralized architecture with increased autonomy of edge devices will help tackling these issues. Furthermore, we motivate the need for electronic cash as a basis for incentivization of participation and trust.

2.1 WHAT IS THE INTERNET OF THINGS?

Semantically, an IOT is a world-wide network network of interconnected objects uniquely addressable, based on standard communication protocols (INFSO 2008). However, in practice the term is used in very different settings and has been evolved into a suitcase word for various technologies. Organizations have given *vision statements* from different perspectives. Atzori, Iera, and Morabito 2010 classify them as *Things*-oriented, *Internet*-oriented, and *Semantic*-oriented, and argues that the Internet of Things can be seen as the convergence of these perspectives. In this thesis, we view the Internet of Things similar to the vision articulated by the CASAGRAS consortium as *a world where things can automatically communicate to computers and each other providing services to the benefit of the human kind* (Atzori, Iera, and Morabito 2010, c.f.). Arguably, this definition is close to the vision Mark Weiser gave, i.e. when the distinction between thing and computer becomes increasingly blurry (Weiser 1991). The Internet of Things extends the digital world into the physical world by providing web services access to physical resources. Sensors are the eyes and ears of the Internet, actuators are the hands. On the other hand, IOT provides a digital shadow for physical objects, extending their local capabilities and providing global capabilities.

In the following, we provide a short review of the evolution of the Internet of Things from the concept to auto-identification to today's consumer IOT. Thereafter, we discuss the current state and issues from different perspectives. We waive the introduction of the various technologies and their individual application scenarios, advantages and disadvantages. The interested reader may consult e.g. (Atzori, Iera, and Morabito 2010; Mattern and Floerkemeier 2010; Gubbi et al. 2013) for excellent overviews. We approach the IOT from a

practical perspective and focus on the issues with the prevailing centralized architecture. Finally, we discuss approaches towards a more decentralized IOT, and the role of digital cash.

2.2 EVOLUTION AND STATUS QUO

2.2.1 *Auto-Identification*

The term Internet of Things was coined by Kevin Ashton in 1999 (c.f Ashton 2009), then director of the Massachusetts Institute of Technology (MIT) Auto-ID Center which started to develop a global inter-corporate RFID infrastructure (Want 2006). RFID chips can be passive, i.e. do not need an own power supply, and can be integrated in all kinds of products to automatically identify individual products. Today, RFID technology is used in key access cards, to track books in libraries, and in supply chain management in general. RFID allows to track and uniquely identify all kinds of objects around us. It may also allow those objects to transmit part of the context they perceive through sensors. Using RFID, physical objects get a digital identity. Often this identity is fixed, sometimes it changes with context, but the interactivity is fairly limited. RFID-enabled products might be the largest category of today's IOT. In this thesis, however, we are concerned with more powerful smart objects and connected devices.

2.2.2 *Wireless Sensor (and Actuator) Networks*

Wireless Sensor Networks (WSN) aim to digitize the physical environment and are based on various kinds of open and proprietary communication technologies, such as LoRaWAN (Adelantado et al. 2016), SigFox[1] or ZigBee (Baronti et al. 2007). WSNs are typically deployed by a single company for a single application. Sensor data is transferred from individual sensing nodes to gateways, connecting local area networks with wide area networks, and finally to corporate databases. Important applications are in agriculture as well as monitoring technical infrastructure such as electricity grids. The term WSN is mostly used in the academic community while the industry has embraced terms such as Industrial IOT (Xu, W. He, and S. Li 2014), smart city (Cocchia 2014), smart agriculture (Aqeel-ur-Rehman et al. 2014) etc. Although the term IOT is often used, we recognize a big gap to the vision stated above.

2.2.3 *The iPhone and the Rise of Consumer IOT*

In 2007, the iPhone heralded a new era. Smartphones are the most pervasive wireless sensor (and computing) platform that has ever existed. The ubiquity

1 https://www.sigfox.com/

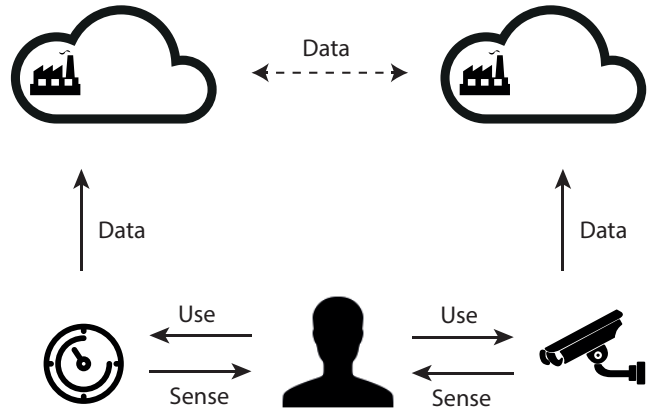


Figure 2.1.: Prevailing cloud-centric architecture in the smart home context.

of smartphones in the developed world, and increasingly in the developing world, came with an unprecedented global supply chain and rapidly decreasing prices for Systems-on-Chip (SoC), combining microprocessors, storage and different communication technologies. The smartphone has become a programmable interface, and its parts low-cost, easy to procure building blocks for all kinds of smart products and connected devices. This consumer IOT has been driven predominantly by startup companies. Crowdfunding platforms such as Kickstarter[2] and Indiegogo[3] allow early adopters to fund new product categories and demonstrate market interest to venture capitalists. Cloud computing on the other hand, has allowed to access computing, bandwidth, and storage infrastructure on a *pay-as-you-go* basis (Armbrust et al. 2010; Vaquero, Rodero-Merino, et al. 2008). Startups need comparably little initial capital to operate backend infrastructure and are able to scale the infrastructure elastically with customer demand. Additionally, IOT platforms have emerged, provided *as-a-Service*, and entailing various functions from device provisioning to data analytics. These developments have led to a predominantly cloud-centric architecture of the IOT. Figure 2.1 illustrates the cloud-centric architecture in the smart home context.

2.3 OPPORTUNITIES AND CHALLENGES

2.3.1 *Industry Perspective*

The IOT ecosystem is a lively mixture of established technology companies, start-up companies, and traditional hardware and consumer goods companies. The digitization of physical objects is stimulated by the aspiration to get real-

2 https://www.kickstarter.com/
3 https://www.indiegogo.com/

time data of products and their life cycle in order to allow for high-resolution management (Fleisch 2010), to provide new services to the consumer, and to explore and implement new business models (Fleisch, Weinberger, and Wortmann 2015).

Off-the-shelf SoC and Systems-in-Package (SiP), as well as a multitude of IOT platforms provided as a service (Mineraud et al. 2016), allow small start-up companies, and non-specialized companies to bring an ever growing number of connected products to market. Analysts predict the IOT will generate trillions of dollars per year within the next decade (Manyika et al. 2015), and there will be tens to hundreds of billions connected devices. Thus, the Internet is determined to undergo a phase transition. Currently, the Internet's dominant applications are social, connecting humans, and human-machine communication such as video and music streaming. However, in the future, the dominant application will most likely be based on autonomous machine-to-machine communications.

Although the marginal cost of connecting a product has become almost negligible, the maintenance and operating costs throughout the product lifetime are of concern. Therefore, vendors aim to use the digital capabilities of a product to reframe the product as a service with the aim to manifest recurring payments in form of subscriptions. Examples are Tado[4], offering a subscription-based thermostat, Comcast[5], offering a subscription-based security system, and also SolarCity[6], offering solar panels based on a pay as you go model. We assume that these models will be even more important in developing countries where upfront costs are an obstacle. However, at the same time financial infrastructure such as banking accounts and credit cards necessary to implement subscription-based services are underdeveloped. Successful examples can already be seen in markets where mobile payment systems, such as M-Pesa, have become pervasive (Hughes and Lonie 2007). M-Kopa[7] is offering solar home systems, a combination of a small solar panel, a battery, a few LED lights, and a mobile phone charger, based on a pay as you go model (Alstone, Gershenson, and Kammen 2015). Each solar home system is connected via cellular networks and can be remotely disabled, in case of late payments.

However, it is rather unlikely that service orientation coupled with subscription-based business models can be applied to all kinds of connected products. Many consumers are reluctant concerning recurrent payments. Tado, for example, started with a subscription-only model but had to offer a traditional one-time payment option soon. Many products that are becoming connected such as LED light bulbs, door locks, washing machines, and refrigerators have a life expectancy of many years. It remains to be seen if enough additional

4 https://www.tado.com
5 https://www.xfinity.com/home-security.html
6 http://www.solarcity.com/
7 http://www.m-kopa.com

value can be created and extracted to pay for operational costs of the backend infrastructure.

In addition, with increasing distribution of connected devices there will be an increasing demand for interconnectivity of these devices. We see this already in the smart home. A simple example is the communication between the Nest protect smoke alarm[8] and the Lifx[9] LED bulbs. If the smoke detector recognizes a fire the light bulbs begin to blink in an alarming red. However, instead of local communication, the communication is facilitated between the backend services of the two vendors (c.f. Figure 2.1). With increasing demand for interconnectivity and communication this cloud-centric mode of operation will produce significant costs for the vendor.

Finally, as data is increasingly viewed as an asset, vendors store huge amounts of customer data. However, only few are able to leverage these data to create value. In any case, the collection of sensitive customer data is also a liability due to the risk of cyber attacks. A recent estimate for the average cost of a data breach involving customer data is $4 million USD[10].

2.3.2 *Consumer Perspective*

Simple connected security systems, user-aware thermostats, health and fitness trackers, and many more connected products provide new levels of convenience, security and data-driven knowledge for self-improvement to consumers. Most IOT products are stand-alone and proprietary, but Application Programming Interface (API)s allow at least for restricted interoperability. These APIs are typically not standardized but hub products and services have emerged, that allow the orchestration of multiple devices from different vendors. Examples of such devices are the Nest learning thermostat[11] and the Amazon Echo[12] voice control. An example of a popular web service to interconnect devices and various web services is IFTTT[13].

Besides convenience, the current instantiation of the IOT has various downsides for the consumer. First, products that are augmented with digital services come with terms of service. Usage of the product, or at least the usage of a significant set of features, requires agreement to these terms. In many cases, these terms are repeatedly subject to change, and the customer has essentially no choice but to agree. These terms of service usually define what data the vendor is allowed to collect and to use. This practice interferes with the privacy of the customer. Even if the vendor itself is trustworthy, data breaches are common[14]. We have briefly discussed the role of IOT in emergent

8 https://nest.com/smoke-co-alarm/meet-nest-protect/
9 http://www.lifx.com/
10 https://www-03.ibm.com/security/data-breach/
11 https://nest.com/thermostat/meet-nest-thermostat/
12 https://www.amazon.com/echo
13 https://ifttt.com/
14 See e.g. https://www.privacyrights.org/data-breach

economies from the vendor perspective. Privacy issues from the consumer perspective are of particularly concern, since privacy and consumer rights are underdeveloped.

Besides data breaches, IOT devices often lack necessary security features or are poorly configured. Vulnerabilities of hundreds of thousands of devices have been demonstrated (Fernandes, Jung, and Prakash 2016; Bodenheim et al. 2014; Costin et al. 2014; Garcia et al. 2016). Vendors are often reluctant to invest in patching older generation devices, and if they do, consumers are often not aware of the need to update. Automatic updates may help, but provide a new opportunity for hackers to centrally attack a large number of devices.

If important features of a product are dependent on services provided solely by the manufacturer's backend, the notion of ownership becomes ambiguous. Even though a customer is in physical possession of a product, the functionality is owned by the manufacturer. We have seen multiple instances of manufactures discontinuing backend services and hence rendering the connected products useless (Cox 2016). While this may be expected only for small startup companies, the case of Revolv, owned by Nest, an Alphabet company, proved the contrary (Gilbert 2016).

2.3.3 *Societal Perspective*

An Internet of Things is the basis of a data-driven society (Alex Pentland 2013). Collection and fusion of real-time data enables a more efficient usage of resources. Smart Grids allow the integration of decentralized renewable energy sources. Connected buildings and smart cities enable more efficient energy usage. Connected traffic control systems and connected cars allow for a more efficient usage of the physical infrastructure. Smart agriculture allows for more precise usage of pesticides and fertilizers. Health trackers and smart pills will lead to a better understanding of human health and allows for earlier interventions, and thus cost savings.

However, the Internet of Things does not come without risks. We have already mentioned the impact on personal privacy in the context of connected consumer products. In a connected world we will leave digital traces everywhere. Since smartphones have cameras, the cost of them has rapidly decreased. In many cases a camera in combination with image recognition is cheaper than adding a specialized sensor. Connected cars have cameras, drones have cameras, humans might have cameras in their visual field. If only a subset of these data sources can be accessed and combined, there is a great risk of establishing a surveillance society.

The interconnectedness, complexity and automated control loops involving connected actuators may allow individual hackers, even without the resources of governmental agencies, to attack physical infrastructure on a large scale. Stuxnet (Langner 2011) was only an early example. Every year at the hacker

conference Defcon[15] new connected devices are getting hacked. The situation is particularly severe, when large parts of the connected infrastructure are centrally controlled with single points of failure.

2.4 DEVELOPMENTS TOWARDS A DECENTRALIZED IOT

Many of the issues encountered in the former sections can be addressed by decentralizing the IOT infrastructure and providing more autonomy to individual devices or sub networks. Mineraud et al. 2016 provide a comprehensive stop gap analysis of the current IOT platform landscape. Out of 39 investigated platforms only 3 are categorized as decentralized. These platforms are research projects and are not ready for productive deployment. LinkSmart and OpenIoT are open source middlewares. Hub of all Things is a personal data store in the early steps of commercialization (Ng 2014). Personal data stores provide means to keep data produced by personal connected devices under control. In particular, if personal data stores allow for selective computation on personal data as proposed by the SafeAnswers system (Montjoye, Shmueli, et al. 2014). Instead of giving service providers access to raw data, service provider are able to send vetted algorithms to the personal data store. However the idea of personal data stores is not new, still there a no successful implementations. A. Narayanan, Toubiana, et al. 2012 discuss this phenomenon based on technical, economical and usability considerations. Neither manufacturers of connected products push these architectures, nor is there a real consumer pull at the moment. Hub of all Things and Enigma (Zyskind, Nathan, and A. Pentland 2015) try to establish multi-sided markets and thus provide financial incentives and positive feedback loops to foster participation. In particular in the case of Enigma, are cryptocurrencies and blockchain technology at the core.

An approach towards autonomy of individual devices can be seen with the Web of Things paradigm (Guinard et al. 2011). Individual devices employ web servers and interact with each other, and with cloud services via Representational State Transfer (REST)ful APIs, and machine and human readable Java Script Object Notation (JSON) documents. Thus, proved security and authentication mechanism from the traditional web such as Secure Socket Layer (SSL)/Transport Layer Security (TLS) and OAuth (Leiba 2012) can be reused. The machine-payable web, as envisioned by 21 Inc. (c.f. Sec. A.4, can be seen as an extension to the web in general, and the Web of Things in particular. The machine-payable web reintroduces the *402 payment required code* and allows to pay for individual Hypertext Transfer Protocol (HTTP) requests using Bitcoin. Thus, resources can be paid for directly instead of using subscriptions and API keys. This allows individuals (i.e. their computers and connected devices) to offer digital goods and services in exchange for payments.

15 https://www.defcon.org/

In the Web of Things and machine-payable web approach connected devices become servers in addition to clients as in a cloud-centric architecture. This provides the basis for the concepts of peer-to-peer (Rodrigues and Druschel 2010) and grid computing (D. P. Anderson 2004). Peer-to-peer computing can be entirely decentralized, where all nodes are equal. However, most peer-to-peer networks are not based on standardized web protocols, but on specialized protocols (c.f. BitTorrent or Bitcoin). In the grid computing paradigm, a network of nodes is considered as one big computer sharing work. However, the work gets typically distributed and aggregated by a central controller. Grid computing has only been successful in the scientific context. People all over the world have contributed CPU cycles to identify extraterrestrial life (David P. Anderson et al. 2002) and to simulate the folding of proteins (Beberg et al. 2009). However, participation based on altruism and scientific interest can not be extended to general application domains.

Another paradigm is known as edge or fog computing, that aims to extend the cloud paradigm to edge devices (Bonomi et al. 2012; Vaquero and Rodero-Merino 2014; Yi, C. Li, and Q. Li 2015). The focus thereby is in increased utilization of resources at the edge, lower upstream bandwidth requirements, and decreased latency. Data analytics at the edge also enables increased privacy since raw data can be contained at the device level and does not need to be transferred and processed in the cloud. Vaquero and Rodero-Merino 2014 proposes a definition: "Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so". They further highlight the need for accountability and monetization in order to provide incentives for device cooperation.

On a similar line, B. Zhang et al. 2015 argues that the cloud-centric architecture does not scale for the IOT and suggests a distributed platform, called the Global Data Plane (GDP). It is based on a data-centric design with focus on transport, replication, preservation, and integrity of streams of data while enabling transparent optimization for locality and quality of service. The foundation is a single-writer append-only log, coupled with location-independent routing, overlay multicast and higher level interfaces such as common access APIs. However, as known from peer-to-peer-based Distributed Hash Table (DHT) like Kademlia (Maymounkov and Mazières 2002), without proper incentivization nodes may leave the network and data may be lost.

Finally, as motivated by F. Giannotti et al. 2012, the Nervousnet architecture envisions the IOT as a distributed participatory sensing platform "to provide real-time data for all and an AppStore for IOT applications" (E. Pournaras, Moise, and D. Helbing 2015). It aims to provide a resilient *planetary nervous*

system to build a digital democracy. Besides aspects of gamification to incentivize user participation, the system aims to embed micropayments as a reward mechanism (Dirk Helbing and Evangelos Pournaras 2015).

We are still in the early days of the evolution towards an IOT. In those days, it is natural that successful commercial applications utilize centralized solutions. Central control and a unified view of data simplifies complexity, and economies of scale decrease costs. However, this may only be true up to a particular scale. Currently, a form of decentralization is achieved by having a large number of vendors and a multitude of platforms, i.e. decentralization by competition at the company level. However consolidation is likely to happen. Furthermore, because standards and interoperability are still issues, individual homes, companies and cities can be expected to settle on a single platform. We already motivated the need for a digital cash equivalent in order to build a decentralized but cooperative IOT – an *Economy of Things* (Pureswaran and Lougee 2015).

2.5 CONCLUSION

We introduced the Internet of Things as the vision of a world where things can automatically communicate to computers and each other providing services to the benefit of the human kind, and followed the evolution through three major epochs: auto-identification, wireless sensor and actuator networks, and the current consumer IOT. We saw that the prevailing architecture of the IOT is characterized by cloud computing and the client-server architecture. While this model allowed for simple and effective scaling during the initial stages of the Internet of Things, it becomes problematic with increasing scale. Current business models may not be sustainable. Edge devices provide minimal security. Raw data is routed through, and collected by central servers, acting as single points of failure and providing lucrative targets for attacks. Thus, potentially undermining privacy of the individual. We presented different avenues that strife for decentralization to mitigate these issues. A common prerequisite of these approaches is the ability for incentivizing participation and fair behavior of unknown participants. Thus, a digital analog of cash is needed.

3

BITCOIN AND BEYOND: TECHNOLOGICAL PERSPECTIVE

Politicians have often laboured under the delusion that money is something created and manipulable by themselves, when in fact it is the spontaneous institution of a free society and will continue to evolve in ways outside their grasp.

– Nick Elliot

The concept of money and currency has evolved over thousands of years. When presenting the origin and use of money, most economists reiterate the thesis formulated by Adam Smith that money originates from the needs of a barter economy and the division of labor. Because demand and supply of two goods rarely coincide in a barter exchange, a commodity that can act as a unit of account and medium of exchange is required. However, ethnographic studies provide no evidence that pure barter economies have ever existed (Graeber 2014). David Graeber formulates an alternative thesis where debt and credit based on informal gift economies provide the cultural breeding ground for the emergence of money (Graeber 2014). In a gift economy, debt and credit are informally stored in the brains of the economic actors. This allows for division of labor, but restricts the size of the economy, because trust and reciprocity are fundamental. Currency allows to formalize and quantify debt obligations precisely and enables economies to transcend trust boundaries. The archetype of currency is the coin forged from a scarce metal and *coined* by a sovereign to guarantee authenticity. Today, most monies are fiat money. Money without intrinsic value, established as legal tender by governmental decree. Physical currency, in form of coins and notes, is only a tiny fraction of the total monetary supply in an economy. Most money is created by commercial banks in form of deposits, and other financial contracts. Hence, most money is custodial and exists only virtually on disparate electronic ledgers of financial institutions. Therefore, the need for trust is reintroduced on a higher level.

Cryptocurrencies introduce a new type of money. Although digital, cryptocurrencies share many characteristics of physical currencies (e.g. being a bearer instrument). In addition, cryptocurrencies have unique and novel characteristics (e.g. programmability), and provide the basis for permissionless innovation on a much faster timescale. As the term suggests, cryptocurrencies are based on a combination of various cryptographic primitives. However equally important, for the design and recent success of cryptocurrencies, are economic principles and mechanism design. Bitcoin, the first cryptocurrency with widespread adoption, is based on a peer-to-peer network with incentive-

compatible rules for participation and disproportionate rewards for early adopters. The nodes of the peer-to-peer network maintain a cryptographically and thermodynamically secured replicated transactional database, called blockchain, that provides a tamper-proof and verifiable memory of all transactions, and thus, the ownership status of bitcoins. Hence, the blockchain provides a precise digitized analog of society's shared accounting system encountered in the gift economy. Moreover, the Low-trust and permissionless nature of cryptocurrencies enables machines to autonomously participate in economic interactions.

In this chapter, we introduce Bitcoin, the first implementation of a peer-to-peer electronic cash system. The introduction is rather technical with a focus on the transaction and scripting system of which we make extensive use in later chapters. The scripting system is deeply integrated in the transaction structure of Bitcoin and allows for programmable transactions. These programs, that determine the validity of a transaction and are executed and verified by all network participants, are the basis for smart contracts – contracts enforced by computer code. However, Bitcoin's scripting language and programming model is fairly restricted. This limits the types of smart contracts that can be instantiated directly in Bitcoin script. The open source nature of Bitcoin has lead to a Cambrian explosion of cryptocurrency and blockchain projects. The cryptocurrency with the second largest market capitalization besides Bitcoin is Ethereum. Ethereum is specifically designed to provide a platform for decentralized applications and smart contracts. We present Ethereum with a focus on its differences to Bitcoin, and provide a concise comparison between both systems. The last section before the conclusion briefly introduces permissioned blockchains because of their importance for the economic consideration in the next chapter.

3.1 BITCOIN: PROGRAMMABLE CASH

In November 2008 an e-mail with the subject: "Bitcoin P2P e-cash paper"[1] was sent to The Cryptography Mailing List. The first e-mail from sender Satoshi Nakamoto. The e-mail presents the abstract and a link to a white paper with the title "Bitcoin: A Peer-to-Peer Electronic Cash System". A few month later, on January 3rd 2009, the first instances of the Bitcoin client software began running the Bitcoin network, and Bitcoin has been running ever since.

As already apparent from the title of the paper, Bitcoin provides a Peer-to-Peer (P2P) digital cash system without reliance on a trusted third party. Since digital assets can be duplicated without cost, previous digital cash systems had to rely on trusted third parties to authorize transactions centrally. This issue is known as the double spending problem. Assume Alice has a file that represents $100, and wants to send this file to Bob. Bob can be sure that the

1 http://satoshi.nakamotoinstitute.org/emails/cryptography/1/

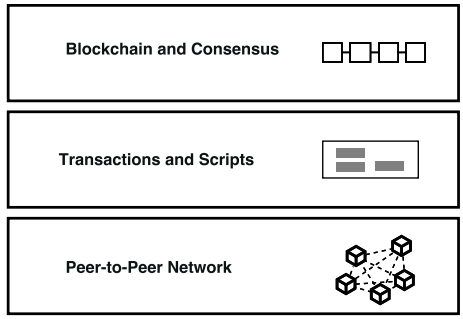


Figure 3.1.: Technological layers of Bitcoin.

file is not counterfeited, if Alice attaches a digital signature to the file, but how can Bob be sure that Alice did not send the file also to Charlie? Traditionally, this issue was solved by a central trusted entity keeping track of balances in a ledger and authorizing transactions. However, this third party has therefore the ability to censor transactions, and to create money at will. Furthermore, this entity represents a single point of failure, susceptible to rent-seeking behavior, coercion, and cybercriminal attacks. In addition, electronic money systems that rely on central authorization can not be considered electronic cash systems, because cash is a bearer asset, i.e. cash is non-custodial. Thus far, before Bitcoin, no centrally-controlled digital money system has been successful on a global scale.

In Bitcoin, by contrast, transactions are accepted and ordered by the entire network with the help of a novel consensus mechanism, called Nakamoto consensus, which involves the expenditure of an exogenous real-world resource – energy. This process, called mining, is also responsible for creating new coins, and provides the necessary economic incentives for an individual *miner* to include only transactions that other network participants deem valid. Hence, a canonical ordering of transactions is achieved, and double spending is prevented. In the following sections the technological underpinnings of Bitcoin are presented in more detail.

3.1.1 *The Technology*

From a technological perspective, Bitcoin can be divided into three layers (see Fig. 3.1). Bitcoin is a peer-to-peer network, consisting of computers running a client software and interacting with each other via specified messages. The most important data structure that is exchanged between the nodes is the Bitcoin transaction. The Bitcoin transaction provides the instructions to induce a state change, i.e. a change of ownership of bitcoins. Finally, a suitable consensus mechanism is required to prevent double spending.

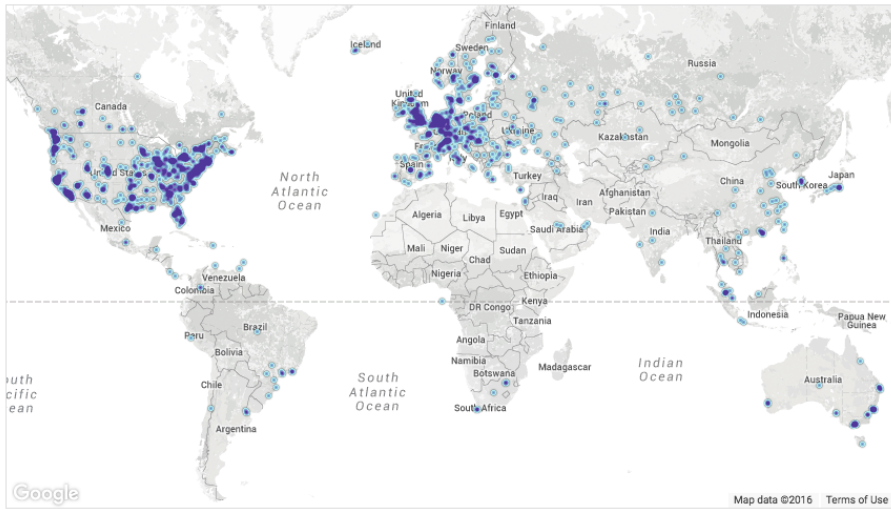


Figure 3.2.: Concentration of Bitcoin nodes around the world (Source: https://bitnodes.21.co, accessed 2016-07-07).

Peer-to-Peer network

The foundation of Bitcoin is a peer-to-peer network of voluntary nodes. Each node validates, relays, and stores all (valid) transactions. Thus, in Bitcoin, everyone validates everything, or at least everyone is able to validate everything. In the beginning, every Bitcoin *user* ran a full Bitcoin node, since it was the only way to keep track of, and to send bitcoins. Today, most users do not directly participate in the peer-to-peer network, because running a full node is becoming increasingly resource intensive. At time of writing, the Bitcoin network consists of approximately 5600 nodes. Nodes are distributed globally, but the majority is in Europe and the coastal areas of the US (see Fig. 3.2).

Transactions and Scripting

To be precise, the naming of *Bitcoin* is misleading. In Bitcoin, there are no actual coins being exchanged between parties. As mentioned earlier, Bitcoin is based on a global ledger replicated across a network of nodes. Every node can calculate the current state of the ledger by applying all transactions sequentially beginning from an initial state. A simplified representation of a transaction is shown in Figure 3.3. A transaction contains essentially a list of inputs and outputs. Inputs reference (unspent) outputs of previous transactions, and provide a means of authentication to prove ownership of these UTXOs. Outputs are tuples entailing values, the amount denominated in satoshi[2], and the condition the receiver has to meet to spend the respective

2 1 bitcoin = 10^8 satoshi

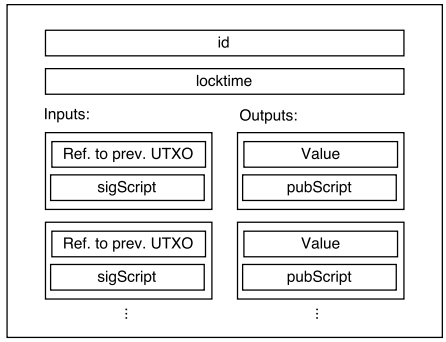


Figure 3.3.: A simplified representation of the Bitcoin transaction data structure.

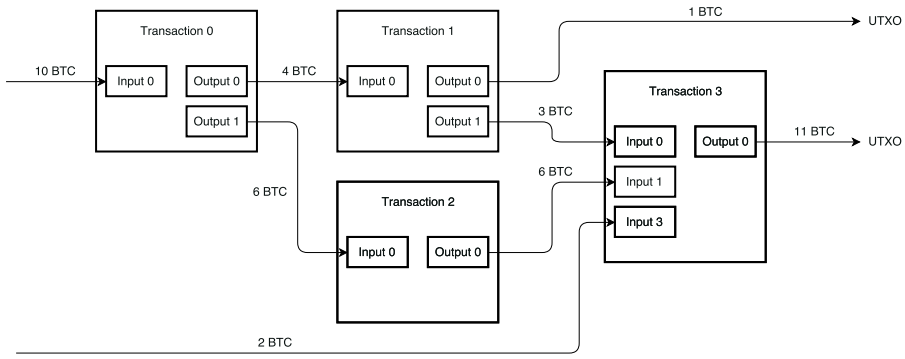


Figure 3.4.: Transaction flow and change of the UTXO set.

output. Consequently, each individual UTXO has a value and an owner. Thus, an individual UTXO can be identified with a *coin*. Each transaction spends one or more UTXO and creates one ore more UTXO. UTXOs have always to be spent entirely. In Figure 3.4 the spending and creating of transaction outputs is illustrated. In comparison to the more intuitive account model, the UTXO model allows to process transactions in parallel and has privacy benefits (Buterin 2016b). Hence, the Bitcoin ledger is not comprised of accounts and balances, but of UTXOs.

A transaction is valid if the combined value of the inputs is greater than, or equal to the combined value of the outputs, and if the conditions of the referenced previous UTXOs are met. In the following we will take a closer look at these conditions and how transactions are authenticated. The corresponding data fields are called *pubScript* and *sigScript* and can contain data and op codes, i.e. programming primitives, of a constrained, stack-based scripting language called *Bitcoin Script*. When validating a transaction, a node concatenates the pubScript of a referenced output with the sigScript of a referencing input, and

executes the combined script. A transaction can only be valid if all scripts evaluate to *true*.

The most important op codes are related to cryptographic primitives such as digital signature verification and secure hash functions. Bitcoin currently employs Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication. In the prevailing case, a derivation of an ECDSA public key is identified with a Bitcoin address, and the corresponding private key is used to create a signature to authenticate the spending of an UTXO corresponding to this Bitcoin address. The pubScript in this basic scenario is called Pay-to-PubKey-Hash (P2PKH).

```
OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

The sigScript that is able spend the output takes the following form:

```
<Sig> <PubKey>
```

where <Sig> denotes the signature corresponding to the particular transaction and the private key corresponding to the public key <PubKey>.

Concatenating and executing works as follows:

```
1. <Sig> <PubKey> OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY
    OP_CHECKSIG
2. <Sig> <PubKey> <PubKey> OP_HASH160 <PubKeyHash> OP_EQUALVERIFY
    OP_CHECKSIG
3. <Sig> <PubKey> <PubKeyHash> <PubKeyHash> OP_EQUALVERIFY
    OP_CHECKSIG
4. <Sig> <PubKey> OP_CHECKSIG
5. TRUE
```

First, OP_DUP duplicates the operand <PubKey>. Second, OP_HASH160 computes the SHA-256 and RIPMED-160 secure hash of <PubKey>. Third, OP_EQUALVERIFY takes two operands and returns true, if they are equal. Fourth, OP_CHECKSIG takes a signature and a public key, and returns true, if the signature validation is successful.

Bitcoin, i.e. the reference implementation Bitcoin core, currently distinguishes five types of *standard* pubScripts:

- Pay To Public Key Hash (P2PKH)
- Pay To Public Key (P2PK)
- Multi-Signature
- Pay to Script Hash (P2SH)
- Null data

If a pubScript can not be categorized in one of these types, it is termed *non-standard*. For some time Bitcoin core nodes would not relay transactions containing non-standard pubScripts because there had been found vulnerabilities in some op codes. However, if a non-standard transaction ended up in the

blockchain because a miner (see Sec. 3.1.1) had a different policy, Bitcoin core nodes still accepted it. Today, Bitcoin core relays transactions with arbitrary non-standard pubScripts.

Pay-to-PubKey (P2PK) is similar to P2PKH and not used very often since a shorter key hash is replaced by a longer public key. Multi-signature outputs demand m-of-n signatures to spend an output. However, today multi-signature functionality is usually implemented as Pay-to-Script-Hash (P2SH). P2SH is the most versatile pubScript type, and thus worth discussing in more detail.

The P2SH pubScript takes the following form:

```
OP_HASH160 <Hash160(redeemScript)> OP_EQUAL
```

Thereby, the pubScript reveals very little information about what is needed to spend the output. The spending party then has to provide a *redeemScript* that hashes to the required 20 byte value. Thus, the pubScript is very small and concise, but the redeemScript in the sigScript can be up to 520 bytes (see (Andresen 2012) for more information).

An example implementing P2PK as P2SH is as follows:

```
pubScript: OP_HASH160 <Hash160{<PubKey> OP_CHECKSIG}> OP_EQUAL
scriptSig: <Sig> {<PubKey> OP_CHECKSIG}
```

Null data pubScripts allow zero value outputs with 80 bytes of arbitrary data. To be standard, each transaction can only have one null data output, and needs to have at least one additional output type. Null data outputs are used for Proof-of-Publication, and as the basis for overlay protocols, as discussed later on. Null data outputs are provably unspendable. This means nodes do not need to consider them in their UTXO set, which is typically held in memory to quickly validate incoming transactions.

The null data pubScript is as follows:

