

# Universität Mannheim

Fakultät für Mathematik und Informatik  
Lehrstuhl für Softwaretechnik – Prof. Dr. Colin ATKINSON

Arbeit zur Erlangung eines  
Diploms an der Universität Mannheim  
im Fach Wirtschaftsinformatik

Betreuer: Prof. Dr. Colin ATKINSON

A Software Engineering Perspective on Cryptocurrencies

Nicolas HOFER

<nicolas.hofer@gmx.de>

Mannheim, den 2. Oktober 2014



---

## Abstract

This work evaluates a **digital cash** phenomenon, called **cryptocurrencies** that was started in early 2009 and has created a buzz at the latest in the year 2013, by the virtual explosion of prices in **currencies** (e.g. prices in EUR or USD) for certain **cryptocurrencies**, especially for **bitcoin**. The evaluation is realized from the perspective of a software engineer. The central research question is derived of a claim made by the **Bitcoin** community: **Bitcoin** does create “a new kind of **money**”. The potential implications for existing **payment systems** would be tremendous if this proves to be substantial. This issue is examined by performing a requirements analysis for **payment systems** in general, thereby establishing certain definitions and understandings of terms, including ‘**money**’. An analysis and evaluation is also done for the **Banking Payment System (BPS)** and the **Cryptocurrency Payment System (CPS)** with the target to be able to draw conclusions for the **CPS** by comparing the two payment systems. Furthermore a detailed evaluation of **Bitcoin** (first letter capitalized!) and its proposed **money** called ‘**bitcoin**’ (non-capitalized!) is carried out by reverse engineering parts of the software project using modeling technology that is known in model driven software development.

---

## Preface

For reasons of the main research question and the research approach, this work is interconnecting multiple different fields of research:

- Cryptography
- Cryptocurrencies
- Monetary theory and banking
- Software engineering

Some of these fields of research are closely interrelated, but that does not make up for the fact that any reader of this work is going to be challenged by the multiple different terminologies that ‘natives’ of each one of these fields readily utilize. In any case it was not deemed reasonable to just assume that every reader of this work is going to be proficient in each one of these different fields’ terminologies. For this reason a glossary was created for this work, even though the ‘natives’ of each of the fields will probably consider ‘their’ terms as trivial and might wonder about their explanatory inclusion in a glossary in a work such as the one at hand.

**Glossary** I hope that readers are going to make great use of the [glossary](#), especially if terms are hit in the work that are not part of the reader’s day-to-day language. Sometimes terms are used in a very specific way in this work, potentially different from what would be expected from an intuitive stance. The most use of the glossary can certainly be made when this work is read in the portable document format (pdf), since the glossary terms are hyper-linked within the document. Two terms I want to specifically mention here: [Bitcoin](#) and [bitcoin](#). While being separated by capitalization only, they have completely different meanings and are used almost all throughout this work. [Bitcoin](#) is standing for the whole project and [bitcoin](#) is used as the term for the [money of account](#) and the [money proper](#) of the [Bitcoin](#) project.

**References** Due to my aesthetic preference and unlike many publications within natural sciences, the bibliography style of this document is set up in a way that citations are consisting only of the last name of the (first) author of any publication and the year of publication. This made it possible to include references to specific pages in reference works in a way I feel aesthetically comfortable with.

Unpublished references were put into the bibliography only if they did provide a distinct date of creation and a specific author. Other references, e.g. wiki pages, are referenced by URL within a footnote only.

**Acknowledgements** I am so thankful for your unconditional love and support, Carolin. This work would just not exist, if it was not for you. Thank you, I love you!

I want to thank Colin for his valuable inputs on the software engineering perspective. Without Colin’s input I would probably have never taken the approach to look at the research issue at hand from the perspective of a software engineer. Cheers, Colin!

I thank my brother Kai-Simon in helping me with proof reading the work at hand. Thank you!

I want to thank Wolfgang and the community at <http://www.dasgelbeforum.net> (DGF), who have greatly inspired my perspective on monetary theory in extensive discussions over many years. I’m especially thankful for all the personal exchange with Wolfgang. Thank you DGF and thank you Wolfgang in particular!

---

I thank Vera for her support in offering me quiet, cool working space when it was most needed. Thank you!

I thank everybody who has supported me consciously and unconsciously in the completion of this work. This includes the German taxpayers and all of their direct and indirect employees providing me with what, I guess, is called an 'education'. I thank all of you.

I am very thankful for the fact that I was able to do my studies the way I did, being a student antecedent to the so-called 'Bologna process' that I personally am rather critical about. So, I guess, I have to be thankful for the year of my birth being early enough. Thank you.

Last, but not least, I thank the creators of the free software and tools, such as TexStudio, UMLet and BibDesk, that helped immensely in the creation of this work. Thanks guys!