```
OP_RETURN <data>
```

Null data pubScripts will be used in Chapter 5 to embed sensor data in Bitcoin transactions.

The Blockchain

In Bitcoin, transactions are batched into a structure, called a block. A block contains a number of transactions as payload, and a header. The header consists essentially of the root of the merkle tree (c.f. Sec. 3.1.1) that is computed from the transaction hashes, the hash of the previous block, and a number called nonce which we will discuss in the next section. Let us assume for the moment that there is only one node in the Bitcoin network. Since each block refers to its predecessor the emergent data structure is a chain of blocks, a blockchain. Note that linking blocks by means of hash pointers leads to an tamper-evident append-only data structure. If a transaction is changed in

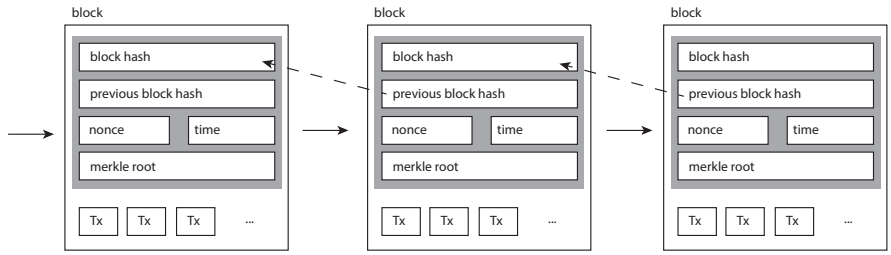


Figure 3.5.: Simplified structure of the Bitcoin blockchain. Each block references its predecessor by a hash pointer. The content of the gray area is the block header.

block i of a blockchain containing n blocks, the chain of hashes breaks, and every hash from block i to block n has to be re-computed.

Therefore, a blockchain is a tamper-evident append-only transaction log that defines the current state by giving its complete history. In Bitcoin, the state that is computed by replaying all transactions is the set of UTXOs, i.e. who owns which coins.

Mining and Nakamoto Consensus

The nodes that batch transactions into blocks are called miners. The reason for this terminology will become clear later in this section. Noteworthy, every participant can decide to be a miner at any time. Thus, there may be (and actually should be) more than one miner in the network, and the set of miners may change continuously. How do we get from local subjective blockchains to one global objective blockchain that implies the global state of the Bitcoin network? In terms of the science of distributed computing: How does the network achieve consensus? The naive (local) rule would be to accept the block that arrives first. However, in a distributed system with finite speed of information propagation, there is no guarantee that two nodes will agree on which block arrived first. Thus, if the two blocks entail conflicting transactions, different nodes will assign the same coins to different owners. This is true even if both nodes are honest. Therefore, consensus protocols are used which involve some form of voting. Traditional consensus protocols such as Paxos (Lamport et al. 2001) and Practical Byzantine Fault Tolerance (PBFT) (Castro, Liskov, et al. 1999) rely on a fixed set of identified consensus nodes. In Bitcoin, however, consensus nodes, i.e. miners, may join or leave the network at any time, and there is no notion of a reliable identity. This is important to provide censorship resistance, and resistance to Denial of Service (DoS) attacks which could stall the network. However, a malicious actor may spin up multiple nodes to gain more weight in the voting process. This is called a Sybil attack (Douceur 2002). The Nakamoto consensus protocol is based on the idea to

couple the weight of a vote with the expenditure of a real-world physical resource, i.e. doing work by spending compute cycles. The approach is inspired by proof-of-work as presented in HashCash (Back 2002). Nodes only consider new blocks to be valid if an appropriate amount of proof-of-work is provided. This is operationalized as follows:

A block has to have an identifying hash that is smaller than a particular target value. Thus, a miner has to vary the block data and re-compute its hash until the target is met. Therefore the consensus protocol is non-interactive and the block generation rate is a function of the combined hash rate of every mining node and the particular target or difficulty. The more hashing power an actor controls, the higher is her chance to find a block. If a node follows the Nakamoto consensus protocol it will broadcast a block as soon as the block has been found. Receiving nodes will check the validity of the transactions, and the proof-of-work. Mining nodes will stop mining their current block, apply the transactions of the new block, and start mining on top of this new block. Thereby, the chain with the highest accumulated amount of work is considered the valid chain which defines the state of the ledger.

If nodes adhere to the Bitcoin mining protocol, the system is resilient to double-spending, and censorship, as long as the majority of hash power is controlled by honest nodes. Eyal and Sirer 2014 shows that a *rational* miner should divert to a different strategy termed *selfish mining*, by which the miner withholds blocks in order to have a head-start to mine the next block, in order to get an unfair advantage. At time of writing, selfish mining has not been observed in the Bitcoin network, but in other blockchain-based cryptocurrencies. It can be argued that a rational miner should follow the selfish mining strategy to maximize her profits. However, the profits are denominated in bitcoin whereas her expenditures are probably to be paid in a national currency. Since the appearance of selfish mining could harm the Bitcoin system as a whole, and thereby have a negative effect on the bitcoin exchange rate, following the Nakamato protocol may still be the rational choice from a long-term ecosystem perspective.

Why would a miner follow the Nakamoto protocol at all considering the resource intensive proof-of-work requirement? This question is closely related to the question: How are bitcoins created?

The answer to both of these questions is that the Bitcoin protocol allows a miner to include one special transaction into each block, a coinbase transaction. These transactions have do not reference any previous transaction outputs, but allow the creation of new coins. The maximum accumulated value of the coinbase outputs of a particular block is fixed in the protocol and gets reduced by 50% every 210,000 blocks. On protocol level, UTXO values are integers denominated in satoshis. 10^8 satoshis correspond to 1 bitcoin. Therefore, after 33 *halvenings* the block reward becomes effectively zero because it will fall below the limit of one satoshi. In addition, the difference between the value of transaction inputs and outputs are identified as transaction fees, and can be

claimed by the miner. However, in 2016, transaction fees play only a minor role which account for less than 2% of a block reward.

In the original white paper proof-of-work was introduced as one-CPU-one-vote. However, as the value of bitcoin appreciated, miners exploited the inherent parallelizability of the proof-of-work algorithm and implemented the algorithm with increasing sophistication. First, on Graphics Processing Unit (GPU)s, followed by Field-Programmable Gate Array (FPGA)s, and eventually on Application-specific Integrated Circuit (ASIC)s (Taylor 2013). Thereby, the energy efficiency has increased by several orders of magnitude. The implication of this evolution is that mining has become subject to economies of scale, and is only profitable with the most-recent specialized hardware, and only with access to cheap electricity. Furthermore, mining is organized in pools. Thereby, individual miners provide their hashing power to a pool operator in exchange for a share of the mining reward. This allows miners to decrease the variance of their expected rewards. Most pools implement centralized transaction selection. This has an inherent risk for censorship if a single pool gets to large or if there is collusion between pool operators. A possible approach to prevent centralized transaction selection are non-outsourceable scratch-off puzzles (Miller et al. 2015).

Merkle Trees and Simplified Payment Verification

In July 2016, the entire Bitcoin blockchain has a size of more than 75 Gigabyte. With a mean block size of 0.8 Megabyte[3], and a block generation rate of 1 per 10 min, the blockchain grows about 115 Megabytes each day. This is a problem even for powerful IOT device categories such as smartphones. Even more limiting is the bandwidth requirement of full nodes, since transaction and block data might be communicated multiple times and to multiple peers.

Bitcoin uses a data structure called a merkle tree (Merkle 1980) to commit transactions to the block header. This allows to efficiently prove the existence of a transaction in a block by communicating only the respective merkle branch, i.e $\mathcal{O}(\log n)$ instead of all n transactions of a block. See Figure 3.6 for an illustration of a merkle tree.

This data structure allows the usage of light clients relying on Simplified Payment Verification (SPV). Light clients only store block headers (80 bytes) and do not validate transactions. Light clients only keep track of transactions involving specific addresses and ask full nodes to provide those transactions together with a merkle proof. The security model of the light client is based on the assumption that only valid transactions are included in the chain of most accumulated proof-of-work.

3 Calculated between May 21 and June 22 2016.

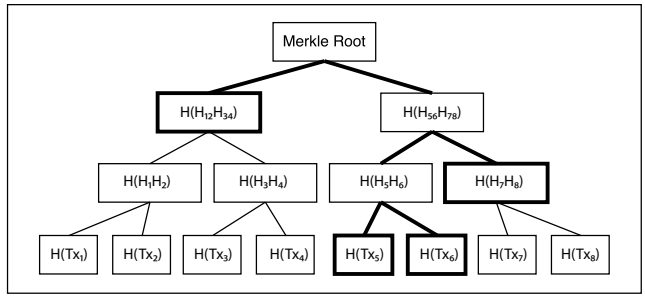


Figure 3.6.: Merkle tree as used in the Bitcoin blockchain. The leaf nodes are cryptographic hashes of transactions. Each parent level consists of pair-wise hashing of the child nodes until the root hash is calculated. In oder to prove the existence of transaction Tx only the marked nodes are needed.

3.1.2 *Interfacing Bitcoin*

The interface to interact with Bitcoin, i.e to spend bitcoins or create addresses to receive bitcoins, is provided by software known as *wallets*. Since Bitcoin is based on public key cryptography and transactions have to be authenticated by ECDSA, storing bitcoins is equivalent to storing and managing cryptographic keys. Wallets have functions to create key pairs, to derive Bitcoin addresses from public keys, and to build and sign transactions based on the available UTXOs.

There exist different models with varying degrees of trust in third parties, depending on who provides the blockchain and who controls the cryptographic keys.

3.1.3 *Challenges and Developments*

Although Bitcoin has repeatedly exceeded a market capitalization of $10 billion it has still to be viewed as an experiment with unknown outcome. Bitcoin faces a multitude of challenges in various fields. Besides the technical challenges, there are also challenges concerning usability, governance, legal status, law enforcement, and even environmental impact. However, it is interesting to note that some challenges concern human-blockchain interaction, and do not apply to machines. In particular, comprehensibility and usability of key management systems are prohibiting Bitcoin adoption of humans, whereas machines do not care.

Scalability

Although often characterized as a distributed system, Bitcoin, and every other current blockchain-based system, is fully replicated. Instead of nodes sharing

work, every (full) node is validating and storing every transaction. Thus far, there is no known alternative in order to provide the same security properties. This global replication has severe consequences on the performance and scalability of the system.

Croman et al. 2016 investigated Bitcoin along key metrics and provided evidence about Bitcoin's scaling behavior without drastic architectural changes. The most important metrics are as follows:

MAXIMUM THROUGHPUT Bitcoin has two main parameters which directly influence throughput, i.e. the number of transactions per second. First, the *block generation rate* which is controlled to have an expectation value of 10 min, and second the *block size limit* which was introduced later to prevent DoS attacks. This leads to an effective throughput of about 3.5 transactions per second. In contrast, VISA handles on average about 2,000 transactions per second and has a peak capacity of 56,000 transactions per second (Visa n.d.).

LATENCY The time until a transaction is *confirmed*, i.e. the time until the transaction is included into a valid block, is roughly 10 min in expectation, but may be longer if the network load is high and a transaction provides not enough fees. For transactions of higher value it is advisable to wait for more than one confirmation, i.e. to wait until the block is sufficiently deep in the blockchain and reorganization attacks become unfeasible. The standard number of confirmations, as stated in the original white paper, is six which translates to one hour.

BOOTSTRAP TIME A node that joins the network has to download and process the entire transaction history. This bootstraping time scales linearly with time and is already on the order of days.

At first sight, it seems that Bitcoin is by no means able to provide a global payment network. However, there are multiple developments and approaches to scalability. One is to decrease the data that has to be stored per transaction. Segregated witness (Lombrozo, Johnson, and Wuille 2015) and new, compressible signature schemes are examples. Another is to increase the block size limit. However this change would involve a hard fork, meaning that non-updated clients would discard the new blocks resulting into two chains, and effectively two currencies. Moreover, increasing the block size or decreasing the latency has two important side effects. First, it increases the load of nodes due to more traffic, increased demand for validation, and storage. Second, due to relatively longer propagation times the rate of orphaned blocks, i.e. blocks that do not end up in the main chain, would increase. This would lead to decreasing miner profitability and thus to an increased centralization of mining. Approaches to mitigate this problem is to include, or at least compensate, orphaned (ommer) blocks (Kiayias and Panagiotakos 2016; Lewenberg, Sompolinsky, and Zohar

2015; Sompolinsky and Zohar 2015), or to allow a block to reference multiple parents, leading to a Directed Acyclic Graph (DAG) instead of a chain.

Another suggestion is to use proof-of-work only for leader selection (Eyal, Gencer, et al. 2016) or group assignment (Kokoris-Kogias et al. 2016) in order to establish temporary identities to use a traditional and faster signature-based byzantine fault tolerant consensus protocol for block ordering.

Ideally, the performance of a distributed system increases with the number of nodes. The traditional method from distributed databases is known as sharding (e.g. Google's Spanner (Corbett et al. 2012)). Thereby, the network is partitioned into smaller committees, each of which processes a disjoint subset of transactions, the shards. Sharding in byzantine environments, however, is an open problem (Croman et al. 2016), and has gained interest in the research community only recently (Luu, V. Narayanan, et al. 2016; Gencer, Renesse, and Sirer 2016).

Besides changing Bitcoin itself, it is possible to achieve higher transaction rates and almost instant confirmations with systems built on top. We will discuss this approach in more detail in chapter 6.

Privacy

Bitcoin provides pseudonymity due to the fact that bitcoin addresses are not linked to real identities, but to self-generated public keys. Every user can create any number of public keys, and today most wallet software avoids the reuse of bitcoin addresses. This means change is sent to a newly generated address. However, the blockchain provides a complete and public visible history of every transaction. Multiple studies have shown that individual profiles can be recovered from the transaction graph (Ron and Shamir 2013; Androulaki, Karame, et al. 2013; Reid and Harrigan 2013; Babaioff et al. 2012; Ober, Katzenbeisser, and Hamacher 2013; Spagnuolo, Maggi, and Zanero 2014).

If this profile can be linked to a real identity at the edges, i.e. at the exchanges or a merchant, privacy is completely lost. There are already companies that specialize on analyzing the transaction graph[4].

Privacy is also important for fungibility. In Bitcoin every coin has a public history, because an UTXO is the edge of a graph. If coins can be connected to illicit usage or theft, regulated parties may not be able to accept them. This means the value of particular coins might be less than others, which breaks fungibility and could lead to a deterioration of trust in Bitcoin as a whole.

Furthermore, there is a privacy risk on the network level by observing from which node a particular transaction originates (P. Koshy, D. Koshy, and McDaniel 2014; Biryukov, Khovratovich, and Pustogarov 2014).

4 The most well known are Skry (`https://skry.tech`) and Chainalysis (`https://www.chainalysis.com`).

There are many developments to increase the privacy of Bitcoin, and cryptocurrencies in general. Since it is possible for multiple parties to create a transaction collaboratively, i.e. multiple parties are able to add inputs and outputs to a single transaction, it is possible to mix coins from various people without trusting a third party service. Depending on the number of mixings and the number of inputs and outputs involved, connecting inputs to outputs becomes infeasible. Methods based on this idea are CoinJoin (Greg Maxwell 2013; Meiklejohn and Orlandi 2015), MixCoin (Bonneau et al. 2014), CoinShuffle (Ruffing, Moreno-Sanchez, and Kate 2014), and BlindCoin (Valenta and Rowan 2015).

Confidential transactions (Gregory Maxwell 2015) allow to keep the transacted amount private between sender and recipient. This enhances the privacy promises of mixing techniques enormously. As of now, the transaction structure and the cryptographic primitives available in Bitcoin script do not permit the implementation of confidential transactions.

Zerocoin (Miers et al. 2013) is a cryptographic extension to Bitcoin to allow fully anonymous transactions based on zero-knowledge proofs. However, this comes at the cost of increased computational complexity of verification and transaction size, such that the system is more of theoretical importance than of practical. Androulaki and Karame 2014 introduce a further extension to Zerocoin with additional properties similar to confidential transactions.

Usability

Bitcoin is based on public key cryptography. Thus, users have to manage private keys securely. Losing a private key means losing access to the bitcoins protected by the respective key. This is different to the common password-based authentication, where the issuing party always has the power to reissue a new password. However, wallet software, the usage of multi-signature accounts, and other methods based on Bitcoin script such as covenants (Möser, Eyal, and Gün Sirer 2016) are evolving to provide increasing usability. A good overview of key management issues and remedies is provided by Eskandari et al. 2015.

Another usability issue for Bitcoin is its volatility in terms of exchange rate to major fiat currencies (Luther and White 2014; Sapuric and Kokkinaki 2014). This aggravates the use of bitcoin as an unit of account, and imposes currency risk on holders. Most merchants accepting bitcoins use payment processing services that allow a flexible disbursement in a mix of bitcoin and fiat. Thus, absorbing some of the currency risk.

Governance

As a decentralized network with various stakeholders there is the question how decisions concerning development and updates are made. Above the consensus level, there is permissionless innovation. Individuals and companies

can build software and services that interface with the Bitcoin network and can compete for users. However, changes of the consensus layer may lead to two different networks, and thus to two different currencies. Changes which further restrict the consensus rules are called soft forks. As long es the majority of mining power accepts the new rules, the network will not split. However, if the changes are such that non-updated clients are not able to accept blocks generated by updated clients, then there will be two chains even if only a minority of mining power stays with the old client. This scenario is called a hard fork.

There is a discord in the community concerning the question, if hard forks are a safe way to do protocol updates, and if there should be hard forks on a regular basis. This discord is in particular visible in the block size debate. A. Narayanan and Miller 2015 stresses the importance of the governance issue, and provide three suggestions for better governance.

Environmental Footprint

Mining bitcoins is energy-intensive on purpose. The mining process ensures scarcity and integrity. O'Dwyer and Malone 2014 estimated the combined electricity consumption of Bitcoin miners to be comparable to the energy consumption of Ireland. Since the time of this estimation, in early 2014, until now, in Q4 2016, the hashing rate has increased by a factor of more than one hundred. Although mining hardware has become more efficient, it is highly probable that the energy consumption has increased further.

Other consensus protocols for decentralized cryptocurrencies have been proposed that omit the continuous expenditure of energy. The most discussed approach is proof-of-stake (BitcoinTalk 2011). The intuition behind proof-of-stake is that each participant has voting power according to her stake in the system, i.e. the number of coins she owns, instead of voting power according to computing power in the proof-of-work scheme. Simple protocols based on this idea have been shown to be flawed (Poelstra 2014). More complex protocols including security deposits and the ability to punish dishonest participants have been developed recently (Kiayias, Konstantinou, et al. 2016; Bentov, Pass, and Shi 2016). However, thus far, there no implementations of these advanced protocols available.

3.1.4 *Building Applications beyond Currency*

In this section we discuss how Bitcoin and the underlying technological paradigm, the blockchain, can be used as a platform for applications beyond electronic cash. As Marc Andreessen put in an op-ed article in the New York Times: "Bitcoin gives us, for the first time, a way for one Internet user to transfer a unique piece of digital property to another Internet user, such that the transfer is guaranteed to be safe and secure, everyone knows that

Challenge	Description	Approaches
Scalability	Max. throughput limited to less than 7 tx/s	Off-Chain payments, DAGs, sharding
Privacy	Public history allows analysis and potential identification	Mixing, homomorphic encryption, zero-knowledge proofs
Usability	Handling of cryptographic keys, comprehensibility, volatility	Covenants, education, integrated applications
Governance	Decision making in decentralized systems	Advisory board, public commenting, consensus on evaluation criteria
Environmental Footprint	Proof-of-work is energy intensive	Proof-of-stake, proof-of-space

Table 3.1.: Summary of Bitcoin challenges and respective approaches.

the transfer has taken place, and nobody can challenge the legitimacy of the transfer. The consequences of this breakthrough are hard to overstate" Andreessen 2014.

Indeed, already in late 2010, the Bitcoin community targeted another type of digital property: domain names [5]. But how to implement a domain name system on Bitcoin?

Altcoins

One approach is to bootstrap a new, application-specific system. In April 2011, the first block of *Namecoin* was mined. Namecoin is a fork of Bitcoin with a transaction structure adapted to need of a domain lookup system. Then there is the question, how to secure the system? In a proof-of-work-based system there has to be enough decentralized hashing power to make 51% attacks unfeasible. Due to the enormous hashing power of the Bitcoin network, there are essentially two possibilities for another proof-of-work-based system: (1) The use of an Bitcoin-incompatible proof-of-work schema and boostrapping an entire new network, or (2) merged-mining which allows Bitcoin miners to reuse their proof-of-work to secure the alternative coin concurrently. Namecoin was the first merge-mined coin, and the network is still running. Kalodner et al. 2015 provides an empirical investigation of Namecoin.

Colored Coins

The idea of colored coins is to taint a bitcoin, i.e. a particular transaction output, in order to attach an external value or meaning to it, independent of its bitcoin value (*Overview of Colored Coins* 2012). The technique has been used to issue and track/trade securities such as company shares and stocks, as well gift cards on the Bitcoin blockchain. There are different non-interoperable protocols that are used in practice. Typically, colored coin transactions utilize a

5 https://bitcointalk.org/index.php?topic=1790.0

null-data output for protocol instructions. Thus, colored coins are transparent for regular Bitcoin clients. Since miners do not validate the specific rules inherent in colored coin transactions, the SPV model breaks for colored coins.

A general problem with securities as colored coins is that those are not digital bearer assets, since they are aways augmented by an issuing counterparty[6]. This is a very important point which is often neglected when various assets are *tokenized and put on a blockchain*.

Virtual Chains and Meta Coins

Virtual chains can be interpreted as embedded consensus system that operate on a layer on top of Bitcoin. Bitcoin is used only for proof-of-publication of virtual chain instructions. The term virtual chain was introduced and formalized as part of Blockstack, the technology stack on which OneName is built (c.f. A.6), after switching from the Namecoin blockchain (Ali et al. 2016).

A similar approach is used by meta coins such as Mastercoin (now Omni[7]) and Counterparty[8] to build a general purpose asset and smart contract platform on top of Bitcoin, and thus leverage the security properties of the Bitcoin blockchain.

However, these embedded consensus systems can not provide the same level of light client security as simplified payment verification in Bitcoin itself. Bitcoin nodes, and miners in particular, are unaware of the consensus rules of the overlay protocol[9].

3.1.5 *Sidechains*

Sidechains are separate networks like alternative coins, however, the value of those coins is *pegged* to the value of bitcoins (Back et al. 2014). The idea is that bitcoins can be frozen on the Bitcoin blockchain, and created or unlocked on the sidechain. At a later stage the sidechain coin can be frozen on the sidechain and again unlocked on the mainchain. Thus, the sidechain does not need to bootstrap value. In order to allow a low-trust *two-way peg*, Bitcoin and the sidechain must be able to perform simplified payment verification of transactions on the other chain. Bitcoin's scripting language does not yet allow SPV. A federated sidechain has been introduced by Blockstream[10] in order to provide fast transfers between Bitcoin exchanges. Furthermore, SPV of Bitcoin transactions in form of an Ethereum contract has been implemented[11].

6 See (Swanson 2015b) for a detailed discussion.
7 http://www.omnilayer.org/
8 http://counterparty.io/
9 Meta coin instructions are typically stored in null-data outputs.
10 https://blockstream.com/2015/10/12/introducing-liquid/
11 http://btcrelay.org/

3.1.6 *Non-currency Blockchains*

Alternative blockchains that do not issue their own currency in order to incentive miners use different consensus system than proof-of-work and therefore have different security guarantees. In most cases, a fixed set of *validators* is assumed which run traditional byzantine-fault-tolerant consensus algorithm. Therefore, non-currency blockchains tend to be federated systems and their value hinges on the structure of the particular federation. The most notable developments are the Hyperledger project[12] under the umbrella of the Linux Foundation, and R3's Corda (Brown et al. 2016). The Hyperledger project is not a single blockchain but aims to establish a modular platform for building distributed ledgers. Corda is an open source distributed ledger platform to meet the requirements of the financial services industry.

3.2 ETHEREUM: PLATFORM FOR SMART CONTRACTS AND DECENTRALIZED APPLICATIONS

In Sec. 3.1.4, we saw that, given Bitcoin, there are essentially two approaches to build a new blockchain-based decentralized application: (1) Building on an abstracted layer on top (virtual chains, meta coins, colored coins), or (2) building a new network aside (altcoins, sidechains) of Bitcoin. Since Bitcoin nodes are blind to overlay consensus rules, light client protocols with comparable security to SPV are not possible, and overlay nodes have to keep track of all transactions. Altcoins, in contrast, face the issue that they have to bootstrap an entire new network.

Ethereum aims to solve this dilemma by providing a blockchain-based platform for building decentralized applications. As the white paper states: "Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions" (Buterin 2014).

The idea of Ethereum was described by Vitalik Buterin in 2013. The crowdsale (c.f. Sec. 4.3.3), which started in July 2014, collected more than $18 million worth of bitcoin. One year later, in July 2015, the initial release of the Ethereum network went online.

In Q4 2016, ether, the native cryptocurrency of Ethereum, is valued at more than $10, and has a market capitalization of more than $1 billion. According to Ethernodes[13], there are almost 10,000 Ethereum nodes globally. Furthermore, private Ethereum networks are being implemented by various big corporation to test blockchain-based applications. This makes Ethereum the second most important cryptocurrency after Bitcoin.

12 https://www.hyperledger.org/
13 https://ethernodes.org

In this section, Ethereum is introduced with a particular focus on its main differences to Bitcoin.

3.2.1 *The Technology*

The review of Ethereum's technology is based on the white paper (Buterin 2014) and the yellow paper (Wood 2014).

Accounts, Contracts and Transactions

As discussed in Sec. 3.1.1, Bitcoin is based on the UTXO model. Transactions spend one or more UTXO and create one or more UTXOs. The implicit global (consensus) state , emerging from the blockchain, is the set of UTXOs. In contrast, Ethereum uses the more intuitive model of accounts, which keep track of an ether balance. There are two types of accounts: externally owned accounts (e.g. user accounts), and internal accounts, called contracts. Externally owned accounts are authenticated by public key cryptography, and can be used to instruct the network by means of transactions. Contracts are arbitrary programs formed by a Turing-complete instruction set, persistent storage, and an ether balance.

A transaction is a cryptographically signed data structure comprising the following data fields:

```
nonce: An incrementing integer value,
gasPrice: Price per unit gas in wei.
gasLimit: Maximal amount of gas to be paid for executions of the
    transaction.
to: Receiver account.
value: Value to be transferred to receiver in wei.
init: Code to instantiate a new contract (optional).
data: Input data.
```

If all we wanted to do were to send ether from one account to another, then it would essentially suffice to specify *to* and *value*. However, due to the account model, the receiver could broadcast the transaction repeatedly to empty the sender's account. To avoid this *replay attack*, the *nonce* is used. Every transaction from an account can only use a nonce once. Transaction fees to miners are paid in gas. The amount of gas needed is determined by the Ethereum Virtual Machine (EVM). The exchange rate between gas and ether can be defined by the user with the *gasPrice* data field. Miners are free to decline the transaction, if the exchange rate is set too low[14]. The *data* field is used to provide input for contract function calls. The *init* field is used to provide the byte code to instantiate a new contract.

14 A review of the gas economics can be found at https://github.com/LeastAuthority/ethereum-analyses/blob/master/GasEcon.md.

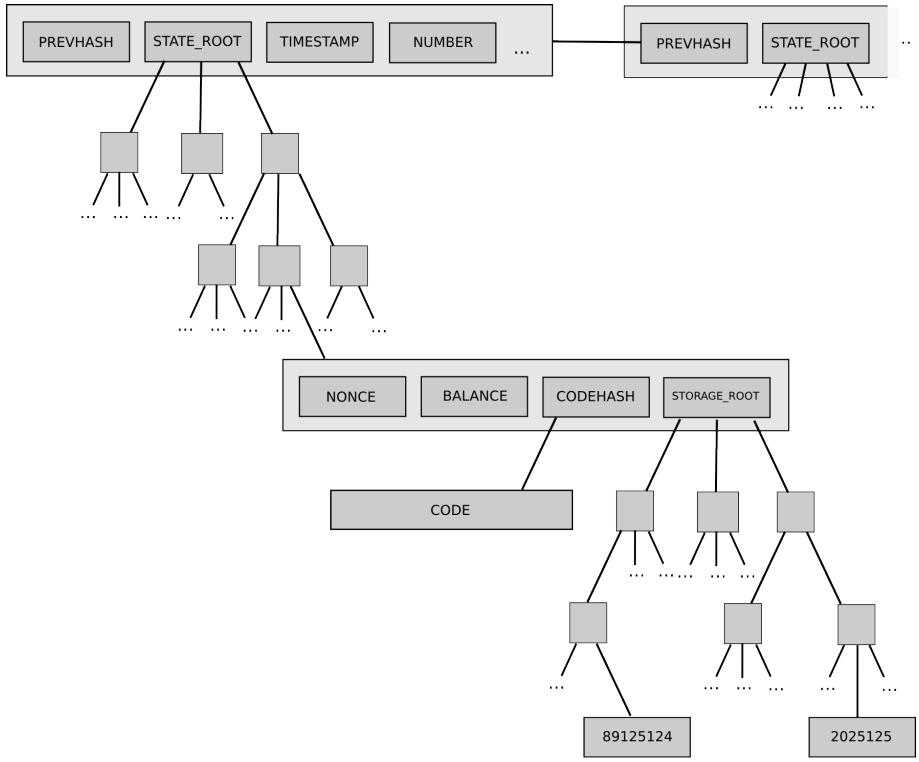


Figure 3.7.: Simplified structure of the Ethereum blockchain (Ethereum Wiki 2016a).

Blockchain and Consensus

Similar to Bitcoin, Ethereum blocks entail a number of transactions. However, in contrast to Bitcoin, block headers commit to the current global state by including the hash of the state trie root. Each leaf node of the state trie corresponds to the current state of an account (see Fig. 3.7). This allows to efficiently prove the balance of an account, or the state of contract variables. However, it requires nodes to recalculate the trie with every transaction. Ethereum uses modified prefix trees (tries), called merkle-patricia trees, instead of simple merkle trees. This allows to efficiently recalculate the state trie based on the effects of individual transactions which only affect a tiny subset of all accounts.

A typical Ethereum block header has a size of approximately 500 bytes, more than 6 times the size of a Bitcoin block header.

Ethereum uses a different proof-of-work algorithm than Bitcoin, which aims to be ASIC-resistant and GPU-friendly[15]. The rationale behind is to enable

15 See https://github.com/LeastAuthority/ethereum-analyses/blob/master/PoW.md for a review of *Ethash*.

commodity hardware (GPUs) to stay competitive. Moreover, Ethereum is scheduled to switch to a proof-of-stake consensus mechanism in 2017.

Inspired by Greedy Heaviest Observed Subtree (GHOST) protocol (Sompolinsky and Zohar 2013), Ethereum partially rewards ommer[16] blocks to disincentive block withholding (i.e. selfish mining (Eyal and Sirer 2014)). However, in contrast to GHOST, transactions in ommer blocks have no effect, i.e they are not processed, and have to be included in another block. This allows a targeted block time of 15 s.

Light clients

Similar to Bitcoin, we define a light client to be a client that only downloads block headers. Because of the extensive use of tries and their root commitment to the block header, Ethereum light clients are much more powerful than Bitcoin light clients. Most importantly, due to the state trie, an Ethereum node can directly prove an account balance to a light client. Furthermore, logs can be used in contracts to make light clients aware of particular events. A full node is then able to prove the exact content of the event. However, the much higher block generation rate and the larger size of block headers requires more resources from light clients.

At time of writing, the first light client implementation[17] just entered public testing.

The Ethereum Virtual Machine

Ethereum allows to deploy stateful Turing-complete contracts. These programs are executed by the EVM. Essentially, the entire Ethereum network emulates a single instance of a trusted virtual machine. However, Turing-completeness implies that a contract may be an infinite loop. This would render the entire network unusable. Thus, the EVM employs the concept of gas. Every execution of an instruction in the EVM costs a specified amount of gas. If the execution of a transaction runs of out gas, the effect of the transaction is rolled back, but the gas is still consumed. Hence, the infinite loop problem is avoided by applying economics.

Developers usually do not work with the stack-based low-level scripting language, but with higher-level abstractions. Currently there exist three of those programming languages, which are all inspired by traditional programming languages. The most popular is Solidity, which is inspired by JavaScript. We will use Solidity to write smart property contracts in Section 7. In addition, there are Serpent (Python) and LLL (Lisp).

16 Ommer is the gender neutral description of aunt/uncle.

17 https://github.com/zsfelfoldi/go-ethereum/wiki/Light-Ethereum-Subprotocol-%28LES%29

3.2.2 *Smart Contracts and Decentralized Applications*

Ethereum contracts are programs that are evaluated by the EVM, and are thus executed by a trusted computing instance. Hence, these programs can be used to emulate functionalities that are traditionally provided by Trusted Third Party (TTP)s. The term *contract* for these programs originates from the concept of *smart contracts* introduced in (Szabo 1997). In this sense, smart contracts are self-enforcing contracts implemented as computer code. A simple example of a smart contract implementing a fungible currency with a fixed supply is given below[18]:

```
contract MyCurrency {

    uint256 public totalSupply;
    uint256 public fx;

    /* Provides a ledger mapping addresses to balances*/
    mapping (address => uint256) public balanceOf;

    /* Initializes MyCurrency with total supply and price in wei */
    function MyCurrency(uint256 _totalSupply, uint256 _fx) {
        totalSupply = _totalSupply;
        fx = _fx;
    }

    /* Allows to buy MyCurrency with ether*/
    function buyMyCurrency() payable {
        if (totalSupply < msg.value/fx) {
            throw;
        }
        else {
            balanceOf[msg.sender] += msg.value/fx;
        }
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) {
        if (balanceOf[msg.sender] < _value) throw;
        if (balanceOf[_to] + _value < balanceOf[_to]) throw;
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
```

18 The code is inspired by https://www.ethereum.org/token

```
    }
}
```

The contract is expressed in Solidity and provides a ledger keeping track of balances. When instantiated, the total supply and the exchange rate between *MyCurrency* and ether (or wei) are set. *MyCurrency* can then be bought from the contract using the *buyMyCurrency* function. Afterwards, the buyer can send *MyCurrency* to arbitrary addresses using the *transfer* function. All functions are invoked by sending transactions to the contract. Data of these transactions can then be accessed from within the contract. For example, `msg.sender` provides the Ethereum address from which the transaction originates, and `msg.value` provides the value in wei of the transaction. Thus, all Ethereum contracts can leverage the built-in public key infrastructure for permissions and authentication, and act as custodians of funds. The simple example could be easily extended with a more interesting monetary policy or an auctioning function. However, many smart contracts need to act on external data, and are thus dependent on the trustworthiness of the data provider. An example could be a data-driven smart contract for application in a pharmaceuticals supply chain. Many pharmaceuticals are required to be kept within certain temperature limits. A trusted temperature sensor could provide temperature measurement data to the contract which only pays the supplier if the temperature was within the limits.

3.2.3 *Challenges and Developments*

Ethereum faces the same challenges as Bitcoin. Many of those challenges are even more severe, e.g. scalability and privacy. In addition, there are unique challenges.

Verifiers Dilemma

Processing and validating transactions in Ethereum can require non-trivial computational effort. A block containing a transaction that requires non-trivial computational effort creates a dilemma for the other nodes. Since the gas is only collected by the miner who includes the transaction into a block, there is no direct economic incentive for other nodes to spend the computation power to validate the transaction. However, if the transaction is invalid, then the node would end up in an incorrect chain, at least temporarily. The verifier's dilemma is presented in (Luu, Teutsch, et al. 2015).

Contract Security

High-level contract programming languages like Solidity give the impression that contract development is easy. However, contract development has unique

challenges. Since contracts mostly have a financial element, as well as multi-party interactions, there is a need for *game-theoretic debugging*, in order to ensure incentive compatibility. Delmolino, Arnett, A. Kosba, et al. 2015 and Delmolino, Arnett, A. E. Kosba, et al. 2015 discuss several possible issues that came up during the first contract development lab class at Cornell University. A more recent example is *The DAO* contract[19], which collected approximately $150 million worth of ether. Mark, Zamfir, and Sirer 2016 points out several game-theoretic attack vectors. Later, an attacker combined two non-game-theoretic, EVM-related bugs in order to attempt to claim more than $50 million worth of ether (Daian 2016). Luu, Chu, et al. 2016 describes three security bugs in contracts. One of them, mishandling of exceptions, was part of the The DAO attack. Furthermore, the papers presents evidence for numerous appearances in deployed contracts.

3.3 COMPARISON

In the last two sections we presented the two most important cryptocurrencies, Bitcoin and Ethereum. To conclude this presentation, we provide a concise comparison and discussion. Table 3.2 compares Bitcoin and Ethereum along a set of dimensions that have appeared explicitly and implicitly during the last two sections. Bitcoin's main scope is to provide a peer-to-peer electronic cash system. Therefore, focus is on security and decentralization. Further important goals are privacy and anonymity, despite the public nature of the system. For many users Bitcoin has taken an additional role of traditional money besides acting as a medium of exchange, it provides a store of value. In order to keep those characteristics, the Bitcoin community[20] is reluctant to controversial changes. From an original design perspective, ether is considered as fuel to provide incentives for a decentralized computing platform. However today, most transactions on the platform are purely financial transactions between externally owned accounts, and do not involve contracts. Even if contracts are involved, their nature is nevertheless mostly financial. This might change in the future. As a developer platform, the focus is on ease of use and developer empowerment (c.f. accounts vs. UTXOs, Solidity vs. Bitcoin script). These characteristics sometimes oppose security, as can be seen most prominently in the DAO incident. Ethereum core developers follow a much more progressive agenda. For example, Ethereum is embracing *hard forks*, protocol upgrades which have to be done by all network participants, in contrast to *soft forks* which require only miners to upgrade. In fact, Bitcoin had his last hard fork, caused by a software bug, in March 2013[21], whereas Ethereum had multiple hard forks in 2016 alone. Due to the employment of a modified version of

19 https://github.com/slockit/DAO

20 It is important to note that the Bitcoin community is a diverse set of actors with diverting goals.

21 See https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki for a detailed description.

	Bitcoin	Ethereum
Main Scope	E-cash	Decentralized computing platform
Focus	Security & Decentralization	Ease of use
Development	Conservative	Progressive
Model	UTXO	Account
Scripting	Restricted	Turing-complete
Block Time	10 min	15 s
Header Size	80 bytes	>500 bytes
Fees	Tx size	Instructions, storage etc.
Incentive Issues	Selfish mining	Verifier's dilemma
Monetary policy	Max 21 million	Uncertain

Table 3.2.: Comparison between Bitcoin and Ethereum.

the GHOST protocol, Ethereum can handle a higher block generation rate, and allows a higher transaction throughput. However, the higher block generation rate in combination with the larger header size leads also to higher bandwidth and storage requirements for light clients. Based on a block generation rate of $\frac{1}{10min}$, a header size of 80 bytes for Bitcoin, and a block generation rate of $\frac{1}{15s}$, and header size of 500 bytes for Ethereum, the requirements for Ethereum are increased by a factor of 250. Another important difference is the monetary policy. Bitcoin is set with an issuance schedule following a geometric function, and accumulated issuance of approximately 21 million bitcoins. Ethereum currently has a fixed reward of 5 ether per block (Ethereum Wiki 2016d)[22]. The reward is not set to decrease automatically as in Bitcoin. However, this will probably be changed with a future hard fork protocol update.

Blockchains and the End-to-End Principle

One of the main design principles that led to the success of today's Internet is the End-to-End (e2e) principle (Saltzer, D. P. Reed, and Clark 1984). The e2e principle states that *intelligence* should be at the edges, at the top of a layered system. This implies the strict separation of application-specific matter, and the actual means of communication and transport. This is what allows the uncountable applications and use cases of the Internet and low cost for communication. Taking this perspective, what does it mean for blockchains? To answer this, we would have to define the basic function of a blockchain. Is it a decentralized, censorship-resistant publication platform? Is it a peer-to-peer value transfer system? Or is it general purpose decentralized state-transition system? We do not know yet, however all blockchains today employ the notion of a digital asset, i.e. value, that can be transferred between accounts augmented by public key cryptography. Therefore, we will take the perspective, that the basic functionality is a peer-to-peer value transfer

22 Plus a possible reward for including a valid ommer block.

system. Just as the Internet transfers duplicate-able information, a blockchain transfers non-duplicable information, i.e. digital scarcity. From this perspective Bitcoin follows the e2e principle. Particular applications, e.g. the transfer of a special asset, is done on a higher level (c.f. colored coins, meta coins and virtual chains), and logic is contained mainly in application-specific clients. In contrast, in Ethereum, every node is aware of every application. Thus, violating the e2e principle.

Internet of Things Perspective

There are two main considerations for the application in the IOT. The first are requirements and capabilities of light clients, since most connected devices are resource constrained, and can not take part in a blockchain network as full nodes. The second are protocol upgrades in form of hard forks. Connected devices are mostly embedded devices with limited direct user interaction. In the future, all kinds of devices might have wallets and transact autonomously with each other. In case of a hard fork every device has to be upgraded, since upgraded and not-upgraded devices are incompatible, and transactions are essentially based on different currencies. However, who decides and who initiates an upgrade? Therefore, avoidance of hard forks could be very important for IOT deployments.

3.4 PERMISSIONED BLOCKCHAINS

Starting in 2014, a common theme in the financial service industry has been *to embrace the blockchain*, however, without the currency aspect. This lead to the notion of permissioned blockchains, and several start ups, large corporations, and industry consortia have been formed around this approach (c.f. Sec. 4.3.4). The main point of permissioned blockchains is that transaction validators (in the sense of miners) are permissioned and known entities, instead of anonymous ad-hoc miners. Non-validating nodes can have adjustable access rights. Individual users, i.e transaction originators, may also be permissioned. This can be done either directly identify individuals or by privacy-preserving permissioning (Thomas Hardjono 2016; Hardjono and N. Smith 2016).

Permissioned blockchains can be implemented in single companies, but more importantly across company and industry boundaries. Since validators are fixed and known entities, there is no need for a mining process, and no need for incentivization through minting new coins. Thus, permissioned blockchains usually do not have a native token or cryptocurrency. Instead of proof-of-work-based consensus, permissioned blockchains are typically based on variants of traditional byzantine fault-tolerant consensus protocols based on leader election such as Practical Byzantine Fault Tolerance (PBFT) (Castro, Liskov, et al. 1999). A prominent representative, Hyperledger (Cachin 2016),

aims for a modular architecture that enables the replacement of a particular consensus protocol.

Swanson 2015a argues that only permissioned blockchains are suitable to track *off-chain assets*, i.e. assets which are not entirely governed by the blockchain, but involve a counterparty. The main reason is that legally-binding transfers of off-chain assets should not be subject to possible blockchain reorganizations caused by anonymous non-accountable miners. However, while this risk exists in principle, in practice the causation of long-range reorganization attacks is very expensive, and has not been observed yet.

Table 3.3 shows a taxonomy based on transaction validation (e.g. mining) and transaction access. Transaction access thereby means who can access transactions. Bitcoin and Ethereum are examples of permissionless and public blockchains. Transaction validation can be done by everyone and is incentivized by cryptocurrency rewards. Access regulation can be built on top of permissionless blockchains. For example colored coin transactions are able to conceal what is actually transacted to the parties involved. Furthermore, multisignature outputs that require the signature of an authorizing party can be used to regulate transactions on a permissionless blockchain. Permissioned blockchains employ authorized transaction processors instead of voluntary miners. Since transaction validators are fixed, they have to be secured against DoS and other cybercriminal attacks. Permissioned blockchains are federated systems and can have public transaction access. An example would be the Interplanetary Database[23]. Security and censorship resistance of such systems are highly dependent on the number and diversity (in terms of incentives, location, legislation etc.), and the governance structure. Most permissioned blockchain projects can be classified as regulated or private, where all participants are permissioned in some form. This allows for greater control and privacy. On the other hand, the security model might approximate that of a traditional shared database (A. Narayanan 2015b).

3.5 CONCLUSION

Bitcoin is the first implementation of an electronic cash system. Digital property has always been reliant on trusted third parties to prevent the double spending problem, and thus digital bearer instruments, such as electronic cash, were impossible. In contrast, Bitcoin is based on open source software that establishes a decentralized peer-to-peer network maintaining a blockchain, a public cryptographically- and thermodynamically-secured replicated database. Bitcoin is programmable money and can be used to express simple forms of self-enforcing smart contracts, and to reduce the need for trust in certain transactions. However, Bitcoin's architecture and programming model is restricted in order to keep the system decentralized, lean and secure. Ethereum

23 https://ipdb.foundation/

		Transaction validation	
		Permissioned	**Permissionless**
Transaction access	**Public**	Transparency but no censorship resistance. Security by traditional means. DoS Attacks possible.	Cryptocurrencies. Censorship resistance and security provided by cryptoeconomics
	Regulated	Stakeholders may have transparency and auditability. No Censorship resistance.	Regulated systems embedded in permissionless blockchains (e.g. authorized securities trading based on colored coins)
	Private	Access limited to transaction validators	Not applicable

Table 3.3.: Taxonomy of blockchain technology by transaction validation and transaction access based on (BitFury 2015).

is generalizing the blockchain concept to provide a decentralized trusted Turing-complete virtual machine with an integrated cryptocurrency, that can act as a platform to implement arbitrary smart contracts and decentralized applications. Public permissionless blockchains currently have limitations concerning scalability, privacy, usability, governance, and their environmental footprint. However, researchers, industry and the open source community are working on improvements.

Cryptocurrencies are subject to network effects, and the technological underpinnings are only one factor for the success of one or the other. Thus, a consideration of the ecosystem is important.

4 BITCOIN AND BEYOND: ECONOMIC PERSPECTIVE

One general law, leading to the advancement of all organic beings, namely, multiply, vary, let the strongest live and the weakest die.

— Charles Darwin

The last chapter introduced Bitcoin and its underlying mechanics, such as the blockchain, from a technological perspective. This chapter provides a complementary perspective by investigating the Bitcoin, cryptocurrency and blockchain business ecosystem. Only rarely have software projects, originating from a single developer, or a small group of developers, such a profound economic impact, spinning up entire ecosystems of start-ups, open source projects, a new age of non-governmental monies, novel business models, and cross-industry consortia involving some of the world's largest corporations. This chapter retraces the evolution of the narrow and extended Bitcoin ecosystem, and provides empirical evidence that Bitcoin is still the dominant cryptocurrency. Venture capital investment is almost non-existent for companies focusing on public blockchains besides Bitcoin. However, Bitcoin implicitly introduced and proved a novel business model, the *appcoin* or *Nakamoto* business model, that may allow to fund and sustain decentralized organizations, applications and software protocols. Although the biggest experiment, The DAO - a form of democratized and decentralized venture capital fund implemented as Ethereum smart contracts, has failed, the combined market capitalization of all blockchain-based coins, excluding Bitcoin, is almost $2 bn USD, and almost $117 million USD have been collected in ICOs since 2013. Although the legal status of these experiments is still undefined and heavily debated, the number of these experiments is likely to increase in the near future.

After the presentation of the blockchain ecosystem based on empirical data, the chapter concludes with a discussion of the economic relevance of cryptocurrencies based of five characteristics, i.e. inclusion, innovation, disintermediation, automation, and new business models.

4.1 DATA AND METHOD

BITCOIN ECOSYSTEM To attain a wide dataset that adequately represents the presently existing Bitcoin start-up ecosystem, a variety of publicly available online sources was utilized: The start-up platform AngelList[1], searched with

1 https://angel.co/

the broad keyword *cryptocurrency*, a compiled list of crypto technology companies by industry expert Mougayar 2015 and Coindesk 2015, a reputable and well known bitcoin news site, which maintains a comprehensive list of venture capital invested into Bitcoin start-ups. To validate and add additional data to the compiled set we used the websites of the individual companies as well as press releases and the Crunchbase database[2]. We identified a total of 704 start-ups and projects in the cryptocurrency and blockchain space, which are existent today. Of these, 599 belong to the Bitcoin ecosystem, and 65 received venture capital. Following an iterative process, we identified representative categories that help understanding the Bitcoin ecosystem.

BLOCKCHAIN ECOSYSTEM BEYOND BITCOIN A list of publicly traded blockchain-based tokens together with market price and market capitalization was acquired from Coinmarketcap[3]. The description was added by the author based on consultation of the relevant project websites.

Venture capital investment numbers are taken from Coindesk Venture Capital[4], and provide only publicly disclosed funding. Classification in different categories (altcoin, appcoin, bitcoin, ethereum, permissioned) was done based on consultation of the relevant project websites. Data concerning ICOs was collected from Cyber Fund[5]. Only completed and successful ICOs were considered. Funding amounts were given in bitcoins, and converted to USD based on the exchange rate at the end date of the respective ICO. The exchange rate was taken from Coindesk. The underlying blockchain of the coins was added by the author based on the project websites.

Data concerning blockchain consortia is based on (Mougayar 2016) and the consortia websites. Consortia with no information about members were excluded.

All data was acquired on 2016-12-11.

4.2 BITCOIN ECOSYSTEM

The open and permissionless nature of Bitcoin has led to a Cambrian explosion of projects and start-up companies. Table 4.1 provides an overview of the diversity of the narrow ecosystem, i.e. the ecosystem comprising Bitcoin companies only. For the companies that are marked as bold, there are concise case studies available in Appendix A. These case studies were also published in (Wörner, Von Bomhard, et al. 2016).

The ecosystem is divided in two main categories (see Figure 4.1 for illustration). The first category consists of start-up companies that act mainly inside the Bitcoin ecosystem itself. Examples are wallet providers, mining operations

2 https://crunchbase.com
3 https://coinmarketcap.com/currencies/
4 http://www.coindesk.com/bitcoin-venture-capital/
5 https://cyber.fund/radar

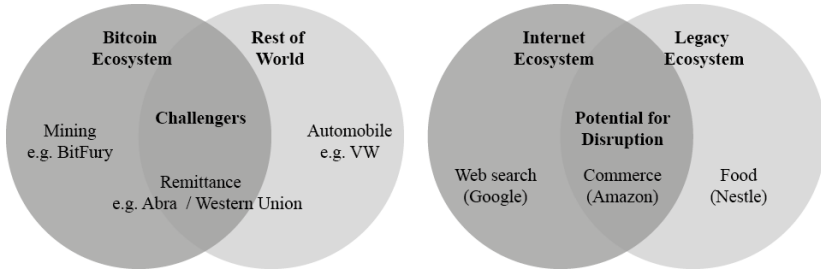


Figure 4.1.: Distinction between challengers and the Bitcoin ecosystem without challengers.

and Bitcoin exchanges. We refer to this category as *Bitcoin ecosystem without challengers*. Hence, the second category is termed challengers. Those are companies that use Bitcoin technology to attack traditional companies and business models outside the Bitcoin ecosystem. Examples of this category are payment processors like BitPay[6], challenging traditional online payment processors like PayPal, and remittance services like Abra[7], challenging incumbents like Western Union by cutting down transaction fees through disintermediation and decreased vulnerability to fraud. The classification of companies is not always clear-cut and may change as the ecosystem develops, but provides a lens to identify sectors, which might get disrupted first.

Interestingly, the challenger category is not limited to the financial service industry. We identified three main sectors beyond financial services where start-up companies use Bitcoin technology to innovate and thereby challenge incumbents: (1) notary services, (2) marketplaces, and (3) digital assets. Notary services use the Bitcoin blockchain as an immutable public database and time-stamping service. Applications are records management, by providing verifiable audit trails and provable data integrity, as well as identity registries, which are not tied to a particular identity provider. Marketplaces provide a decentralized infrastructure where physical as well as digital goods and services can be traded for bitcoins. The digital assets sector is concerned with the management of "anything that exists in a binary format and comes with the right to use" (Wikipedia 2016). This entails digital art, photographs, music, but also coupons and tickets. Furthermore, we extend the concept of digital assets to incorporate IOT), since IOT is concerned with the digital representation of physical devices.

Figures 4.2 and 4.3 give an overview of the temporal evolution of the ecosystem. Figure 4.2 shows the evolution of all Bitcoin projects whereas Figures 4.2 and 4.3 is restricted to the venture-backed Bitcoin start-up ecosystem. Notably, there are no companies that were founded in the first two years of Bitcoin's existence. Projects that started earlier have not survived (e.g. Mt.

6 https://bitpay.com/
7 https://www.goabra.com/

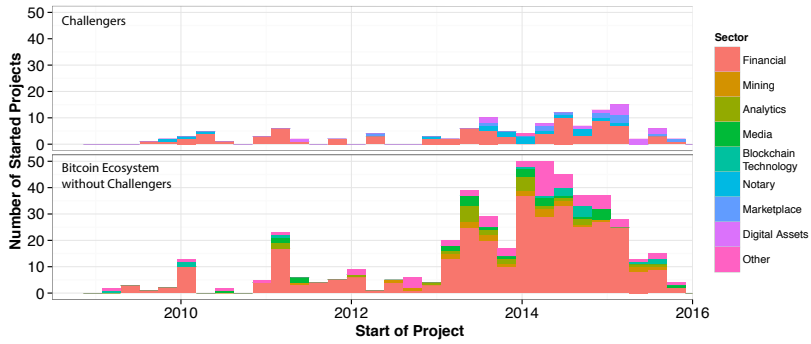


Figure 4.2.: Number of new Bitcoin projects over time.

	Sector	Subsector	Description	Representatives	Revenue Mechanics
Challengers	Digital Assets	Internet of Things	Service to register and manage connected devices	**Filament**	Product as a Service
		Intellectual property	Service to register and manage IP like music and art	**Ascribe**, Monegraph	Transaction fee (%)
		Generic Platform	Generic platform to register and manage all kinds of digital assets	Colu, CoinSpark	Not applicable
	Marketplace	E-Commerce	Buying and selling of physical or digital products	**OpenBazaar**	Not applicable
		Digital (micro) Commerce	Buying and selling digital micro services like individual API usage	**21**	Device sales (currently)
	Notary	Records Management	Service to provide data integrity and auditability	**Factom**	Token sale
		Identity	Service to provide an identity to authenticate across the Intern	**Onename**, Shocard	Not applicable
	Financial	Payment processor	Payment services for merchants	BitPay, Coinbase	Transaction fee (%)
		Remittance	Global transfer of money across boarders	Abra	Flat fee for depositing/withdrawing funds
		Over-The-Counter (OTC) Infrastructure	Infrastructure to trade commodities, financial instruments, and derivatives	Symbiont, Mirror	Project based
		Lending	Service to facilitate peer-to-peer lending	Bitbond	Transaction fee (%)
		Crowdfunding	Service to facilitate crowdfunding for projects	Koinify, Swarm	Transaction fee (%)
Bitcoin Ecosystem without Challengers	Blockchain Technology	Sidechains	Technology for custom blockchains compatible with bitcoin	Blockstream	Project based
	Analytics	Analytics	Service for browsing and analyzing blockchains	Coinalytics, Blockchain	Project based
	Mining	Mining	Manufacture and/or operating mining infrastructure	BitFury, KnC Miner	Bitcoin mining, device sales
	Financial	Wallet / Vault	Product or service to store private keys, and optionally initiate transactions	Case, Xapo, Coinbase	Ancillary services
		Exchange	Service to trade bitcoins for other currencies	Bitstamp, Kraken, Gem	Transaction fee (%)
		Compliance	Service to provide Know Your Customer (KYC) and Anti-Money Laundering (AML) compliance	Elliptic, BlockScore	Transaction fee (%)

Table 4.1.: A categorization of the (venture-backed) Bitcoin start-up ecosystem.

Gox, 2015) or have not attracted venture capital. The following three years (2011-2013) are characterized by companies building the infrastructure for the Bitcoin ecosystem. From 2014 on, the number of challengers has grown, and the sectors beyond the financial sector have gained traction. Hence, these companies are very young and are just in the process of entering the market.

4.3 BLOCKCHAIN ECOSYSTEM BEYOND BITCOIN

4.3.1 *Altcoins and Metacoins*

Bitcoin is an open source project with a public code base. Since a blockchain is defined by its history starting from the genesis block, new cryptocurrencies can be created with the same code base starting from a new genesis block. However, more interesting than cryptocurrencies based on the same code base are variations thereof. An early example of a Bitcoin-derived, and reasonably successful, cryptocurrency is *Litecoin*. Litecoin has different technical

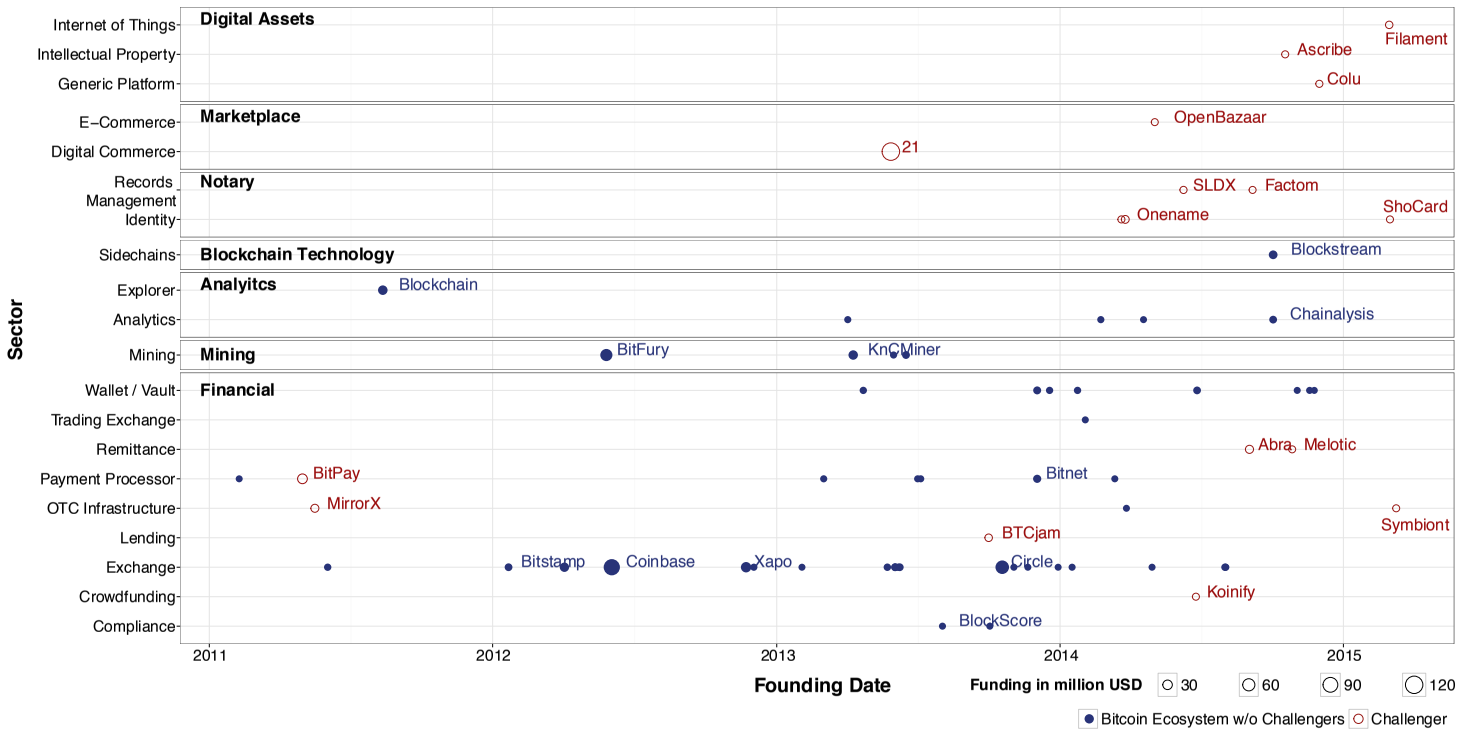


Figure 4.3.: Evolution of the venture-backed Bitcoin start-up ecosystem.

Coin	Market Cap. [M USD]	Price [USD]	Description
Bitcoin	12,330	768.87	First decentralized cryptocurrency
Ethereum	704	8.11	Decentralized app and smart contract platform
Ripple	243	0.001	Interbank value transfer
Litecoin	178	3.65	Early bitcoin clone with slight changes
Monero	106	7.81	Cryptocurrency with improved privacy
Ethereum Classic	82	0.95	Pre-hard-fork Ethereum
Dash	63	9.16	Cryptocurrency with improved privacy
Steem	47	0.21	Appcoin for social media platform
Augur	35	3.17	Equity token for prediction markets platform
MaidSafe	30	0.07	Appcoin for decentralized app and storage platform

Table 4.2.: Overview of the ten coins with highest market capitalization (2016-12-11).

parameters (e.g. block generation rate is targeted at 2.5 min instead of 10 min) and parameters concerning monetary policy (84 million litecoins instead of 21 million bitcoins). Other *coins* started from the idea of Bitcoin but are not based on a mere fork of the code base. Instead, they are developed form a new code base to build improved technical features, such as increased privacy. For example, *Monero* uses one-time ring signatures and stealth address to create an opaque blockchain in order to resist analysis and to provide unlinkable transactions. Similarly, Ethereum was started from scratch, but adopted many ideas from Bitcoin. Table 4.2 displays the ten *crypto coins* with the highest market capitalization. The market capitalization of a given coin is calculated by its current available supply multiplied by its exchange rate. The term *coin* is used deliberately in contrast to currency, since three of the listed coins have special roles in particular decentralized applications.

Steem[8] is a token to reward participation on a social media platform.

Augur[9] is a prediction market platform built on top of Ethereum, and the corresponding token is a form of equity, which entitles the bearer to a share of the market fees on the prediction market platform. In addition, the token obliges the bearer to provide oracle services to the platform, i.e provide truthful information to clear particular prediction markets. Neglecting the obligation or providing erroneous information, as determined by a contest, leads to automated confiscation of tokens by the platform.

MaidSafe[10] denotes the token in the SAFE network. These tokens can be earned by providing storage or computation resources to the network, and have to be spent to access and consume resources by the network.

8 https://steem.io/
9 https://augur.net/
10 https://maidsafe.net/

Ripple[11] is a partly permissioned network targeting financial institutions, such as banks, to lower costs of cross-boarder payments, and to replace the traditional correspondent banking system. Partly permissioned, because nodes can, in principle, individually decide with whom they interact, in practice, however, Ripple Labs, the company behind the software, provides the majority of nodes, and suggests a list of trusted nodes per default. According to Ripple Labs, the tokens (XRPs) serve two main functionalities. First, DoS prevention by requiring to *burn* a certain amount of XRP with transaction. Second, a counterparty-free, network-native, currency to provide liquidity between entities aiming to facilitate cross-currency transactions.

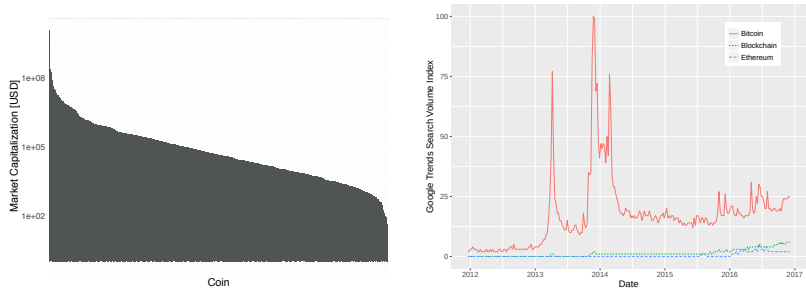
Ethereum Classic is also worth noting. After *The DAO* contract got drained by an anonymous attacker, capturing about $ 50 million USD worth of ether, parts of the Ethereum community, supported by Vitalik Buterin and the Ethereum Foundation decided to return the funds by means of a software upgrade leading to a hard fork (Buterin 2016a). This decision was controversial and parts of the community decided to continue with the *classic* software. Hence, there are now two Ethereum blockchains and currencies sharing the same history up to block no. 1920000.

These coins are only the tip of the iceberg. In December 2016, Coinmarketcap lists 705 different coins of which 565 are actively traded, and thus have a market price. Figure 4.4a shows the market capitalization of all 565 coins ordered from highest to lowest on a logarithmic scale. There is a broad regime of exponential decay enclosed by two regimes of super-exponential decay at the edges. This shows that the value of all currencies and tokens is mainly centered on a few, with Bitcoin at the top. Figure 4.4c shows the share of Bitcoin of the total market capitalization of all coins over time. The long term trend is slightly downwards. However, recently Bitcoin's dominance is strongly increasing from 80% to more than 85%. Figure 4.4b shows the Google Trends query index of the search terms *Bitcoin*, *Blockchain*, and *Ethereum* from 2012 to December 2016. "The query index is based on query share: the total query volume for the search term in question within a particular geographic region divided by the total number of queries in that region during the time period being examined. The maximum query share in the time period specified is normalised to be 100, and the query share at the initial date being examined is normalised to be zero" (Choi and Varian 2012). *Bitcoin* clearly dominates the Google search queries. This is even more so for emerging economies like China, India, and Africa. The term *blockchain* appreciates almost linearly since fall 2015. The interest in *Ethereum* appreciated in the beginning of 2016 and maxed in the summer with the rise and fall of *The DAO*.

Of course there are more metrics than the ones presented here. In particular metrics that might have a predictive element, such as number of develops working on core and ancillary software, number and size of meetups and

11 https://ripple.com/

conferences around the world. Furthermore, the combined market price of an underlying coin together with meta coins could be interesting to consider.



(a) Coins listed on Coinmarketcap sorted by market capitalization.

(b) Google Trends Search Volume Index comparison.

(c) Share of Bitcoin of the total market cap of all cryptocurrencies. Source: https://coinmarketcap.com/charts/%23btc-percentage

Figure 4.4.: Illustration of Bitcoin's dominance in the cryptocurrency space.

4.3.2 *Venture Capital Investments*

Figure 4.5 shows the monthly venture capital investments in the entire blockchain ecosystem. The date denotes the actual investment. Companies with multiple investment rounds show up repeatedly. In the figure, the blockchain ecosystem is divided into five distinct categories. Companies are allocated to a given category if they focus on a specific ecosystem or technology. Exchanges are classified as Bitcoin, although trading of additional coins is possible in general.

In most months the investments in Bitcoin start-ups vastly outweighs investments into the other categories. The only category with increasing investments is that of permissioned blockchains.

Table 4.3 lists the ten start-ups with highest venture capital funding. Only three of them can be clearly allocated to the field of permissioned blockchains. *Ripple Labs* was already briefly discussed in the last section. *Digital Asset Holdings*[12] builds a distributed and cryptographically-secured network of known

12 https://digitalasset.com/

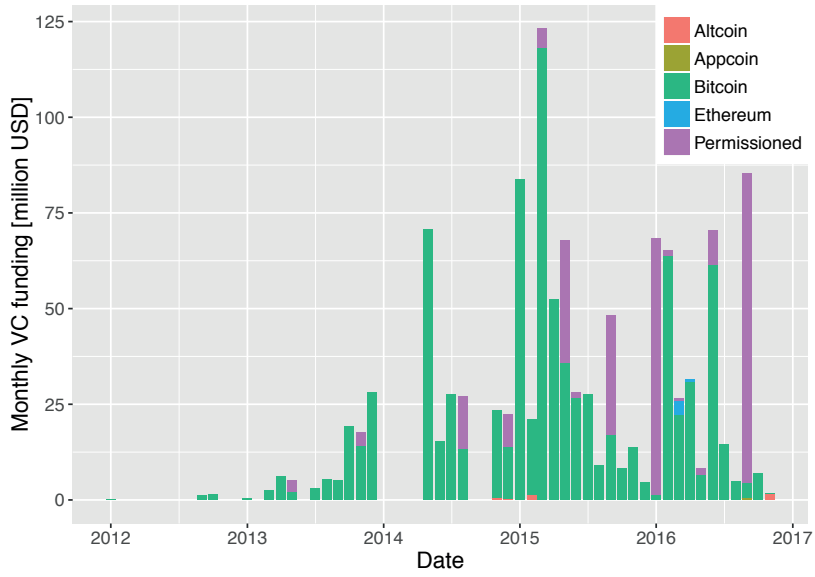


Figure 4.5.: Monthly venture capital investments in blockchain start-ups.

entities to improve efficiency, security, compliance and settlement speed in capital markets with applications in post trade and repo clearing. *Chain*[13] started as a Bitcoin company, providing developer APIs to the Bitcoin network, but pivoted towards permissioned blockchain technology, and recently released their open source blockchain solution *Chain Core*[14]. According to their website, "Chain works with leading companies including Visa, Nasdaq, Fiserv, Citigroup, Capital One, Orange, State Street, MUFG, and many more"[15].

Thus, venture capital is focused on Bitcoin companies and on companies that build blockchain-based infrastructure with applications to financial services and capital markets.

4.3.3 *Nakamoto Business Model, Crowdsales and Initial Coin Offerings*

Bitcoin essentially proved a new type of business model that can also be used as a financing mechanism. Brener 2016 introduced the term *Nakamoto Business Model* for a business model that allows a company or a team of developers behind a decentralized network or a software protocol to generate income. The basis of this business model is that the network or the application is tied to a particular coin. Brener 2016 explains the standard procedure of the Nakamoto

13 https://chain.com/
14 https://github.com/chain/chain
15 https://chain.com/faq/ accessed 2016-12-15

Company	Funding [M USD]	Sector	Blockchain
Circle Internet Financial	136	Financial Services	Bitcoin
21 Inc	121.05	Infrastructure / Marketplace	Bitcoin
Coinbase	116.5	Wallet / Vault / Exchange	Bitcoin
Ripple Labs	96	Infrastructure	Permissioned
Blockstream	76	Infrastructure	Bitcoin
Digital Asset Holdings	60	Infrastructure	Permissioned
BitFury	60	Mining / Technology	Bitcoin
Chain	43.7	Infrastructure	Permissioned
Xapo	40	Wallet / Vault / Exchange	Bitcoin
BitFlyer	33.94	Exchange	Bitcoin

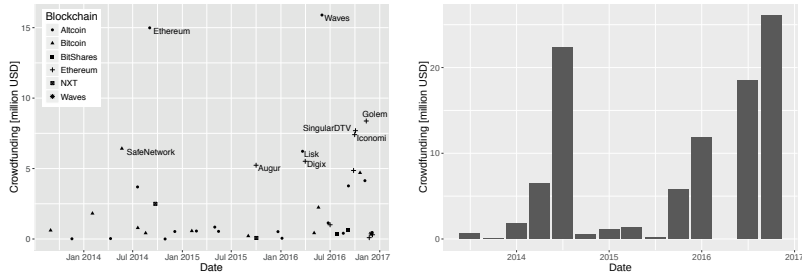
Table 4.3.: Overview of the ten most venture-backed start-up companies in the blockchain ecosystem.

Business Model as follows (Brener 2016 uses the term user token instead of appcoin):

1. Publish a white paper defining the specifications of the network and a road map for its future development.
2. Publicly announce the token and release the source code prior to creating the first token.
3. Deploy the network and secure user tokens via mining. Alternatively, allocate a portion of the pre-sale tokens to the founding team as a reward for ideating and developing the network.
4. Advertise the network and sell user tokens to anyone, anywhere.
5. Work to grow the number of people using, building apps on top, and maintaining the network.

In the case of Nakamoto and Bitcoin this model created an organization with a market capitalization of more than $12 bn USD and estimations are that Satoshi Nakamoto owns around 1 million bitcoins (Lerner 2016). A related concept is that of a crowdsale or ICO. The most prominent ICO was that of Ethereum. During an online crowdsale during July to August 2014 global investors bought ether with a value of more than 30 000 bitcoins, a year before the actual network went live. The bitcoins were held by the Swiss non-profit foundation *Stiftung Ethereum* and were used to fund the development. The investor accounts were then included into the genesis block of the Ethereum blockchain. Ethereum itself allows to implement coins, often termed tokens, and crowdsales in form of smart contracts (see also Chapter 7). The most prominent was *The DAO* which collected more than $150 million USD worth of ether in April and May 2016. Although this project ended early and badly, because of a software bug in the contract code that allowed hackers to steal the

collected ether funds, the crowdsale model on top of Ethereum is increasingly utilized. Figure 4.6a gives an overview of successful crowdsales, classified by the underlying platform to implement the coins or tokens. In particular since 2016, most of the crowdsales with a sum above $5 million USD are on the Ethereum network. Besides that, the Bitcoin or Ethereum crowdsale model with native network coins is still popular. Figure 4.6b shows the combined quarterly of value of the crowdsales. It can be seen that the importance of this model has significantly increased over the last year.



(a) Individual projects classified by underlying blockchain.