## Contents

<b>Abstract</b>	<b>ii</b>
<b>Preface</b>	<b>iv</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>x</b>
<b>Glossary</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation	1
1.2 Research objective	1
1.3 Research approach	1
1.4 Structure	2
<b>2 Cryptocurrencies - a quick overview</b>	<b>3</b>
2.1 Digital cash	3
2.2 What is a cryptocurrency?	4
<b>3 Cryptography</b>	<b>7</b>
3.1 Symmetric cryptography	7
3.2 Asymmetric cryptography - public key cryptography	7
3.3 Cryptographic hashing	8
3.4 Digital signature algorithms	9
<b>4 Software engineering</b>	<b>10</b>
4.1 Requirements and Design	10
4.2 Software engineering models and methods	12
4.2.1 Modeling	13
4.2.2 Software engineering methods	13
4.2.3 Modeling tools used in this work	13
<b>5 Currency, money, payment systems</b>	<b>14</b>
5.1 Money	14
5.2 Currency	14
5.3 Payment Systems	16
5.4 Synopsis on ‘money’	17
<b>6 Requirements analysis</b>	<b>19</b>
6.1 Basic functions of a payment system	19
6.1.1 Users of a payment system	20
6.1.2 Use case: Paying with money	20
6.1.3 Use case: Pricing in money	22
6.1.4 Use case: Holding on to money	23
6.2 Medium of exchange	24
6.2.1 Barter exchange	24
6.2.2 Money - facilitating barter trades	24

6.2.3	Lifting the veil of barter	25
6.2.4	Final settlement of debt	25
6.2.5	Money proper	25
6.3	Unit of account	25
6.3.1	Pricing is voluntary	26
6.3.2	Choosing a unit of account	26
6.3.3	Potential motivations to choose a unit	26
6.3.4	Specification of contracts	27
6.3.5	Credit agreements	27
6.3.6	Discovering the Money Meta Infrastructure	28
6.3.7	Money of account	28
6.4	Store of value	28
6.4.1	Funding liquidity	28
6.4.2	Market liquidity	28
6.4.3	No new requirements	29
6.5	Hierarchy of requirements	29
6.5.1	Pricing first, then payments	29
6.5.2	Hoarding the money proper	29
6.5.3	Conclusion on the hierarchy	29
6.5.4	Relevance for Bitcoin	30
6.6	Money Meta Infrastructure	30
6.6.1	Central hypothesis	30
6.6.2	Credit agreements	31
6.6.3	Interim Conclusion	33
6.6.4	Relevance for Bitcoin	34
6.7	Synopsis	34
<b>7</b>	<b>Bitcoin</b>	<b>37</b>
7.1	Introducing the ‘coin’	37
7.2	Fundamentals	40
7.2.1	Clients, nodes and wallets	40
7.2.2	Transactions, addresses and transaction points	40
7.2.3	Blocks and the blockchain	43
7.2.4	Mining	45
7.2.5	What a bitcoin is eventually	48
7.3	Cryptocurrency Payment System	50
7.3.1	Use case diagrams	50
7.3.2	Users	53
7.3.3	Use Case: Paying with bitcoin	54
7.3.4	Use Case: Pricing in bitcoin	57
7.3.5	Use Case: Holding on to bitcoin	58
7.4	Bitcoin EcoSystem	59
7.4.1	Including the EcoSystem	59
7.4.2	Use case: ‘Trading in currencies’	60
7.4.3	Users	61
7.4.4	One nested use case diagram to include it all	61
7.4.5	Specific risks	62
7.5	Beyond a payment system	65
7.5.1	Some preliminary use cases	65
7.5.2	Resembling (parts of) the MMI	66
7.6	MMI for a CPS	67

---

<b>8</b>	<b>The money view on actuality</b>	<b>68</b>
8.1	A brief distinction . . . . .	68
8.1.1	Conventional view on money . . . . .	68
8.1.2	Applicability of conventional view . . . . .	69
8.2	Fundamentals of the money view . . . . .	69
8.2.1	Swap of IOUs . . . . .	69
8.2.2	Inside money and outside money . . . . .	72
8.2.3	Money or credit . . . . .	72
8.3	Banking Payment System . . . . .	74
8.3.1	How banks create inside money . . . . .	75
8.3.2	Usage and acceptance of bankmoney . . . . .	75
<b>9</b>	<b>Conclusion</b>	<b>77</b>
9.1	Recapitulation . . . . .	77
9.2	On Bitcoin . . . . .	78
<b>10</b>	<b>Future research</b>	<b>81</b>
	<b>Bibliography</b>	<b>xviii</b>
	<b>Ehrenwörtliche Erklärung</b>	<b>xxi</b>