(b) Total quaterly crowdfunding results

Figure 4.6.: Successful crowdsales and ICOs

Ehrsam 2016 argues that this new business model will lead to "that businesses that are based on network effects will start to be built *decentralized first*" (emphasize in the original). The appcoin model allows to offset the network effect. In the beginning of a new application, dependent on network effects or a software protocol, the value to a user is low. The value increases with every new user, but the incentive to join in the beginning is low, resulting in a chicken-and-egg problem. Appcoins can provide this missing incentive. In the beginning of Bitcoin, every user could mine with her personal computer with a reasonable chance to find blocks, and thus receive newly minted bitcoins. Although bitcoins had no price in the beginning, because there was no market, users with some confidence in the product were incentivized to provide service to the system in two ways. First, to provide security with mining, and second, by advertising the system and recruit new users. Thus, growing the usefulness and value of the network organically. In other words, the intrinsic speculative element of a tradable token required to use the network can help to bridge the gap until network effects alone provide enough value to users. Bitcoin not only proved the fact that such a model could work, but also provided the first infrastructure, namely pseudonymous electronic cash, to provide value and funding for the next generation of decentralized infrastructure and applications.

However, the same model can be misused as a *pump and dump pyramid scheme*, where the developers and early users are only interested in a quick financial gain instead of building a sustainable product.

4.3.4 *Permissioned Blockchains and Consortia*

Permissioned blockchains are blockchains with known participants that do not necessarily fully trust each other. Thus, the likely use of permissioned blockchains is predominately in inter-business relationships with static memberships. Therefore, over the course of the last year, a number of consortia around the development, standardization and use of blockchain technology have been established. Figure 4.7 shows 21 blockchain consortia based on the founding date and the country of incorporation. Furthermore, the size and sector is depicted.

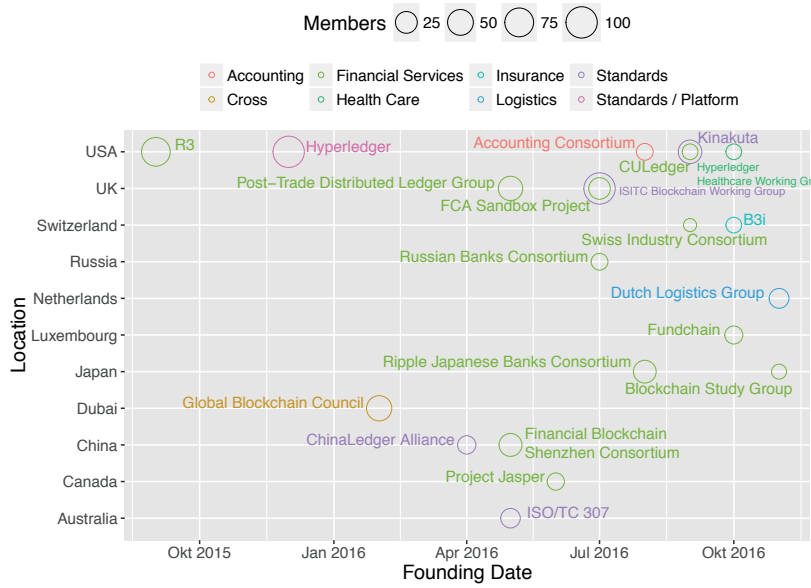


Figure 4.7.: Overview of blockchain consortia.

The largest consortia are around standards and the technological infrastructure. The *Hyperledger* project is an open source collaborative effort to advance cross-industry blockchain technologies. It is hosted by the Linux Foundation and has 100 member companies such as Accenture, IBM, Intel, J. P. Morgan, and the Deutsche Börse Group. Currently the Hyperledger project entails four distinct open source projects:

- **Blockchain Explorer:** A user-friendly web application to view and query Hyperledger blockchains.

- **Fabric:** A modular blockchain architecture originating from Digital Asset Holdings and IBM.

- **Iroha:** A blockchain infrastructure contributed by Soramitsu, Hitachi, NTT Data, and Colu.

- **Sawtooth Lake:** Blockchain with novel consensus protocol (proof of elapsed time) using Intel's trusted computing platform SGX (Costan and Devadas 2016).

The largest domain-specific consortium is R3 with about 70 of the world's biggest financial institutions. In collaboration with development teams of member companies *Corda* was developed and open sourced on November 30. 2016 (Brown et al. 2016). The Corda distributed ledger is explicitly designed to record the state of deals and obligations between institutions and people. In contrast to the general design of blockchains, Corda keeps most of the data private between the interacting parties and potential notaries.

All projects of these consortia are still in the exploration or proof of concept phase. To the authors knowledge non is in production.

4.4 ECONOMIC RELEVANCE OF CRYPTOCURRENCIES

Although permissioned blockchains and related consortia have gotten a lot of attention during the last year, applications of these systems are still in the ideation or proof of concept phase. Bitcoin, in contrast, is used productively by thousands of companies and at least hundreds of thousands of individuals. Hence, this thesis is concerned with the potential of cryptocurrencies. In the following, five important characteristics provided by cryptocurrencies with potential economic relevance are presented. These characteristics are accompanied with the open and permissionless nature of cryptocurrency blockchains, and can hardly be replicated on private or permissioned systems.

4.4.1 *Inclusion*

Public cryptocurrencies, such as Bitcoin or Ethereum, aim to be inclusive and censorship resistant. They provide a financial and payment infrastructure reaching as far as the Internet. This provides a novel alternative for inhabitants of developing countries, who are often excluded from financial services (Chaia, Goland, and Schiff 2010) or are dependent on a monopoly (e.g. M-Pesa). Cryptocurrencies are built solely on cryptography and mechanism design, cast in executable computer code, instead of human institutions, trust or reputation. This provides the basis for anonymity and the equal participation of humans and machines.

4.4.2 *Innovation*

Financial institutions have grown over centuries and the financial system is large, complex, and opaque. In contrast, cryptocurrencies are open source, and all code and transactions are publicly verifiable. Individuals and companies are able to create products and services interfacing. or building on top of cryptocurrencies. Everyone is able to review code, and to suggest changes and advancements. Effective governance structures are still missing (A. Narayanan 2015a), but there is always the possibility to fork the code, implement the changes, and deploy a new network. Since everything happens in public, the community can learn efficiently from each other, and an environment of cumulative innovation can evolve.

4.4.3 *Disintermediation*

Non-duplicable transfer of value over the Internet has always involved a number of trusted and regulated intermediaries such as banks, credit card companies, payment service providers (e.g. PayPal or Stripe). These intermediaries provide valuable service based on the current financial infrastructure, but are also gatekeepers extracting fees. Cryptocurrencies allow to circumvent these intermediaries in many cases, and decentralized application platforms allow to implement certain functions of formerly trusted third parties entirely in code.

4.4.4 *Automation*

Cryptocurrencies are programmable money in two ways. First, cryptocurrency transactions can include their own verifiable rules under which the transaction is valid or invalid. This allows low-trust atomic transactions in some cases (e.g. in the case of trade of blockchain-based tokens, trade involving smart property (Hearn 2011a), or zero-knowledge contingent payments (Gregory Maxwell 2016)), or high granularity (micropayments) to minimize losses and build up trust, or the inclusion of a competitive market of low-trust arbitrators, who can be consulted in case of a dispute. Second, cryptocurrencies provide a native interface for machines. No legal identity is needed to participate, and machines can reason about the finality of payments due to proof-of-work. Traditional payments, except physical coins, are essentially reversible by individual humans, companies or institutions. With a cryptocurrency, a whole ecosystem of miners, individuals, and companies would have to decide to rollback the blockchain. These characteristics allow automation at the edges with machines acting as autonomous economic actors.

4.4.5 *Novel Business Models*

The Nakamoto business model, and variations thereof, provide a novel mechanism to finance and create infrastructure and applications based on network effects. This business model does not need the formal creation of a company and includes users naturally as shareholders, thus offsetting network effects and naturally incentivizing users to recruit others.

In addition, automation and disintermediation decrease literal transactions costs. Thus, micropayments become technically viable. Although human micropayments are psychologically discouraged because of mental transaction costs (Szabo 1999), machines and software agents can make use of direct micropayments to enable new kinds of economic interactions and share resources efficiently and securely.

4.5 CONCLUSION

This chapter provided an overview of the Bitcoin and related blockchain ecosystem that originated with the unexpected success of the first decentralized cryptocurrency. First, the narrow Bitcoin ecosystem is considered. The ecosystem is classified in different sectors and restricted to companies with venture-capital funding. Besides financial services three further important sectors are identified: (1) digital assets, (2) marketplaces, and (3) notary services. Thereafter, the broader blockchain ecosystem is considered. Numerous alternative coins with their own blockchains, but also meta coins, existing on host chains, have emerged. Still, Bitcoin is dominant according to a variety of metrics. Venture capital in the broader ecosystem is almost non-existent, except investments into permissioned blockchains. The emergence of permissioned blockchains without cryptocurrencies goes hand in hand with the emergence of industry and cross-industry consortia exploring the applicability of blockchain technology in a variety of sectors. However, although little venture funding flows into companies behind and around alternative public blockchains, the sector blooms because of new forms of cryptocurrency-powered crowdfunding, and variations of the Nakamoto business model. The chapter closes with a discussion of five characteristics of cryptocurrencies with potential economic relevance: (1) inclusion, (2) innovation, (3) disintermediation, (4) automation, and (5) novel business models. These characteristics further motivate the investigation of the application of cryptocurrencies in the Internet of Things.

5 S2AAS ON THE BITCOIN BLOCKCHAIN

5.1 CONTEXT AND MOTIVATION

In 2016, there are about 6.4 bn connected devices, 4 bn of them are in possession of individuals and private households (Gartner 2015a). In 2020, the number is expected to surpass 20 bn (Gartner 2015a). In addition, there are currently more than 3.4 bn smartphone subscriptions globally. Even in developing countries the adoption of smartphones has surpassed the adoption of mobile phones (Ericsson 2016). Connectivity and computing is spreading around the globe faster and more pervasive than energy grids or clean water. Smartphones are the largest sensing platform the world has ever seen. A multitude of physical and virtual sensors allow to digitize an ever more precise portrayal of the context the device, and its user, exists in. This allows applications to provide the user the services and information that are relevant for her *here* and *now*. The IOT extends this paradigm to ever more objects around us. Humans are augmented with sensors measuring their steps, their heartbeat, their breath, their stress levels, their interactivity and much more. Homes are augmented with sensors measuring energy consumption, air quality, temperature, and occupancy. Cities are augmented with fixed sensing infrastructure but also by cars and individuals. However, the combination of the sensing capabilities with computing and connectivity allows the creation of digital services that can be delivered not only to the local user, but to anybody and anything around the globe at the speed of light. We expect this will eventually enable a new business model pattern called Sensing-as-a-Service (S²aaS).

The New Deal on Data (Alex Pentland 2009) emphasizes that data are worth more when shared, but due to technical and legal issues most data ends up in siloed corporate databases. The New Deal further points to the need for an equivalent to property rights for data. Individuals need to be empowered to control and monetize the data they generate.

Therefore, we propose a bottom up approach by incentivizing individual devices and their users to provide sensing capabilities on a global market, such that the collected data can be used to create the greatest economic and societal value. The underlying infrastructure has to be open and permissionless in order to scale organically. Everyone and everything should be able to participate. Furthermore, financial incentives and competition have the potential to improve the quality and expressiveness of data.

In this chapter, we investigate the suitability of Bitcoin to provide the basis of such an infrastructure. To this end, we discuss the most important characteristics of Bitcoin from the vantage point of a Sensing-as-a-Service (S²aaS)

infrastructure, and present the most basic building block: The concept of exchanging data for electronic cash via the Bitcoin blockchain. Based on this concept, we analyze, and derive, requirements and present a prototypical implementation. We find that Bitcoin provides an open, permissionless infrastructure with pseudonymous identification, allowing disintermediated frictionless machine-to-machine payments. Integrated cryptographic primitives can be used for authenticity, integrity and confidentiality of data exchanges. However, while the blockchain may be used as a notary service for some important data, the Bitcoin network and protocol are not suitable for data transport. Furthermore, Bitcoin's high-latency eventual consistency, the UTXO model, and the block size limit inhibit secure low-trust micropayments in form of *direct* Bitcoin transactions. The structure of this chapter is as follows: After the presentation of a motivating example, we provide background to the S²aaS and crowdsensing concepts (Sec. 5.2). In Sec. 5.3 the relevant Bitcoin characteristics are presented. Thereafter, we describe the concept (Sec. 5.4), analyze, refine and derive requirements (Sec. 5.4.1). Based on this conceptual work, we present an implementation (Sec. 5.5), and evaluate the concept (Sec. 5.6). The key findings and a brief discussion is provided in Sec. 5.7. Sec. 5.8 presents the related work that has been done since this work has originally been done. Finally, we end the chapter with a conclusion (Sec. 5.9).

Motivating Example

To illustrate the here presented concept, we use a personal connected weather station like Netatmo (Denmead 2013) as an example. Personal weather stations are typically equipped with multiple sensors to continuously measure, for instance, temperature, humidity, wind speed, wind direction, solar radiation, and air pressure. Some even measure air pollution in terms of particulate matter, which concerns people's health. The most obvious use for this data is to inform the owner of the weather station with the data it generates. Another example for the application of measurement data from a weather station is its use to feed the control system of the owner's household heating system. For the investment into a personal weather station and heating control, its operation and maintenance, the owner gains the benefit of a well-tempered house and saves energy and money (Dong and Lam 2014).

Clearly, there is more overall use to the data, if it is shared with others. For instance, neighbors could use the exact same data to control their heating systems, however, only given the condition that the owner of the weather station is willing to share the generated data. Even for the rather simple case of a weather station, there are many other useful applications of the data. Fitness platforms like Runkeeper[1] and Nike+[2] could aggregate data from many owners of personal weather stations to monitor air pollution and

1 https://www.runkeeper.com
2 https://nikeplus.nike.com

generate running tracks with optimal air quality in real-time. Meteorologists could improve weather forecasts with the high-resolution data from many personal weather station owners. Also, researchers could use the data to track climate change (Hijmans et al. 2005).

There are already enthusiasts who share their data freely and without monetary incentive, e.g. (Weather Underground, Inc n.d.). However, to enable applications of greater use, there has to be extensive supply of sensor data from many weather stations all around the world. Arguably, this can only be achieved by providing monetary incentive to the suppliers (Bohli, Sorge, and Westhoff 2009).

5.2 BACKGROUND

Sensing-as-a-Service and Crowdsensing

The Everything-as-a-Service (Banerjee et al. 2011) paradigm originates from cloud computing (Armbrust et al. 2010) with its Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service models. Instead that every company or individual has its own computing and software infrastructure, the infrastructure is concentrated at one or a few data centers and can be accessed globally – and thus consumed as a service. This provides economies of scale for the provider, and low capital costs for the consumer. This model has been extremely successful, such that most computing infrastructure is provided by a small number of providers such as Amazon, Microsoft and Google. The rearrangement from capital costs to operational costs in the form of *pay as you go* revenue models has led to the creation of countless start-up companies.

The S²aaS model is different in the sense that the resource is (often) unique. Sensing is specific to a particular context. A weather station measures the temperature at a specific location and at a specific time. A virtual sensor, measuring the installed applications on a smartphone, is specific to an individual and a time. On the other hand, S²aaS allows to provide these unique resources globally as a service, and is the basis of a multi-sided market for sensor data (Fleisch, Weinberger, and Wortmann 2014). Perera et al. 2014 provides a discussion of the applications in the context of smart cities and presents a number of benefits of the model. The main points thereby are the reduction of data acquisition costs and the availability of data that has been unavailable before.

S²aaS may entail wireless sensor networks operated by companies, but also connected consumer devices. That latter concept is also often termed *crowdsensing*. Ganti, Ye, and Lei 2011 define the term as "where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest" (Ganti, Ye, and Lei 2011). Predominately, crowdsensing has been engaged in the form of mobile crowdsensing, utilizing the ubiquity of smartphones, in application-specific

scenarios. Examples are transit tracking (Thiagarajan et al. 2010), road and traffic monitoring (Mohan, Padmanabhan, and Ramjee 2008), site characterization (Chon et al. 2012), and on-street parking (Chen, Santos-Neto, and Ripeanu 2012; Coric and Gruteser 2013). Besides these small-scale academic field studies, notable examples for successful large-scale, application specific mobile crowdsensing application are Google Maps and Waze[3] which was acquired by Google in 2013.

Guo et al. 2015 provides an extensive recent review of the various mobile crowdsensing paradigms, their applications, as well challenges and opportunities. The main challenges that have been identified repeatedly (see also (D. He, Chan, and Guizani 2015)) are privacy and incentives. (Christin 2015) provides an analysis of the privacy implications and threats, as well as a survey of available privacy-preserving techniques.

In the application-specific scenario participants are typically incentivized by receiving an application-specific service. In the case of Google Maps, a participant gets routing and traffic information in exchange for providing location and speed information. Especially in the case of participatory sensing where explicit user input is required, gamification (Deterding et al. 2011) has been used successfully to incentivize participation (e.g. Waze). However, incentivization by service provision leads to contributions only from users who are in need of that particular service, and only during specific times. In contrast, monetary incentives are a general purpose incentive mechanism. Thus far, monetary incentives have mostly been studied in small-scale, localized field studies, as well as in form of game-theoretic incentive mechanism design (Yang et al. 2015).

In recent years, a multitude of, predominantly cloud-based, architectures enabling general purpose crowdsensing and S²aaS applications have been presented (Hu et al. 2013; Cardone et al. 2013; Fosca Giannotti et al. 2012; Haderer et al. 2015; Merlino et al. 2016). However none of them has reached reasonable scale.

What has been mainly neglected is how monetary incentives can be provided efficiently under the conditions of a global platform. Individual rewards are rather small and data requesters as well as data providers might be distributed globally. Furthermore traditional online payment mechanisms provide additional means for de-anonymization of participants. Hence, we investigate the application of Bitcoin, a global peer-to-peer cryptocurrency.

5.3 BITCOIN CHARACTERISTICS WITH RELEVANCE TO S2AAS

Bitcoin is more than just a currency. It is programmable money and a permissionless public platform for innovation based on cryptographic primitives and mechanism design. This section presents characteristics of Bitcoin with

3 https://www.waze.com/

relevance to S²aaS and motivates the concept and implementation provided in the later sections.

5.3.1 *Global, Permissionless and Censorship-resistant*

Comparable to HTTP as a protocol for transfer of data, cryptocurrencies can be viewed as a protocol for the transfer of value on the Internet. It is based on a decentralized design and consequently has no single point of failure and no single point of trust. Just like HTTP, anyone is free to use it and build applications on top of it. This has important ramifications for its use in S²aaS applications. For data exchange it is important that two clients are using a common protocol and data format. The same is true for value exchange. Bitcoin is by far the most pervasive, secure, and stable cryptocurrency (from a protocol perspective as well as from a exchange value perspective).

Cryptocurrencies are based on peer-to-peer networks and novel anonymous consensus protocols like the proof-of-work-based Nakamoto consensus protocol. Consensus nodes (miners) can join and leave the network at any time without notice and permission, and without any identification. This is the basis for censorship-resistance. If one miner tries to exclude a transaction, another one will be greedy enough to step in and include the transaction to take the fees. Consequently, no central authority can systematically exclude someone or something from participating. This represents a crucial difference to classical payment networks (e.g. Visa, MasterCard, or PayPal) that can ban anyone from using their services (as happened to Wikileaks in 2010 and Russian bank customers in 2014). Using Bitcoin as payment layer for S²aaS applications brings censorship-resistance to sharing sensor data. Nobody and nothing could systematically be excluded to buy or sell data.

Without counterparty risk of an intermediary to process both payments and data transfer, applications leveraging on a Bitcoin enabled S²aaS environment do not carry the risk of self-interested (even justified) policy changes by central entities. For instance, policy changes by Twitter forced some third-party developers using the Twitter API to shut down their operations (Sippey 2012). Something like this cannot happen using Bitcoin, as there is no central authority able to change the rules out of self-interest. Using Bitcoin as a payment network is completely platform independent. This should give entrepreneurs, sensor data providers (like the personal weather station owners in the example), and established platforms alike confidence in the stability, longevity and availability of S²aaS services built on top of the Bitcoin protocol and stir innovation.

5.3.2 *Pseudonymous Identification*

A viable S²aaS network requires that all entities have to be uniquely identified and authenticated. Cryptocurrencies rely on public key cryptography for authentication. Addresses, or account numbers are self-assigned and derived from cryptographic public keys. Therefore, ownership of an address is provable with public key cryptography and allows for pseudonymous authentication of sensor data.

As soon as an address is credited with at least one bit of value, i.e. 1 satoshi in the case of Bitcoin, the address is stored on the blockchain, and thus replicated across all nodes of the network.

As Bitcoin addresses are not directly connected to an identity and do not need to be registered at some central entity in the network, they guarantee pseudonymity of the owner. This can be favorable for S²aaS applications, because data providers may not want to expose their identity. However, it does not necessarily mean that owners are anonymous. As all transactions are publicly available on the block chain, any payment can be traced to an address that can possibly be connected to an identity at some point (c.f. Sec. 3.1.3).

Bitcoin addresses cannot only be assigned to persons and used as an equivalent to a bank account. Objects like cars, fridges, houses and - like in the example – personal weather stations – can have a Bitcoin address, effectively enabling them to send and receive money. As the Bitcoin network does not incorporate any intermediaries, Bitcoin transactions can be carried out completely automated. In fact, Bitcoin transactions can be carried out equally well by machines and humans enabling direct machine-to-machine payments.

5.3.3 *Low Fees and Friction*

Cryptocurrencies are in principle arbitrarily divisible. Bitcoin is currently divisible down to 8 decimal places. Ethereum is divisible down to 18 decimal places. Thereby, Bitcoin can, in principle, scale down payments to very low amounts allowing for trade of very small exchangeable units. It can be expected that low fees and friction in the Bitcoin payment network can lead to a whole wave of IOT innovations, because the programmatic exchange of arbitrary amounts of cash without human intervention and intermediaries allows for a generation of IOT applications that has not been feasible before.

Using Bitcoin technology as a payment network for S²aaS applications may allow for purchases of single data points, costing way less than the smallest available units of any traditional currency. This would not be possible using traditional payment networks, at least not without the introduction of additional processes. Typical intermediaries in a classical payment network like Visa, MasterCard collect fees of one to three percent (Chakravorti 2003). Average fees for international payments (e.g. with Western Union or MoneyGram) even account for more than eight percent of the transaction amount (World

Bank 2013). Even Internet-native payment company PayPal already considers payments below $10 as micropayments[4], and the fees for micropayments are 5% + $0.05 in June 2016. Hence, for payments below $1 the fees are at least 10%, and payments on the order of cents are not viable.

Bitcoin transactions are not free either, but rather compete against each other for space in the blockchain in something similar to a bidding process. Senders of Bitcoin transactions can include a voluntary, so called miner fee with their transactions. Transactions with higher fees are given higher priority by miners and are consequently processed faster than transactions with lower fees. By exposing Bitcoin transactions to these simple supply and demand market dynamics transaction costs are no longer dictated by the gatekeepers of payment networks and price efficiency in the processing of transactions for all types of applications will eventually be established.

5.3.4 *Programmability*

Cryptocurrencies introduced the concept of programmable money. Bitcoin is programmable money in the sense that transactions are scriptable. For instance, the validity of a transaction, and hence its clearing, can be bound to certain conditions. By combining multiple transaction messages and conditions, rather complex contracts, requiring no trust between the parties, can be established. Traditionally such contracts have been enforced by intermediaries. Bitcoin allows to establish contracts and enforce them completely without intermediaries. The instruction set of Bitcoin script is restricted on purpose, but it contains powerful cryptographic primitives. On the other hand, Ethereum's instruction set is Turing-complete and the programming model is much more convenient. However, this approach has been shown repeatedly to give rise to an increasing number of attack vectors.

An example, which exemplifies the power of the scriptability of Bitcoin for data payments is the concept of Zero Knowledge Contingent Payments (ZKCP) (Gregory Maxwell 2016; Banasik, Dziembowski, and Malinowski 2016). It combines hash-locked Bitcoin transactions with an external zero-knowledge proof-verification protocol in order to atomically bind the payment with the release of these particular data. Gregory Maxwell 2016 explains the concept as follows: "ZKCP is a transaction protocol that allows a buyer to purchase information from a seller using Bitcoin in a manner which is private, scalable, secure, and which does not require trusting anyone: the expected information is transferred if and only if the payment is made. The buyer and seller do not need to trust each other or depend on arbitration by a third party". However, sensor data is typically not suitable for ZKCP because generating and verifying the proofs is a computational intensive task.

4 https://www.paypal.com/us/webapps/mpp/merchant-fees

5.3.5 *Cryptographic Verifiability*

Bitcoin uses digital signatures to prove ownership of bitcoins. Using the same technique, persons, or in this case sensors, can authenticate themselves by signing a message, proving ownership of their Bitcoin address. This means data can be readily authenticated proving integrity to a buyer. Certificates provided and signed by the manufacturer of the sensor could assert capabilities of the sensor. A buyer can then verify that the data originated from a sensor with the particular capabilities as attested by the manufacturer.

Confidentiality of the data to be sent can be guaranteed by encrypting the data with the public key of the receiver. However, encryption using (asymmetric) public key cryptography is much less efficient than symmetric encryption. Thus, in practice more complex protocols such as Diffie-Hellman key exchange (Diffie and Hellman 1976) have to be used.

5.3.6 *Immutability and Timestamping*

Proof-of-work blockchains are (practically) immutable, tamper-evident append-only logs. This is useful to publish important public data. After publishing data to the blockchain, the data gets globally replicated. This provides high availability. Due to the immutability and append-only structure of the blockchain, published data inherits this immutability and obtains a publicly verifiable timestamp. Although the timestamp might not be exact, it provides an absolute time ordering of published data.

In many applications data should not be public and should thus not directly get published to the blockchain. Assume the example of a connected car which continuously logs mileage data. The owner of the car does not want to publicly publish the data, but might need to prove to his insurance or to a party interested in buying the car that the current mileage has not been tampered with. Instead of directly publishing the data m to the blockchain, the car blinds the data first by calculating a secure hash $M = H(m|s)$[5] where s is a random bit string with appropriate entropy and $|$ denotes concatenation, and publishes M. An observer is not able to learn anything about m by knowing M because secure hash functions are one-way functions. The random bit string s prevents that an observer is able to infer m by trial and error. At a later time, the car (or its owner) can prove that the mileage has not been changed afterwards by presenting that the transaction containing M is in a particular block (providing a rough timestamp), and by presenting m and s.

Table 5.1 provides a summary of the presented characteristics with a concise description. These characteristics motivate the application of Bitcoin as a basic layer of economic interactions in the context of S²aaS. The most basic element

5 We assume that m and s are in a specified, unambiguous encoding.

Characteristic	Description
Global, Permissionless, Censorship-resistant	Everyone and everything can participate
Pseudonymous Identification	Sensors and requesters do not need a continuous identity
Low Fees and Friction	No human involvement and competition instead of gatekeepers
Programmability	Transactions contain tiny programs to condition payments and to enable shared control
Cryptographic Verifiability	Built-in cryptographic primitives enable authentication, data integrity and confidentiality
Immutability and Timestamping	Proof-of-work blockchain is a decentralized timestamping service that can be used as a notary service for important data

Table 5.1.: Characteristics of Bitcoin with relevance to S²aaS.

of this interaction is the concept of exchanging a single datum for electronic cash using Bitcoin.

5.4 CONCEPT: EXCHANGING DATA FOR CASH USING BITCOIN

Consider the scenario in which two machines trade a single datum for cash. The simplified process is illustrated in Figure 5.1. Given the example above, the requesting machine A could be the Nike+ smartphone application which requires the current air pollution at the user's typical running track to optimize air quality during the run. The sensor C in this case would be an air pollution sensor in the vicinity of the running track. Both, the requesting machine and the sensor have a key pair which provides unique identification and allows them to transfer cash as well as private data over the blockchain B as the decentralized public ledger. In the illustrated scenario, the requesting machine A sends a payment to the Bitcoin address of sensor C. This involves the generation of a transaction that gets included in the blockchain (1). In a second step, sensor C notices the receipt of the payment (2). After that, sensor C creates a transaction to the Bitcoin address of requester A, including its most current datum encrypted with A's public key (3). Finally, requester A notices the receipt of the transaction that includes the requested datum and decrypts it using its private key (4).

5.4.1 *Analysis and Requirements*

Analysis

In the basic concept, we suggest the Bitcoin blockchain as the sole medium of exchange. Both payment and data is *transferred* via the blockchain. However, the blockchain is essentially a replicated append-only log that provides the

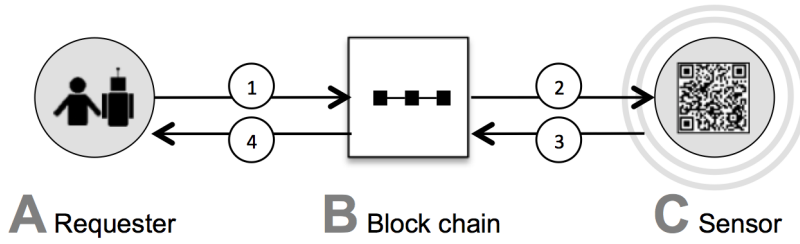


Figure 5.1.: Schema for the basic S²aaS process of exchanging a single datum for cash using Bitcoin.

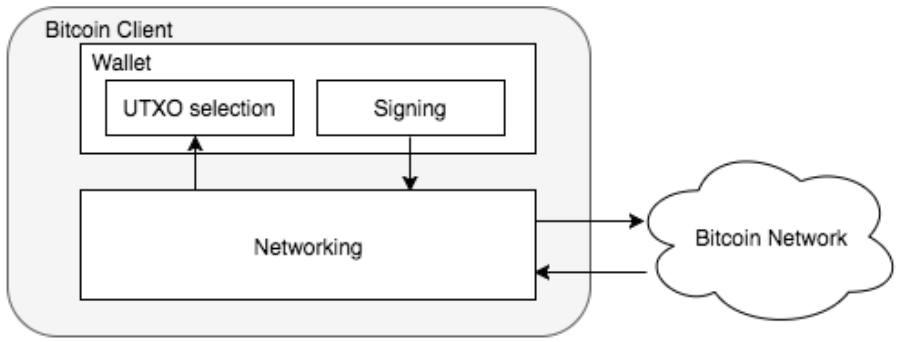


Figure 5.2.: Simplified functional structure of a Bitcoin client. A client can be divided into a networking part and a wallet. The wallet is responsible to keep track spendable coins (UTXOs) as well as the creation and signing of transactions.

basis for inferring ownership of bitcoins. Thus, *transfer* is equivalent to the respective clients noticing state changes due to the inclusion of relevant transactions into blocks.

In order to interact with the Bitcoin network directly, requester and sensor have to run a Bitcoin client (c.f. Fig. 5.2). The standard client software, Bitcoin Core, implements a full Bitcoin node. Thus, it validates and relays all transactions that are broadcasted to the network. In order to validate the transactions, it keeps a full copy of the blockchain (more than 85 GB in late 2016). Requirements for storage, but also communication and computation are thus exceeding the capabilities of many IOT devices.

Let us observe the required functionalities. We assume the requester knows the Bitcoin address S of the sensor and the price p of the data. In the first step of the concept, the requester creates a P2PKH transaction that pays p to S. Therefore, the requester needs to select n UTXOs with a values v_i $i \in 1, n$ such that $\sum_{i,1,n} v_i \geq p$. These UTXOs will provide the input for the transaction. In addition, the requester has to add at least one output with value p redeemable

by the sensor. If the value of the combined inputs exceeds p, then the requester will add another output to credit itself with the change. Finally, the requester provides the signatures for the inputs and broadcasts the transaction.

In step (2), the sensor needs to learn about the payment transaction. Instead of being a regular peer on the network, the sensor can opt to only receive transactions involving its own address(es). Full Bitcoin nodes accept filters to serve this requirement. After the payment transaction propagates through the network and the sensor has received the transaction, the sensor can prepare the data transaction. Therefore, the sensor creates a transaction with a Null-Data output:

```
OP_RETURN <data>
```

However, a transaction with just a Null-data output is not valid. Furthermore, the requester needs to be aware of the data transaction. In order to satisfy both requirements, the sensor provides a small payment ϵ back to the requester by providing inputs with UTXOs surpassing ϵ and adding an P2PKH output with value ϵ, spendable by the requester. The sensor provides signatures for the input(s) of the data transaction and broadcasts it to the network. Similar to the sensor, the requester needs in principle only to know about transactions involving its address(s). After some time the requester will receive the transaction from a connected node and is able to parse the data. This concludes the process.

Deriving Requirements

Recapitulating the process we find that requester and sensor need the following functionalities:

- Receive transactions involving specific addresses (Communication).
- Store UTXOs that are redeemable (Storage).
- Calculate (ECDSA) signatures (Computation).
- Broadcast transactions (Communication).

Furthermore, we find that the actual price p the requester is paying needs to be at least $p' = p + \epsilon$. We can also conclude that typical connected products including sensors are capable of providing the aforementioned functionalities. Typical wallet software or wallet libraries (e.g. Bitcore[6] (JavaScript), 21[7] (Python), Bitcoinj[8] (Java)) can be used to implement functionalities (2)-(3). Functionalities (1) and (4) need either a communication with the Bitcoin network or a mediating web service has to be used. Communication with the Bitcoin network adds some overhead but is necessary to be immune against censorship, single points of failure, and web service deprecation.

6 https://github.com/bitpay/bitcore
7 https://github.com/21dotco/two1-python
8 https://github.com/bitcoinj/bitcoinj

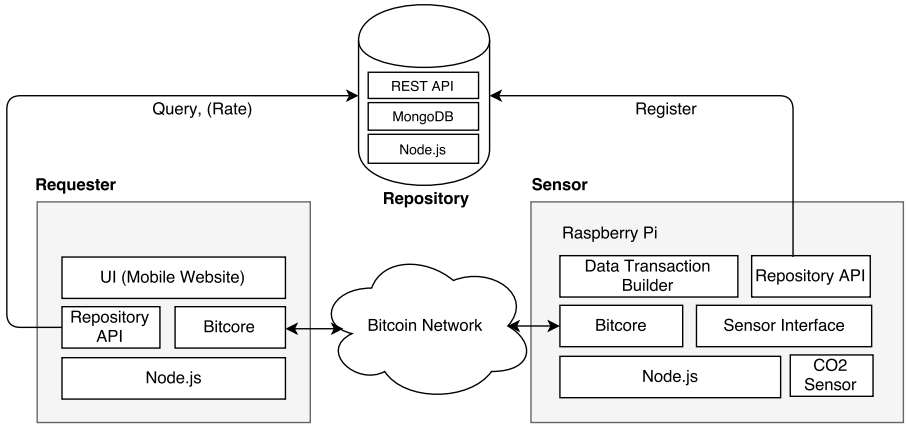


Figure 5.3.: Architecture of the S²aaS implementation. Besides requester and sensor, a repository is implemented to register, query, and rate sensors.

5.5 IMPLEMENTATION

5.5.1 *System Overview*

An overview of the system architecture is depicted in Figure 5.3. The main components are the requester client, a sensor client, and the Bitcoin network itself (c.f. Fig. 5.1). In addition, we introduce a central repository. In the preceding discussion, we have assumed that the requester already knows the Bitcoin address of the sensor. The central repository provides a means for a requester to find sensors. In the following, we describe the main components in more detail.

5.5.2 *Requester*

The requester client is implemented as a node.js[9] application and provides a human user interface in form of a web application, which is run locally. The web application interfaces with the REST API of the central repository to query sensors of interest. The repository returns JSON document including the Bitcoin address of the sensor and the price of a measurement. We use the Bitcore JavaScript libraries to interact with the Bitcoin network, and to create the payment transaction. After broadcasting the payment transaction the requester waits for the data transaction. After receiving the data transaction, the requester may rate the sensor using the repository web service.

9 https://nodejs.org/

```
{
    "name": "Air Quality Zurich Downtown"
    "datatype" : "int",
    "type"     : "co2",
    "unit"     : "ppm",
    "price"    : 10,
    "location" : "47.37246913,8.54426892"
    "description": "Zurich air quality measurements"

}
```

Figure 5.4.: Example of meta data for an air quality sensor.

5.5.3 *Sensor*

The sensor is built based on a Raspberry Pi[10] embedded Linux computer. The actual sensor is a CO_2 air quality sensor. The Raspberry Pi runs a node.js application similar to the requester client. However, without a graphic user interface. In the prototypical implementation a configuration file specifies the meta data (see Fig. 5.4). On first start, the sensor client creates a new key pair and derives a Bitcoin address. The address gets added to the meta data object and embedded into a JSON web token entailing a signature created by the private key (Jones, Bradley, and Sakimura 2015). This web token is used to register with the repository.

The sensor client implements a *Data transaction builder* function which creates a data transaction based on the most recent measurement. The transaction builder function is called upon the arrival of a payment function.

5.5.4 *Repository*

The repository is essentially a bulletin board that sensors can use to advertise their services and requesters can use to find sensors. However, the repository is not required, and is not involved in the actual S²aaS process. The repository is based on a MongoDB[11], a document-oriented database, and a REST API with a custom authentication layer. As described in Sec. 5.5.3, sensors register by providing a signed web token. This allows to authenticate the sensor, if it wants to update the entry in the repository, e.g. to update the price. Furthermore, the repository could require a proof of exchange from the requester together with a potential rating.

10 https://www.raspberrypi.org/
11 https://www.mongodb.com/

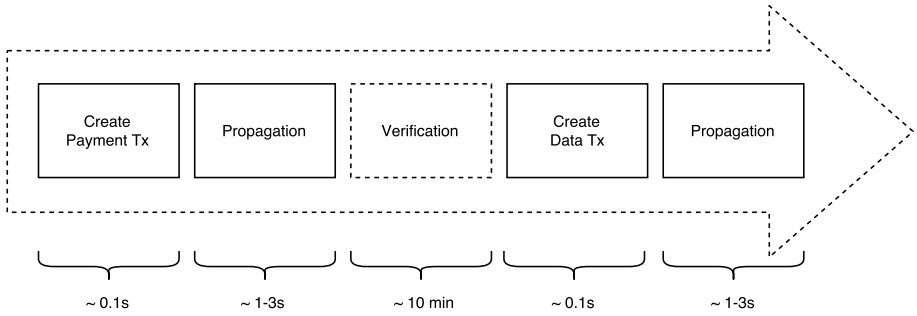


Figure 5.5.: Duration of the various steps of the S²aaS process. Data based on (Karame, Androulaki, and Capkun 2012; Croman et al. 2016)

5.6 EVALUATION

We evaluate the concept and its implementation based on the following criteria:

- Latency
- Transaction costs
- Viability of small payments
- Hardware Requirements

5.6.1 *Latency and Double-spending Risk*

In order to give a rough estimate for the duration of a typical S²aaS process, we divide the process into its basic components and consider them separately. Fig. 5.5 provides a graphical representation of the main components, as well as a rough estimate of the duration of each component. Creating and signing a transaction takes a few hundred milliseconds on a typical CPU. After broadcasting a transaction to the Bitcoin network, it takes on average 1-3 s until it reaches a specific node. In principle, the sensor could immediately prepare and broadcast the data transaction, and the data would reach the requester within 5 s approximately. However, accepting a so called *zero confirmation* transaction involves a significant risk to get defrauded by a double-spending attack. In other words, the requester could simultaneously create and broadcast another transaction spending the same UTXOs and crediting itself. In order to decrease the risk of a double-spending attack, the sensor should wait until the transaction is at least included in a block. If we assume the available space in a block is much larger than the size of transactions on the network within the time between two blocks, then it should take 5 min on average until a transaction is included in a block. However, this assumption is currently not true as discussed in Sec. 3.1.3. In contrast, there might be

a significant transaction backlog. In this case, the probability of inclusion is highly dependent on the attached transaction fee, which miners can claim by including the transaction into a block. We will discuss transaction costs in the next section in more detail. On the one hand, we can assume that in a typical S²aaS task a timely delivery of a measurement is very important. This means a latency of 5 min, or even more, is inhibiting. On the other hand, a requester does not lose anything by trying to double-spend, since it is able to create a new identity, if the double-spend is noticed.

One way to mitigate the risk is to add a rating system for requesters. A sensor would only accept a zero confirmation payment if the requester's rating is sufficient. A sensor could rate a requester by providing proof of a successful transaction or by proving a double-spending attack. These proofs can be provided by presenting the signed transactions. The repository can ensure itself by checking with the blockchain.

5.6.2 *Transaction Costs*

As indicated in the last section, transactions typically have to include a fee to get considered by miners. In particular, because block space has become scarce. Miners *have to* make a selection which transactions to include, and this selection is rationally done by considering the fees they are able to claim. Furthermore, the reference implementation Bitcoin core includes a rule for transaction relaying. If the included transaction fee is too low, the transaction might not get relayed by nodes in the network. This is a measure against DoS attacks, by which the network gets flooded by low value transactions in order spam the network and the blockchain, and as such increase the costs for nodes. The minimum relay fee is a adjustable parameter, but the default is set to 1000 satoshi[12].

In Bitcoin, data size matters. Block space is scarce (c.f. Figure 5.6), and data is what has to be communicated with all nodes, and eventually stored in the blockchain. Thus, transaction costs are proportional to the byte size of the transaction instead of its monetary value. While this is beneficial for the transfer of large monetary value, it is an issue for low-value transactions as encountered in the S²aaS setting. In the following, we will look at the typical size of our *payment transactions* and *data transactions* in order to provide a estimate for the typical transaction costs of the data for electronic cash exchange.

PAYMENT TRANSACTION The payment transaction needs at least one input and has most probably two P2PKH outputs. One for crediting the sensor and one for the change, since typically there is no UTXO available that covers the needed amount exactly. We assume that the UTXO that will be consumed as

12 https://github.com/bitcoin/bitcoin/blob/master/src/main.h

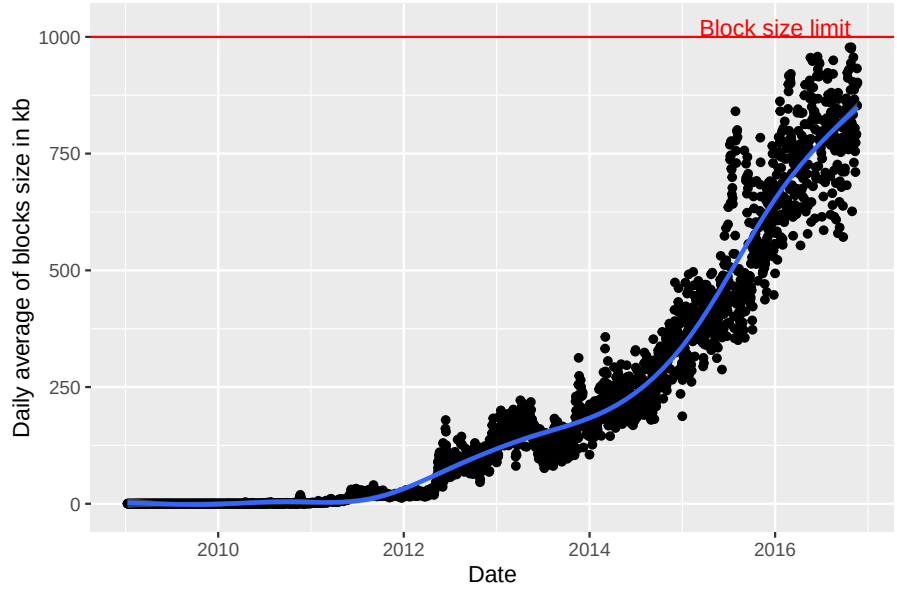


Figure 5.6.: Daily averages of the size of Bitcoin blocks. Space in blocks has become increasingly scarce.

Field	Description	Size [Bytes]
nVersion	Transaction version	4
vin	Vector of inputs	1 + variable (if <255 inputs)
vout	Vector of outputs	1 + variable (if <255 outputs)
nLockTime	Lock time	4

Table 5.2.: Data fields of a Bitcoin transaction and their sizes.

the input to the transaction is a P2PKH output as well. Based on the protocol definition in the source code[13], we can infer that a Bitcoin transaction is comprised of the fields listed in Table 5.2.

Each input consists of a 36 byte reference to the output being consumed, a 4 byte sequence field, a scriptSig and a 1 byte (in our case) script length field. The scriptSig to spend a P2PKH is 72 byte for the signature plus 33 byte for the public key. This adds up to 146 byte per input. Each output consists of a 8 byte value field, the pubScript, and again a 1 byte script length field. As discussed in Sec. 3.1.1, a P2PKH pubScript consists of 4 op codes (3 bytes each) and a 20 byte Bitcoin address. Hence, each output contributes 33 bytes, and the total transaction size in bytes of a P2PKH transaction can be calculated as follows:

13 https://github.com/bitcoin/bitcoin/blob/src/primitives/transaction.h

$$size_{Tx} = 10 + 146 * n_{in} + 33 * n_{out}$$

With $n_{in} = 1$ and $n_{out=2}$ we get 222 bytes as the minimal size of a payment transaction.

DATA TRANSACTION The data transaction needs at least one input, two P2PKH outputs and one Null-Data output. The Null-Data output's pubScript consists of one op code (4 bytes) and a maximum of 80 bytes of data. Consequently, the size of the payment transaction is:

$$\begin{aligned} size_{DataTx}(data) &= 10 + 146 * 2 + 33 * 2 + 8 + 1 + 3 + bytes(data) \\ &= 380 + bytes(data) \end{aligned}$$

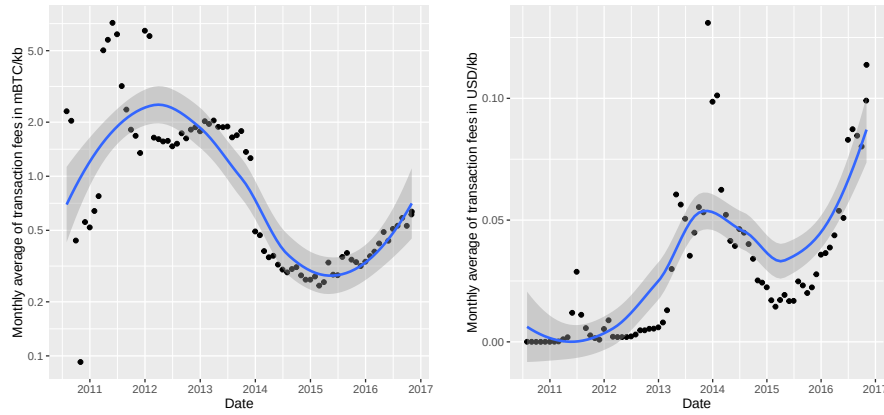
Noteworthy, Bitcoin core only relays transaction with a maximum of one Null-Data output. Thus, the payload of the data transaction is limited to 80 bytes. While this is sufficient for a measurement value, it is an issue for encrypting the measurement value. A single measurement might be 2 to 8 bytes. If we include a compressed timestamp (e.g. elapsed seconds since midnight) another 2 bytes would be needed. Encrypting the measurement value using Elliptic Curve Integrated Encryption Scheme (ECIES), which is based on the cryptographic infrastructure Bitcoin natively provides, however, leads to payloads greater than 80 bytes.

Taking both transactions together, we get a minimal size of 610 byte. In Figures 5.7a and 5.7b we show the monthly average of transaction fees paid in mBTC/kB and USD/kB. The blue lines are trend lines. The exchange rate data is provided by the CoinDesk Price Index[14]. We see that that the average transaction fees/kB have surpassed $0.1, i.e. the process would cost at least 6 cents.

5.6.3 *Minimal Payment and Dust Limit*

Bitcoin is divisible to eight decimal places. Thus, with an exchange rate of around $ 650 per bitcoin an individual transaction can in principle transfer value of less than 1 mCents. Above, we saw that the transaction costs are already surpassing this value by a factor of more than 1000. Another problem is the so called *dust limit*. If we consider a sensor that gets paid for individual measurements, then we expect that the sensor will get a large number of tiny individual payments. Instead of accumulating money in one account, the UTXO model of Bitcoin leads to large collection of individual UTXOs with tiny amounts of value. Every UTXO included in a transaction adds 33 byte and thus the necessary fees on the order of hundreds of satoshis. Hence,

14 http://www.coindesk.com/price/



(a) Monthly average of transaction fees paid in mBTC/kB.

(b) Monthly average of transaction fees paid in USD/kB.

spending UTXOs below a certain value costs more fees than the actual value. Thus, receiving a payment below this dynamic limit, is of no use.

The Bitcoin core wallet calculates the dust limit as follows:

$$L_{Dust} = 546 * minRelayTxFee/1000$$

With a default minRelayTxFee of 1000 satoshis, we get a dust limit of around 0.5 cents.

5.6.4 *Hardware Requirements*

From the requirement analysis in Sec. 5.4.1, we can infer that the sensor needs to run a SPV node to connect to the Bitcoin network, and to request transactions concerning its address(es). SPV nodes retrieve and store only block headers. Block headers in Bitcoin have a fixed size of 80 bytes. With an average block generation rate of $\frac{1}{10min}$, we get an average communication and storage requirement of 11.52 kB per day, or 4.2 MB per year. Each payment adds at least another 222 bytes for the transaction itself, and $log_2(2000) * 32 = 351$ bytes for the merkle branch in order to prove the inclusion of the transaction in a particular block. The number 2000 in the logarithm is the typical number of Bitcoin transactions in a single block, and 32 bytes is the size of a transaction Id. These merkle proofs can be deleted later on. Furthermore, block header data could be pruned on a regular basis.

Computationally, the most demanding task is to calculate signatures. ECDSA is an efficient digital signature scheme than can run on restricted devices (Rifà-Pous and Herrera-Joancomartí 2011). Notably, for ECDSA the signature operation is faster than the verify operation. Since we rely on SPV, the sensor does only need to do the signature operation and not the verify operation. Sig-

nature creation takes less than 0.5 s on a typical low-power constrained IOT microprocessor such as the ARM Cortex-Mo (Tschofenig and Pegourie-Gonnard 2015). Thus, the hardware requirements are met by a typical connected device and further optimizations are possible.

5.7 DISCUSSION

In this section we synthesize the findings of Sec. 5.3 and Sec. 5.6.

LOW LATENCY COMES WITH SECURITY RISK Although transactions propagate the network in a matter of seconds Bitcoin's consensus process provides only eventual consistency. Conflicting transactions will eventually be resolved, but practical finality may only achieved on the order of tens of minutes. In many cases, sensor data is time critical. Thus, a sensor requiring practical payment finality before releasing data might not be competitive. On the other hand, if the sensor does not wait for a confirmation by the network, a malicious requester could perform a double-spending attack.

LOW FRICTION BUT HIGH TRANSACTION COSTS Bitcoin payments can be completely automated. If a transaction is included in a block with sufficient depth, a machine can be sure that it has control over the coins. Just like a vending machine can be sure as soon as someone inserts a coin. All other forms of online payments can be disputed and reverted, and the owner or some delegated third party service has to take care – adding friction. Thinking of billions of sensors, the traditional systems do not scale. Thus, in principle Bitcoin provides a superior basis. Nevertheless, we have seen that over the years transaction costs have risen tremendously because block size got scarce. Furthermore, the dust limits inhibits individual transaction below the order of cents.

COMBINATION BETWEEN ON- AND OFF-CHAIN IS NEEDED The concept illustrated in Fig. 5.1 suggests the usage of the Bitcoin network as medium for money *and* data exchange. In this case no further communication channels are needed. However, latency, transaction costs and the need for confidentiality suggest that additional means for communication are required.

DECENTRALIZING THE REPOSITORY In the prototypical implementation, we introduced a centralized repository that provides a means for discovery between sensors and requesters. Authentication based on Bitcoin's public key cryptography provides built-in integrity of the records, and as soon as a requester knows about the existence of a particular sensor, the requester is not dependent on the repository anymore. The actual exchange does not depend on the repository. However, the repository may censor specific sensors, and poses a target for DoS attacks. We believe, that a decentralized repository is

not the critical part of the infrastructure, but could be implemented based on technologies like Blockstack (Ali et al. 2016) or by means of a DHT (c.f. OpenBazaar in Sec. A.3).

5.8 RELATED WORK

At the time we presented the concept in (Noyen et al. 2014) and provided a first prototypical implementation (Wörner and Bomhard 2014), there was no work concerning the usage of cryptocurrencies to build an infrastructure for S²aaS or other business models that benefit from machine-to-machine payments. In this case, related work refers not to work that has already existed at the point the research was conducted and presented, but on work that has been done since then. Filament aims to implement the concept of S²aaS in business and industrial settings instead of consumer devices. Direct bitcoin payments to the device will be possible, but the default model will be to have a backend service, either run by Filament or by the client company, that can issue payment receipts. These payment receipts can then be presented to the device in exchange for data. While this architecture might be suitable for sensor networks that are mainly in operation for one or a few companies, this does not scale to consumer devices with millions of individual device owners. Notably, Filament is developing a new light-weight communication protocol[15] with end-to-end encryption and without the need for a centralized messaging broker. 21 Inc. provides another interesting basis for S²aaS. They built easy to use command line tools and open source software libraries to interact with the Bitcoin network, either directly or by using propriety services provided by the 21 backend infrastructure. The libraries allow to conveniently build bitcoin-payable RESTful HTTP API endpoints. There are three different payment schemes which client and server can negotiate: (1) normal bitcoin payments, (2) payment channels (see Sec. 6.2.2), and (3) so called *BitTransfers*. BitTransfers are *off-chain* payments with 21 Inc. as the custodian. Essentially, 21 keeps an internal ledger that is updated with every BitTransfer, allowing to transfer earned bitcoins to an *on-chain* address in bulk at a later time. Conveniently, users of the 21 Inc. software typically opt-in to be part of a Software Defined Network (SDN) which allows to overcome the typical issues to connect with devices behind routers and Network Address Translation (NAT). On the 21 Inc. overlay network all devices are addressable as if they were on a local network. This allows to establish direct communication channels between requesters and sensors. We developed a S²aaS system based on the 21 Inc. infrastructure (Wörner 2016), but the reliance on proprietary infrastructure counteracts its use as a global S²aaS platform. More information on Filament and 21 can be found in the case studies in Appendix A.

15 http://telehash.org/

Besides these industry endeavors, there has also been related work in academia. Y. Zhang and Wen 2015 presents the concept of an *IoTcoin*. Ownership of an IoTcoin allows, for example, to access data generated from a particular device. The IoTcoin is implemented using the colored coins protocol and could be issued by a sensor owner and sold to a requester. This model is essentially a way to pre-pay for the extended usage of a sensor. Thus, the IoTcoin acts like a payment aggregation. Furthermore, an IoTcoin owner is in principle able to sell the IoTcoin to another requester. However, larger upfront payments require additional trust in the sensor, and hence reinforce the need of a reputation system.

Delgado-Segura, Tanas, and Herrera-Joancomartí 2016 propose the *PaySense* framework which uses Bitcoin for rewards and as a reputation system. The framework introduces a Data Collection Server (DCS) and an Address Certification Authority (ACA). A sensor has to acquire a certificate from the ACA in order to receive tasks and rewards from the DCS. The sensor gets only paid by the DCS, if the measurement task fulfills specific quality requirements. Hence, the payment can be interpreted as a reputation. The clever part of the system is that a sensor can transfer the reputation to a new address, i.e. a new identity, in a privacy-preserving way by using a CoinJoin-based mixing service. The system is not permissionless, since the ACA decides which sensors to certify, but the ACA can not link the sensor with its identity in the system. Furthermore, in principle, multiple interoperable ACA (and DCS) are possible.

5.9 CONCLUSION

The IOT is expected to consist of billions of sensor nodes bridging the gap between the physical and digital world. Based on the idea that not only the one who generates data can profit from it, the concept of S²aaS foresees global sensor data markets. However, to carry this idea from theory to practice, there are manifold systemic hurdles to overcome.

Individual sensor owners have to be incentivized to make their data publicly available in a well-structured and meaningful format, such that machines can automatically identify relevant data providers and procure data without human due diligence. The underlying infrastructure should be open, permissionless and censorship-resistant in order to foster participation and innovation.

In this chapter, we identified and described characteristics of cryptocurrencies, and Bitcoin in particular, that serve the need of a global S²aaS infrastructure. We described the concept of exchanging data for electronic cash using the Bitcoin infrastructure exclusively. Based on this concept, we implemented a prototypical system based on a sensor application and a requester application. The prototype was extended by a central repository to provide a means for discovery as well as a simple reputation system. The analysis of the concept and the evaluation of the prototype showed that cryptocurren-

cies are a promising economic layer for machine-to-machine payments, and S²aaS in particular. The global and permissionless payment infrastructure allows machines to accept payments with finality, and can thus deliver service without counterparty risk. This provides a secure basis for an automated machine economy. The built-in cryptographic primitives of cryptocurrencies can be used to authenticate sensors and their measurement data to ensure integrity (ECDSA), and can be used as a basis for confidentiality by encryption (ECIES). Critical data can be blindly committed to the blockchain to provide an immutable proof of existence at a particular time. Moreover, S²aaS based on cryptocurrencies may provide the missing incentives to equip connected devices with dedicated hardware for cryptographic operations and secure key storage, and make security a first class priority.

In general, communication and data delivery should happen on other channels than the Bitcoin network itself. Data transfer on the Bitcoin network adds additional costs, and can not provide confidentiality due to size constraints. Furthermore, the Bitcoin network is an unstructured, dynamic network that provides resilience while sacrificing latency.

However, secure payments have high latency on the order of tens of minutes, and transaction fees have risen dramatically during the last month. These fees are proportional to the size (in bytes) of a transaction instead of their value (in bitcoins or dollars). Furthermore, the dust limit, which is related to Bitcoin's UTXO architecture, prohibits payments below cents. Hence, low-latency micropayments, which would allow a low-minimized automated machine economy to flourish, seem to be out of reach. In the next chapter, we will see how the programmability of Bitcoin allows to setup contracts between untrusted parties to enable Bitcoin micropayments.

6 ENABLING INSTANT MICROPAYMENTS FOR CROWDSENSING APPLICATIONS

Everything must be made as simple as possible. But not simpler.

— Albert Einstein

6.1 CONTEXT AND MOTIVATION

One of the main drivers to utilize a cryptocurrency for applications like S²aaS is the opportunity to exchange data for cash without additional counterparty risk or long term contractual relationships. In the last chapter, we saw that Bitcoin's blockchain-based consensus and the UTXO architecture interfere with low-latency micropayments, which are essential for many sensing tasks. In this chapter, we discuss approaches to enable low-latency micropayments based on Bitcoin with the focus on going *off-chain* without losing the security an openness of Bitcoin. One method is known as payment channels. These are simple yet powerful smart contracts allowing low-latency micropayments between a buyer and a seller. While direct payment channels are useful for particular use cases in which a buyer is only interested in buying repeatedly from a single seller, S²aaS and mobile crowdsensing, however, require in general the simultaneous trades between a buyer and a large number of sellers and vice versa. This would require to establish contracts between each trading pair which may lock up significant capital and cause significant transaction costs. Therefore, we develop an enhancement of payment channels in order to scale to a large number of participants. To achieve this, we make a compromise on decentralization and introduce a hub. The hub is responsible for trust-minimized mediation of payments and data delivery. Although we sacrifice censorship-resistance and resilience, the power of the hub is strictly limited. We refer to this limitation by *trust-minimized mediation* which is operationalized as follows. Trust-minimized mediation of payments is achieved by means of HTLCs. HTLCs are expressed in Bitcoin's scripting language to form a smart contract that provides atomicity of two payments in two distinct payment channels. Trust-minimized mediation of data delivery is achieved by encryption and digital signatures provided by the cryptographic primitives underlying Bitcoin. We implement a proof-of-concept prototype targeted at the usage in crowdsensing scenarios. Smartphones are the largest sensing platform with an unprecedented scale and global distribution. The application allows smartphone users to offer built-in sensors as a service in

exchange for payment with minimal human involvement. The protocols can be open and implemented into other smartphone applications, or ported to other sensor platforms or connected devices.

The structure of the chapter is as follows. In Sec. 6.2, we provide an overview of methods to enable low-latency micropayments with bitcoin. The main focus thereby is on payment channels. In Sec. 6.3, we present the Trust-minimized mediation of payment channels using HTLCs and briefly discuss the trust-minimized mediation of data delivery and discovery in Sec. 6.4. Thereafter, we present the system and its main processes in Sec. 6.5. Sec. 6.6 provides implementation details. In Sec. 6.7 we evaluate the concept mostly based on criteria already introduced in Chapter 5. In Sec. 6.8 we present a discussion concerning the costs of running a hub, developments towards bidirectional payment networks and a comparison to Ethereum. Furthermore, we briefly recapitulate the key findings and set the results in a broader picture by synthesizing with Chapter 5. The chapter ends with a conclusion (Sec. 6.9).

6.2 LOW-LATENCY MICROPAYMENTS WITH BITCOIN

As long as the scalability issues concerning latency and transaction throughput of decentralized cryptocurrencies are not solved (c.f. Sec. 3.1.3), we need efficient methods of payment aggregation to enable low-latency micropayments with Bitcoin. In the following, the state of art is presented.

6.2.1 *Off-Chain Micropayments with Trusted Third Parties*

Centralized third party services have emerged allowing zero fee microtransactions between their users. Examples are Coinbase[1] and ChangeTip[2]. Thereby users deposit coins in addresses controlled by the third party service. The service then handles transactions between users internally, i.e. off-chain. In principle, we could setup our own service that provides an internal ledger with accounts for requesters and sensors. After a requester credits its account with a regular Bitcoin transaction of sufficient value, it could transact with all registered sensors, and payments could be handled in milliseconds. However, with this approach we lose most of the characteristics that Bitcoin provides in the first place. The system is now entirely dependent on the TTP. This might have far reaching consequences. First, we lose censorship-resistance. Hence, the service provider can ban parties from accessing the system. Moreover, the TTP is situated in a particular jurisdiction which might require the compliance to certain KYC/AML laws. This would imply human involvement and impacts the privacy of participants. Second, the TTP is custodian over all user funds, and may provide an attractive aim for hackers. In addition,

1 https://www.coinbase.com
2 https://www.changetip.com

the existing services are proprietary and may try to engage in rent-seeking behavior. Furthermore, requesters and sensors have to communicate with proprietary APIs which might be subject to change. In the worst case the TTP might go out of service. Implementing the service as an open source platform such that there is competition between TTPs would be beneficial, but network effects may lead to centralization.

6.2.2 *Payment Channels*

The idea to use Bitcoin contracts to establish payment channels between two parties was introduced by Hearn and Spilman (Hearn 2013; Spilman 2013). Payment channels are based on a timelocked 2-of-2 multi-signature output that is funded by the payer. The transaction creating this multi-signature output is called *funding transaction* and can be represented as $Tx_{funding}(addr_{payee}, T_{expiry}, C)$ (see Fig. 6.2 for the pubScript). C is the amount locked in the multi-signature output and represents the capacity of the channel. The need for a channel lifetime, as given by T_{expiry}, will be discussed later on. As soon as the funding transaction has sufficient confirmations, the payee can be sure that the funding value is locked, and the channel is established. In order to pay the payee, the payer creates a transaction that consumes the multi-signature output and splits the value in two outputs, one spendable by the payer, and one spendable by the payee. The payer sends the transaction to the payee via some arbitrary communication channel. The payee is now able to add its signature and broadcast the transaction to the Bitcoin network in order to close the channel. However, the payee can also just keep the transaction, and wait for another payment transaction that increases its share of the value in the multi-signature output. The important point is, that the payee can be sure about the payment as soon as it gets the transaction. Hence, after initial setup, individual payments are low-latency, and are exchanged off-chain. Furthermore an incremental update of the shares can be as low as 1 satoshi, and thus individual payments are true micropayments.

So far, the payee can close the channel at any time by broadcasting any of the private payment transactions to the Bitcoin network. However, the payer can not, since it has only partially signed and thus invalid transactions. In order to protect the payer against a payee that aborts the protocol leading to locked funds, the output can also be spent by the payer alone using a *refund transaction* after time T_{expiry}, the effective lifetime of the channel.

Every payment channel involves two on-chain transactions. The funding transaction, and one payment transaction or the refund transaction for unilateral closing of the channel after the lifetime expires.

Noteworthy, such a payment channel is unidirectional. If the payee would at some point create a payment transaction that decreases its share and increases the payer's share, provide the partial signature, and send it to the payer,

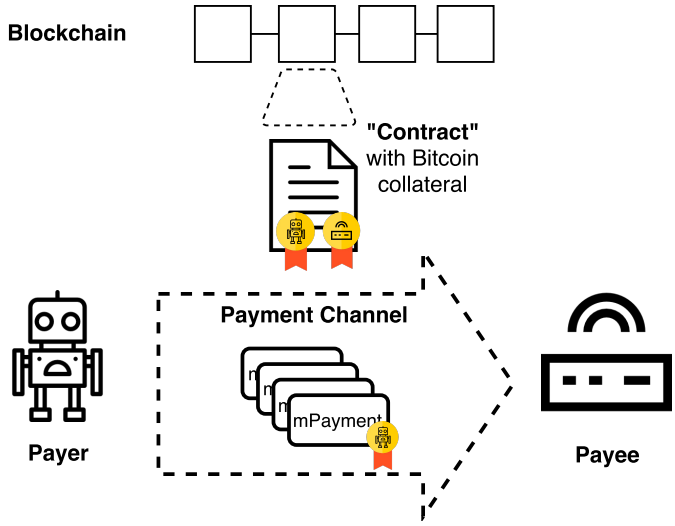


Figure 6.1.: Abstract illustration of an unidirectional payment channel between a payer (A) and a payee (B). The contract is a 2-of-2 multi-sig output on the blockchain, that can either be spent by B after time T_{expiry}, or immediately by A with one of the payment transactions which update the share between A and B.

then there could be a race condition, since both parties would have valid transactions with different allocations of the funds.

A simplified graphical representation of the protocol is depicted in Figure 6.1. Payment channels have been implemented in various clients and libraries. Although all exchanged transaction are valid Bitcoin transactions, the communication protocol and the state machines of payer and payee are not necessarily interoperable. Hence, standardization is necessary.