## List of Figures

1	Object diagram: Requirements . . . . .	12
2	Object diagram: Creditor pays out loan . . . . .	16
3	Object diagram: creditor-debtor relationship . . . . .	16
4	Object diagram: Debtor pays Creditor . . . . .	17
5	Use case diagram: Functional Requirements . . . . .	19
6	Object diagram: Users of a payment system . . . . .	20
7	Domain model: Hierarchy on Requirements . . . . .	30
8	Activity Diagram: Credit agreement . . . . .	32
9	Initial use case diagram: Functional Requirements for a Payment System . . . . .	34
10	Use case diagram: Including detail on money proper and money of account . . . . .	34
11	Use case diagram: Including detail on the hierarchy of requirements . . . . .	35
12	Use case diagram: Including the Money Meta Infrastructure (MMI) . . . . .	36
13	A coin: a chain of digital signatures . . . . .	38
14	Communications diagram: single-input/single-output transaction . . . . .	42
15	Class diagram: Blockchain, Blocks, Transactions . . . . .	43
16	Class diagram: Blockchain in more detail . . . . .	43
17	Object Model: Mining . . . . .	46
18	Coinbase transaction . . . . .	47
19	Use case diagram: Bitcoin used as a payment system . . . . .	50
20	Use case diagram: Bitcoin as CPS, including the hierarchy . . . . .	51
21	Nested systems diagram: Bitcoin as CPS enclosed by Network . . . . .	52
22	Nested systems diagram: Bitcoin as CPS enclosed by Network including MMI . . . . .	53
23	Nested systems diagram: Bitcoin Network including hierarchy and MMI . . . . .	54
24	Object Model: payment transaction . . . . .	56
25	Communications diagram: multi-input/output transaction . . . . .	57
26	Nested systems diagram: the EcoSystem . . . . .	60
27	Object diagram: Users of a CPS . . . . .	61
28	Nested systems diagram . . . . .	63
29	Diagram on Bitcoin use cases that go beyond Bitcoin as a payment system . . . . .	65
30	Class diagram showing a creditor-debtor relationship . . . . .	70
31	Use case diagram on the BPS . . . . .	74

**List of Tables**

1	Alice’s and Bob’s stylized balance sheets: Creditclaim . . . . .	70
2	Alice’s and Bob’s balance sheets: swap of IOUs . . . . .	71
3	Bank and customer balance sheet: credit agreement . . . . .	75
4	Bank and customer balance sheet: payment . . . . .	75
5	Bitcoin creation as outside money . . . . .	78

## Glossary

### address

is the **cryptographic hash** of the public key of a private/public key-pair; the **address** conceptually 'holds' **bitcoins** by providing access to the **bitcoins** at a **transaction point**. **xi**, **xvi**, **xvii**, **9**, **41**, **54**, **55**, **59**

### ASIC

application-specific integrated circuit. **61**

### asset

is a thing or a right that is priced at the current valuation of expected cash flows (e.g. generated income or potential sale). **xiii–xv**, **5**, **11**, **15**, **20**, **24–30**, **34**, **44**, **51–53**, **58**, **60**, **70–72**, **79**

### BaFin

Bundesanstalt für Finanzdienstleistungsaufsicht. **78**

### bankmoney

is a term used for the **money proper** created by commercial banks by expanding their balance sheets in an exchange of IOUs with their debtors. **18**, **48**, **60**, **61**, **68**, **69**, **71–76**, **79**, **82**

### Bitcoin

the **cryptocurrency** project that calls its **money proper** and its **money of account** "bitcoin"; in this work the **Bitcoin** project serves as the example for **cryptocurrencies** in general. **iii**, **iv**, **ix**, **xi**, **xiii**, **xvi**, **1–7**, **9**, **12**, **14**, **19**, **27**, **30**, **34**, **35**, **37**, **39**, **40**, **42–45**, **47–54**, **56**, **58–62**, **64**, **65**, **67–69**, **74**, **77–79**

### bitcoin

refers to the **money proper** and to the **money of account** of Bitcoin. **iii**, **iv**, **xi–xiii**, **xvi**, **xvii**, **2**, **4**, **5**, **9**, **14**, **26**, **30**, **34**, **35**, **37**, **40–48**, **50–62**, **64**, **65**, **68**, **69**, **74**, **76**, **78**, **79**

### block

is a file that contains a link to the previous **block** and holds a set of **transactions** that have not been included in any preceding **block** in the **blockchain**. **xi–xvi**, **9**, **39**, **42–48**, **57**, **60**, **61**