```
OP_IF
   OP_2 <PubKey A> <PubKey B> OP_2 OP_CHECKMULTISIG
OP_ELSE
  <T_expiry> OP_CHECKLOCKTIMEVERIFY OP_DROP
  OP_DUP OP_HASH160 <PubKeyHash A> OP_EQUALVERIFY OP_CHECKSIG
OP_ENDIF
```

Figure 6.2.: pubScript of the funding transaction for a payment channel. The first branch of the conditional needs both signatures and is used for the payment transactions. The second branch can be used by the payer after T_{expiry} for refund.

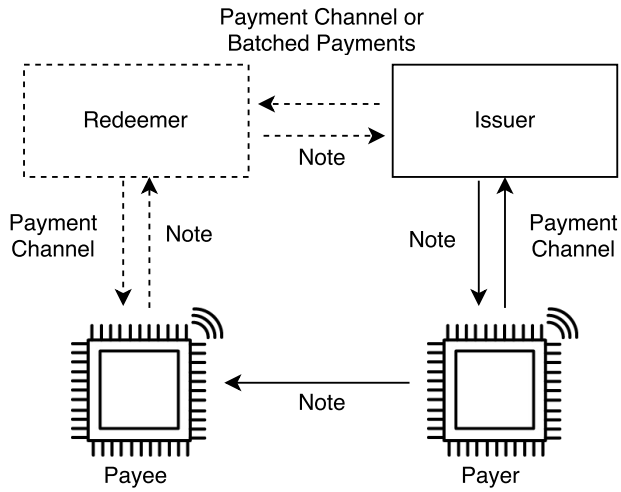


Figure 6.3.: Possible participants, relationships and transaction flows in a micropayment scheme based on promissory notes.

6.2.3 *Further Bitcoin Micropayment Technologies*

Digital Promissory Note

In general, a promissory note is a written financial instrument, in which an issuer promises the bearer to pay a specific amount of money under specified terms. Thereby, a payee is able to accept payments by different payers without having to trust them individually, but by only having to trust the issuer, that it will be able to redeem the promissory notes eventually. Micropayments in form of promissory notes can be aggregated, and redeemed at the issuer as soon as a substantial amount is reached. An implementation of this concept is the Stroem protocol (Fransson 2015). In this protocol, promissory notes are dedicated for a specific purpose, i.e. a specific purchase. Thus, the payer needs to procure the promissory note at time of the transaction. Procurement of promissory notes is a suitable application for unidirectional payment channels. A payer can open a payment channel with an issuer and purchase low-value promissory notes in order to keep the counterparty risk low. Depending on how many parties accept the promissory notes of the particular issuer, the effective reach and usefulness of the payment channel may be increased significantly. An illustration of payments based on promissory notes is shown in Fig. 6.3.

The Stroem protocol is partly proprietary and there is limited counterparty risk.

Method	Counterparty Risk	Scaling
Off-chain	high	good
Payment channel	no	bad
Promissory Notes	medium	good
Probabilistic Payments	low	medium

Table 6.1.: Comparison of Bitcoin micropayment schemes.

Probabilistic Payments

Probabilistic payments (Wheeler 1996; Rivest and Shamir 1996; Rivest 1997) are lottery-based payments. The lottery is biased in such a way that a single draw has an expected value according to the aspired micropayment. Probabilistic payments are only fair, i.e. the actual value paid approaches the value that should be paid, for a large number of payments on the order of tens of thousands individual payments. Pass and 2015 presents such a lottery-based micropayment scheme for ledger-based transaction systems, and discusses implementations in Bitcoin. However, an implementation that does not rely on a partially-trusted third party would need a new signature verification primitive in the Bitcoin scripting language. The partially-trusted third party acts as an escrow service and signs a transaction in case the payee is able to provide a winning ticket. In contrast to payment channels, a single UTXO could be used to pay more than one party.

Noteworthy, probabilistic payments could be combined with payment channels in order to allow for sub-satoshi payments.

Table 6.1 shows a brief comparison of the presented schemes. The off-chain scheme with a TTP has the greatest counterparty risk. Technically, it is the easiest to implement, and payments could in principle be arbitrarily fast and cheap. If a significant number of requesters and sensors have accounts with a single provider, then one Bitcoin transaction can be used to buy data from a large number of sensors. However, a TTP needs to be trusted and needs to fulfill these expectations. Payment channels are peer-to-peer and do not involve any intermediary. However, every requester-sensor pair would need its own payment channel. Thus, transaction costs and locked up capital of a requester scales proportional to the number of sensors it would like to transact with. The scheme based on promissory notes allows for better scaling of payment channels and reduces the counterparty risk of having only one centralized TTP. Nevertheless there needs to be an infrastructure of semi-trusted service providers, and thus far, there is none. Probabilistic payments do have a low counterparty risk since the service provider cannot steal money. However, scaling to many sensors is not much better than with payment channels.

Combining payment channels with a central provider, a hub, allows to scale the reach of a single channel almost indefinitely while simultaneously keeping

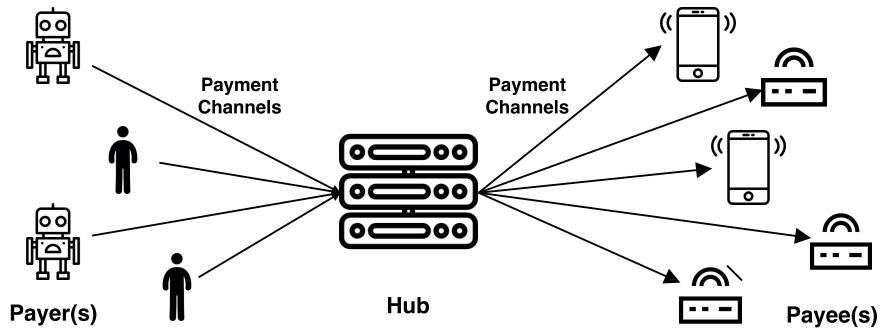


Figure 6.4.: Scaling payment channels by connecting them in a hub and spoke architecture.

the counterparty risk low. However, we can decrease the counterparty risk even further by using another type of smart contract.

6.3 TRUST-MINIMIZED MEDIATION OF UNIDIRECTIONAL PAYMENT CHANNELS

In the early days of telephony, individual communication lines were merged at large switchboards where operators connected these individual communication channels to establish temporary end-to-end communication channels between two parties. As briefly indicated at the end of last section, we can do the same for payment channels. Figure 6.4 illustrate the resulting architecture. A payer can minimize its counterparty risk with the hub by making each micropayment individually, and waiting for the paid-for data from the sensor before making the next micropayment. But this would involve a lot of sequential transactions, if a large number of sensors need to be queried simultaneously. Thus, the payer has to balance between counterparty risk and efficiency.

However, we can make use of a slightly more complicated smart contract that allows to interconnect payment channels temporarily, such that an individual micropayment taking that path is atomic, i.e. the payment either travels through both channels or no payment happens at all. We can achieve this behavior by using HTLCs which were introduced conceptually as an enabler of the Lightning Network (Poon and Dryja 2015). We can express a HTLC between two parties in common words as follows: I pay you if you can provide a secret within a certain time period. By conditioning two payments in two individual payment channels on the same secret, we are able to atomically connect both payments.

Figure 6.5 illustrates the flow of an individual payment over two payment channels. We assume that A and C, as well as C and B already have established payment channels, i.e. there exist shared multi-signature outputs with time-

locks T_0 and T_0'[3]. In order to make it more concrete, we assume that the shared funding outputs have a value of 1 BTC each, and A wants pay B an amount of 0.1 BTC. Moreover, we assume that the channels are fresh, i.e. no payment transactions have been exchanged. The protocol is as follows: First, the final recipient B creates a payment-specific random secret S, computes $H = hash(S)$, and communicates H to C and A. H act as the hashlock for both HTLC outputs and will provide the atomicity of the process. A creates a payment transaction that consumes the shared multi-signature output and creates two outputs: (1) an output assigning 0.9 BTC to her, and (2) a HTLC output with a value of 0.1 BTC[4]. Figure 6.6 shows the pubScript of such a HTLC output as used in the payment transactions. A provides her signature for the transaction and sends it to C. C may now sign the transaction as well, and broadcast it to the Bitcoin network. However, without S, C will not be able to claim the 0.1 BTC. Thus C stores the transaction and creates another payment transaction addressing C with a HTLC output requiring the same secret. C signs the transaction and sends it to B. B could sign the transaction and broadcast it to the network. Since B knows the secret, B could claim the 0.1 BTC locked in the HTLC output. But would have to do it before T_2. Otherwise B would be able to claim the output. B would claim the HTLC output by broadcasting a transaction that entails the S. Thus, S would be public and C could use it to claim the HTLC output of the payment transaction from A. To ensure that C is always able to do that before A is able to reclaim the value, T_1 has to be sufficiently later than T_2. If timelocks are selected appropriately, a payment between A and B, mediated by C using HTLCs is atomic, i.e. either both payments succeed or both payments fail. This means also that B can just communicate the secret privately to C and A, and they update their shares accordingly. Hence, A can create a transaction that replaces the HTLC output with an output granting C the 0.1 BTC. After signing and sending the transaction to C, C would do the same concerning the HTLC with B, and the payment is concluded without any on-chain transaction.

If a payer wants to pay many payees simultaneously, each payment would be represented by an individual HTLC output in the payment transaction.

If all parties cooperate, then there will be no payment transactions with HTLC outputs ending up in the blockchain. From the perspective of the Bitcoin network the situation is equivalent to two independent channels, providing additional transactional privacy for requesters and sensors.

6.4 TRUST-MINIMIZED MEDIATION OF DATA EXCHANGE AND DISCOVERY

In the last chapter, we found that data exchange via the Bitcoin network is not viable due to latency, cost, and confidentiality. In the last section,

3 The actual timelocks are not important. However, they have to be longer than the HTLC timelocks T_1 and T_2.

4 We neglect necessary fees in order to provide a cleaner explanation.

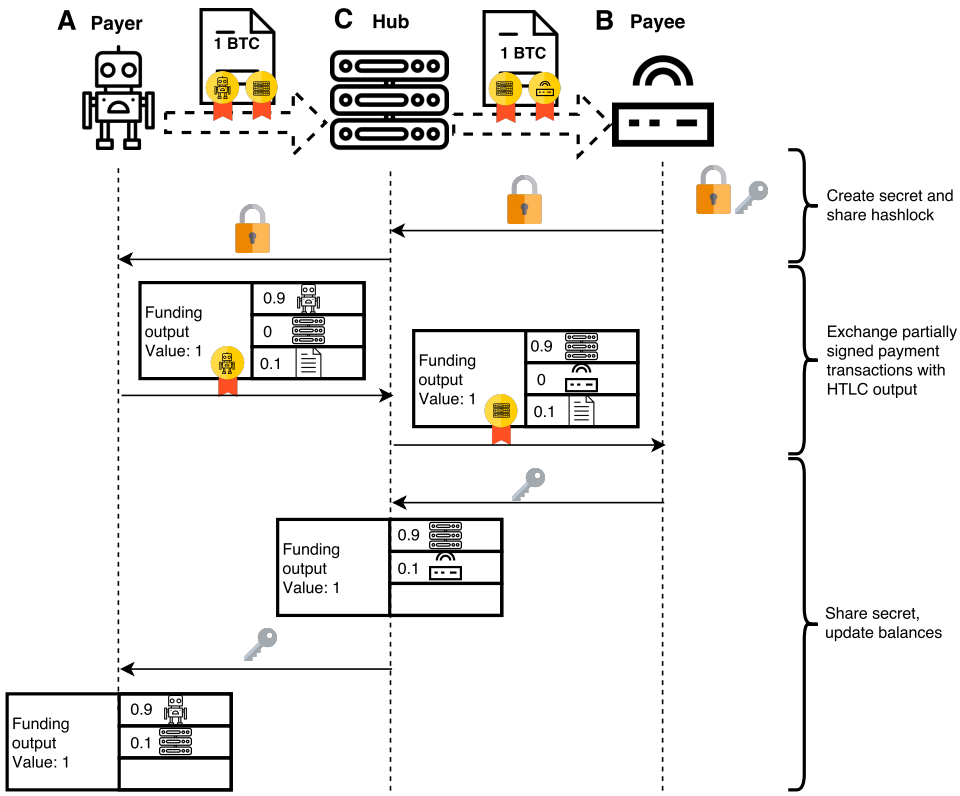


Figure 6.5.: Protocol for an individual mediated payment. In contrast to normal payment channels, the payment transactions have at least one HTLC output (denoted as the contract sheet).

```
OP_IF
  OP_DUP OP_HASH160 <PubKeyHash B (C)> OP_EQUALVERIFY OP_CHECKSIG
  OP_HASH160 <Hash160 (secret)> OP_EQUAL
OP_ELSE
  OP_DUP OP_HASH160 <PubKeyHash A (B)> OP_EQUALVERIFY OP_CHECKSIG
  <T1 (T2)> OP_CHECKLOCKTIMEVERIFY OP_DROP
OP_ENDIF
```

Figure 6.6.: HTLC pubScript of a payment transaction in an unidirectional mediated payment channel setup. The first branch of the conditional can be used by the recipient to claim the output by providing the secret, the second branch can be used by the sender to reclaim their funds after T_1 (T_2).

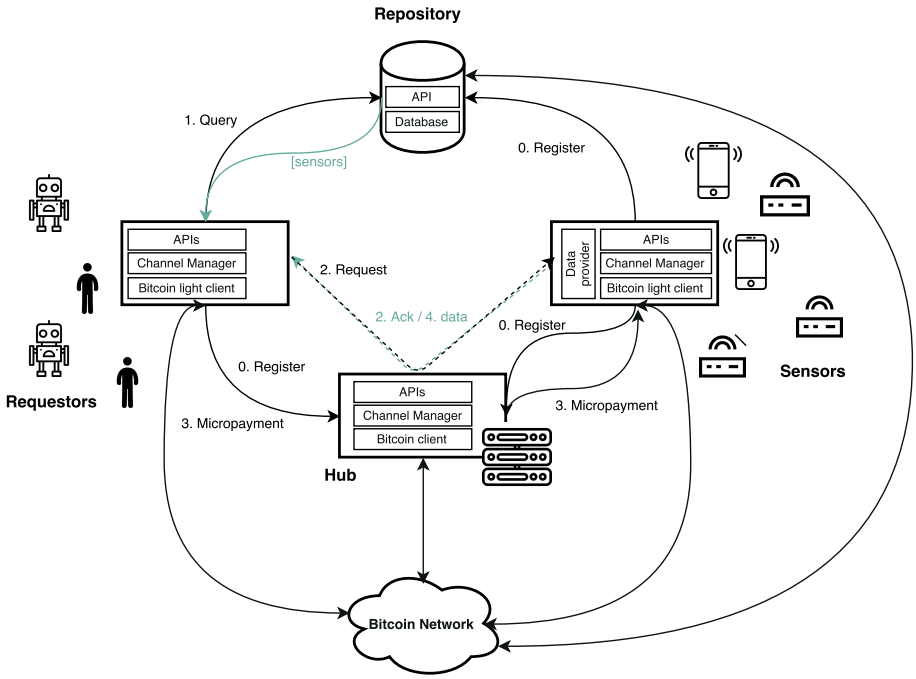


Figure 6.7.: Overview of system's architecture.

we introduced a payment hub in order to allow efficient trust-minimized micropayments between a larger number of requesters and sensors. The hub needs to keep connections with requesters and sensors in order to be able to exchange Bitcoin transactions off-chain, i.e. privately and not via the Bitcoin network. These communication channels can also be used to deliver the measurement data.

Although the communication between a sensor and a requester might be mediated by the hub, we do not necessarily need to trust the hub. The sensor can encrypt the data using ECIES based on an Elliptic Curve Cryptography (ECC) public key of the requester, and sign the data using ECDSA. Encryption ensures confidentiality, i.e. the hub does not see what it is transferring. Digital signatures ensure authenticity and integrity, i.e. the hub can not manipulate the data it is transferring. Furthermore, digital signatures can be used to verify the authenticity of entries in the repository, as already discussed in the last chapter.

6.5 SYSTEM OVERVIEW

In the following ,we briefly present the participating entities and describe the processes. A graphical illustration of the system is shown in Fig. 6.7.

6.5.1 *Entities*

The system consists of a requester client, a sensor client, a repository and a hub. The repository and the hub may be one entity, but need not be.

THE REPOSITORY stores sensor meta data and provides a queryable API for requesters.

REQUESTERS may query the repository and want to procure data from one or more sensors. Requesters have to be able to fund and manage a payment channel, and to provide payments in form of HTLC payment transactions.

SENSORS provide timestamped measurement data for requesters. The data is delivered confidentially and authenticated.

THE HUB provides Trust-minimized mediation of micropayments and data exchange. Thereby, the hub has to be scalable and secure in order to manage a potentially large number of payment channels.

6.5.2 *Processes*

Register and Channel Setup

Before a sensing task can be performed, requesters and sensors have to register with the hub in order to setup payment channels. This process is asymmetric, since the introduced payment channels are unidirectional, and are directed from the requester towards the sensor.

The requester essentially registers by establishing a payment channel with the hub. The procedure is listed below.

Procedure 6.1 Register with Payment Hub (Requester)

1: **Request** $pubKey_{hub}$
2: **Create** funding transaction $Tx_{funding}(pubKey_{hub}, T_{expiry}, C)$
3: **Broadcast** $Tx_{funding}$
4: **Send** ID($Tx_{funding}$) to hub
5: **Register event** for Tx_{refund} at T_{expiry}
6: **Wait until** confirmation from hub
7: **Channel established**

The sensor has to request a payment channel from the hub. In the simplest case it provides its public key and waits until the funding transaction has sufficient confirmations in the blockchain (c.f. Procedure 6.2)

In practice, the hub might require additional security from the sensor. We discuss this is in Sec. 6.7.6.

Procedure 6.2 Register with Payment Hub (Sensor)

1: **Request** channel by providing $pubKey_S$
2: **Wait until** ID($Tx_{funding}$)
3: **Wait until** confirmation of $Tx_{funding}$ (Bitcoin Network)
4: **Send** Channel established to hub

Data Procurement and Payment

Once payment channels are established, a requester can request data from a sensor. The request entails the requester's public key and is signed. The sensor performs the sensing task and provides the hashlock $H(S)$ as an authenticated message. Once the requester receives the message, it adds an HTLC output with a value corresponding to the price of the data to the current payment transaction. The requester sends the transaction to the hub, and the hub does the same based on the current payment transaction to the sensor. Then, the sensor encrypts the data with the requester's public key and creates a signed message that contains the encrypted data and the secret S. When the message reaches the requester, the requester can verify the authenticity of the message and decrypt the data with its private key. Now the process can begin again. Otherwise the requester removes the HTLC output and adds the value to the output belonging to the hub and sends the updated payment transaction to the hub. The hub does likewise. An illustration of the process is shown in Fig. 6.8.

Closing a Channel

The requester has to send a request to the hub, if it wants to close the channel before the channel capacity is depleted. The hub will then sign the latest payment transaction and broadcast it to the Bitcoin network. If the hub does not cooperate, then the requester can reclaim the entire value locked in the channel as soon as the channel expires. The sensor, on the other hand, can always close a channel autonomously by broadcasting the latest payment transaction itself.

The receiving side of the channel has to make sure that it closes the channel with sufficient margin to the expiry time. Otherwise a race condition with a refund transaction can occur.

6.6 IMPLEMENTATION

In order to leverage the largest mobile sensing platform available today, we implemented the sensor client as an Android smartphone application. The Android API has a lot of similarities to the Java API. Hence, we used Bitcoinj[5], a

5 https://bitcoinj.github.io/

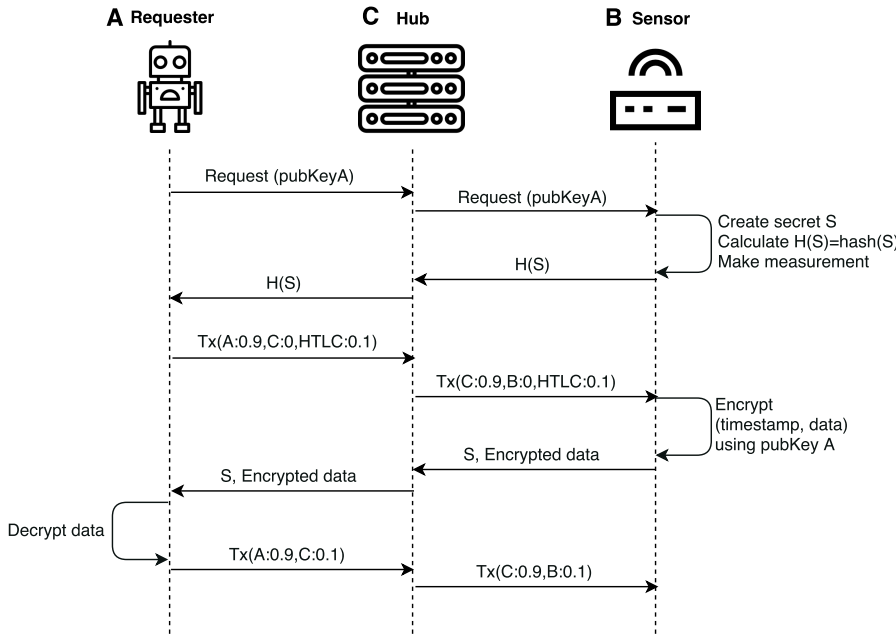


Figure 6.8.: Process of requesting, paying, and delivering data. Tx($P_i : v_i$) denotes a transaction with output of value v_i redeemable by party P_i, and $P_i = HTLC$ denotes a HTLC output. Authentication is not illustrated explicitly.

Sensor	Type	Application
Barometric pressure	physical	Weather prediction
Location	both	People flow
Network strength	physical	Coverage maps
Installed apps	virtual	Inter-app correlations
Transportation mode	virtual	City monitoring
Steps	virtual	Health monitoring

Table 6.2.: Examples of virtual and physical sensors available in smartphones.

Java library for working with the Bitcoin protocol, as a starting point. Besides being able to run on multiple platforms, Bitcoinj provides the first implementation of a SPV client and the first payment channel implementation. In fact, at time of implementation (Q2-Q3 2015) there was no other implementation of payment channels available. Thus, Bitcoinj provides a suitable basis for a cross-platform implementation of HTLC-based mediated payment channels.

In order to implement HTLCs, we extended the four layers of Bitcoinj's payment channels with a fifth layer that is concerned with keeping track of the HTLC flow. Messages and transactions are serialized using Google protocol buffers[6] and are exchanged between the components over Transmission Control Protocol (TCP) connections. Embedding those connections into higher level protocols such as HTTP or Extensible Messaging and Presence Protocol (XMPP) is straightforward. The payment channel implementation is based on a client-server architecture. The payer is instantiated as a client, and the payee is instantiated as a server. The hub has to instantiate both, a server for the requesters and a clients for the sensors. In the following, we briefly present the individual entities.

6.6.1 *Sensor*

The sensor is implemented as an Android smartphone application. The application has two main parts. First, a user interface that allows a user to select sensors, which are offered as a service (c.f Fig. 6.9). Furthermore the user is able to set prices. The proof of concept implementation does only provide the standard sensors of the Android API. However, more interesting data sources could be added. See Table 6.2 for some interesting sensors and their applications. Second, a service which is running in the background. The service implements a payment channel server and is responsible for communicating with the hub and the Bitcoin network.

6 https://developers.google.com/protocol-buffers/

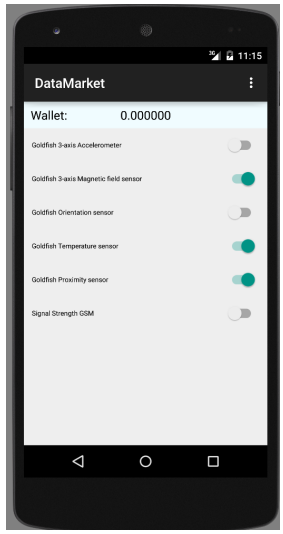


Figure 6.9.: Main screen of the crowdsensing smartphone application. It allows to offer available sensors and a Bitcoin wallet, able to keep track of payment channel states.

6.6.2 *Requester*

The requester client provides a console to query the repository and to procure sensor data in a simple Structured Query Language (SQL)-like language. The main commands and a their descriptions are shown in Table 6.3.

6.6.3 *Hub and Repository*

The hub and the repository are in principle two separate entities. In particular, because the sensor might be connected to different hubs or provide endpoints to setup direct payment channels. However, for the sake of simplicity, the proof

Command	Description
REGISTER <capacity> <lifetime>	Registers with Hub by opening payment channel
CLOSE	Requests to close the channel
STATUS	Returns the status of the channel
STATS NODES	Returns all connected sensor nodes
STATS SENSORS	Returns available sensor types
SELECT SENSOR=<type>	Returns sensors of type <type> with pricing information
BUY <type> FROM <node>	Requests the current measurement value of sensor <type> from node <node>

Table 6.3.: Main commands of the data requester console.

concept implementation combines both functions into one Java application based on Bitcoinj. The hub provides endpoints for requesters and sensors to initiate TCP connections. These TCP connections are used to exchange signed messages which are serialized into Google Protocol Buffers. Payment channel states are identified by the public key of the initiator (requester) or recipient (sensor). Channel states and the most recent payment transactions are written to disk regularly to provide a means of recovery in case the system fails. In a production setting we would use a dedicated database. The hub creates a new public key for every channel using a hierarchical deterministic wallet (Wuille 2012) in order to minimize the loss, if a private key gets compromised. Further security measures can be implemented such as using different keys for the payment channel and the HTLCs. This would allow to keep the (incoming) payment channel keys in cold storage, i.e. stored on a secure computer without network access.

FINAL NOTE At time of implementation the `CHECKLOCKTIMEVERIFY` op code (Todd 2014) was not available in Bitcoin script. Thus, the implementation of payment channels, and HTLCs in particular, was more involved than presented in Section 6.3, and implied additional security risks, because refund transactions had to be created collaboratively (c.f. malleability discussion in Sec. 6.8.2). We evaluate the concept based on the implementation of the current capabilities of Bitcoin Script, i.e with `CHECKLOCKTIMEVERIFY` as described in Sec. 6.3. The actual implementation is described in (Bungiu 2015) and the source code is available on GitHub[7].

6.7 EVALUATION

In the following, we evaluate the concept of trust-minimized mediation of payments and data exchange for crowdsensing applications. We base the evaluation on the criteria used in the last chapter. In addition, we discuss some criteria like scalability, confidentiality, and privacy more explicitly.

6.7.1 *Latency and Double-spending Risk*

We assume that channels are already established. Then, the latency is determined by the latency of communication between the parties, and by the speed of the cryptographic operations. Each step shown in Fig. 6.8 involves signing and signature verification. Depending on the hardware, the whole process is on the order of seconds, and can be accelerated by using specialized cryptographic processors. On-chain Bitcoin contracts prevent double-spending, and signed transactions can be accepted immediately by the receiver. Hence,

7 https://github.com/domwoe/datamarket

propagation and conformation times of the Bitcoin network are no bottleneck anymore.

6.7.2 *Transaction Costs*

Each payment channel requires at least two on-chain transactions. One to open the channel, and one to close the channel. If a channel is closed unilaterally, by broadcasting a payment transaction with HTLC outputs, then additional on-chain transactions may be required for settlement. However, this adds additional costs to the parties able to broadcast the payment transactions (i.e. the receiving parties), and cooperation is thus strongly incentivized. The number of transactions that can be facilitated via these two on-chain transactions is only limited by the capacity and lifetime of the channel. Thus, the marginal transaction cost per procured sensor datum paid to the Bitcoin network approaches zero. However, the hub acts as a service provider and has costs as well. These costs have to be covered by fees billed to the requester. We will discuss this in Sec. 6.7.8.

6.7.3 *Scalability*

Scalability is essentially only limited by the capital in bitcoins available to the hub. Requesters have to provide the coins to open a channel with the hub themselves, but the hub needs to advance the coins to open channels with each sensor. Thus, each additional sensor requires locking up capital and paying fees to the Bitcoin network. Of course, managing a large number of connections, and generating and verifying digital signatures for each payment requires sufficient processing power and bandwidth, but the hub can be a distributed system itself. Hence, in principle, the system can scale indefinitely based on hardware requirements alone.

However, there is an additional bottleneck. With the current block size limit and block rate the transaction throughput of the Bitcoin network is limited to 3-7 transactions per second. Although payment transactions are off-chain, the funding and settlement transactions are on-chain. Thus, even if there would be no other transaction on the network it would take about 2 days to setup channels with 1 million sensors. In reality, the Bitcoin network is already almost at maximal transaction throughput (see Fig. 6.1).

6.7.4 *Confidentiality and Integrity*

Confidentiality of data can be provided by Bitcoin's built-in ECC in form of ECIES. Data is then encrypted based on the public key of the requester. However, this implies that the sensor needs to know the public key of the requester. Since the public key is also communicated via the hub, the hub

could generate a new key pair and provide the corresponding public key to the sensor. In order to minimize the risk, the sensor could check the blockchain for a funding transaction corresponding to the public key, and only accept the public key if there is one.

Integrity is provided by signing messages using ECDSA. Thus the entire infrastructure for integrity and confidentiality is directly provided by Bitcoin.

6.7.5 *Privacy*

In the case of direct Bitcoin payments as discussed in the last chapter, every transaction is recorded on the blockchain. Although identities on the blockchain are pseudonymous, in most cases at some point a non-pseudonymous entity can be linked. This happens for example at a regulated crypto-fiat exchange or at a merchant. Mediated payment channels provide additional privacy, at least to an outside observer. Only funding and closing transactions are recorded on the blockchain. Thus, only the links to the hub are public, and not the individual transactions between requesters and sensors. However, the hub knows about every transaction, but not about the data content. Of course, there are privacy risks for the sensor owner depending on the actual data offered. Data that is enriched with location information can quickly lead to identification of the smartphone owner in the case of mobile crowdsensing (Montjoye, Hidalgo, et al. 2013).

6.7.6 *Malicious Sensors*

Crowdsensing and S²aaS schemes are dependent on a large number of sensors. Thus, it is very important to keep the friction of participation low. The here presented scheme does require no more than installing an application. In particular, the sensors do not need any bitcoins to begin with. However, each registered sensor requires the hub to initiate a payment channel with a particular capacity and lifetime. Hence, in the worst case this locks up bitcoin corresponding to the capacity for the entire lifetime of the channel, and consumes fees for funding and refund transactions. If we provide a crowdsensing application via the Google Play store, then we can be reasonably sure that only genuine devices try to register. However, we aim for an open permissionless platform, and here a malicious party could try to register sensors until the capital of the hub is depleted. In practice, a permissionless system is hard to achieve. One approach would be to adopt the idea of Bitcoin's proof-of-work and require a sensor to provide a reasonable proof-of-work based on a challenge from the hub.

6.7.7 *Censorship-resistance and Resilience*

Although the hub provides trust-minimized mediation of payments and data exchange, the introduction of a centralized facilitator has consequences. The hub is able to exclude particular sensors and requesters, either on its own terms or compelled by regulation or law enforcement. Furthermore, a single hub provides an attack surface for DoS attacks which could take down the entire system. Ideally, the system would be able to provide a fallback to regular on-chain Bitcoin transactions or to direct payment channels. This would imply to have a means for direct communication between requesters and sensors. Protocols such as BitMessage[8] or Telehash[9] could be used. More general overlay networks such as utilized by 21 Inc. are another alternative.

6.7.8 *Hub Costs and Revenue Mechanics*

Requesters and sensors have an obvious value in taking part in the system. The hub allows to maximized the benefit of individual payment channels and simplifies communication. But why would someone provide this service and run a hub? In other words, the hub needs a business model. In order to have a viable business model, the hub needs to cover at least its cost. The costs of the hub can be divided into the following categories.

HARDWARE AND OPERATIONS consists of hardware requirements and its operational expenses. Infrastructure-as-a-Service could be used to provide a scalable infrastructure without capital costs. There are some base loads such as running a Bitcoin node. In addition, there is a variable load that mostly depends on the number of transactions. However, a typical server is able to handle thousands of transaction per second.

BITCOIN NETWORK FEES have to be provided for on-chain Bitcoin transactions. In the best case, the hub needs to open channels with sensors and close channels from requesters. However, if there are sensors with no demand, then the hub has to close these channels after expiry by broadcasting a refund transaction. These costs are independent of the value transferred through the channels.

COST OF LOCKED CAPITAL is based on the sum of the capacities of all outgoing payment channels and their lifetimes.

The value of the system depends mainly on the participating data providers. Hence, onboarding a sensor should be as frictionless as possible, and fees should be taken from the requesters. The technical implementation of fees is straightforward. The hub just requires a higher amount from the requester

8 https://bitmessage.org/
9 http://telehash.org/

than the sensor demands for each payment. Although more sensors are better in principle, each additional sensor adds costs in all of the above categories. Thus, a hub will try to leverage available information, such as current and historical or reputation measures, if available, to prefer sensors that provide high quality data being high in demand. The hub aims to use payment channels as efficiently as possible and has to balance between lock up of capital and paying Bitcoin transaction costs for establishing and settling of channels.

6.8 DISCUSSION

6.8.1 *Key Findings*

BITCOIN CONTRACTS AND OFF-CHAIN TRANSACTIONS ALLOW LOW-LATENCY MICROPAYMENTS The programmability of Bitcoin transactions allows the creation of smart contracts between parties. These smart contracts have on-chain collateral and allow two parties to exchange and accept Bitcoin transactions directly without relying on the Bitcoin network and the blockchain. Hence, individual payments on the order of μUSD are possible. Each channel, connecting two parties, locks bitcoins and requires two on-chain transactions, one for establishing the channel, and one for settlement. Thus, payment channels alone are not sufficient for crowdsensing scenarios in which a requester is buying data from thousands of individual sensors.

SCALING CAN BE ACHIEVED BY ADDING A HUB WITH LIMITED POWER HTLCs can be used to atomically connect payments in two channels that meet at one party. This allows to route payments via a hub without having to trust the hub. A malicious hub is only able to lock the coins in a channel for the channel lifetime, but cannot steal any funds. The hub can simplify discovery and communication between requesters and sensors, and provide additional privacy against a blockchain observer. Digital Signatures and encryption ensure integrity and confidentiality. However, the hub may censor individual participants or transactions for legal or economic reasons. Hence, fallback mechanisms are required.

OFF-CHAIN SCALING ONLY IN COMBINATION WITH ON-CHAIN SCALING A S^{2}aaS infrastructure would need to scale to millions or even billions of nodes. The maximal throughput of the current Bitcoin network is 3-7 transactions per second. Even though we could handle almost all transactions off-chain, it would take years to setup the channels, and they would need sufficiently long expiry times. Lightning channels (Poon and Dryja 2015) can be open indefinitely and could be of advantage longterm. However, they are not functional given the current Bitcoin network (see Sec. 6.8.2).

RUNNING A HUB CAN BE EXPENSIVE S²aaS and crowdsensing schemes are only valuable if a large number of data providers are available which are able to attract requesters. Hence, participation needs to be as frictionless as possible. In the presented system a smartphone user needs only to download and install and application in order to participate. Ad-hoc micropayments are enabled because the hub finances a payment channel to each requesting sensor. Hence, each additional sensor implies locking of funds inside contracts and paying Bitcoin network fees.

6.8.2 *Comparison with Bidirectional Payment Networks*

There have been more complex payment channel designs invented which allow reversing the payment direction, and hence creating bidirectional channels (Decker and Wattenhofer 2015; Poon and Dryja 2015). In addition, Lightning channels (Poon and Dryja 2015) can stay open indefinitely. Bidirectional payment channels in combination with some form of HTLCs can be used to build routable payment networks in which payments can be transported via an arbitrary number of trust-minimized intermediaries. McCorry et al. 2016 provides a comparison of the two designs and discusses the issues towards payment networks. Besides the routing problem itself, which is trivial when the topology is reduced to a hub and spoke architecture as in our case, both payment channel designs, and the HTLC designs rely on creating chains of partially signed transactions. This makes the protocols vulnerable to transaction malleability (Andrychowicz et al. 2015). If a child transactions is created collaboratively, i.e. requires signatures of both parties, before the parent transaction, i.e. the transaction of which the child spends an output, is committed to the blockchain, then a party can unilaterally commit an altered parent to the blockchain, which invalidates the child. A malicious party is thus able to lock funds indefinitely or even steal funds. Bitcoin Improvement Proposal (BIP) 141 (Lombrozo, Johnson, and Wuille 2015) solves this issue but the actual date of deployment is uncertain. Furthermore, there is disagreement in the community about deploying the enhancements via soft fork, and it is thus uncertain if the necessary majority for a deployment can be reached. The here presented payment channel and HTLC design on the other hand is not vulnerable to transaction malleability.

6.8.3 *Comparison with Ethereum*

Ethereum was still in its pre-release stage at time of implementation. At time of writing this thesis, we can review what it would mean to implement a S²aaS or crowdsensing with Ethereum. Ethereum has a much faster block time of approximately 15 s in comparison to Bitcoin's 10 min. However, 15 s is still a significant latency for a sensing scheme. Hence, an analog to payment

channels is needed. Generalizations of Bitcoin payment channel networks are currently under development[10]. In addition, Bylica et al. 2015 present a lottery-based micropayment scheme implemented as an Ethereum contract that is particularly suited for one-to-many payments if payees are expected to receive a large number of payments from different payers. In general, Ethereum has some advantages to Bitcoin in terms of micropayments. First, Ethereum's account model does not have the dust issue that cryptocurrencies based on UTXOs have. A sensor with an Ethereum account would get all micropayments in the same account instead of multiple UTXOs. Second, Ethereum is divisible down to 18 decimal places, and third, Ethereums's exchange rate is significantly lower. On the other hand, although there are features that make Ethereum light client friendly, there is currently no release version of a light client. Furthermore, the fast block rate and significantly bigger block headers need more bandwidth and storage. Finally, Ethereum suffered from various attacks and bugs which have lead to hard forks. Thus, clients would have to be updated regularly, which is always an issue for dedicated connected devices. However, less so for smartphone applications. Hence, as of now, there are no substantial advantages to use Ethereum, and the stability, diffusion and maturity of Bitcoin provides a firmer basis for such a platform.

6.8.4 *Towards a Common Architecture*

We saw that the basic elements of a S²aaS scheme are discovery, data exchange, and payments. Discovery allows a requester to find sensors that are able to provide data which are of interest. We implemented discovery in form of a central repository. Digital signatures ensure integrity of the meta data, but availability and censorship-resistance are not guaranteed. In the last paragraph of Section 5.7, we briefly discussed that discovery could also be implemented in form of a DHT or as a virtual chain on top of Bitcoin, as implemented by Blockstack (Ali et al. 2016). Data exchange can be done via the Bitcoin network, but only for very important data points requiring trusted timestamping and global availability. Peer-to-peer data exchange requires addressability and suitable communication protocols. SDN together with HTTP or Constrained Application Protocol (CoAP) (Shelby, Hartke, and Bormann 2014), BitMessage (Warren 2012) or Telehash can be used to allow peer-to-peer communication between parties in different networks and despite NAT. Mediated data exchange via central server (hub) is simpler and more efficient. Digital Signatures and encryption ensure that the server is not able to tamper with the data or learn about the content of the data. However, the issues are similar to the central repository. Availability and censorship-resistance can not be guaranteed. A middle ground can be found by using the central server to initiate peer-to-peer to communication (c.f. WebRTC (Bergkvist, Burnett, and Jennings 2012)). If a

10 https://github.com/raiden-network/raiden

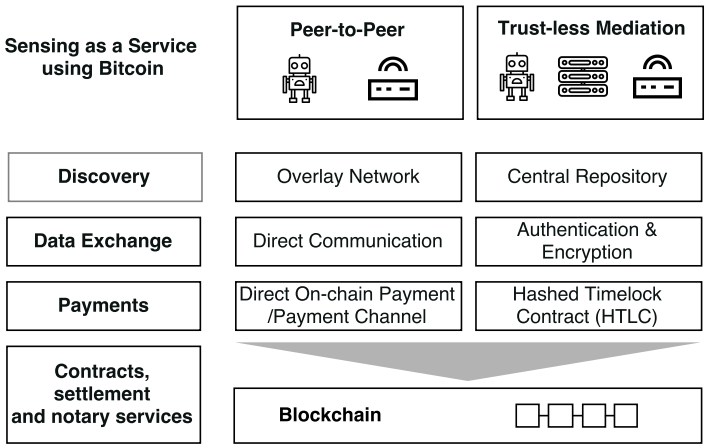


Figure 6.10.: Functions of the system and their technological enablers.

requester needs to buy data from a large number of sensors, or the sensors vary over time , then mediated payment channels are necessary. However, if direct communication channels between requesters and sensors are available, then direct payment channels could be established for repeated payments. For one time payments of higher value data, on-chain payments are a fallback solution. Figure 6.10 provides an overview of the different solutions for the main functions. In principle, an integrated architecture could support both peer-to-peer services and competitive mediation.

6.9 CONCLUSION

The main contribution of this chapter is the introduction of a low-latency micropayment scheme that fits the S²aaS scheme. We concentrated on payment channels which leverage Bitcoin's built-in programmability to implement a simple form of a smart contract. However, these contracts are bilateral and based on an on-chain multi-signature output. Thus, in m-to-n relationships, m*n contracts – and 2*m*n on-chain transactions – would be needed. We introduce a hub, able to interconnect payments in two channels via HTLCs. We discussed how discovery and data exchange can be mediated as well, and implemented a prototypical crowdsensing system based on this concept. By downloading a smartphone application, a user is able to offer the sensing capabilities of her phone to requesters all over the world in exchange for Bitcoin micropayments. However, running the mediating hub locks up capital proportional to the number of sensors. Furthermore, Bitcoin's current block size and rate limitation would inhibit a large scale deployment of the scheme. Even though most transactions are happening off-chain, and mediation decreases the number of necessary channels tremendously, channel opening and closing

does still need on-chain transactions. Progress towards bi-directional payment channel networks which can be open indefinitely (lightning channels), and on-chain scalability improvements could make S²aaS with Bitcoin viable on a large scale.

7 TOWARDS ECONOMIC DEVICES: INSIGHTS FROM AN BLOCKCHAIN-ENABLED DISPLAY

If we wish to preserve a free society, it is essential that we recognize that the desirability of a particular object is not sufficient justification for the use of coercion.

— Friedrich August Hayek

7.1 CONTEXT AND MOTIVATION

In the last two chapters the application of Bitcoin as a platform, and as medium of exchange between connected devices was discussed and explored by building prototypical systems. Bitcoin contracts were introduced to enable instant micropayments between a large number of data requesters and data providers. These *smart contracts* allow to automate rules for which a trusted third party was required traditionally. The direct exchange of digital goods and services for digital cash is only one possible economic interaction enabled by cryptocurrency and smart contract platforms. In this chapter, the term *smart contract* is explored in a broader sense. Furthermore, the related concept of *smart property* is presented. Based on these two foundations, the concept of *economic devices* is introduced. Economic devices extend the capabilities of connected devices with economic capabilities independent of the manufacturer and third party services. These capabilities can be divided into active and passive capabilities. Active capabilities entail the trade and exchange of digital goods and services, such as the buying and selling of sensor data, but also indirect contractual relationships. These relationships can be encoded as *tokens*, and thus be tradable itself. An example would be an IOU that enables the bearer to access sensor data at a later point in time. Passive capabilities are based on the notion of smart property. In this regard, the economic device is the passive object of economic interactions and not the active subject. These capabilities allow the low-trust trade of an economic object, as well as its capitalization, i.e. using the economic object as collateral for loans. This is of particular significance for developing countries where large parts of the population are not served by traditional financial institutions, and private property rights are underdeveloped and not enforceable. In addition, economic devices could provide the building blocks for future sharing economy applications, and an increasingly autonomous and interoperable IOT.

The concept of economic devices is illustrated and investigated with a prototype of a Blockchain-enabled display. The prototype implements two

active economic capabilities. First, it offers the service to show user-selected content on a pay-per-time basis, payable with cryptocurrency. Second, it issues tradable tokens comparable to shares in a company. However, token-holders receive a share of revenue in real-time instead of yearly dividends. The prototype is based on an Ethereum contract. This allows to encode the entire business logic on the blockchain and minimizes the required trust in the device itself. An alternative implementation based on Bitcoin is discussed briefly.

7.2 BACKGROUND

7.2.1 *Smart Contract*

The term *smart contract* has already been introduced briefly as a contract enforced by code (c.f. Chapter 3. The thesis provided a few examples of contracts enforced by Bitcoin script. The two most important examples were payment channels (Sec. 6.2.2) and HTLCs (Sec. 6.3). These contracts work by posting collateral and programmatically define how the collateral can be spent. In the case of a payment channel, for example, the collateral posted by the payer is frozen for a specified time and can only be redistributed in the meantime collaboratively by the payer and payee. Thereby, the blockchain acts a programmable custodian of the collateral. Traditionally, either a trusted third party would have to be employed as a custodian, or a legal contract would have to be created between the parties. Smart contracts, in this sense, are an example of regulation by code (Lessig 2009). Performance of the contract is enforced automatically. Whereas performance of a traditional contract has to be audited and requires a judicial system for dispute resolution, and an executive system for enforcement. This is a manual and expensive process. In particular, for agreements that cross jurisdictional borders, as is usual for transactions on the Internet. Hence, the long tail of transactions has to rely on trust and reputation. However, only very few transactional agreements can be encoded completely in Bitcoin contracts. One example are zero knowledge contingency payments (Sec. 5.3.4 and HTLCs. For example, a payment channel itself is only useful, if it is used to perform micropayments in exchange for goods or services. However, these exchanges can rarely be made atomic, and thus, either buyer or seller has to advance performance. But because individual payments can be made as tiny as 1 satoshi, the financial loss is vanishingly small for the payer, and the loss of reputation for the payee is arguably larger than her financial gain. Ethereum allows, in principle, to deploy arbitrary programs with a sovereign ether balance and memory on a blockchain (c.f. Sec 3.2.1 and Sec. 3.2.1). These programs are called contracts, and thus, programs on a blockchain are often generally identified with smart contracts. However, although Ethereum programs can encode formal agreements between multiple parties, i.e. contracts, they do not have to. The original notion of a smart

contract was developed by Nick Szabo. Szabo defined smart contracts as machine-readable transaction protocols which create a contract with predefined terms. He writes "Many kinds of contractual clauses (such as collateral, bonding, delineation of property rights, etc.) can be embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive (if desired, sometimes prohibitively so) for the breacher". Although certain kinds of smart contracts could already be implemented using trusted hardware, blockchains and cryptocurrencies are important building blocks to advance what is possible. Szabo's generic example of a smart contract is the vending machine. "A canonical real-life example, which we might consider to be the primitive ancestor of smart contracts, is the humble vending machine. Within a limited amount of potential loss (the amount in the till should be less than the cost of breaching the mechanism), the machine takes in coins, and via a simple mechanism, which makes a freshman computer science problem in design with finite automata, dispense change and product according to the displayed price. The vending machine is a contract with bearer: anybody with coins can participate in an exchange with the vendor. The lockbox and other security mechanisms protect the stored coins and contents from attackers, sufficiently to allow profitable deployment of vending machines in a wide variety of areas." (Szabo 1997). Just as the vending machine is able to autonomously accept cash, machines are now in general able to accept electronic cash using cryptocurrencies. Furthermore, Ethereum provides an open trusted computing platform which can either provide a programmable trusted third party or provide a trusted partial identity of a machine. Another aspect of smart contracts are smart legal contracts (Grigg 2004; Clack, Bakshi, and Braine 2016). These focus on the expressibility of legal contracts in code. Although, the goal is also automation and programmatic execution of contract terms, they embrace that most traditional contracts cannot be enforced entirely by code. Hence, they focus on documenting and providing tamper-proof evidence of contract breaches for traditional dispute mediation within the judicial system. Cryptographic signatures and tamper-evident append-only logs, as provided by blockchains, are a valuable tool in this regard.

7.2.2 *Smart Property*

The term smart property was also introduced by Nick Szabo. He defined smart property as "Software or physical devices with the desired characteristics of ownership embedded into them; for example devices that can be rendered of far less value to agents who lack possession of a [cryptographic] key" (Szabo 1997). A simple example of smart property is a car. Modern cars already employ cryptographic protocols and use electronic keys to authenticate the owner. Electronic immobilizers enforce that only the bearer of the key is able to start the engine. Smart property can be seen as a particular type of a smart contract. However, in the case of the car, integration in a legal system is

still desirable, since key theft should not imply the transfer of ownership to the thief. On the other hand, sophisticated methods for key revocation and recovery are being developed which might enable a pure technological implementation of the concept of ownership. Smart property benefits tremendously from cryptocurrencies. One simple example for illustration is the concept of an atomic sale (see Appendix B for a more detailed presentation of concepts related to smart property and their implementation based on Bitcoin and Ethereum). Assume a car is represented on the Bitcoin blockchain as a simple P2PKH UTXO. In order to open or start the car, the car demands a challenge to be signed with the private key corresponding to the UTXO. Spending this UTXO, and assigning it to another key would allow the bearer of this key to prove the transfer of ownership to the car. Given this setup, it is now possible to construct a single transaction that transfers ownership and bitcoins. Thus, the sale is atomic. Either the trade is successful and the buyer becomes the new owner and the seller receives the coins, or the trade is unsuccessful and no exchange happens at all. Thus, no party has to advance, and required trust is minimized. In the case of a car sale, this might not be so important because buyer and seller would typically meet in person due to the high value that is on stake. However, the model could be attractive for deployed sensors.

7.3 ECONOMIC DEVICES

Today, things are passive participants of an economy. Things can be sold and bought. Things can be rented, and in some cases things can be capitalized, e.g. they can be used as collateral for loans. Connected, or smart, things have embedded communication and information technology. They can be digitally upgraded with additional functionality and services, and they can act as a physical point of sales for goods and services (Fleisch, Weinberger, and Wortmann 2015). In the latter sense, they instantiate a smart contract similar to the vending machine. An example is the Amazon Echo, a voice-controlled speaker. It is connected to a personal Amazon account which has access to credit card details, and allows to order items from Amazon with a mere voice command. There is a high degree of automation, but this automation is based on trust. The user gives Amazon the ability to debit an arbitrary amount of money from her credit card, because the user expects that Amazon will only debit the agreed amount. Furthermore, the user expects Amazon to deliver the respective product. If Amazon would debit more money, the users trust that the bank or credit card company is able to provide a refund, and as a last resort the user can go to court. If the user likes to sell an Amazon Echo on the secondary market, the user has to be permissioned by Amazon. Most of the functionality that the Echo provides is based on cloud services connected to a personal identity on Amazon. Possession of the physical device is only valuable if it is connected to the Amazon identity, and this identity is owned and controlled by Amazon.

Cryptocurrencies and blockchains allow to minimize necessary trust and enable machine-based transactions that were not possible before. Machine-based transactions can denote transactions between humans or corporations that are mediated by machines, but also autonomous transactions between machines. The discussed S²aaS model is an example. Cryptocurrencies allow a machine to reason about the finality of a payment, and micropayments allow the payer to advance payments with little financial risk. Smart contracts allow machines to enforce contractual relationships, and smart property, in particular, allows the enforcement of property rights on connected devices without permissions of a company or the requirement of a judicial system. Hence, cryptocurrencies enable software agents to do autonomous economic interactions. Ideally, these software agents are implemented on low-trust computing platforms. Ethereum provides such a low-trust platform. However, software agents implemented purely on Ethereum are limited to actions within the platform, and their complexity and performance is limited, due to the replicated architecture. Tamper-resistant hardware and trusted computing environments allow the implementation of trustworthy software agents in connected devices. The combination of trustworthy software in connected devices and economic interactions based on cryptocurrencies and smart contracts provides the basis for economic devices.

7.3.1 *Capabilities of Economic Devices*

Capabilities of economic devices can be distinguished between passive and active capabilities. An overview of these capabilities is illustrated in Fig. 7.1 and will be explained in the following.

Passive Capabilities

Passive capabilities are related to the smart property concept. Economic devices can be traded atomically. This can be facilitated either by representing the ownership on a public blockchain, or by relying on trustworthy computing on the device itself. The former has been discussed in Sec. 7.2.2. The latter relies again on the fact that cryptocurrencies empower machines to reason about the finality of payments. Thus, the device can change its ownership status based on a payment.

If the ownership of a device can be reassigned based on cryptocurrency transactions, then the device natively can be used as collateral for loans. Hearn 2011b describes a simple protocol to implement the concept with Bitcoin. If the debtor does not repay the creditor as defined in a smart contract, the device becomes unusable for the debtor. A real example of such a model is given by M-Kopa[1]. M-Kopa offers solar home systems to the rural population of Western Africa. Solar home systems contain a small solar panel, a battery,

1 http://www.m-kopa.com/

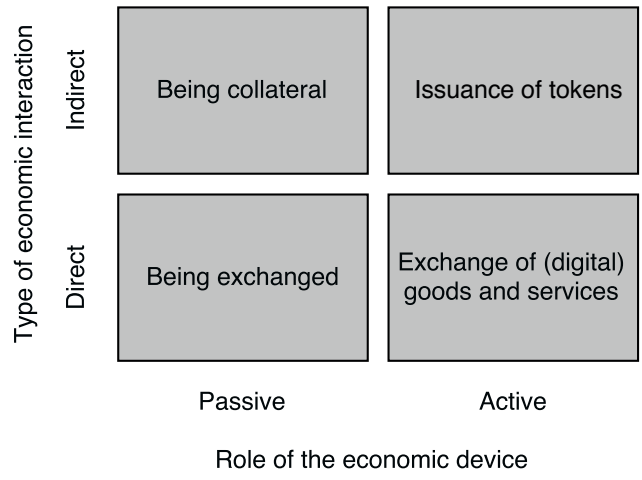


Figure 7.1.: Capabilities of economic devices.

and a few loads, such as LED lights and an USB charger. M-Kopa offers these solar home system using pay-as-you-go models in order to attract low-income households. In a pay-as-you-go model, only a small down-payment is made in the beginning, and then the customers pays for usage. The typical model employed by M-Kopa is the lease-to-own model, which transfers ownership to the customer after sufficient payments. Typical pay-as-you-go models are based on a centralized service that can be denied without payment. However, a solar home system is self-sufficient and does only need exposure to the sun. Thus, it is cumbersome to prevent usage without payment. Therefore, M-Kopa embeds connectivity in the device which allows to lock the device remotely - rendering it useless for the customer. After the customer has paid for the device, she is able to return to the pay-as-you-go model in order to get a loan to buy selected products. Hence, M-Kopa acts a gatekeeper to provide financing. In contrast, economic devices based on cryptocurrencies allow an open market for loans based on programmable collateral.

Active Capabilities

Active capabilities entail the direct exchange of (digital) goods and services for cryptocurrencies. These enable the basic interactions of what IBM calls the *Economy of Things* (Pureswaran and Lougee 2015). This capability depends heavily on the possibility to perform micropayments. In particular, this allows to pay for each API call directly and enables autonomy and interoperability between web services.

In addition, economic devices can enter more complex contractual relationships. A standard way to implement such relationships are tokens. Tokens are meta coins 4.3.1 created on a blockchain. Tokens can be used to grant the bearer specified usage rights. For example, a sensor might offer a token that grants access to data for a particular time period. Moreover, the sensor could issue a token that grants the bearer a share of its revenue. Tokens can be tradable, or ownership transfer can be permissioned. If the tokens are tradable, then the trade against other tokens or cryptocurrency can be made atomic, i.e. trust-minimized.

Economic devices can either be deterministically preprogrammed to determine their behavior in economic interactions, or they can adapt their behavior based on artificial intelligence. For example, a sensor might have preprogrammed economic parameters, such as the price for measurement data, but the price could also be defined dynamically based on supply and demand. Furthermore, tokens can be used to distribute voting rights. Then a voting mechanism can be used to change parameters defining the behavior of the device.

7.4 APPLICATIONS AND SIGNIFICANCE

Arguably, the greatest potential for economic devices is in developing countries. Developing countries often lack a formal property system, and large parts of the population lack access to formal financial services. Nearly 2.2 bn adults in Africa, Asia, Latin America, and the Middle East are *unbanked*. In subsaharan Africa this culminates to 80% of the adult population (Chaia, Goland, and Schiff 2010). On the other hand, Internet access and smartphone ownership is rising (Poushter 2016) fueled by the diminishing costs of computation and communication technology. This provides access to cryptocurrencies and smart contract platforms, and thus to digital financial and legal systems purely based on open source software. Productive assets, like solar home systems, are ideal applicants for economic devices, combining active and passive capabilities. The system could offer electricity in exchange for cryptocurrency payments, and issue tokens that collect a share of the revenue. These tokens could be sold to global investors, and thus reduce the upfront cost of the system for local users. The same model can be applied to other productive assets like small wind turbines or communication infrastructure such as small cells. Furthermore, economic devices are ideal building blocks for the sharing economy. The German start-up company Slock.it[2] presented the prototype of an economic device called *slock*. A slock is a cryptocurrency and smart contract enabled lock. For example a bike owner could use a slock to lock her bike and allow others to rent the bike. A renter would have to pay a deposit to the smart contract deployed on the Ethereum network. The deposit minus

2 http://wwww.slock.it

the renting fees is then returned to the renter if evidence of proper return is provided to the smart contract. A similar scheme is presented in (Bogner, Chanson, and Meeuw 2016).

7.5 AN ETHEREUM-ENABLED PUBLIC DISPLAY

In order to gain insights into the current possibilities and limits of economic devices, we developed a prototype with similar characteristics as the solar home system described above. However, we decided to develop a prototype of a public display which offers the service to show user-selected content in exchange for cryptocurrency payments. The display is better suited for demonstrations and provides more means for user interaction. The system was demonstrated at the MIT Media Lab Demo days in Fall 2016. The display is able to issue tokens that entitle the bearer to a share of the generated revenue. In analogy to a corporation, the tokens are called *shares*, and the bearers *shareholders* or sometimes investors. In contrast to the solar home system, which is typically used by an individual or a household, the public display is inherently a multi-user system comparable to a solar micro utility. In addition, there needs to be a stakeholder to install and maintain the system. We call this stakeholder *entrepreneur* or *operator*. The *entrepreneur* is also a shareholder in order to have *skin in the game*. The entrepreneur does not need to provide the entire financing. In contrast, the entrepreneur and the manufacturer can initiate a public crowdsale (c.f. 4.3.3) for the shares in order to close the financing gap. Furthermore, share issuance is adapted such that new shares are issued as a function of revenue, and are awarded to the entrepreneur. This provides further incentives for the entrepreneur to maximize the utility of the system, and over time shares accumulate with the entrepreneur.

Shareholders, as well as prospective shareholders, are able to observe the performance of the economic device transparently. Thus, a fair market price of the shares can emerge.

The prototype is built on the Ethereum blockchain. Financial interactions are implemented as Ethereum smart contracts. The public display is augmented by an embedded Linux computer, and reacts to changes in the smart contract code. The focus of the prototype is on the software implementation. The hardware itself is not tamper-resistant. In addition, a possible implementation based on Bitcoin is briefly described.

7.6 SYSTEM ARCHITECTURE

In the following we present the system, the stakeholders and their primary roles. Figure 7.2 provides a simplified illustration of the stakeholders and their roles.

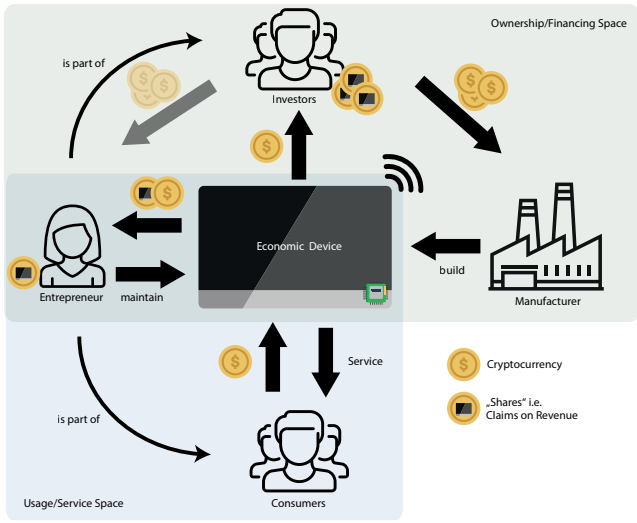


Figure 7.2.: Stakeholders and their primary roles in the model.

7.6.1 *Stakeholders*

MANUFACTURER The manufacturer produces the economic object, and is the beneficiary of the initial sale of shares.

ENTREPRENEUR / OPERATOR The entrepreneur identifies a potentially profitable location for a public display and takes care of proper operation such that maximal revenue is generated. The entrepreneur has to have a double role as investor to have financial skin in the game, and is incentivized by being awarded additional shares depending on generated revenue.

INVESTORS Investors seek a profitable investment. Hence, they provide capital to close the entrepreneur's financing gap in return for a share of revenue.

CUSTOMERS Customers are individuals or software agents that are willing to pay for the service the economic device provides.

7.6.2 *Economic Device*

The economic device is represented by a large screen augmented with an embedded Linux computer and a representation on the Ethereum blockchain. The economic device provides the service to screen user-defined content on a pay-per-time basis. Customers pay for the service using cryptocurrency

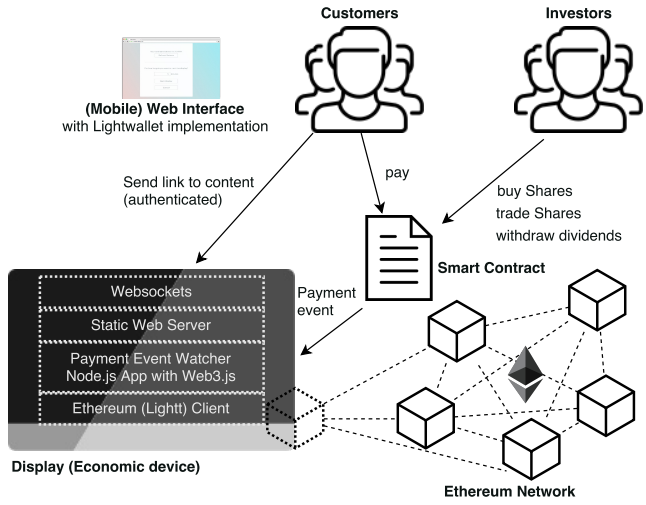


Figure 7.3.: Illustration of the implemented system after deployment by the entrepreneur and the manufacturer.

(ether). The revenue is distributed immediately and securely to shareholders. Shares are initially sold in a crowdsale and can be freely traded.

7.7 IMPLEMENTATION

The economic device has a physical and a digital instantiation. Although the physical instantiation is important to provide security against physical tamper, the focus is on the digital instantiation. Ethereum allows to implement application logic as stateful programs which are stored and executed on the Ethereum network (c.f. Section 3.2.1). These programs are called (smart) contracts. By linking a connected device with a smart contract, we are able to implement an economic device. In order to interface with the Ethereum network, a Geth client (Ethereum Wiki 2016b) is running on the device. The Geth client is an implementation of an Ethereum node in Go and provides an Remote Procedure Call (RPC) interface that can be accessed conveniently from Node.js via the web3.js library (Ethereum Wiki 2016c). Figure 7.3 provides an illustration of the system as implemented. The aspects of the initial crowdsale are not shown.

7.7.1 *Economic Device Contract*

The contract is implemented using Solidity, the JavaScript-like higher-level smart contract language that can be compiled to EVM byte code. In the

following, we describe the functions of the economic device contract providing the economic interactions.

SHARES AS TOKENS Shares are represented as tokens. We base the smart contract on the standard token contract[3]. The token contract is essentially a simple accounting system that maps account numbers to values, i.e. the number of *tokens* an account owns. A *transfer function* then allows an account owner to debit her balance and credit another. This provides the basis for investors to trade shares.

INITIAL FINANCING BASED ON TOKEN SALE Initial share allocation is achieved with an adapted crowdsale contract[4]. The amount of initial shares is fixed and corresponds to the price the manufacturer demands (see Table 7.1. A certain amount of shares need to be bought by the entrepreneur as a down payment. Revenue of the crowdsale is credited to the manufacturer after a successful crowdsale. The crowdsale has a predefined maximal duration. If not enough shares are sold during this period, the contract automatically refunds the investors, and a new contract with different parameters can be deployed.

Instead of having a fixed price per share or a defined goal, an auction-based crowdsale could be implemented.

PAYMENTS, DIVIDENDS AND DILUTION The contract exposes a public *pay function* (see Procedure 7.1). The function checks if the device is currently rented. If so, the customer is refunded. If not, internal functions to pay dividends and to increase the entrepreneur's share are called. Finally, a *payment event* is created. The payment event carries the data tuple (*payer*, *paidUntil*), where payer denotes the Ethereum account address of the customer invoking the pay function, and *paidUntil* denotes the timestamp until the customer has paid for renting the display.

Since share issuance is handled within the contract handling payments, the overhead is low. There is no need for an additional transaction, but only an update of the token balance. Thus, there is no need to aggregate payments until a certain threshold, before the entrepreneur gets credited with additional shares.

Noteworthy, there are two possibilities to pay dividends to shareholders. The obvious approach would be to let the contract actively send dividends to the shareholders' Ethereum accounts. However, sending value from a contract is expensive in comparison to updating an internal balance, and the cost would have to be provided by the customer invoking the pay function. To avoid this, the contract keeps track of the dividend balances and we add a *withdraw*

3 https://github.com/ConsenSys/Tokens/blob/master/Token_Contracts/contracts/StandardToken.sol

4 https://www.ethereum.org/crowdsale

Procedure 7.1 Pay for service

```
1: if now ≥ paidUntil then
2:     creditDividends(value)
3:     dilute(value)
4:     paidUntil = now + value/pricePerTimeUnit
5:     Event Payment(customer,paidUntil)
6: else
7:     return money to customer

8: function creditDividends(value)
9:     for sh in shareholders do
10:        balanceEther[sh] += balanceShares[sh]/totalShares*value

11: function dilute(value)
12:    newShares = value * supplyIncreaseRate
13:    balanceShares[indexOfEntrepreneur] += newShares
14:    totalShares += newShares
```

Parameter	Description
sharePrice	Price per share in Wei.
initialShares	Total supply of shares in sale. initialShares*sharePrice is paid to the manufacturer.
initialSharesEntrepreneur	Number of shares the entrepreneur has to buy (down payment).
issuanceRate	Shares issued to the entrepreneur per revenue
pricePerTimeUnit	Price of service

Table 7.1.: Financial parameters of the contract.

function. The withdraw function allows shareholders to collect their dividends whenever they wish. Thereby, costs for withdrawing have to be borne by the withdrawing shareholder, and shareholders may decide for themselves when collecting dividends is appropriate.

FINANCIAL PARAMETERS The contract defines several financial parameters. Table 7.1 provides a listing with a short description of each parameter. In the current implementation all these parameters are fixed in the contract. However, especially the price of the service might need to be adjusted based on competition and demand. These adjustments would have to happen under defined rules since otherwise the entrepreneur could set a low price, rent the display herself, and offer the service. Thus, investors would be defrauded.

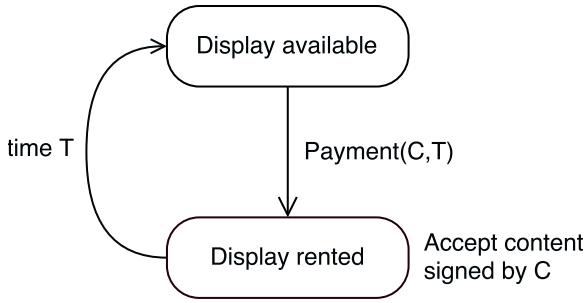


Figure 7.4.: State machine of the device. There are two main states: available and rented. If available, the pay function can be invoked and the payment event notifies the device that customer C has paid to cast content until time T.

CONTRACT-DEVICE COMMUNICATION The application running locally on the device has essentially only one job: watching for payment events executed by the respective contract instantiation. Solidity events provide an interface to the logging facilities of the EVM, and can be used to trigger client application logic with data payload. The payment event triggers a state transition of the device (c.f. Figure 7.4).

Logs are committed to the block header and are thus accessible for light clients (c.f. Sections 3.2.1 and 3.2.1). Thus, a future implementation can be based on an Ethereum light client with much lower computation, storage and bandwidth requirements - an important point for possible applications in developing countries.

7.7.2 *Interacting with the Device*

In principle, all interactions could be facilitated via the Ethereum network. In particular with the availability of the P2P messaging protocol Whisper (Ethereum Wiki 2016e), and the Mist browser[5]. However, these tools are at a very early stage. Instead, the economic device provides a web server to serve graphical user interfaces in form of (mobile) static single-page web applications.

The client does not necessarily need an Ethereum client, which is important as long as fully functioning light clients are not available. Here, the Lightwallet library[6] is used to keep the wallet and private keys locally on the client. Communication with the Ethereum network is achieved via an Ethereum client with a public RPC interface.

In the following, we briefly present the steps a customer needs to take in order to rent the display:

5 https://github.com/ethereum/mist
6 https://github.com/ConsenSys/eth-lightwallet

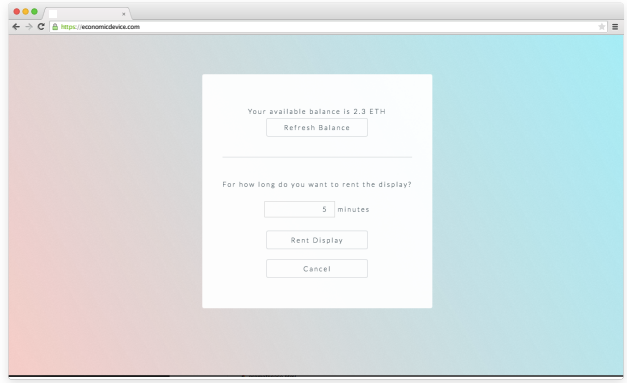


Figure 7.5.: An example view of the customer interface. In this view the customer can make the payment.

1. Open *personal* website of the display.
2. Open an existing wallet by providing a seed and a password or create a new wallet. Make sure that the wallet has sufficient funds.
3. Make a payment.
4. Send links to content directly to the device[7]. The messages are authenticated, such that the device can verify if the sender is the same account as the one that paid to the contract. The content can be changed as long as the period paid for is not expired.

An illustration of a view of the customer interface is shown in Figure 7.5.

7.7.3 *Implementation with Bitcoin*

The Bitcoin ecosystem is much more mature. The exchange rate to fiat currencies is more stable, there are many more ways to procure bitcoins, and there are already multiple implementations of light clients. Hence, an implementation based on Bitcoin would be desirable. In what follows we briefly discuss the main points.

Shares could be implemented based on a *Colored Coins* specification (see Section 3.1.4). However, there would be no way to issue shares based on revenue automatically. This role would have to be taken over by an application running on the device with access to the necessary cryptographic key. Since the entrepreneur has physical access to the device, it is important that the device is tamper-proof. Otherwise the entrepreneur could gain access to the keys and start issuing shares.

7 This done via a websocket connection (Fette and Melnikov 2011)

The crowdsale could be implemented as an assurance contract (Hearn 2011a) (see also the Lighthouse app, a P2P crowdfunding app using Bitcoin assurance contracts[8]).

Distribution of dividends could be implemented in multiple ways: (1) In order to pay, the customer (i.e the web application the customer is using to use the service) parses the blockchain for current shareholders and pays the shareholders directly. The customer is then able to prove the correct payment to the device. However, if individual payments are small and there are many shareholders, the transaction would have to entail a proportionally large fee, and investors end up with many small UTXOs. (2) The device could act as a payment hub (c.f. Section 6) between customers and investors. However, this implies that the device would need the necessary capital to pre-fund the payment channels to the investors. (3) Customers pay to an address controlled by the device itself. The device aggregates payments over a specified time period and distributes the dividends according to this schedule. Since investors have to trust the device to securely store the share issuance key anyways, this would be the most practical implementation.

In the Ethereum implementation, all financial parameters were defined and immutably stored in the smart contract. Since a trusted device is already assumed, the financial parameters and other contract terms can also be stored on the device. To guarantee immutability, the secure hash of the contract terms can be published to the Bitcoin blockchain.

In conclusion, an implementation based on Bitcoin is in principle possible. In contrast to Ethereum, where all fiduciary code and business logic are handled by smart contracts deployed on the network, in Bitcoin an application running on the device has to execute the business logic. Therefore, tamper-proof hardware with secure storage of cryptographic keys is indispensable.

7.8 KEY FINDINGS

7.8.1 *Economic devices allow trust-minimization but are not trust-less*

Cryptocurrencies enable machines to autonomously participate in economic interactions with humans and other machines. These economic interactions can be distinguished between active and passive interactions. The basis for this is that machines can reason about the finality of a payment or the state of a contract. This allows for many low-trust interactions. In cases where low-trust interactions are not possible, efficient micropayment schemes enable payments of unprecedented granularity. Thus, necessary trust in the service providing counterparty is minimized because individual payments can be advanced without significant financial risk. In the example of the public display, a customer has to pay before she is able to provide content that is

8 https://github.com/vinumeris/lighthouse

shown on the display. In principle, there is no assurance that the display will show the content. However, the customer is able to probe the device using very small amounts of money. The public payment history to the device can also act as a reputation system. Initial investors, who take part in the crowdsale, need to trust the manufacturer to produce a functional device and need to trust the entrepreneur to operate and maintain the device such that it generates as much revenue as possible. Investors can follow the strategy to invest only tiny amounts in an individual economic device, and to diversify their investment in many economic devices from various manufacturers. We can also conceive the establishment of securitization contracts. These contracts would offer Investors tokens with varying risk profiles and invest their capital in a pool of dividend-paying tokens.

7.8.2 *Distributing Logic between the Blockchain and the Device*

Ethereum allows to deploy arbitrary programs on the blockchain itself. Thus, the economic device can be implemented as a thin economic client. In the prototype, the entire business logic is implemented as an Ethereum contract. Execution of contract terms is initiated by transactions from shareholders or customers. In this case, the device does not even need an external Ethereum account, i.e. a local private key with access to an ether balance. However, it is worth stressing that Ethereum functions in Ethereum contracts can only be triggered with transactions originating from other contracts or external accounts. These transactions have to provide the gas required to execute the function. Thus, if the economic device needs to actively initiate an economic interaction (e.g. paying for a service) an external account is required, and the economic device would need to hold the private key. The model of thin economic devices allows to keep a lot of functionality on the trustworthy Ethereum platform. However, this comes with a cost and latency for execution. In addition, most economic interactions require some trust in the device anyways. Thus, developers of economic devices have to think carefully about how to split logic between the device and the Ethereum contract. The brief discussion of an implementation based on Bitcoin and Colored Coins exemplified the *thick economic client model*. Here, software running on the physical device is responsible for executing much of the business logic. Thus, tamper-resistant hardware, and vetted open source software is indispensable.

7.8.3 *Immutability and Governance*

One of the characteristics of blockchains is immutability[9]. Thus, contracts have to be defined completely at time of deployment. There are obvious issues concerning technical bugs in the contract code, but also more subtle issues

9 At least practically and in the absence of hard forks.

concerning game theoretical aspects. A simple example is price setting. In the prototype, the price of the service, i.e. renting the display for a specific time, is predetermined. Assume an entrepreneur set up a display at a crowded location and is generating a lot of revenue. Another entrepreneur is able to observe the blockchain for such business opportunities, and may decide to set up another display with a lower price next to the original one. Now, customers will prefer the new, cheaper service, and the original display would need to adjust the price in order to stay competitive. It is possible to implement a function that is able to set a new price, but who should have the rights to call it. If we would allow the entrepreneur to set the price, she might be tempted to lower the price dramatically and to try to trick customers to pay a higher price on a side channel. Thus, the investors would not get a fair share of the profit. Another possibility would be to implement a voting system. However, coordination might be a problem, and the entrepreneur would get to much importance over time. A third possibility would be to implement an intelligent system to adjust the price. However, this increases the complexity, increasing the chance of bugs and ways to exploit even more. Defining and writing smart contracts is challenging and errors might be not fixable. Thus, an open collaboration between open source community, companies and academia is necessary to develop guidelines, tools, and templates that can be ever more defined and tested. For economic devices it is of particular importance that errors can not be exploited remotely, such that the same error can be exploited in many devices simultaneously.

7.8.4 *Cryptocurrencies and Developing Countries*

Economic devices might be an approach to mitigate some problems concerning the underdeveloped financial and legal systems in developing countries. In particular, productive assets providing foundational infrastructure for connected devices, such as solar home systems and micro utilities, as well as small cells or other communication access points, provide interesting applications. Therefore, cryptocurrency ecosystems have to evolve such that people of developing countries get access to cryptocurrencies. Although a smartphone is enough to receive and send cryptocurrencies around the globe, the problem is the exchange with traditional fiat currencies. Typical cryptocurrency exchanges depend on the traditional financial infrastructure and require customers to deposit money using credit cards or wire transfers. However via remittance payments and innovative services like Abra[10], cryptocurrency can reach the rural population. Additionally, online services are emerging that allow to earn bitcoins directly. An example is 21 tasks[11], a platform for micro tasks which are compensated with bitcoin payments. Another example is OpenBazaar (see A.3).

10 https://www.goabra.com/
11 https://21.co/tasks/

7.9 CONCLUSION

In this chapter, the concept of economic devices was introduced. The concept combines passive economic capabilities, based on the notion of smart property, with active economic capabilities, such as trading of digital goods and services, and indirect contractual agreements encoded as tokens. These capabilities enable economic interactions with connected devices as the object, as well as the subject, with minimal trust requirements in the respective counterparties. Thus, enabling more automated economic interactions and new types of interactions that were not possible before, because trust requirements were inhibiting. Economic devices may contribute to a sharing economy that is less dependent on centralized intermediaries and can help to bridge the financing gap for connected productive assets in developing countries. The latter application was illustrated with an Ethereum-enabled public display issuing tokens that function as claims of revenue to be bought by global investors. Fiduciary code is handled by a smart contract deployed on the Ethereum network. Thus, shares of revenue are distributed automatically and transparently, decreasing the necessary trust that investors have to bring up. Even though fiduciary code is executed on a trusted computing platform, limited trust in the manufacturer and the entrepreneur is still necessary. Before economic devices are useful for developing countries, cryptocurrencies itself need to be more prevalent in these countries.

8 CONCLUSION

> *That's been one of my mantras - focus and simplicity. Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.*
>
> — Steve Jobs

8.1 SUMMARY AND KEY FINDINGS

The Internet of Things has made a leap over the last decades. Smartphones have become ubiquitous in developed countries and developing countries are catching up. The scale of the smartphone supply chain made SoCs, combining computing, memory and various communication technologies, cheap and pervasive. Cloud computing, in combination with the *as-a-Service* business model, allows to scale backend infrastructure for connected devices without large capital requirements and expertise. These developments allowed manufacturers of traditional *things*, as well as start-up companies to bring an ever growing number of connected devices to market. Connected devices in combination with artificial intelligence and data analytics have great potential to make life more comfortable and efficient, to save energy, optimize industrial manufacturing, and enable new business models. However, much value can only be unlocked, if connected devices and web services from various vendors are interoperable (Manyika et al. 2015). Furthermore, connected devices pose a thread to individual as well as national security and privacy. The prevailing architectural paradigm of the IOT is a tight link between connected devices and corresponding cloud services. Data is streamed from the device to a data center without control of the individual, and corporations have to maintain large data centers in order to keep connected devices functional over their lifetime. Enabling devices to be more autonomous, i.e. to be less dependent on backend infrastructure, is a promising approach towards a more secure and sustainable IOT. In analogy to how money, contracts, and the market enabled decentralization and specialization in the human economy, it can be expected that the same can be true for the machine economy.

Cryptocurrencies are based on cryptographic primitives, peer-to-peer networks, and a consensus mechanism that employs economic (or game theoretic) considerations. This enables decentralized and global digital money, in combination with a payment network independent of any trusted third party. The technological underpinnings are vaguely conflated to the term *blockchain*

technology. Blockchain technology enables digital bearer instruments which can be controlled by cryptographic keys. Thus, machines are able to facilitate autonomous economic interactions. The Bitcoin network was initiated in the beginning of 2009. At time of writing, the market capitalization of Bitcoin is on the order of $10 billion. Due to its nature as open source software, Bitcoin gave rise to a flourishing ecosystem of alternative coins, start-up companies, and industry endeavors. Although there are hundreds of alternative decentralized currencies, Bitcoin's supremacy is unsolicited. Bitcoin transactions provide a built-in scripting language which allows to implement self-enforceable *smart contracts*. One of the most basic, yet powerful, are multi-signature transactions enabling the implementation of low-trust escrow contracts, and provide the basis for payment channels.

In order to investigate the impact of cryptocurrencies on the IOT, one the most foundational processes, the exchange of sensor data, and its business model manifestation Sensing-as-a-Service was selected. The unique, pseudonymous Bitcoin addresses can be used to identify and address a sensor, enabling a global and permissionless sensing platform with a built-in public-key infrastructure for data authentication and encryption. In a first prototypical system, Bitcoin was used as a medium of value **and** data exchange. Data payload was injected directly in Bitcoin transactions. This limits the maximal payload per transaction to 80 bytes. This is enough space for a typical data point, but already too little to use Bitcoin's native encryption scheme ECIES for data encryption. Thus, transmitted sensor data would always be public. A severe problem for the S^2aaS scheme on one hand. However, on the other hand, a benefit for certain applications since the data gets authenticated, timestamped, and globally replicated. The broadcast-based, peer-to-peer transport protocol and the proof-of-work-based consensus protocol lead to significant latency in the process. Depending on the sensor's trust in the payer, the process may take from seconds (unconfirmed transaction with double spending risk) to a period on the order of tens of minutes. Bitcoin transaction fees are, in principle, market-based. Increasing transaction volume in combination with a limited transaction throughput has lead to an increase in transaction fees. Due to the limitation in the block size, Bitcoin transaction fees are proportional to their size in bytes instead of their value. Thus, micropayments and data transactions are affected in particular, leading to transaction fees on the order of $ 0.1 USD for the data exchange process.

The initial prototype illustrated the concept of an autonomous sensor earning money by selling (authenticated) measurement data on an open market, but pointed out the limitations concerning confidentiality, latency, and micropayments. These limitations can be mitigated by introducing an additional communication layer. Bitcoin's cryptographic primitives and the programmability of Bitcoin transactions allow to move most communication *off-chain* without changes in the trust model, i.e without sacrificing security. Bitcoin payment channels allow for instant micropayments between two parties based

on a timelocked multi-signature escrow. This means a payer locks funds (capacity of the channel) with a fixed payee. Updates of the balances, with a granularity as low as 1 satoshi (on the order of $ 0.00001 USD), can then be made securely and instantly via a direct communication channel. S²aaS, and in particular crowdsensing applications, require the collection of data points from a large number of individual sensors. Thus, direct payment channels between a data requester and each sensor are inhibiting in many cases. We introduce a hub, which cryptographically interconnects payment channels in order to mediate between data requesters and sensors. Thereby, a protocol based on HTLCs is developed that allows for low-trust atomic payments over two interconnected channels. In contrast to the more advanced lightning network protocol (Poon and Dryja 2015), the presented protocol is not impeded by transaction malleability, and can thus be used securely on the current Bitcoin network. However, payment channels are unidirectional and have a fixed maximal lifetime. A variant of the protocol has been implemented as part of a mobile crowdsensing system, that enables users of a smartphone application to sell various sensor data to interested parties without having to sign-up, or disclose any additional personal or financial information. Still, there remain two main issues. First, in practice hubs may introduce *Know Your Sensor* requirements, since establishing (and closing) of channels with sensors consumes transaction fees, and open channels lock capital. A viable business models for hubs requires to optimize payment channel lifetimes, capacities, and sensor selection. Second, block size scarcity and the low block generation rate limits the number of possible channel openings/closings. Thus, setting up channels for a million sensors on the current Bitcoin network would take at least a few months.

Scaling issues aside, cryptocurrencies enable connected devices to perform autonomous economic interactions. Direct trading of digital goods and services, such as data, computation or storage, are but one aspect. Drawing from the concepts of smart contracts and smart property, the general concept of economic devices is introduced. Economic devices provide passive and active economic capabilities. These capabilities enable economic interactions with connected devices as the object as well as the subject with minimal trust requirements in the respective counterparties. Passive capabilities involve atomic trading and renting of the device, as well as *capitalizing* it in form of collateral for loans, by supporting the enforcement of property rights by code and direct financial incentives or penalties, instead of the legal system and human enforcement. The concept is illustrated with a prototype of a public display that is instantiating active economic capabilities. The display provides the service to screen customer-selected content on a pay-per-time basis in exchange for cryptocurrency payments. Furthermore, the display is able to issue tradable shares, which provide owners access to a real-time share of revenue. The prototype is implemented based on an Ethereum smart contract. In contrast to Bitcoin, Ethereum allows to deploy autonomous programs with

access to cryptocurrency on the network itself. Thus, a trusted representation of the economic device can be instantiated, whereas an implementation based on Bitcoin would involve to executing business logic on the device hardware itself. Thus, tamper-resistant hardware and trusted execution environments would be indispensable. It is argued that economic devices could be beneficial to finance productive assets in developing countries where financial services and legal systems are underdeveloped.

8.2 IMPLICATIONS FOR RESEARCH AND PRACTICE

Incentives for Sensing-as-a-Service and Crowdsensing

The important role of smartphones and other connected devices for S²aaS and crowdsensing has long been identified (Ganti, Ye, and Lei 2011). However, large scale deployments are always application-specific. Participants trade data, or provide sensing tasks, in exchange for particular services. An example is Waze, where participants contribute traffic information in exchange for a routing service. This model has a number of disadvantages. First, it can not be applied in general, since it has the same issue as the barter economy – the need for a double coincidence of demand. Second, data is only available to a single service provider. The service provider may aggregate, refine, and resell the data on a secondary market, but this is an opaque process form the perspective of the data originator. This thesis discusses the application of cryptocurrencies in general, and Bitcoin in particular, to provide a bottom up approach for the direct exchange of data and digital cash, such that either humans or machine can trade without intermediaries or with intermediaries governed by smart contracts. This provides the basis for a global market for real-time data. An open, global market for real-time data provided by smartphones and a plethora of other connected devices would be a promising resource for academia, industry and individual entrepreneurs alike.

Low-Latency Bitcoin Micropayments

Due to latency, transaction fees, and the UTXO model are direct Bitcoin transactions not a viable option for trades of real-time data between untrusted parties. The same holds for other transactions concerning digital goods and services between machines. Payment channels have been developed to enable recurring low-latency micropayments between a single payer and a single payee (Spilman 2013; Hearn 2013). Payment channel clients are already implemented in various libraries. However, S²aaS and crowdsensing applications require the procurement of data from a large number of individual sensors. With simple payment channels, each requester-sensor pair would need an individual payment channel, where each channel locks Bitcoin collateral and

requires costly on-chain transactions for funding and settlement. Lightning channels allow bidirectional payments, and an infinite channel lifetime (Poon and Dryja 2015). However, they require a fix for Bitcoin's malleability issue (Decker and Wattenhofer 2014). The fix is already developed, but will only be adopted by the network if 95% of the total hash rate in the Bitcoin network signals support. Currently, support stagnates at about 25%[1]. In this thesis, payment channels are combined with the concept of HTLCs to provide simple and practical low-latency micropayments tailored for the scenario to connect a large number of payers and payees. However, the solution requires the operation of a hub which has to provide collateral for out-going channels, and provides a target for attacks. The system provides a useful application of multi-party smart contracts that can be expressed with the restricted Bitcoin scripting language.

Low-latency Micropayments are the basis for a machine-payable web. This paradigm has been coined by 21 Inc. Today, the web is mostly used by humans, and websites are monetized by advertisement. In principle, advertisement has allowed the web to stay open. If there would have been no advertisement, websites needed paywalls and subscriptions to be profitable. In the years to come, the web will be ever more dominated by machines. However, machines are not susceptible for advertisement. Currently, the equivalent of paywalls is used for machines. Humans have to provide personal information and payment details to buy API keys for their machines, such that machines are able to access a particular web service. This cumbersome process prevents interoperability and competition. Furthermore, it endangers human privacy.

Micropayments are in general indispensable for machine-to-machine payments between machines without a preprogrammed trust relationship, since micropayments allow to advance payments with negligible financial risk.

Device-centric Business Models

The discussion of S²aaS and the concept of economic devices has shown that cryptocurrencies allow novel device-centric business models. Billing can be handled on the device itself, without backend infrastructure and third party financial services providers. Furthermore, revenue-sharing agreements can be encoded with smart contracts. Thus, a connected sensor could be programmed to share revenue from data sales with its owner and its manufacturer. An early example of a comparable model is provided by the 21 BitShare chip, a Bitcoin mining chip with hardcoded addresses for revenue distribution (Srinivasan 2015). One address is controlled by 21, and one address is controlled by the device, and thus by the user. Cryptocurrencies, smart contracts, and tokens provide hardware manufacturers novel tools to bring the as-a-service approach into the physical world. This might be of particular importance for

1 https://blockchain.info/de/charts/bip-9-segwit (Accessed: 2017-02-23)

products targeted towards emerging economies. Large upfront investments for consumers can be converted into continuous tiny payments. This is only possible if payments are frictionless and intermediaries do not consume the margin. Furthermore, economic devices provide a much a better basis for the sharing economy and secondary markets for products, since ownership or control can be provably transfered with atomic transactions, an independent of third party permissions.

Choosing a Cryptocurrency for an IOT Project

When the research for this thesis was started in late 2013, the number and variations of cryptocurrencies were small. Alternatives to Bitcoin, such as Litecoin, were direct descendants of Bitcoin's code base, and differences were mainly in the selection of certain parameters. As illustrated in Section 4.3.1, today, there are hundreds of different cryptocurrencies and special purpose tokens. While most of them are still clones, a small number introduces innovative features. Most notably, there is Ethereum with the second largest market capitalization and a growing ecosystem of developers, ICOs, and industry interest (see Chapter 4). Ethereum was introduced in Section 3.2. Particular emphasize was given to the differences to Bitcoin. Ethereum provides a much shorter block time (15 s in contrast to 10 min), and a sophisticated virtual machine with support for expressive programming languages and autonomous programs called contracts. The shorter block time is only a minor advantage for IOT applications, because those predominantly have real-time requirements. Thus, trust-minimizing off-chain solutions have to be employed in any case. On the contrary, a high block rate implies increased resource requirements for light clients – another point of consideration for many IOT applications. In addition, resource-constrained connected devices often lack the ability to be updated continuously. Furthermore, in many cases these devices are not actively maintained. Progressive cryptocurrencies with frequent, breaking changes, i.e. hard forks, such as Ethereum, are thus problematic, since each hard fork effectively introduces a new currency. Software updates of Bitcoin have been legacy-compatible for the last years. Non-updated devices might not be able to benefit from new features and optimizations, but they are still able to transact with participants using a newer software under the old protocol rules, which provide a common denominator. The prototype in Sec. 7.5 shows that with Ethereum fiduciary code and business logic related to economic devices can be implemented entirely on the trustworthy blockchain network. In the discussion of an implementation based on Bitcoin, it was shown that this can not be achieved with Bitcoin. In this case, fiduciary code has to be be executed on the device itself or by using multisignature constructs. This increases the requirements of trustworthy and secure hardware, and thus the cost of the hardware.

8.3 OUTLOOK AND FUTURE WORK

With the advent of cryptocurrencies, machines are becoming economic actors. Connected devices can become smart property, which can be securely rented, traded and collateralized (c.f. Chapter 7 and Section B. Machines will be able to trade digital goods and services with each other on a fine-granular level, enabling a higher utilization of resources, increased interoperability, and enhanced security. However, cryptocurrencies are complex socio-technological constructs, which require interdisciplinary research and development to build a secure and scalable infrastructure, as well as to understand the consequences.

The most pressing area of research is scalability. Approaches were discussed already in Sec. 3.1.3. One important area of research and development are payment channel networks. In Chapter 6, a hub-and-spoke network was introduced. There the question already arose, how a sustainable business model for a hub might look like, and if centralization and the pressure to be economically efficient might lead to exclusion of market participants[2]. HTLCs can be used not only to create a one-hop hub-and-spoke topology, but an arbitrary network topology where payments can be routed over multiple untrusted intermediary nodes (Poon and Dryja 2015; Decker and Wattenhofer 2015). An establishment of such a network could lead to a phase-transition for blockchain-based cryptocurrencies. Instead of using on-chain transactions, which have an inherent latency, the vast amount of payments will happen almost instantly on fully collateralized payment channels. Still, as discussed in Sec. 6.8, although payment channel networks are able to increase the number of transactions between network participants tremendously, on-chain scaling is still required for onboarding of new network participants and security of the collateral (McCorry et al. 2016).

There are novel cryptocurrency designs, which try to address some of the scalability issues of blockchains. One of these approaches, IOTA[3], aims in particular at being a cryptocurrency for IOT applications. Instead of a blockchain, IOTA introduces a data structure called *tangle* (Popov 2016). The tangle is a DAG of transactions, and nodes have to validate at least two prior transactions and perform proof-of-work to create a valid transaction that can be attached to the tangle. A DAG structure allows for higher throughput, since transactions can be processed in parallel, and discarding blocks and miners eliminates transaction fees. However, the project is still in an early stage and the viability of this approach is uncertain. Formal proofs, simulations and real-world network tests have to be performed in order to provide evidence for the security and practical scalability.

A related field of research is how to come to an agreement on protocol updates in general (A. Narayanan 2015a), and how to implement them safely, since there is a high risk of network splits and exclusion of participants.

2 In this case data providers.

3 https://iotatoken.com

An example is the split of Ethereum into two independent networks and currencies – Ethereum and Ethereum Classic. Cryptocurrencies are subject to strong network effects. Thus, there is a tendency towards a winner-takes-all dynamic (c.f. Fig. 4.4a). However, for machines these dynamics might be different, if inter-currency exchanges become frictionless. Centralized services such as Shapeshift[4] already provide APIs to exchange various cryptocurrencies without the need for any kind of registration or user account. However, in order to truly own the currency, machines need to be part of respective cryptocurrency network. Thus, the resource requirements grow with every additional currency. An alternative are meta-currencies living on top of other cryptocurrency networks. This scenario is likely, if a general cryptocurrency platform such as Ethereum is able to scale, and can become predominant.

For mobile crowdsensing, the next step would be to perform a large-scale field study to address social and economic questions. The here presented prototype is only a technical proof of concept. There are interesting questions, about what kind of smartphone-based data would people sell, and for what price. Would people require to know the identity of buyers, and who would be interested in these data? However, since the hub would require quite some capital, it would be advisable to address these questions with an off-chain payment system until commercial hubs or systems like the lightning network are sufficiently distributed.

For the implementation of cryptocurrency-enabled, or economic devices, there are the questions of how they can, and should interact with cryptocurrency networks, and what code should be executed on the device itself and what should be executed on the network. There are a lot of trade-offs that have to be made. Some came up in the discussion in Sec. 7.7.3 and Sec. 7.8.2, but more application-specific prototypes have to be build and analyzed. While there are already valuable applications of *dumb* economic devices, e.g. smart property or the more complex blockchain-enabled public display discussed in Chapter 7, the most promising applications are autonomous machine-to-machine transactions. For these, digital bearer assets, micropayments, atomic transactions, and smart contracts are necessary. But these are not sufficient. Self-learning software agents embedded into the devices and the cryptocurrency platforms are needed to establish an *Economy of Things*.

4 https://www.shapeshift.io

A

BITCOIN START-UP ECOSYSTEM: REPRESENTATIVE CASES

In (Wörner, Von Bomhard, et al. 2016), we investigated six promising start-up companies that are active in the Bitcoin ecosystem with applications beyond financial services. Most of these companies appear repeatedly in this thesis. Therefore, the case studies are presented here for reference.

A.1 FILAMENT (INTERNET OF THINGS)

Filament[1] provides wireless sensor networks for the (industrial) IOT, e.g. for smart cities or smart agriculture use case. Most IOT platform providers follow a centralized approach by connecting all devices to their respective cloud-infrastructure. This has the major disadvantage that devices depend on a central infrastructure in order to operate. Moreover, it can be argued that this approach cannot keep up technically and economically with the increasing number connected devices.

Filament is one of the first companies that develop a fully decentralized IOT infrastructure, which encompasses three blockchain-related aspects: (1) Each device is registered on the blockchain providing a verifiable and immutable identity. This enables discovering of and authenticating with other devices/services without the need of a dedicated backend infrastructure. Therefore, devices are technically autonomous and are able to operate independently of Filament. (2) Each device is governed by a *smart* contract, which manages agreements of device control/ownership, data access and financial agreements concerning the device. Ownership can be transferred permanently or temporarily by a simple transaction on the blockchain. Filament implements the financial agreements as a Product-as-a-Service, which means that the owner gets paid directly for the ongoing use of the device. (3) Furthermore, each device is able to transfer value in form of bitcoins to other devices in order to get access to data or request some service.

As described, devices can be operated and governed by using only the blockchain as a backend and therefore without any technical dependence on the platform creator (Filament) or other third parties. This might bare great benefits for customers needing to deploy large Industrial IOT applications with a lifetime of 5-10 years. Because they want to minimize the risk of a lock-in with a specific company. Moreover, according to Filament, customers prefer paying continuously on a real-time basis instead of an upfront investment,

1 https://filament.com/

which can be solved efficiently by Bitcoin micropayments. Ownership is decoupled from usage and both are independent of the manufacturer or a platform provider.

Filament itself is a venture capital-backed company formerly known as Pinocc.io. They claim to have their first deployments with Fortune 500 companies in 2016. They will get paid for the ongoing use of their devices (by owning the smart contract). Moreover, they work on a licensing model, i.e. Customers can attach a module version to their own devices, which will give them all the described benefits of Filament in return to a small fraction of the payments for the ongoing use of the devices. However, since all protocols will be open and there is no dependence to Filament by design, other companies could use that and build their own hardware without Filament.

A.2 ASCRIBE (INTELLECTUAL PROPERTY)

Ascribe[2] aims to provide provenance of intellectual property. Digital work, like art, photos, and music, can be registered publicly on the bitcoin blockchain together with its accompanying terms and conditions. The technological basis is the spool protocol[3], an overlay protocol that uses Bitcoin transactions to represent unforgeable ownership transfers and licensing agreements for digital work. Thus, authenticity of ownership and usage rights can always be proven. Ascribe focuses on digital art in particular. Although art is in principle copyrighted by the time of creation, it is currently cumbersome and expensive to officially register work and prove ownership (see e.g. http://copyright.gov/docs/fees.html). Therefore, hardly any digital art gets registered. Ascribe aims to change the status quo by providing a virtually free and automatable registration process. Moreover, the unique representation of digital work based on cryptography and the blockchain is the basis for the creation of a secondary market for digital work. For example the spool protocol enables creating limited editions of digital work. So far, this has not been possible without relying on some central institution.

The spool protocol is open source and can be used by anybody. In principle, the only costs are bitcoin transaction fees. However, Ascribe wraps the protocol in convenient web services and provides tools adapted to particular customer groups, e.g. individual creators, museums and marketplaces. Ascribe's revenue model is then based on a share of the rentals and sales of registered digital work that are facilitated by their APIs.

The core innovation is the application of the bitcoin blockchain to provide commoditized provenance of intellectual property. Provenance is demonstrated by relying on the immutability of the bitcoin blockchain, instead of an authority.

2 https://www.ascribe.io/
3 https://github.com/ascribe/spool

A.3 OPENBAZAAR (E-COMMERCE MARKETPLACE)

OpenBazaar[4] is an open source project consisting of a protocol and a reference implementation that enables a decentralized peer-to-peer e-commerce marketplace. In comparison to traditional e-commerce market places like eBay and Amazon, there is no central server or authority that is running the market place. Thus, there is no middleman who is able to charge fees or to restrict offered products and services. Everyone with Internet access is able to set up a shop by running a network client. Payments are facilitated using Bitcoin transactions. Therefore, no payment provider or banking account is needed. This lowers payment transaction fees and increases the global reach. Besides sellers and buyers there are notaries and arbiters for dispute resolution participating in the marketplace. Those latter participants are involved by using bitcoin multi-signature transactions. Thus, OpenBazaar unbundles the functions of traditional marketplaces. Trades on the OpenBazaar network are based on Ricardian Contracts (Grigg 2004), i.e. an electronic document that defines the terms of a trade such that it is readable by computers and humans, and is cryptographically signed. Apart from selling physical and digital products, OpenBazaar can also be used to trade speculative contracts, which can be readily represented by Ricardian Contracts.

The main value proposition of OpenBazaar to sellers as well as to buyers is the elimination of fees and restrictions. Since marketplaces are subject to network effects most users will most probably not switch immediately from traditional marketplaces to OpenBazaar. However, OpenBazaar could set foot in under-served niche markets. Examples could be digital goods, developing countries with limited access to traditional payment services, and prohibited goods.

OpenBazaar itself is not a company, but its main developers founded the venture capital-backed company OB1. Their current focus is on developing the OpenBazaar protocol and its reference implementation. As outlined above, OB1 is not able to profit directly from transactions on the marketplace in the way traditional revenue mechanics on centralized marketplaces work.

A.4 21 (DIGITAL MICRO COMMERCE MARKETPLACE)

At first sight, the categorization of 21 Inc.[5] as a challenger seems odd, since 21 is an infrastructure and platform provider for the bitcoin ecosystem. Indeed, it is often categorized as a mining company. However, we argue that 21 is better classified as a marketplace for digital micro services, which has the potential to challenge traditional Internet business models.

With a funding of $121M, 21 supersedes every other start up in the bitcoin ecosystem. They have developed an embeddable ASIC mining chip that they

4 https://www.openbazaar.org/
5 https://www.21.co

are using in their own mining operations, but which is also embeddable into arbitrary connected devices. In November 2015 they released their first product, the 21 Bitcoin computer. Essentially the 21 Bitcoin computer is a full-stack development platform to build bitcoin-payable digital services, which can be published and discovered on 21's digital marketplace. Individual service consumptions, like an API call, can be billed at as little as 1 satoshi. The embedded mining chip, which is currently coupled to a mining pool operated by 21, supplies the device with a continuous stream of satoshis.

21 aims to embed their chips into any connected device (e.g. smart phones) to establish bitcoin as a system resource like CPU, bandwidth or disk space, but for the purpose of buying and selling digital goods and services (Srinivasan 2015). It is crucial to understand that it does not make sense to sell the small amounts of mined bitcoins for Fiat currency on an exchange. Instead, the idea is to supply every device with a continuous stream of bitcoin from the point of commissioning on so that it can directly operate on the marketplace.

Having such an infrastructure of bitcoin-enabled devices in place at scale, could offer compelling new opportunities and even disrupt traditional business models of the Internet. For example, it has been difficult for news sites to directly monetize their content on the Internet. It is still tedious for users who want to read just one article to signup, enter their credit card information and buy a subscription. Thus, most news sites still depend on indirect revenue by advertisements, which becomes more problematic with the increasing spread of ad-blockers. These problems could be eliminated with the diffusion of a bitcoin-enabled infrastructure for frictionless micropayments. Similarly, a bitcoin-enabled IOT device, e.g. for automatic irrigation of farms, could pay a weather service API in return for accurate weather prediction data without the need for signup. Moreover, the device could search automatically for the cheapest (and best in terms of reputation) weather service API on the marketplace.

The 21 platform is a mixture of a centralized and a decentralized model. As of today, the mining power is bound to 21, and returns get allocated to a wallet owned by 21. However, it is announced that this will change in the future. The marketplace for digital services on the other hand is in principle decentralized and trades can be conducted peer-to-peer. Currently, 21 generates revenue by private mining operations and by sales of the bitcoin computer. In the future, however, all kinds of interesting revenue models are imaginable: selling and licensing of mining chips, revenue sharing of embedded mining operations or tiny transaction fees for off-chain transactions to name just a few.

In conclusion, 21 could change how resources in the Internet are paid for, and thereby also contribute to making new resources available, which have not been available yet because of missing incentives

A.5 FACTOM (RECORDS MANAGEMENT)

Factom[6] is an open source software project that provides businesses with the ability to prove data integrity and to create verifiable and immutable audit trails.

While data integrity could be achieved by directly adding a hash of the data to a bitcoin transaction and thereby timestamp the data on the bitcoin blockchain, this method does not scale. On one hand, this is because of inherent scalability issues of bitcoin, and on the other hand because of transaction fees. Therefore, Factom consists of a peer-to-peer network that is independent of the bitcoin network. Customers of Factom generate hashes of their data and send them for recordkeeping to the Factom network. There, all hashes are compressed to a single hash by building a Merkle tree and taking the root of the tree. This single hash value is then stored in the bitcoin blockchain. This provably timestamps all individual records without having to write all records individually into the bitcoin blockchain. The network maintains its own cryptocurrency, *factoids*, which is used to incentivize participants of the peer-to-peer network to provide their resources. A factoid can be transformed into entry credits, which can be used to submit new records. The price of a factoid depends on the market value, but the price of an entry credit is fixed to 1/10th of a cent.

In summary, Factom provides a decentralized platform for data provenance with a permanent, timestamped record of an unforgeable reference to the data anchored in the blockchain. This offers an efficient and cheap alternative for businesses, institutions and governments to have a proof of existence, proof of process or proof of audit for their data. Their first publicly announced project is using Factom for an official land title registry in partnership with the government of Honduras (Batlin 2015). In fact, Factom is an interesting option for governments in developing countries. They often face mismanagement and corruption, but cannot afford or enforce infrastructure and processes to guarantee compliance of their administration. Moreover, the Factom solution offers also an opportunity for small businesses/start-ups to have more auditing and to prove compliance with regulations without hiring expensive professional companies for that.

A.6 ONENAME (IDENTITY)

Onename[7] allows registering identities on the Bitcoin blockchain. This blockchain identity can be connected to various online identities like Facebook, Twitter, Pretty Good Privacy (PGP) keys and a Bitcoin address. Thus, it provides a probabilistic identity which reliability grows with the number of verified connected accounts. Onename is based on an open source overlay protocol called

6 http://factom.org/

7 https://onename.com/

Blockchain ID. Everyone is able to register his identity without having to rely on Onename services. Blockchain identities are independent of the company Onename, are referenced on the Bitcoin blockchain and are therefore owned by the holder of the respective private key.

Blockchain IDs will allow signing up on third party websites comparable to Facebook Login or Google Sign-In. There is no need for passwords since authentication is done using digital signatures. Blockchain ID is also used as an identity provider for OpenBazaar.

Moreover, it is possible to add additional namespaces. In this sense Blockchain IDs could represent not only humans, but also machines. Thus, Blockchain IDs might be the basis of a decentralized Domain Name Service (DNS) system or an IOT registry.

Essentially, the technology allows individuals to own their online identities, rather than being dependent central institutions. Holding, i.e. registering and prolonging, a Blockchain ID requires fees that directly support the Bitcoin ecosystem. One part of the fee is a typical Bitcoin transaction fee that will be collected by a miner. The other part of the fee is particular to the protocol and leads to *burning* of bitcoins. Since the number of bitcoins is constrained, the elimination of bitcoins theoretically increases the value of existing bitcoins. While it might seem that our current identities are free, we actually pay with our personal data. Every time we use Facebook to login into a third party website, we give away more information about us. Identity is also subject to network effects. Institutions will only start to accept Blockchain IDs if there are enough people using them and demand acceptance.

Onename basically follows the same strategy as OB1. The initial focus is on developing open source protocols and advocating their adoption, instead of having a revenue model in place.

B

IMPLEMENTING SMART PROPERTY

The prototype in Section 7 implemented only active capabilities of economic devices. Here, supplementary implementations of passive economic capabilities based on the smart property concept. The following concepts are implemented with Bitcoin and Ethereum.

- Low-trust atomic trades
- Low-trust renting
- Liquid property (using smart property as collateral for loans)

B.1 LOW-TRUST ATOMIC TRADES

Smart property can be sold via the Internet in an atomic process without third party escrow. Transfer of ownership[1] and transfer of money happen at once.

B.1.1 *Implementation based on Bitcoin*

The simplest way to achieve atomicity in Bitcoin is by executing both parts of the trade in a single transaction. In Fig. B.1 an atomic trade protocol based on Hearn 2011b is shown. The protocol may either be initiated by the seller (A) or by the buyer (B). We assume A initiates the protocol and sends the price, a Bitcoin address where she would like to receive the payment, and a reference to the UTXO that represents the smart property (ownership output) to B. B creates a new ownership key pair and prepares a transaction that sends the price to A and transfers the ownership to the newly created ownership key. B signs the transaction and sends it to A. To accept the offer A signs the ownership input and broadcasts the transaction to the Bitcoin network.

After a number of confirmations, B can provide a SPV proof to the smart property. Therefore, B provides a selection of subsequent block headers entailing the header of the block the transaction is in, plus its merkle proof. The smart property is able to reason about the amount of proof-of-work that was spent to generate the blocks. In case of doubt, the smart property can ask for more block headers.

In this protocol the smart property is not part of the Bitcoin network, but is able to verify SPV proofs provided by a potentially untrusted party.

1 We do not necessarily mean legal rights of ownership, but the ability to control the property

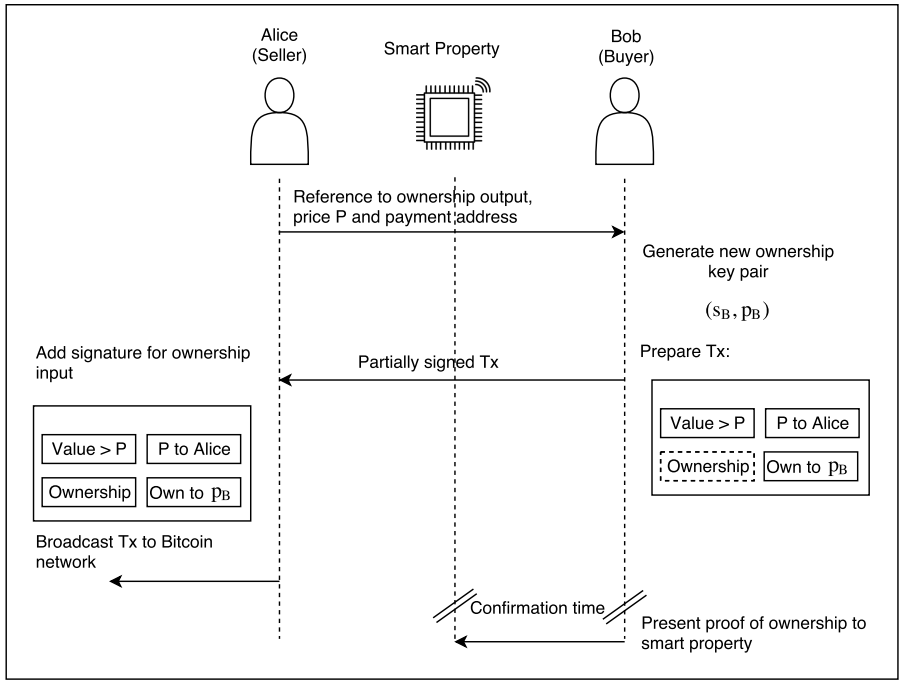


Figure B.1.: Protocol of atomic trade of a smart property on the Bitcoin blockchain. The smart property does not need to interact with the Bitcoin network itself.

B.1.2 *Implementation based on Ethereum*

The following contract illustrates a contract representing a tradable smart property in Ethereum. The original issuer of the contract is identified as the owner. In practice this would be the manufacturer. Ethereum does not support collaborative transactions, i.e. transactions with operations authorized by different entities. Therefore, selling has to be implemented in a two-stage process. In the contract below, we do this by implementing a *sell function* which can be only called by the current owner (through the *onlyOwner modifier*). The *sell function* has two parameters: the price, and a buyer address. The *buyer parameter* allows to explicitly state a seller. Otherwise everyone who would pay the price using the *buy function* would be able to buy the smart property.

```
contract SmartProperty {

    address owner;
    address buyer;
    bool onlyBuyerFlag;
    bool isOnSale;
    uint price;

    modifier onlyOwner() {
        if (msg.sender != owner) throw;
    }

    modifier onlyBuyer(bool flag) {
        if (flag && msg.sender != buyer) throw;
    }

    function SmartProperty() {
        owner = msg.sender;
    }

    function sell(uint _price, address _buyer) onlyOwner() {
        isOnSale = true;
        if (_buyer != 0) {
            onlyBuyerFlag = true;
            buyer = _buyer;
        } else {
            onlyBuyerFlag = false;
        }
        price = _price;
    }

    function stopSale() onlyOwner() {
        isOnSale = false;
    }

    function buy() onlyBuyer(onlyBuyerFlag) {
        if (isOnSale) {
            if (msg.value == price) {
                isOnSale = false;
                owner = msg.sender;
                owner.send(msg.value);

            } else {
                msg.sender.send(msg.value);
            }
        } else {
            msg.sender.send(msg.value);
        }
    }
}
```

B.2 LOW-TRUST RENTING

Time-restricted transfer of ownership with adjustable counterparty risk. Smart property can be the basis for a peer-to-peer sharing[2] ecosystem.

B.2.1 *Implementation based on Bitcoin*

The idea is to combine shared ownership and the unidirectional payment channel. Therefore, the parties create a transaction that creates a multi-signature ownership output for A (owner) and B (renter), and a 2-of-2 multi-signature output where B deposits some amount. Furthermore, both parties create a timelocked refund transaction that allocates the ownership output back to A, and the deposit back to B. The timelock should cover the maximal renting period, and the deposit should cover the renting price for that period. B can now pay in small increments with off-chain payment transactions that

2 In the sense of Uber and AirBnB.

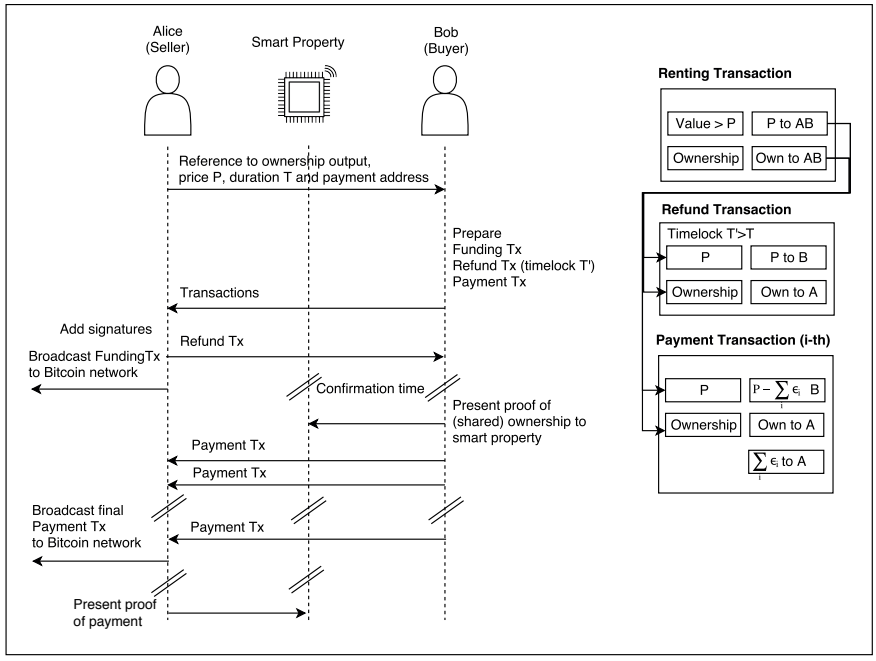


Figure B.2.: Protocol for trust-minimized renting of smart property using a Bitcoin payment channel.

entail an additional ownership output that assign the car ownership back to A. Because of Bitcoin's current malleability issue it is important that the order of providing signatures is such that only A is able to broadcast the funding transaction. Furthermore, A needs also a (tiny) payment transaction from B before broadcasting the funding transaction. Otherwise B could just use the smart property without paying for the entire maximal renting period, since A would not have a transaction that is immediately valid. Figure B.2 illustrates the protocol in more detail.

B.2.2 *Implementation based on Ethereum*

The following contract implements low-trust renting of smart property on Ethereum. Note that the *Rentable Property* contracts inherits properties and functions from the *SmartProperty* contract. When deploying the contract the owner sets a deposit a renter has to provide and a time based renting price. The main functions are a *rent* and a *returnPropery* function. The *rent* function allows an arbitrary account or contract to rent the smart property by providing a deposit. The deposit is held in the contract, and can only be released by the rules of the contract. Neither the owner, nor the renter have control over the deposit. The contract keeps also track of the block in which the rent function

was executed. This *startBlock* serves as the begin of the renting period as measured in time of blocks. The Ethereum network has an average block generation rate of approximately 14 s. The software running locally on the smart property has now to be notified that it must obey orders signed by the renter for a maximal period defined by the price and the deposit. The renter can later call the *returnProperty* function which calculates the price for the renting period and distributes the deposit accordingly. If the renter does not return the property in time her access/control rights expire.

```
contract RentableProperty is SmartProperty {

    address renter;
    uint startBlock;
    uint pricePerBlock;
    uint deposit;
    bool isRentable = true;

    modifier onlyRenter() {
        if (msg.sender != renter) throw;
    }

    function RentableProperty(uint _pricePerBlock, uint _deposit) {
        owner = msg.sender;
        pricePerBlock = _pricePerBlock;
        deposit = _deposit;
    }

    function notRentable() onlyOwner() {
        isRentable = false;
    }

    function rentProperty() {
        if (isRentable) {
            if (msg.value >= deposit) {
                renter = msg.sender;
                startBlock = block.number;
                isRentable = false;
                if (msg.value > deposit) {
                    msg.sender.send(msg.value - deposit);
                }
            } else throw;
        } else throw;
    }

    function returnProperty() onlyRenter {
        renter = 0;
        if (deposit > price*(block.number-startBlock)) {
            owner.send(price*(block.number-startBlock));
            msg.sender.send(deposit-price*(block.number-startBlock));
        } else {
            owner.send(deposit);
        }
        isRentable = true;
    }
}
```

B.3 LIQUID PROPERTY

Smart property can be used as collateral for loans. The underlying principle is that property ownership is transferred to the lender if repayment terms are not met. Thereby the loan gets securitized by the smart property. Because of the global permissionless nature of cryptocurrencies, a global market for loans on individual smart properties can emerge which lowers the cost of loans. Moreover, since transactions are public, a borrower is able to prove timely payments of earlier loans.

```
OP_IF
    2 <pubKeyA><pubKeyB> 2
    OP_CHECKMULTISIG
OP_ELSE
    <T> OP_CHECKLOCKTIMEVERIFY OP_DROP
OP_END
```

Figure B.3.: PubScript of timelocked 2-of-2 multi-signature ownership output.

B.3.1 *Implementation based on Bitcoin*

We will implement the following contract. The owner of a smart property (B) wants a loan of size L and provides the property as security for a creditor (A). If the owner (and debtor) does not repay the loan (plus interest) until time T, ownership of the property will be transferred to the creditor.

After A and B agreed on terms, B creates the *loan transaction* spending his ownership output and creating a timelocked 2-of-2 multi-signature ownership output that can be redeemed either collaboratively by A and B, or by A alone after time T. Furthermore, B adds an output that credits him with loan L, leaving the input, providing the loan, for A to add.

B sends the partial transaction to A, who prepares the *settlement transaction*. The settlement transaction reassigns the ownership back to B, and credits A with the loan plus interest. A completes the loan transaction and partially signs the settlement transaction, and sends both transactions back to B. B can then broadcast the loan transaction to the Bitcoin network. If B can provide an input to the settlement transaction covering $L + \Delta$ before time T, B can complete the settlement transaction and regain sole ownership of the property. However, if the settlement transaction does not enter the blockchain before T, then A is able to claim sole ownership.

Another Bitcoin-based protocol for smart property as a collateral for loans is described in Hearn 2011b. However, the protocol has the problem that a creditor has the ability to resell the property immediately without giving the debtor a chance to repay the loan. In the protocol described above this is prevented by use of the multi-signature.

B.3.2 *Implementation based on Ethereum*

A simple implementation of property liquidification defines a `loan` and a latest `payDay`[3]. The owner of the property deploys the *LiquidProperty* contract, and a lender can provide a loan using the *giveLoan* function. The loan is immediately credited to the debtor who can use the *pay* function to pay back the loan. Either lender or debtor can call the *enforce* function to enforce the

3 Here represented as a Ethereum block number.

contract. If the loan is paid, then the contract state return to the initial state, if the loan is not paid and it is later than `payDay`, then the ownership gets transferred to the lender.

```
contract LiquidProperty is SmartProperty {

    address lender;
    uint loan;
    uint payDay;
    uint paid = 0;

    bool isLiquid = false;

    function LiquidProperty(uint _loan, uint _payDay) {
        owner = msg.sender;
        loan = _loan;
        payDay = _payDay;
    }

    function giveLoan() {
        if (msg.value < loan || isLiquid) throw;
        isLiquid = true;
        lender = msg.sender;
        owner.send(msg.value);
    }

    function pay() {
        if (!isLiquid) throw;
        paid = paid + msg.value;
        lender.send(msg.value);
    }