### blockchain

is a collection of **blocks** that are each linked to exactly one other **block**; the current **blockchain** contains all **blocks**, beginning with the **genesis block** leading to the most recent generated **block**. **xi–xiii**, **xv**, **4**, **39**, **42–48**, **54**, **57**, **59**, **65**

### BPS

Banking **Payment System**. **iii**, **ix**, **1**, **2**, **5**, **10–12**, **19**, **35**, **37**, **49**, **53**, **61**, **62**, **64**, **66**, **68**, **69**, **71**, **74**, **76**, **78**, **79**

### buyer

is in this work specifically an actor that is buying **bitcoins** for **currency** on exchanges that are part of the **Bitcoin EcoSystem**. **60**, **61**

### ciphertext

is encrypted information. **Plaintext** becomes **ciphertext** by encryption. By decryption **ciphertext** becomes **plaintext** again. **xi**, **7**

**client**

is an application that implements all the functions necessary to be able to operate as a **node** within the **Network**. [xv](#), [xvii](#), [4](#), [40](#), [41](#), [47](#), [55](#)

**coinbase**

is a name for the first output **transaction** in a **generated block** that actually does create brand new **bitcoins**, which are given to the **miner** as a reward for successfully **hashing** the **block**. [44](#), [46–48](#)

**CPS**

**Cryptocurrency Payment System**. [iii](#), [ix](#), [1](#), [2](#), [5](#), [10–12](#), [19](#), [30](#), [35](#), [37](#), [44](#), [49–56](#), [61](#), [62](#), [64](#), [65](#), [67](#), [69](#), [72](#), [74](#), [78](#)

**cryptocurrency**

is a recent (since 2009) type of **digital cash** scheme that does make use of **cryptography** to prevent its users from **double-spending** the digital **money proper**, employing a public ledger file called **blockchain**. [iii](#), [xi–xiii](#), [xv](#), [xvi](#), [1–5](#), [7–10](#), [13–15](#), [19](#), [26–28](#), [37–41](#), [50](#), [54](#), [58](#), [62](#), [64](#), [66–69](#), [73](#), [74](#), [76–82](#)

**cryptographic hash function**

is a **hash function** that is considered to be practically impossible to invert. Meaning there is no practical way to recreate the original data from the **message digest**. [8](#), [9](#)

**cryptographic protocol**

is a protocol that defines what has to be done by what agent and in what sequence to reach a certain goal; in the **cryptocurrency** context it defines what is required for the cryptocurrency to work properly. [4](#), [6–8](#), [12](#), [48](#), [64](#), [66](#)

**cryptography**

is the practice and study of techniques for encoding and decoding information, for the purpose of secure communication in the presence of third parties that are to be excluded from the communication. [xii](#), [xiii](#), [1](#), [2](#), [6](#), [7](#), [14](#), [15](#), [82](#)

**currency**

type of **money** that is supported by a government or a nation state by declaring it as **legal tender** within its jurisdiction. [iii](#), [xi](#), [xiii](#), [xvi](#), [2](#), [4](#), [5](#), [14–16](#), [18](#), [26](#), [27](#), [29](#), [40](#), [58–61](#), [64](#), [72](#), [73](#), [76](#), [78](#), [79](#)

**debt**

is the, potentially intentional, result of the avoidance of a **payment**; a certain amount of **money proper**, specified in the **money of account**, is payable by the debtor at a certain date to the creditor. [xiv](#)

**deep modeling**

is a term coined at the Chair for Software Development of the University of Mannheim, Germany; is a type of **modeling** that i.a. - contrary to conventional **meta-modeling** - has a systematic place for domain meta-types and therefore allows their seamless inclusion within the models. [2](#)

**difficulty**

is the relative measure  $T_{min}/T_{act}$  with  $T_{act}$  being the **target** of the block at hand and  $T_{min}$  the smallest possible **target**. [47](#)

**digital cash**

is a digital form of **money proper** that is digitally created, saved and spent. [iii](#), [xii](#), [3](#), [4](#)

**domain**

is a research area that has common requirements, functionality and terminology. E.g. the domain of this work is **cryptocurrencies** and the terminology of it can be found within this glossary. **xiv, 2, 11, 13, 40, 64, 73, 77, 82**

**double-spending**

is a failure-mode of digital **payment systems**; if digital **money proper** can be duplicated the way files can be, its functionality is undermined; if **double-spending** is possible the whole **payment system** fails; the way **double-spending** is prevented in **cryptocurrencies** using a public ledger called the **blockchain** by employing **cryptography** technology is the key innovation of **Bitcoin**. **xii, xiii, 3, 4, 37–39, 44–46, 48, 55, 59**

**ECDSA**

Elliptic Curve Digital Signature Algorithm. **xvii, 9, 41**

**EcoSystem**

the most widely conceivable **Bitcoin** related system that includes the **Bitcoin Network** as well as businesses and service providers that account for additional use cases, for example allow buyers and sellers of **bitcoin** to trade in other (foreign) **currency**. **xi, xvi, 4, 35, 53, 59–62, 64, 76**

**fiat money**

is a term used to indicate that the **money proper** of a **payment system** viewed as a simple **asset**, without the properties that made it **money proper**, would have a price no different from or very close to zero. As an example may serve the proverbial paper note that itself is just a piece of paper but gains value through the properties that made it **money proper** as a paper bill. **26, 27**

**funding liquidity**

is the ability to put off **payments** when they come due, by paying someone else to make them. **28**

**generated block**

a **block** is *generated* if it was successfully **mined**. **xi, xii, xvi, 43–48**

**genesis block**

is the first **block** in a **blockchain**; the **genesis block** for **Bitcoin** was created including information about the date of creation (by means of a mainstream media headline related to monetary policy matters of that date), to maintain that no previous **mining** and thereby creation of **bitcoin** had happened. **xi, xiii, 4, 43, 44**

**hash function**

is any function that can be fed with arbitrary length inputs, to create fixed length outputs. Receiving slight changes in the input, useful hash functions produce dramatic changes in the output, and allow successful mapping with (almost) no collisions. **xii, xiv**

**hashing**

is the creation of a **message digest** using a **cryptographic** hash function on data of arbitrary size. **xi, xii, xiv–xvi, 39, 41, 42, 44–46**

**inside money**

is some form of credit within the hierarchy of money, yet its transfer is used and accepted

as a means of final **settlement of debt**, thereby making it **money proper**. 69, 72, 74, 75, 78, 79

**IOU**

acronym for ‘I owe you’. Indicates a creditor-debtor relationship. x, 48, 69–71, 75

**ISO**

International Standards Organization. xvii

**legal tender**

is by law the **money proper** for all **debts**, public and private that are denominated in the according **money of account**; legal tender does *not* mean that the corresponding **money of account** *must* be used to nominate private contracts, but *if* the corresponding money of account was used, then legal tender must be accepted in **settlement of the debt**. xii, 15, 18, 26, 58, 68, 74

**market liquidity**

is the ability for a holder of an **asset** to realize positive cash flows from selling the **asset** before the **asset** itself would produce positive cash flows of this amount. 28, 64, 79

**message digest**

is the ‘hash value’, the result of fixed length by applying a **hash function** on data of arbitrary length. xii, xiii, xv, xvi, 8, 9, 37, 39, 41, 43, 45, 56, 61

**meta-modeling**

is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for **modeling** a system. xii

**miner**

is a **node** that decided to engage in **hashing** the current **block**, which is called ‘**mining**’. xii, xvi, 39, 42–48, 52–54, 59, 61, 62, 65

**mining**

is the **hashing** of a **block** according to a specific block hashing algorithm. xiii–xvi, 9, 39, 42–48, 50, 52–54, 60–62, 64, 78

**MMI**

**Money Meta Infrastructure**. ix, 5, 23, 28, 30, 31, 33–37, 53, 54, 61, 62, 64–67, 72–74, 76–81

**model**

is an abstract representation of knowledge and of activities of **systems** in a structure that is defined by a consistent set of rules. xiv, 12

**modeling**

is the creation of **models**; **modeling** is done within a certain **domain**, with the goal to facilitate the comprehension of this knowledge and the activities or particular parts of it. xii, xiv, 1, 2, 13

**money**

is mentioned in reference to one of its two emergent properties **money of account** and **money proper**; depending on the context the reference to **money** is a reference either to **money** as the **money of account** or as the **money proper**. iii, xii, xiv, xv, 1–3, 14, 15, 17–21, 23–30, 34, 35, 48, 50–52, 58, 59, 61, 65, 67–69, 71–74, 77–82

**money meta infrastructure**

is a term invented particularly for but not limited to the legal, civil and fiscal infrastructure concerning **money** that makes users of a **money** confident enough to use and accept it at all. This includes working land registers, contract enforcement measures (e.g. bailiff), police, courts, etc. [ix](#), [xiv](#), [5](#), [23](#), [28](#), [30](#), [62](#), [77](#)

**money of account**

is the unit that is voluntarily used to price **assets** and services; e.g. if € 100 in cash is due, the unit of account is '€'; in this work a **money of account** is a unit of account that is used entirely voluntarily by virtually all private economic units, because it is embedded within a certain legal and fiscal infrastructure. [iv](#), [ix](#), [xi](#), [xii](#), [xiv](#), [xv](#), [11](#), [15](#), [18–20](#), [23–30](#), [33–35](#), [51–53](#), [58](#), [59](#), [62](#), [64](#), [73](#), [76–79](#), [81](#), [82](#)