    function enforce() {
        if (!isLiquid) throw;
        if (paid >= loan) {
            isLiquid = false;
        } else if (block.number > payDay) {
            owner = lender;
            isLiquid = false;
        }
    }
}
```

BIBLIOGRAPHY

Adelantado, Ferran, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, and Joan Melia (2016). „Understanding the limits of LoRaWAN.“ In: *arXiv preprint arXiv:1607.08011* (cit. on p. 8).

Ali, Muneeb, Jude Nelson, Ryan Shea, and Michael J Freedman (2016). „Blockstack: A Global Naming and Storage System Secured by Blockchains.“ In: *2016 USENIX Annual Technical Conference (USENIX ATC 16)* (cit. on pp. 33, 80, 104).

Alstone, Peter, Dimitry Gershenson, and Daniel M. Kammen (2015). „Decentralized energy systems for clean electricity access.“ In: *NATURE CLIMATE CHANGE* 5.4, 305–314 (cit. on pp. 4, 10).

Anderson, D. P. (2004). „BOINC: a system for public-resource computing and storage.“ In: *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pp. 4–10 (cit. on p. 14).

Anderson, David P., Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer (2002). „SETI@Home: An Experiment in Public-resource Computing.“ In: *Commun. ACM* 45.11, pp. 56–61. URL: http://doi.acm.org/10.1145/581571.581573 (cit. on pp. 2, 14).

Andreessen, Marc (2011). „Why Software Is Eating The World.“ In: *Wall Street Journal* 20 (cit. on p. 4).

Andreessen, Marc (2014). *Why Bitcoin Matters*. [Accessed 2016-07-11] (cit. on p. 32).

Andresen, Gavin (2012). *BIP 16: Pay to Script Hash*. https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki (cit. on p. 23).

Androulaki, Elli and Ghassan O. Karame (2014). „Hiding Transaction Amounts and Balances in Bitcoin.“ In: *Trust and Trustworthy Computing: 7th International Conference, TRUST 2014, Heraklion, Crete, June 30 – July 2, 2014. Proceedings*. Ed. by Thorsten Holz and Sotiris Ioannidis. Cham: Springer International Publishing, pp. 161–178. URL: http://dx.doi.org/10.1007/978-3-319-08593-7_11 (cit. on p. 30).

Androulaki, Elli, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun (2013). „Evaluating User Privacy in Bitcoin.“ In: *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*. Ed. by Ahmad-Reza Sadeghi. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 34–51. URL: http://dx.doi.org/10.1007/978-3-642-39884-1_4 (cit. on p. 29).

Andrychowicz, Marcin, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek (2015). „On the Malleability of Bitcoin Transactions.“ In: *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected*

Papers. Ed. by Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–18. URL: http://dx.doi.org/10.1007/978-3-662-48051-9_1 (cit. on p. 103).

Aqeel-ur-Rehman, Abu Zafar Abbasi, Noman Islam, and Zubair Ahmed Shaikh (2014). „A review of wireless sensors and networks' applications in agriculture." In: *Computer Standards & Interfaces* 36.2, pp. 263–270. URL: http://www.sciencedirect.com/science/article/pii/S0920548911000353 (cit. on p. 8).

Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia (2010). „A View of Cloud Computing." In: *Commun. ACM* 53.4, pp. 50–58. URL: http://doi.acm.org/10.1145/1721654.1721672 (cit. on pp. 1, 9, 63).

Ashton, Kevin (2009). „That 'internet of things' thing." In: *RFiD Journal* 22.7, pp. 97–114 (cit. on p. 8).

Atzori, Luigi, Antonio Iera, and Giacomo Morabito (2010). „The Internet of Things: A survey." In: *Computer Networks* 54.15, pp. 2787–2805. URL: http://www.sciencedirect.com/science/article/pii/S1389128610001568 (cit. on p. 7).

Babaioff, Moshe, Shahar Dobzinski, Sigal Oren, and Aviv Zohar (2012). „On Bitcoin and Red Balloons." In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. EC '12. Valencia, Spain: ACM, pp. 56–73. URL: http://doi.acm.org/10.1145/2229012.2229022 (cit. on p. 29).

Back, Adam (2002). „Hashcash-A Denial of Service Counter-Measure." In: (cit. on p. 25).

Back, Adam, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille (2014). „Enabling Blockchain Innovations with Pegged Sidechains." In: (cit. on p. 33).

Banasik, Wacław, Stefan Dziembowski, and Daniel Malinowski (2016). *Efficient Zero-Knowledge Contingent Payments in Cryptocurrencies Without Scripts*. Cryptology ePrint Archive, Report 2016/451. http://eprint.iacr.org/2016/451 (cit. on p. 67).

Banerjee, Prith, Richard Friedrich, Cullen Bash, Patrick Goldsack, Bernardo Huberman, John Manley, Chandrakant Patel, Parthasarathy Ranganathan, and Alistair Veitch (2011). „Everything as a Service: Powering the New Information Economy." In: *Computer* 44.3, pp. 36–43 (cit. on p. 63).

Barham, Paul, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield (2003). „Xen and the art of virtualization." In: *ACM SIGOPS Operating Systems Review*. Vol. 37. 5. ACM, pp. 164–177 (cit. on p. 1).

Baronti, Paolo, Prashant Pillai, Vince W.C. Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu (2007). „Wireless sensor networks: A survey on

the state of the art and the 802.15.4 and ZigBee standards." In: *Computer Communications* 30.7. Wired/Wireless Internet Communications, pp. 1655–1695. URL: http://www.sciencedirect.com/science/article/pii/S0140366406004749 (cit. on p. 8).

Batlin, Alex (2015). *Crypto 2.0 Musings - Digital Business Models*. [Accessed 2015-11-25] (cit. on p. 137).

Beberg, A. L., D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande (2009). „Folding@home: Lessons from eight years of volunteer distributed computing." In: *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1–8 (cit. on pp. 2, 14).

Bentov, Iddo, Rafael Pass, and Elaine Shi (2016). *Snow White: Provably Secure Proofs of Stake*. Cryptology ePrint Archive, Report 2016/919. http://eprint.iacr.org/2016/919 (cit. on p. 31).

Bergkvist, Adam, D Burnett, and Cullen Jennings (2012). „A. Narayanan," WebRTC 1.0: Real-time Communication Between Browsers." In: *World Wide Web Consortium WD WD-webrtc-20120821* (cit. on p. 104).

Biryukov, Alex, Dmitry Khovratovich, and Ivan Pustogarov (2014). „Deanonymisation of Clients in Bitcoin P2P Network." In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: ACM, pp. 15–29. URL: http://doi.acm.org/10.1145/2660267.2660379 (cit. on p. 29).

BitcoinTalk (2011). *Proof of Stake instead of Proof of Work*. [Accessed 2016-10-20] (cit. on p. 31).

BitFury (2015). *Public versus Private Blockchains*. [Accessed 2016-07-13] (cit. on p. 44).

Black, John (2013). „Developments in Data Security Breach Liability." English. In: *BUSINESS LAWYER* 69.1, 199–207 (cit. on p. 1).

Bodenheim, Roland, Jonathan Butts, Stephen Dunlap, and Barry Mullins (2014). „Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices." In: *International Journal of Critical Infrastructure Protection* 7.2, pp. 114–123. URL: http://www.sciencedirect.com/science/article/pii/S1874548214000213 (cit. on p. 12).

Bogner, Andreas, Mathieu Chanson, and Arne Meeuw (2016). „A Decentralised Sharing App Running a Smart Contract on the Ethereum Blockchain." In: *Proceedings of the 6th International Conference on the Internet of Things*. IoT'16. Stuttgart, Germany: ACM, pp. 177–178. URL: http://doi.acm.org/10.1145/2991561.2998465 (cit. on p. 114).

Bohli, Jens-Matthias, Christoph Sorge, and Dirk Westhoff (2009). „Initial observations on economics, pricing, and penetration of the internet of things market." In: *ACM SIGCOMM Computer Communication Review* 39.2, pp. 50–55 (cit. on p. 63).

Bomhard, Thomas von, Dominic Wörner, and Marc Röschlin (2014). „Towards Smart Individual-room Heating for Residential Buildings." In: *Computer Science-Research and Development*, pp. 1–8 (cit. on p. viii).

Bonneau, Joseph, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten (2014). „Mixcoin: Anonymity for Bitcoin with Accountable Mixes." In: *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 486–504. URL: http://dx.doi.org/10.1007/978-3-662-45472-5_31 (cit. on p. 30).

Bonomi, Flavio, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli (2012). „Fog Computing and Its Role in the Internet of Things." In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. MCC '12. Helsinki, Finland: ACM, pp. 13–16. URL: http://doi.acm.org/10.1145/2342509.2342513 (cit. on pp. 2, 14).

Brener, Demian (2016). *On Tokens and Crowdsales: How Startups Are Using Blockchain to Raise Capital*. [Accessed 2016-15-12] (cit. on pp. 53, 54).

Brown, Richard Gendal (2013). *On the Blockchain, Nobody knows you're a Fridge*. [Accessed 2016-08-01] (cit. on p. 3).

Brown, Richard Gendal, James Carlyle, Ian Grigg, and Mike Hearn (2016). *Corda: An Introduction*. [Accessed 2016-15-12] (cit. on pp. 34, 57).

Bungiu, Francisc (2015). *Towards Datamarkets with Bitcoin*. URL: https://github.com/domwoe/datamarket/blob/master/report/MasterThesis_Bungiu_Francisc_Nicolae.pdf (cit. on p. 98).

Buterin, Vitalik (2014). *Ethereum White Paper*. [Accessed 2016-07-20] (cit. on pp. 4, 34, 35).

Buterin, Vitalik (2016a). *Onward from the Hard Fork*. [Accessed 2016-12-17] (cit. on p. 51).

Buterin, Vitalik (2016b). *Thoughts on UTXOs*. [Accessed 2016-10-18] (cit. on p. 21).

Bylica, Pawel, Lukasz Glen, Piotr Januik, Aleksandra Skrzypczak, and Artur Zawlocki (2015). *A Probabilistic Nanopayment Scheme for Golem*. [Accessed 2016-08-18] (cit. on p. 104).

Cachin, Christian (2016). „Architecture of the Hyperledger blockchain fabric." In: *Workshop on Distributed Cryptocurrencies and Consensus Ledgers* (cit. on p. 42).

Cardone, G., L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola (2013). „Fostering participaction in smart cities: a geo-social crowdsensing platform." In: *IEEE Communications Magazine* 51.6, pp. 112–119 (cit. on p. 64).

Castro, Miguel, Barbara Liskov, et al. (1999). „Practical Byzantine fault tolerance." In: *OSDI*. Vol. 99, pp. 173–186 (cit. on pp. 24, 42).

Chaia, Alberto, T Goland, and Robert Schiff (2010). „Counting the world's unbanked." In: *McKinsey Quarterly* 2, pp. 98–99 (cit. on pp. 57, 113).

Chakravorti, Sujit (2003). „Theory of credit card networks: A survey of the literature.“ In: *Review of Network Economics* 2.2 (cit. on p. 66).

Chen, Xiao, Elizeu Santos-Neto, and Matei Ripeanu (2012). „Crowdsourcing for On-street Smart Parking.“ In: *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. DIVANet '12. Paphos, Cyprus: ACM, pp. 1–8. URL: http://doi.acm.org/10.1145/2386958.2386960 (cit. on p. 64).

Choi, Hynyoung and Hal Varian (2012). „Predicting the Present with Google Trends.“ In: *Economic Record* 88, pp. 2–9. URL: http://dx.doi.org/10.1111/j.1475-4932.2012.00809.x (cit. on p. 51).

Chon, Yohan, Nicholas D. Lane, Fan Li, Hojung Cha, and Feng Zhao (2012). „Automatically Characterizing Places with Opportunistic Crowdsensing Using Smartphones.“ In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. UbiComp '12. Pittsburgh, Pennsylvania: ACM, pp. 481–490. URL: http://doi.acm.org/10.1145/2370216.2370288 (cit. on p. 64).

Christin, Delphine (2015). „Privacy in mobile participatory sensing: Current trends and future challenges.“ In: *Journal of Systems and Software*. URL: http://www.sciencedirect.com/science/article/pii/S0164121215000692 (cit. on p. 64).

Clack, Christopher D., Vikram A. Bakshi, and Lee Braine (2016). „Smart Contract Templates: foundations, design landscape and research directions.“ In: *CoRR* abs/1608.00771. URL: http://arxiv.org/abs/1608.00771 (cit. on p. 109).

Coase, R. H. (1937). „The Nature of the Firm.“ In: *Economica* 4.16, pp. 386–405. URL: http://dx.doi.org/10.1111/j.1468-0335.1937.tb00002.x (cit. on p. 2).

Cocchia, Annalisa (2014). „Smart and Digital City: A Systematic Literature Review.“ In: *Smart City: How to Create Public and Economic Value with High Technology in Urban Space*. Ed. by Renata Paola Dameri and Camille Rosenthal-Sabroux. Cham: Springer International Publishing, pp. 13–43. URL: http://dx.doi.org/10.1007/978-3-319-06160-3_2 (cit. on p. 8).

Cohen, Bram (2003). „Incentives build robustness in BitTorrent.“ In: *Workshop on Economics of Peer-to-Peer systems*. Vol. 6, pp. 68–72 (cit. on p. 2).

Coindesk (2015). *Bitcoin Venture Capital*. [Accessed 2015-10-08] (cit. on p. 46).

Corbett, James C., Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, JJ Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, Wilson Hsieh, Sebastian Kanthak, Eugene Kogan, Hongyi Li, Alexander Lloyd, Sergey Melnik, David Mwaura, David Nagle, Sean Quinlan, Rajesh Rao, Lindsay Rolig, Yasushi Saito, Michal Szymaniak, Christopher Taylor, Ruth Wang, and Dale Woodford (2012). „Spanner: Google's Globally-Distributed Database.“ In: *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*. Hollywood, CA: USENIX Association, pp. 261–264. URL: https://www.usenix.

org/conference/osdi12/technical-sessions/presentation/corbett (cit. on p. 29).

Coric, V. and M. Gruteser (2013). „Crowdsensing Maps of On-street Parking Spaces." In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 115–122 (cit. on p. 64).

Costan, Victor and Srinivas Devadas (2016). *Intel SGX Explained*. Cryptology ePrint Archive, Report 2016/086. http://eprint.iacr.org/2016/086 (cit. on p. 57).

Costin, Andrei, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti (2014). „A Large-Scale Analysis of the Security of Embedded Firmwares." In: *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, pp. 95–110. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/costin (cit. on p. 12).

Cox, Catie (2016). *TCP Disconnects "Smart" Lightbulb Servers, Leaves Buyers In The Dark*. [Accessed 2016-08-20] (cit. on p. 12).

Croman, Kyle, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, and Emin Gün (2016). „On scaling decentralized blockchains." In: (cit. on pp. 28, 29, 74).

Daian, Phil (2016). *Analysis of the DAO exploit*. [Accessed 2016-07-21] (cit. on p. 40).

Decker, Christian and Roger Wattenhofer (2014). „Bitcoin Transaction Malleability and MtGox." In: *19th European Symposium on Research in Computer Security (ESORICS), Wroclaw, Poland* (cit. on p. 129).

Decker, Christian and Roger Wattenhofer (2015). „A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels." English. In: *Stabilization, Safety, and Security of Distributed Systems*. Ed. by Andrzej Pelc and Alexander A. Schwarzmann. Vol. 9212. Lecture Notes in Computer Science. Springer International Publishing, pp. 3–18. URL: http://dx.doi.org/10.1007/978-3-319-21741-3_1 (cit. on pp. 103, 131).

Delgado-Segura, Sergi, Cristian Tanas, and Jordi Herrera-Joancomartí (2016). „Reputation and Reward: Two Sides of the Same Bitcoin." In: *Sensors* 16.6, p. 776. URL: http://www.mdpi.com/1424-8220/16/6/776 (cit. on p. 81).

Delmolino, Kevin, Mitchell Arnett, Ahmed E Kosba, Andrew Miller, and Elaine Shi (2015). „Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab." In: *IACR Cryptology ePrint Archive* 2015, p. 460 (cit. on p. 40).

Delmolino, Kevin, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi (2015). *A programmer's guide to ethereum and serpent* (cit. on p. 40).

Denmead, Ken (2013). *Netatmo Is the Weather Station for the Rest of Us. Wired Magazine*. [Accessed 2014-04-02] (cit. on p. 62).

Deterding, Sebastian, Dan Dixon, Rilla Khaled, and Lennart Nacke (2011). „From Game Design Elements to Gamefulness: Defining "Gamification"."

In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. MindTrek '11. Tampere, Finland: ACM, pp. 9–15. URL: http://doi.acm.org/10.1145/2181037.2181040 (cit. on p. 64).

Diffie, W and ME Hellman (1976). „New Directions in Cryptography." In: *IEEE Transactions on Information Theory* 22.6, 644–654 (cit. on p. 68).

Dong, Bing and Khee Poh Lam (2014). „A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting." In: *Building Simulation*. Vol. 7. 1. Springer, pp. 89–106 (cit. on p. 62).

Douceur, John R (2002). „The sybil attack." In: *Peer-to-peer Systems*. Springer, pp. 251–260 (cit. on pp. 2, 24).

Ehrsam, Fred (2016). *App Coins and the dawn of the Decentralized Business Model*. [Accessed 2016-15-12] (cit. on p. 55).

Ericsson (2016). *Ericsson Mobility Report*. [Accessed 2016-11-21] (cit. on p. 61).

Eskandari, Shayan, David Barrera, Elizabeth Stobert, and Jeremy Clark (2015). „A first look at the usability of bitcoin key management." In: (cit. on p. 30).

Ethereum Wiki (2016a). *Ethereum Development Tutorial*. [Accessed 2016-12-19] (cit. on p. 36).

Ethereum Wiki (2016b). *Geth*. [Accessed 2016-11-08] (cit. on p. 116).

Ethereum Wiki (2016c). *JavaScript API*. [Accessed 2016-11-08] (cit. on p. 116).

Ethereum Wiki (2016d). *Mining*. [Accessed 2016-11-08] (cit. on p. 41).

Ethereum Wiki (2016e). *Whisper*. [Accessed 2016-11-08] (cit. on p. 119).

Evans, Benedict (2015). *The Smartphone is the New Sun*. [Accessed 2016-08-16] (cit. on p. 1).

Evans, Benedict (2016). *The End of a Mobile Wave; The Dell of Mobile*. [Accessed 2016-08-16] (cit. on p. 1).

Eyal, Ittay, Adem Efe Gencer, Emin Gun Sirer, and Robbert Van Renesse (2016). „Bitcoin-NG: A Scalable Blockchain Protocol." In: *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, pp. 45–59. URL: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal (cit. on p. 29).

Eyal, Ittay and Emin Gün Sirer (2014). „Majority Is Not Enough: Bitcoin Mining Is Vulnerable." In: *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 436–454. URL: http://dx.doi.org/10.1007/978-3-662-45472-5_28 (cit. on pp. 3, 25, 37).

Fernandes, E., J. Jung, and A. Prakash (2016). „Security Analysis of Emerging Smart Home Applications." In: *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 636–654 (cit. on p. 12).

Fette, I. and A. Melnikov (2011). *The WebSocket Protocol*. RFC 6455. RFC Editor, pp. 1–71. URL: https://tools.ietf.org/html/rfc6455 (cit. on p. 120).

Fleisch, Elgar (2010). „What is the internet of things? An economic perspective.“ In: *Economics, Management, and Financial Markets* 2, pp. 125–157 (cit. on p. 10).

Fleisch, Elgar, Markus Weinberger, and Felix Wortmann (2014). „Business Models and the Internet of Things.“ In: *Whitepaper of the Bosch Internet of Things and Services Lab, a Cooperation of HSG and Bosch* (cit. on p. 63).

Fleisch, Elgar, Markus Weinberger, and Felix Wortmann (2015). „Business Models and the Internet of Things (Extended Abstract).“ In: *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*. Ed. by Ivana Podnar Žarko, Krešimir Pripužić, and Martin Serrano. Cham: Springer International Publishing, pp. 6–10. URL: http://dx.doi.org/10.1007/978-3-319-16546-2%5C%7B_%7D2 (cit. on pp. 10, 110).

Fransson, Jarl (2015). *A Protocol for Microtransactions*. [Accessed 2016-07-09] (cit. on p. 87).

Frey, Remo M., Dominic Wörner, and Alexander Ilic (2016). „Collaborative Filtering on the Blockchain: A Secure Recommender System for e-Commerce.“ In: *22nd Americas Conference on Information Systems (AMCIS)*. San Diego, CA (cit. on p. viii).

Ganti, Raghu K, Fan Ye, and Hui Lei (2011). „Mobile crowdsensing: current state and future challenges.“ In: *IEEE Communications Magazine* 49.11, pp. 32–39 (cit. on pp. 63, 128).

Garcia, Flavio D, David Oswald, Timo Kasper, and Pierre Pavlidès (2016). „Lock It and Still Lose It–On the (In) Security of Automotive Remote Keyless Entry Systems.“ In: *25nd USENIX Security Symposium (USENIX Security 2016), to appear. USENIX Association* (cit. on p. 12).

Gartner (2015a). *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015*. [Accessed 2016-11-21] (cit. on p. 61).

Gartner (2015b). *Hype Cycle for Emerging Technologies*. [Accessed 2016-08-16] (cit. on p. 1).

Gencer, Adem Efe, Robbert van Renesse, and Emin Gün Sirer (2016). „Service-Oriented Sharding with Aspen.“ In: *CoRR* abs/1611.06816. URL: http://arxiv.org/abs/1611.06816 (cit. on p. 29).

Giaglis, George M. and Kalliopi N. Kypriotaki (2014). „Towards an Agenda for Information Systems Research on Digital Currencies and Bitcoin.“ In: *Business Information Systems Workshops: BIS 2014 International Workshops, Larnaca, Cyprus, May 22-23, 2014, Revised Papers*. Ed. by Witold Abramowicz and Angelika Kokkinaki. Cham: Springer International Publishing, pp. 3–13. URL: http://dx.doi.org/10.1007/978-3-319-11460-6%5C%7B_%7D1 (cit. on p. 3).

Giannotti, F., D. Pedreschi, A. Pentland, P. Lukowicz, D. Kossmann, J. Crowley, and D. Helbing (2012). „A planetary nervous system for social mining and collective awareness.“ In: *The European Physical Journal Special Topics* 214.1,

pp. 49–75. URL: http://dx.doi.org/10.1140/epjst/e2012-01688-9 (cit. on p. 14).

Giannotti, Fosca, Dino Pedreschi, Alex Pentland, Paul Lukowicz, Donald Kossmann, James Crowley, and Dirk Helbing (2012). „A planetary nervous system for social mining and collective awareness." In: *The European Physical Journal Special Topics* 214.1, pp. 49–75 (cit. on p. 64).

Gilbert, Arlo (2016). *The Time that Tony Fadell sold me a Container of Hummus*. [Accessed 2016-08-20] (cit. on p. 12).

Graeber, David (2014). *Debt-Updated and Expanded: The First 5,000 Years*. Melville House (cit. on p. 17).

Grigg, I. (2004). „The Ricardian contract." In: *Electronic Contracting, 2004. Proceedings. First IEEE International Workshop on*, pp. 25–31 (cit. on pp. 109, 135).

Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami (2013). „Internet of Things (IoT): A vision, architectural elements, and future directions." In: *Future Generation Computer Systems* 29.7, pp. 1645–1660. URL: http://www.sciencedirect.com/science/article/pii/S0167739X13000241 (cit. on p. 7).

Guinard, Dominique, Vlad Trifa, Friedemann Mattern, and Erik Wilde (2011). „From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices." In: *Architecting the Internet of Things*. Ed. by Dieter Uckelmann, Mark Harrison, and Florian Michahelles. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 97–129. URL: http://dx.doi.org/10.1007/978-3-642-19157-2_5 (cit. on p. 13).

Guo, Bin, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou (2015). „Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm." In: *ACM Computing Surveys (CSUR)* 48.1, p. 7 (cit. on p. 64).

Haderer, Nicolas, Fawaz Paraiso, Christophe Ribeiro, Philippe Merle, Romain Rouvoy, and Lionel Seinturier (2015). „Crowdsourcing: Cloud-Based Software Development." In: ed. by Wei Li, N. Michael Huhns, Wei-Tek Tsai, and Wenjun Wu. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. A Cloud-Based Infrastructure for Crowdsourcing Data from Mobile Devices, pp. 243–265. URL: http://dx.doi.org/10.1007/978-3-662-47011-4_13 (cit. on p. 64).

Hardjono, Thomas and Ned Smith (2016). „Cloud-Based Commissioning of Constrained Devices Using Permissioned Blockchains." In: *Proceedings of the 2Nd ACM International Workshop on IoT Privacy, Trust, and Security*. IoTPTS '16. Xi'an, China: ACM, pp. 29–36. URL: http://doi.acm.org/10.1145/2899007.2899012 (cit. on p. 42).

Hayes, Brian (2008). „Cloud Computing." In: *Commun. ACM* 51.7, pp. 9–11. URL: http://doi.acm.org/10.1145/1364782.1364786 (cit. on p. 1).

He, Daojing, Sammy Chan, and Mohsen Guizani (2015). „Privacy and incentive mechanisms in people-centric sensing networks.“ In: *Communications Magazine, IEEE* 53.10, pp. 200–206 (cit. on p. 64).

Hearn, Mike (2011a). *The Bitcoin Wiki: Contracts*. [Accessed 2016-11-08] (cit. on pp. 58, 121).

Hearn, Mike (2011b). *The Bitcoin Wiki: Smart Property*. [Accessed 2016-07-13] (cit. on pp. 4, 111, 139, 144).

Hearn, Mike (2013). *Anti DoS for tx replacement*. [Accessed 2016-10-11] (cit. on pp. 85, 128).

Helbing, Dirk and Evangelos Pournaras (2015). „Society: Build digital democracy.“ In: *Nature* 527, pp. 33–34 (cit. on p. 15).

Hijmans, Robert J., Susan E. Cameron, Juan L. Parra, Peter G. Jones, and Andy Jarvis (2005). „Very high resolution interpolated climate surfaces for global land areas.“ In: *International Journal of Climatology* 25.15, pp. 1965–1978. URL: http://dx.doi.org/10.1002/joc.1276 (cit. on p. 63).

Hu, X., T. H. S. Chu, H. C. B. Chan, and V. C. M. Leung (2013). „Vita: A Crowdsensing-Oriented Mobile Cyber-Physical System.“ In: *IEEE Transactions on Emerging Topics in Computing* 1.1, pp. 148–165 (cit. on p. 64).

Hughes, Nick and Susie Lonie (2007). „M-PESA: mobile money for the “unbanked” turning cellphones into 24-hour tellers in Kenya.“ In: *Innovations* 2.1-2, pp. 63–81 (cit. on p. 10).

INFSO, D (2008). „Networked Enterprise & RFID INFSO G. 2 Micro & Nanosystems, in co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future [R].“ In: *Information Society and Media, Tech. Rep* (cit. on p. 7).

Jones, Michael B., John Bradley, and Nat Sakimura (2015). *JSON Web Signatures (JWS)*. RFC 7515. RFC Editor, pp. 1–59. URL: http://www.rfc-editor.org/rfc/rfc7515.txt (cit. on p. 73).

Kalodner, Harry, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau, and Arvind Narayanan (2015). „An empirical study of Namecoin and lessons for decentralized namespace design.“ In: *Workshop on the Economics of Information Security (WEIS)*. Citeseer (cit. on p. 32).

Karame, Ghassan O., Elli Androulaki, and Srdjan Capkun (2012). „Double-spending Fast Payments in Bitcoin.“ In: *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. CCS '12. Raleigh, North Carolina, USA: ACM, pp. 906–917. URL: http://doi.acm.org/10.1145/2382196.2382292 (cit. on p. 74).

Kiayias, Aggelos, Ioannis Konstantinou, Alexander Russell, Bernardo David, and Roman Oliynykov (2016). *A Provably Secure Proof-of-Stake Blockchain Protocol*. Cryptology ePrint Archive, Report 2016/889. http://eprint.iacr.org/2016/889 (cit. on p. 31).

Kiayias, Aggelos and Giorgos Panagiotakos (2016). „On Trees, Chains and Fast Transactions in the Blockchain.“ In: (cit. on p. 28).

Köhler, Marcus, Dominic Wörner, and Felix Wortmann (2013). „Platforms for the Internet of Things – An Analysis of Existing Solutions.“ In: *5th Bosch Conf. Syst. Softw. Eng.* Technical Report (cit. on p. vii).

Kokoris-Kogias, Eleftherios, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford (2016). „Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing.“ In: *CoRR* abs/1602.06997. URL: http://arxiv.org/abs/1602.06997 (cit. on p. 29).

Koshy, Philip, Diana Koshy, and Patrick McDaniel (2014). „An Analysis of Anonymity in Bitcoin Using P2P Network Traffic.“ In: *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 469–485. URL: http://dx.doi.org/10.1007/978-3-662-45472-5_30 (cit. on p. 29).

Lamport, Leslie et al. (2001). „Paxos made simple.“ In: *ACM Sigact News* 32.4, pp. 18–25 (cit. on p. 24).

Langner, R. (2011). „Stuxnet: Dissecting a Cyberwarfare Weapon.“ In: *IEEE Security Privacy* 9.3, pp. 49–51 (cit. on p. 12).

Leiba, Barry (2012). „OAuth Web Authorization Protocol.“ English. In: *IEEE Internet Computing* 16.1. Copyright - Copyright IEEE Computer Society Jan 2012; Zuletzt aktualisiert - 2012-06-29; CODEN - IESEDJ, pp. 74–77. URL: http://search.proquest.com/docview/914302361?accountid=12492 (cit. on p. 13).

Lerner, Sergio Demian (2016). *The Well Deserved Fortune of Satoshi Nakamoto, Bitcoin creator, Visionary and Genius*. [Accessed 2016-15-12] (cit. on p. 54).

Lessig, Lawrence (2009). *Code: And other laws of cyberspace*. ReadHowYouWant.com (cit. on pp. 3, 108).

Lewenberg, Yoad, Yonatan Sompolinsky, and Aviv Zohar (2015). „Inclusive Block Chain Protocols.“ In: *Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*. Ed. by Rainer Böhme and Tatsuaki Okamoto. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 528–547. URL: http://dx.doi.org/10.1007/978-3-662-47854-7_33 (cit. on p. 28).

Lombrozo, Eric, Lau Johnson, and Pieter Wuille (2015). *BIP 141: Segregated Witness*. https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki (cit. on pp. 28, 103).

Luther, William J. and Lawrence H. White (2014). „Can Bitcoin Become a Major Currency?“ In: *SSRN Electronic Journal*. URL: http://dx.doi.org/10.2139/ssrn.2446604 (cit. on p. 30).

Luu, Loi, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor (2016). *Making Smart Contracts Smarter*. Cryptology ePrint Archive, Report 2016/633. http://eprint.iacr.org/2016/633 (cit. on p. 40).

Luu, Loi, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena (2016). „A Secure Sharding Protocol For Open Blockchains.“

In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, pp. 17–30. URL: http://doi.acm.org/10.1145/2976749.2978389 (cit. on p. 29).

Luu, Loi, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena (2015). „Demystifying Incentives in the Consensus Computer." In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, pp. 706–719. URL: http://doi.acm.org/10.1145/2810103.2813659 (cit. on p. 39).

Manyika, J, M Chui, P Bisson, J Woetzel, R Dobbs, J Bughin, and D Aharon (2015). „Unlocking the Potential of the Internet of Things." In: *McKinsey Global Institute* (cit. on pp. 1, 2, 10, 125).

Mark, Dino, Vlad Zamfir, and Emin Gün Sirer (2016). *A Call for a Temporary Moratorium on "The DAO"*. [Accessed 2016-07-21] (cit. on p. 40).

Mattern, Friedemann and Christian Floerkemeier (2010). „From the Internet of Computers to the Internet of Things." In: *From Active Data Management to Event-Based Systems and More: Papers in Honor of Alejandro Buchmann on the Occasion of His 60th Birthday*. Ed. by Kai Sachs, Ilia Petrov, and Pablo Guerrero. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 242–259. URL: http://dx.doi.org/10.1007/978-3-642-17226-7_15 (cit. on p. 7).

Maxwell, Greg (2013). *CoinJoin: Bitcoin privacy for the real world*. [Accessed 2016-08-01] (cit. on p. 30).

Maxwell, Gregory (2015). *Confidential Transactions*. [Accessed 2016-10-18] (cit. on p. 30).

Maxwell, Gregory (2016). *The first successful Zero-Knowledge Contingent Payment*. [Accessed 2016-11-01] (cit. on pp. 58, 67).

Maymounkov, Petar and David Mazières (2002). „Kademlia: A Peer-to-Peer Information System Based on the XOR Metric." In: *Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers*. Ed. by Peter Druschel, Frans Kaashoek, and Antony Rowstron. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 53–65. URL: http://dx.doi.org/10.1007/3-540-45748-8_5 (cit. on p. 14).

McCorry, Patrick, Malte Möser, Siamak F. Shahandashti, and Feng Hao (2016). *Towards Bitcoin Payment Networks*. Cryptology ePrint Archive, Report 2016/408. http://eprint.iacr.org/2016/408 (cit. on pp. 103, 131).

Meiklejohn, Sarah and Claudio Orlandi (2015). „Privacy-Enhancing Overlays in Bitcoin." In: *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Ed. by Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 127–141. URL: http://dx.doi.org/10.1007/978-3-662-48051-9_10 (cit. on p. 30).

Merkle, Ralph C (1980). „Protocols for Public Key Cryptosystems." In: *IEEE Symposium on Security and privacy*. Vol. 122 (cit. on p. 26).

Merlino, Giovanni, Stamatis Arkoulis, Salvatore Distefano, Chrysa Papagianni, Antonio Puliafito, and Symeon Papavassiliou (2016). „Mobile CrowdSensing as a Service: a platform for applications on top of Sensing Clouds." In: *Future Generation Computer Systems* 56, pp. 623–639 (cit. on p. 64).

Metcalfe, Bob (2013). „Metcalfe's Law after 40 Years of Ethernet." In: *Computer* 46.12, pp. 26–31 (cit. on p. 3).

Miers, I., C. Garman, M. Green, and A. D. Rubin (2013). „Zerocoin: Anonymous Distributed E-Cash from Bitcoin." In: *Security and Privacy (SP), 2013 IEEE Symposium on*, pp. 397–411 (cit. on p. 30).

Miller, Andrew, Ahmed Kosba, Jonathan Katz, and Elaine Shi (2015). „Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions." In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, pp. 680–691. URL: http://doi.acm.org/10.1145/2810103.2813621 (cit. on p. 26).

Mineraud, Julien, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma (2016). „A gap analysis of Internet-of-Things platforms." In: *COMPUTER COMMUNICATIONS* 89-90, 5–16 (cit. on pp. 10, 13).

Mohan, Prashanth, Venkata N. Padmanabhan, and Ramachandran Ramjee (2008). „Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones." In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. SenSys '08. Raleigh, NC, USA: ACM, pp. 323–336. URL: http://doi.acm.org/10.1145/1460412.1460444 (cit. on p. 64).

Montjoye, Yves-Alexandre de, César A Hidalgo, Michel Verleysen, and Vincent D Blondel (2013). „Unique in the Crowd: The privacy bounds of human mobility." In: *Nature srep.* 3 (cit. on p. 100).

Montjoye, Yves-Alexandre de, Erez Shmueli, Samuel S. Wang, and Alex Sandy Pentland (2014). „openPDS: Protecting the Privacy of Metadata through SafeAnswers." In: *PLoS ONE* 9.7, pp. 1–9. URL: http://dx.doi.org/10.1371%2Fjournal.pone.0098790 (cit. on p. 13).

Möser, Malte, Ittay Eyal, and Emin Gün Sirer (2016). „Bitcoin Covenants." In: ed. by Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, pp. 126–141. URL: http://dx.doi.org/10.1007/978-3-662-53357-4_9 (cit. on p. 30).

Mougayar, William (2015). *The Global Landscape of Blockchain Companies in Financial Services*. [Accessed 2015-10-08] (cit. on p. 46).

Mougayar, William (2016). *State of Global Blockchain Consortia with Interactive Map*. [Accessed 2016-12-11] (cit. on p. 46).

Nakamoto, Satoshi (2008). „Bitcoin: A peer-to-peer electronic cash system." In: p. 28 (cit. on p. 2).

Narayanan, Arvind (2015a). *Bitcoin faces a crossroads, needs an effective decision-making process*. [Accessed 2016-12-15] (cit. on pp. 58, 131).

Narayanan, Arvind (2015b). *"Private blockchain" is just a confusing name for a shared database*. [Accessed 2017-01-04] (cit. on p. 43).

Narayanan, Arvind and Andrew Miller (2015). *Bitcoin faces a crossroads, needs an effective decision-making process*. [Accessed 2016-08-18] (cit. on p. 31).

Narayanan, Arvind, Vincent Toubiana, Solon Barocas, Helen Nissenbaum, and Dan Boneh (2012). „A Critical Look at Decentralized Personal Data Architectures." In: *CoRR* abs/1202.4503. URL: http://arxiv.org/abs/1202.4503 (cit. on p. 13).

Ng, IC (2014). „Engineering a Market for Personal Data: The Hub-of-all-Things (HAT), A Briefing Paper." In: *WMG Service Systems Research Group Working Paper Series* (cit. on p. 13).

Noyen, Kay, Dirk Volland, Dominic Wörner, and Elgar Fleisch (2014). „When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin." In: *CoRR* abs/1409.5841. URL: http://arxiv.org/abs/1409.5841 (cit. on pp. vii, 6, 80).

Ober, Micha, Stefan Katzenbeisser, and Kay Hamacher (2013). „Structure and Anonymity of the Bitcoin Transaction Graph." In: *Future Internet* 5.2, p. 237. URL: http://www.mdpi.com/1999-5903/5/2/237 (cit. on p. 29).

O'Dwyer, K. J. and D. Malone (2014). „Bitcoin mining and its energy footprint." In: *Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET*, pp. 280–285 (cit. on p. 31).

Overview of Colored Coins (2012). [Accessed 2016-07-15] (cit. on p. 32).

Pass, Rafael and abhi shelat abhi (2015). „Micropayments for Decentralized Currencies." In: *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. Denver, Colorado, USA: ACM, pp. 207–218. URL: http://doi.acm.org/10.1145/2810103.2813713 (cit. on p. 88).

Pentland, Alex (2009). „Reality mining of mobile communications: Toward a new deal on data." In: *The Global Information Technology Report 2008–2009*, p. 1981 (cit. on p. 61).

Pentland, Alex (2013). „The Data-Driven Society." In: *Scientific American* 309.4, pp. 78–83 (cit. on p. 12).

Perera, Charith, Arkady B Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos (2014). „Sensing as a service model for smart cities supported by Internet of Things." In: *Trans. Emerging Telecommunications Technologies ()* 25.1, pp. 81–93 (cit. on p. 63).

Poelstra, Andrew (2014). *Distributed Consensus from Proof of Stake is Impossible*. [Accessed 2016-10-20] (cit. on p. 31).

Poon, Joseph and Thaddeus Dryja (2015). „The Bitcoin Lightning Network." In: (cit. on pp. 89, 102, 103, 127, 129, 131).

Popov, Serguei (2016). „The Tangle." In: (cit. on p. 131).

Pournaras, E., I. Moise, and D. Helbing (2015). „Privacy-Preserving Ubiquitous Social Mining via Modular and Compositional Virtual Sensors." In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, pp. 332–338 (cit. on p. 14).

Poushter, Jacob (2016). „Smartphone ownership and Internet usage continues to climb in emerging economies." In: *Pew Research Center* (cit. on p. 113).

Pureswaran, Veena and Robin Lougee (2015). *The Economy of Things*. IBM Institute of Business Value. https://www-935.ibm.com/services/us/gbs/thoughtleadership/economyofthings/ (cit. on pp. 15, 112).

Reed, David P (1999). „That sneaky exponential—Beyond Metcalfe's law to the power of community building." In: *Context magazine* 2.1 (cit. on p. 3).

Reid, Fergal and Martin Harrigan (2013). „An Analysis of Anonymity in the Bitcoin System." In: *Security and Privacy in Social Networks*. Ed. by Yaniv Altshuler, Yuval Elovici, B. Armin Cremers, Nadav Aharony, and Alex Pentland. New York, NY: Springer New York, pp. 197–223. URL: http://dx.doi.org/10.1007/978-1-4614-4139-7_10 (cit. on p. 29).

Rifà-Pous, Helena and Jordi Herrera-Joancomartí (2011). „Computational and Energy Costs of Cryptographic Algorithms on Handheld Devices." In: *Future Internet* 3.1, p. 31. URL: http://www.mdpi.com/1999-5903/3/1/31 (cit. on p. 78).

Rivest, Ronald L (1997). „Electronic lottery tickets as micropayments." In: *International Conference on Financial Cryptography*. Springer, pp. 307–314 (cit. on p. 88).

Rivest, Ronald L and Adi Shamir (1996). „PayWord and MicroMint: Two simple micropayment schemes." In: *International Workshop on Security Protocols*. Springer, pp. 69–87 (cit. on p. 88).

Rodrigues, Rodrigo and Peter Druschel (2010). „Peer-to-peer Systems." In: *Commun. ACM* 53.10, pp. 72–82. URL: http://doi.acm.org/10.1145/1831407.1831427 (cit. on pp. 2, 14).

Ron, Dorit and Adi Shamir (2013). „Quantitative Analysis of the Full Bitcoin Transaction Graph." In: *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*. Ed. by Ahmad-Reza Sadeghi. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 6–24. URL: http://dx.doi.org/10.1007/978-3-642-39884-1_2 (cit. on p. 29).

Ruffing, Tim, Pedro Moreno-Sanchez, and Aniket Kate (2014). „CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin." In: *Computer Security - ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II*. Ed. by Mirosław Kutyłowski and Jaideep Vaidya. Cham: Springer International Publishing, pp. 345–364. URL: http://dx.doi.org/10.1007/978-3-319-11212-1_20 (cit. on p. 30).

Saltzer, J. H., D. P. Reed, and D. D. Clark (1984). „End-to-end Arguments in System Design." In: *ACM Trans. Comput. Syst.* 2.4, pp. 277–288. URL: http://doi.acm.org/10.1145/357401.357402 (cit. on p. 41).

Sanford J. Grossman, Oliver D. Hart (1980). „Takeover Bids, The Free-Rider Problem, and the Theory of the Corporation." In: *The Bell Journal of*

Economics 11.1, pp. 42–64. URL: http://www.jstor.org/stable/3003400 (cit. on p. 2).

Sapuric, Svetlana and Angelika Kokkinaki (2014). „Bitcoin Is Volatile! Isn't that Right?" In: *Business Information Systems Workshops: BIS 2014 International Workshops, Larnaca, Cyprus, May 22-23, 2014, Revised Papers*. Ed. by Witold Abramowicz and Angelika Kokkinaki. Cham: Springer International Publishing, pp. 255–265. URL: http://dx.doi.org/10.1007/978-3-319-11460-6_22 (cit. on p. 30).

Schwab, Klaus, Alan Marcus, JO Oyola, William Hoffman, and M Luzi (2011). „Personal data: The emergence of a new asset class." In: *An Initiative of the World Economic Forum* (cit. on p. 1).

Shelby, Z, K Hartke, and C Bormann (2014). *The Constrained Application Protocol (CoAP)(RFC 7252)* (cit. on p. 104).

Simon, Herbert Alexander (1982). *Models of bounded rationality: Empirically grounded economic reason*. Vol. 3. MIT press (cit. on p. 2).

Sippey, Michael (2012). *Changes coming in Version 1.1 of the Twitter API*. [Accessed 2014-04-02] (cit. on p. 65).

Smith, Adam and Joseph Shield Nicholson (1887). *An Inquiry Into the Nature and Causes of the Wealth of Nations...* T. Nelson and Sons (cit. on p. 2).

Sompolinsky, Yonatan and Aviv Zohar (2013). *Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains*. Cryptology ePrint Archive, Report 2013/881. http://eprint.iacr.org/2013/881 (cit. on p. 37).

Sompolinsky, Yonatan and Aviv Zohar (2015). „Secure High-Rate Transaction Processing in Bitcoin." In: *Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers*. Ed. by Rainer Böhme and Tatsuaki Okamoto. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 507–527. URL: http://dx.doi.org/10.1007/978-3-662-47854-7_32 (cit. on p. 29).

Spagnuolo, Michele, Federico Maggi, and Stefano Zanero (2014). „BitIodine: Extracting Intelligence from the Bitcoin Network." In: *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 457–468. URL: http://dx.doi.org/10.1007/978-3-662-45472-5_29 (cit. on p. 29).

Spilman, Jeremy (2013). *Anti DoS for tx replacement*. [Accessed 2016-10-11] (cit. on pp. 85, 128).

Srinivasan, Balaji (2015). *A Bitcoin miner in every device and in every hand*. [Accessed 2015-10-08] (cit. on pp. 129, 136).

Sundararajan, Arun (2016). *The Sharing Economy: The End of Employment and the Rise of Crowd-Based Capitalism*. MIT Press (cit. on p. 4).

Swanson, Tim (2015a). *Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems* (cit. on p. 43).

Swanson, Tim (2015b). *Watermarked tokens and pseudonymity on public blockchains*. [Accessed 2016-07-11] (cit. on p. 33).

Szabo, Nick (1997). *The idea of smart contracts*. [Accessed 2016-08-01] (cit. on pp. 4, 38, 109).

Szabo, Nick (1999). „Micropayments and mental transaction costs." In: *2nd Berlin Internet Economics Workshop* (cit. on p. 59).

Szabo, Nick (2005). *Trusted Third Parties Are Security Holes*. [Accessed 2016-08-01] (cit. on p. 2).

Taylor, Michael Bedford (2013). „Bitcoin and the age of bespoke silicon." In: *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. IEEE Press, p. 16 (cit. on p. 26).

Thiagarajan, Arvind, James Biagioni, Tomas Gerlich, and Jakob Eriksson (2010). „Cooperative Transit Tracking Using Smart-phones." In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. SenSys '10. Zürich, Switzerland: ACM, pp. 85–98. URL: http://doi.acm.org/10.1145/1869983.1869993 (cit. on p. 64).

Thomas Hardjono, Sandy Pentland (2016). *On Privacy-Preserving Identity within Future Blockchain Systems*. [Accessed 2016-07-26] (cit. on p. 42).

Todd, Peter (2014). *BIP 65: OP_CHECKLOCKTIMEVERIFY*. https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki (cit. on p. 98).

Tschofenig, Hannes and Manuel Pegourie-Gonnard (2015). *Performance of State-of-the-Art Cryptography on ARM-based Microprocessors*. [Accessed 2016-11-06] (cit. on p. 79).

Valenta, Luke and Brendan Rowan (2015). „Blindcoin: Blinded, Accountable Mixes for Bitcoin." In: *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Ed. by Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 112–126. URL: http://dx.doi.org/10.1007/978-3-662-48051-9_9 (cit. on p. 30).

Vaquero, Luis M. and Luis Rodero-Merino (2014). „Finding your Way in the Definition Fog: Towards a Comprehensive of Fog Computing." In: *ACM SIGCOMM COMPUTER COMMUNICATION REVIEW* 44.5, 27–32 (cit. on p. 14).

Vaquero, Luis M., Luis Rodero-Merino, Juan Caceres, and Maik Lindner (2008). „A Break in the Clouds: Towards a Cloud Definition." In: *SIGCOMM Comput. Commun. Rev.* 39.1, pp. 50–55. URL: http://doi.acm.org/10.1145/1496091.1496100 (cit. on p. 9).

Visa. *Visa Inc. at a Glance*. [Accessed 2016-07-10] (cit. on p. 28).

Von Bomhard, Thomas and Dominic Wörner (2016). „Design and Evaluation of a "Smart Individual-room Heating IS" to Improve Comfort And Save Energy in Residential Buildings." In: *European Conference on Information Systems (ECIS)*. Istanbul, Turkey (cit. on p. viii).

Want, R. (2006). „An introduction to RFID technology.“ In: *IEEE Pervasive Computing* 5.1, pp. 25–33 (cit. on p. 8).

Warren, Jonathan (2012). *Bitmessage: A peer-to-peer message authentication and delivery system*. [Accessed 2014-04-02] (cit. on p. 104).

Weather Underground, Inc. *Weather Underground: Weather Forecast and Reports*. [Accessed 2014-04-02] (cit. on p. 63).

Weiser, Mark (1991). „The Computer for the 21st Century.“ In: *Scientific American* 265.9, pp. 66–75 (cit. on pp. 1, 7).

Wheeler, David (1996). „Transactions using bets.“ In: *International Workshop on Security Protocols*. Springer, pp. 89–92 (cit. on p. 88).

Wikipedia (2016). *Digital Asset*. [Accessed 2015-10-08] (cit. on p. 47).

Wood, Gavin (2014). *Ethereum: A secure decentralised generalised transaction ledger*. [Accessed 2016-07-16] (cit. on p. 35).

World Bank (2013). *Remittance Prices Worldwide*. [Accessed 2014-04-02] (cit. on p. 66).

Wörner, Dominic (2016). „Design of a Real-Time Data Market Based on the 21 Bitcoin Computer.“ In: *Tackling Society's Grand Challenges with Design Science: 11th International Conference, DESRIST 2016, St. John's, NL, Canada, May 23-25, 2016, Proceedings*, pp. 228–232. URL: https://www.springer.com/us/book/9783319392936 (cit. on pp. vii, 80).

Wörner, Dominic and Thomas von Bomhard (2014). „When Your Sensor Earns Money: Exchanging Data for Cash with Bitcoin.“ In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. UbiComp '14 Adjunct. Seattle, Washington: ACM, pp. 295–298. URL: http://doi.acm.org/10.1145/2638728.2638786 (cit. on pp. vii, 6, 80).

Wörner, Dominic, Thomas von Bomhard, Marc Röschlin, and Felix Wortmann (2014). „Look Twice: Uncover Hidden Information in Room Climate Sensor Data.“ In: *International Conference on the Internet of Things (IOT)*. Cambridge, MA, pp. 25–30 (cit. on p. vii).

Wörner, Dominic, Thomas von Bomhard, and Felix Wortmann (2013). „Occupancy-based Heating Control for Residential Buildings Using Environmental Sensors.“ In: *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. BuildSys'13. Roma, Italy: ACM, 28:1–28:2. URL: http://doi.acm.org/10.1145/2528282.2528313 (cit. on p. vii).

Wörner, Dominic, Thomas Von Bomhard, Yan-Peter Schreier, and Dominik Bilgeri (2016). „The Bitcoin Ecosystem: Disruption Beyond Financial Services?“ In: *European Conference on Information Systems (ECIS)*. Istanbul, Turkey (cit. on pp. vii, 6, 46, 133).

Wuille, Pieter (2012). *BIP 32: Hierarchical Deterministic Wallets*. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki (cit. on p. 98).

Xu, L. D., W. He, and S. Li (2014). „Internet of Things in Industries: A Survey.“ In: *IEEE Transactions on Industrial Informatics* 10.4, pp. 2233–2243 (cit. on p. 8).

Yang, D., G. Xue, G. Fang, and J. Tang (2015). „Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones.“ In: *Networking, IEEE/ACM Transactions on* PP.99, pp. 1–13 (cit. on p. 64).

Yi, Shanhe, Cheng Li, and Qun Li (2015). „A Survey of Fog Computing: Concepts, Applications and Issues.“ In: *Proceedings of the 2015 Workshop on Mobile Big Data*. Mobidata '15. Hangzhou, China: ACM, pp. 37–42. URL: http://doi.acm.org/10.1145/2757384.2757397 (cit. on pp. 2, 14).

Zhang, Ben, Nitesh Mor, John Kolb, Douglas S. Chan, Ken Lutz, Eric Allman, John Wawrzynek, Edward Lee, and John Kubiatowicz (2015). „The Cloud is Not Enough: Saving IoT from the Cloud.“ In: *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*. Santa Clara, CA: USENIX Association. URL: https://www.usenix.org/conference/hotcloud15/workshop-program/presentation/zhang (cit. on p. 14).

Zhang, Y. and J. Wen (2015). „An IoT electric business model based on the protocol of bitcoin.“ In: *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pp. 184–191 (cit. on p. 81).

Zyskind, G., O. Nathan, and A. Pentland (2015). „Decentralizing Privacy: Using Blockchain to Protect Personal Data.“ In: *Security and Privacy Workshops (SPW), 2015 IEEE*, pp. 180–184 (cit. on p. 13).

ACRONYMS

ACA	Address Certification Authority
AML	Anti-Money Laundering
API	Application Programming Interface
ASIC	Application-specific Integrated Circuit
BIP	Bitcoin Improvement Proposal
CoAP	Constrained Application Protocol
DAG	Directed Acyclic Graph
DCS	Data Collection Server
DNS	Domain Name Service
DHT	Distributed Hash Table
DoS	Denial of Service
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
EVM	Ethereum Virtual Machine
FPGA	Field-Programmable Gate Array
GHOST	Greedy Heaviest Observed Subtree
GPU	Graphics Processing Unit
HTLC	Hashed Timelock Contract
HTTP	Hypertext Transfer Protocol
ICO	Initial Coin Offering
IOT	Internet of Things
JSON	Java Script Object Notation
KYC	Know Your Customer

NAT	Network Address Translation
OTC	Over-The-Counter
P2SH	Pay-to-Script-Hash
P2PK	Pay-to-PubKey
P2PKH	Pay-to-PubKey-Hash
P2P	Peer-to-Peer
PBFT	Practical Byzantine Fault Tolerance
PGP	Pretty Good Privacy
REST	Representational State Transfer
RFID	Radio-Frequency-Identification
RPC	Remote Procedure Call
SDN	Software Defined Network
SiP	Systems-in-Package
SoC	Systems-on-Chip
SPV	Simplified Payment Verification
SQL	Structured Query Language
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTP	Trusted Third Party
UTXO	Unspent Transaction Output
WSN	Wireless Sensor Networks
XMPP	Extensible Messaging and Presence Protocol
ZKCP	Zero Knowledge Contingent Payments

COLOPHON

This document was typeset in LaTeX using the typographical look-and-feel `classicthesis`. The bibliography is typeset using `biblatex`.