**money proper**

is the actual means of final **settlement of debt**; the **money proper** is a means of payment that is accepted *without discount*; e.g. if € 100 in cash is due, the transfer of a € 100 bill settles this debt, the € 100 bill itself is the **money proper**. [iv](#), [ix](#), [xi–xvii](#), [3](#), [5](#), [11](#), [18–20](#), [25–30](#), [34](#), [35](#), [51–53](#), [55](#), [59](#), [62](#), [68](#), [72–74](#), [76–78](#), [82](#)

**MOOC**

Massive Open Online Course. [68](#)

**Network**

is the label for the communication of **nodes** amongst each other and about **transactions** and **blocks**; provides for a multitude of use cases, e.g. 'Pay in **XBT**'. [ix](#), [xii](#), [xiii](#), [xv](#), [4](#), [37](#), [38](#), [40–48](#), [50](#), [52–55](#), [57](#), [59–62](#), [64–66](#), [72](#), [78](#), [79](#)

**node**

is a computer, running a **cryptocurrency** related application called a **client**. [xii](#), [xiv](#), [xv](#), [4](#), [39](#), [40](#), [42–45](#), [47](#)

**nonce**

is a random number within the header of a **block** that is incremented after each **hashing** of the header to receive a different **hash** for the **block** even if the other properties of the header stay exactly the same (even the timestamp may be updated only every couple of seconds). [45](#), [46](#)

**orphan**

is a **block** that was successfully **mined** and added to the **blockchain**, but is no longer part of the longest **blockchain** and therefore **transactions** within the **orphan block** have to be rebroadcast to the **Network** by the **client** (the standard **client** does this automatically). [xv](#), [43](#), [44](#)

**outside money**

is an asset that is no ones liability and that is used and accepted as means of final **settlement of debt** and therefore is **money proper**. [26](#), [69](#), [72](#), [78](#), [79](#)

**payment**

is the performance of a duty, promise or obligation, in this work we emphasize specifically the **settlement of debt**. [xii](#), [xiii](#), [xvi](#), [3–5](#), [10](#), [11](#), [15–19](#), [23](#), [25](#), [26](#), [28](#), [29](#), [35](#), [37](#), [42](#), [44](#), [50–56](#), [58](#), [60](#), [65](#), [67](#), [68](#), [71–76](#), [78–81](#)

**payment system**

is a system that provides the service of facilitating **payments** for its users. **iii, ix, xi–xiii, 1, 2, 5, 10–12, 14, 16–20, 23, 25, 28–30, 33–37, 42, 44, 50–54, 61, 64, 65, 67, 68, 72, 74, 77, 81**

**plaintext**

is the information that the sender wants a recipient to obtain. For textual information this means that the text is human readable. **xi, 7, 8**

**property**

is a *right* (not a thing) to possess *and* use a thing to the exclusion of others. **5, 30, 33, 66, 81**

**requirement**

is defining what a **system** shall or should do and what it shall not or shouldn't do.. **xvi, 9–12, 14, 19, 29**

**RUP**

Rational Unified Process. **11**

**satoshi**

is the name for the smallest fraction of a **bitcoin**; 1 **satoshi** =  $10^{-8}$  **bitcoin**. **xvi, 48**

**seller**

is in this work specifically an actor that is selling **bitcoins** for **currency** on exchanges that are part of the **Bitcoin EcoSystem**. **60, 61**

**settlement of debt**

the specified amount of **money proper** is transferred by the debtor to the creditor at or before the specified point in time. **xiv, xv, 15, 25, 34, 35, 52, 72, 76**

**stakeholder**

is a person, organisation or role that is significantly affected by a **system** in any way, thereby able to contribute **requirements** or help to understand them. **11, 19, 35**

**system**

in the systems engineering context, a **system** is an interacting combination of elements to accomplish a defined objective. These include hardware, software, firmware, people, information, techniques, facilities, services and other support elements, as defined by the International Council on Software and Systems Engineering. **xiv, xvi, 10–12, 17, 19**

**target**

is a property of each **block**; if the **hash value**, created by **hashing** the header of the current **block**, is smaller than the current **target** that is shared among all **miners**, the **block** is successfully **mined** and therefore becomes a **generated block**. **xii, xvi, 39, 45–47**

**trader**

in this work a **trader** is an actor buying and/or selling (trading) **bitcoins** on exchanges within the **Bitcoin EcoSystem**. **xvi, 4, 53, 61**

**transaction**

in the **cryptocurrency** context is a form of communication that involves two or more **addresses**; **money proper** can be 'sent' from one **address** to another by a **transaction**. **xi, xii, xv–xvii, 4, 9, 37–48, 54–57, 59, 62, 65, 67, 78–80**



**transaction point**

is a **Elliptic Curve Digital Signature Algorithm (ECDSA)** private-/public-key-pair that allows to access (by means of the private key) a **bitcoin** value that is stored at the **address** (public key) of the **transaction point**; used as a simplifying concept to explain a **bitcoin** transfer by a **transaction** from one **address** to another. **xi**, **xvii**, **40–44**, **48**, **54–57**, **59**, **62**, **65**

**UML**

Unified Modeling Language. **12**, **13**, **52**, **70**

**wallet**

is a file created by a **client** that contains mainly private-/public-key-pairs called **transaction points**, which provide access to **bitcoins** (**money proper**) that are located at **addresses**; a **wallet** can be imagined much like a physical wallet containing bills (**money proper**); sometimes a **client** can be (misleadingly) called ‘**wallet**’. **xvii**, **40–42**, **54**, **55**, **59**, **62**

**XBT**

is a (non-official) **International Standards Organization (ISO)** form for **bitcoin**; US Dollar in **ISO** form is USD, Euro is EUR, Gold is XAU and **bitcoin** in **ISO** form is XBT. **xv**, **35**, **47**, **48**

## 1 Introduction

The financial crisis of 2007/2008 was a crisis of financial institutions. In November 2008, shortly after the heydays of the crisis, a white paper was released, claiming that - by combining open source, peer-to-peer technology and **cryptography** - a **payment system** could be established that does not require financial institutions or any central agency, service or counterparty (see [Nakamoto, 2008](#)).

### 1.1 Motivation

Is it possible that network technology could revolutionize yet another business branch? The inception of E-Mails put pressure on postal services, the utilization of file-sharing systems forced content providers to rethink their business models and while ‘blogging’ is still partly ridiculed by established mass media outlets, hardly one can be found that does not also publish information by using web-logs today (see [Falkvinge, 2011](#)).

Could the inception of **cryptocurrencies**<sup>1</sup> mean that “a new kind of money”<sup>2</sup> was created? Does it force a redefinition of business models for financial institutions? To what extent could this redefinition be necessary? Could the decentralized **payment system** based on **cryptography**, we call the **Cryptocurrency Payment System (CPS)** change or even replace the currently used **Banking Payment System (BPS)** that is based on financial institutions and their cooperation worldwide? Could this render financial crisis’ obsolete? These and other questions arise, if cryptocurrencies are a new kind of money. This work’s aim is to contribute to finding an answer to the question if **cryptocurrencies** indeed *are* a new kind of **money**.

### 1.2 Research objective

The work present is generic in nature, and is therefore concerned with **cryptocurrencies** and their capabilities in general. However the first and oldest of the cryptocurrency projects, the **Bitcoin** project, which exists publicly since early 2009, is used as the prime example of a cryptocurrency in this work. All specific examples and models that involve a cryptocurrency are based on the properties of the Bitcoin project. The central objective for this work is to examine the claim of the Bitcoin community that with the creation of cryptocurrencies “a new kind of money” was created. To be able to do this kind of examination the Bitcoin project needs to be looked at from an information systems perspective, as well as from a monetary theory perspective.

### 1.3 Research approach

Since the claim to examine is that the invention of cryptocurrencies created “a new kind of *money*”, it is necessary to establish an understanding of the term **money**. We approach this issue from a requirements engineering perspective on payment systems in that we ask the question: What does a **payment system** have to be able to provide for it to be used as such by a significantly large share of the population? What indeed are the requirements for a **payment system**?

In software development projects **modeling** technology is used to reduce complexity in the development process by being able to strip business logic complexity of the one that the implementation itself comprises. In this way modeling technology facilitates the creation of yet to develop software systems. In this work we use modeling technology to visualize properties of systems that already do exist. In this case the worldwide **payment system** that is operated by financial institutions, we call **BPS**, and the decentralized **CPS** that is claimed to potentially be

---

<sup>1</sup>One of the first indications of the term ‘cryptocurrency’ in a mainstream media outlet can be found on <http://www.forbes.com> - accessed May 13th 2014.

<sup>2</sup>This is a claim commonly formulated by the Bitcoin community: “Bitcoin is an innovative payment network and a new kind of money” - [www.bitcoin.org/en](http://www.bitcoin.org/en) - accessed May 13th 2014.

able to replace it or parts of it.<sup>3</sup> By using modeling technology we hope to be able to show in detail what the capabilities of the CPS are and then examine and compare these capabilities to those offered by the BPS. We approach these payment systems mainly from the viewpoint of an end-user or customer of these systems.

## 1.4 Structure

The first part of this work is a very short introduction to the cryptocurrency world and the payment system that operates on cryptocurrency technology in section 2. It is followed by the brief section 3 on cryptography, since cryptographic technology is used in the realization of cryptocurrencies and comprehending certain cryptographic procedures is not avoidable for a concise understanding of the working of cryptocurrencies. Finishing the basic sections is section 4 that is outlining specific areas of software engineering fundamentals that encompass procedures or techniques that were used in this work to further analyse the mentioned payment systems and their capabilities. Specifically modeling technology is going to be depicted, including not only conventional or 'shallow' modeling technology, but also deep modeling is going to be delineated.

In section 5 we are trying to find definitions for very key terms: *money*, *currency* and *payment systems*.

The approach in section 6 is to further inquire the term money using requirements analysis of a hypothetical payment system. We do this to be able to proceed with the examination of both the currently most widely used payment system, we call the BPS and the payment system based on cryptocurrency technology, we call the CPS.

Section 7 is an extensive assessment of Bitcoin by going into detail on fundamental principles of cryptocurrencies. Bitcoin thereby serving as our example for a CPS. We are starting with the explanation of important terms that are essential for the understanding of cryptocurrencies and then outlining the inner workings of Bitcoin using modeling and requirements engineering approaches.

In section 8 we are examining the BPS after we introduce what we call 'the money view on actuality', which is a view effectively developed throughout the course of this work. We are focusing on the BPS, keeping in mind that it is currently the most dominant payment system and therefore must be considered to fulfil the requirements elicited for a payment system in section 6.

We draw a conclusion on our findings of the work, especially for the cryptocurrency examined in this work, called Bitcoin, in section 9. Finally, we touch upon multiple possibilities for future research in this domain that look promising.

---

<sup>3</sup>"Can Bitcoin replace PayPal?" - <http://edition.cnn.com/2013/12/10/business/will-bitcoin-replace-paypal/> - accessed May 14th 2014.

## 2 Cryptocurrencies - a quick overview

In this introductory section we want to give a quick overview over **digital cash** and the specific type of **digital cash** that is subsumed under the label ‘**cryptocurrency**’. We start with some reflections on **digital cash** in general and the specific challenges the attempted digitalization of **money** poses. Then we introduce **cryptocurrencies** that embody a form of **digital cash** scheme that offers a decentralized solution to the challenges that **digital cash** schemes face. We also give a quick overview on the current state of use of the oldest and most widespread **cryptocurrency** that is called **Bitcoin**.

### 2.1 Digital cash

At the very first glance the demand for **digital cash** seems obvious, since for about twenty years now, we are on the road to ‘the digital economy’ (see [Tapscott, 1995](#)) and what better way to **pay** in a digital economy than with digital cash? At second glance however the question arises how exactly ‘**digital cash**’ would actually work.

**Specific challenges for digital cash** If Alice is able to pay Bob<sup>4</sup> by the transfer of a specific sequence of bits, then what is going to stop Alice from simply copying the sequence and ‘pay’ Carol, too? In fact, what would stop her from copying the bitstring over and over again? Or looking at it from the opposite angle: what would stop a counterfeiter to reproduce the bitstring that actually is Alice’s and thereby steal from her? This is but a small fraction of challenges that revolve around the potential susceptibility of digital cash for fraud, security breaches and counterfeiting. A central issue for digital cash schemes to resolve and that we want to focus on here briefly is the so called issue of **double-spending**.

**Double-spending - a central issue for digital cash** Very important yet trivial for successful **payments** in general is the inability for a payer to pay two (or more) payees with the very same **money proper**<sup>5</sup>. In other words: if it is possible for a payer to **double-spend** the identical **money proper**, then this type of **money proper** is not suitable for **payments**. A **money proper** that can be **double-spent** would be immediately rejected in future attempts to pay. **Double-spending** is a very crucial challenge for digital cash schemes to overcome, since it is very easy to simply copy digital information.

**Central authority as solution** If a central authority was part of the **digital cash** scheme and it was trusted by all participants then the **double-spending** problem could be resolved by checking the legitimacy of each **payment** transaction with the central authority. In a scheme of this kind the **digital cash** would be an actual digital token that was completely administered by the central authority, which would therefore act much like a bank. While a perfectly working central party alleviates the problems mentioned, including **double-spending**, it might be considered a great risk for the entire **digital cash** scheme to be totally dependent on the soundness of just one central authority. If it failed in any way, the entire system would be affected. In the past the digital cash schemes based on tokens and a central authority have hardly even succeeded as the mere customer loyalty systems they have been so far, let alone succeeding as an actual cash

---

<sup>4</sup>Alice and Bob are commonly used placeholder names, see [http://en.wikipedia.org/wiki/Alice\\_and\\_Bob](http://en.wikipedia.org/wiki/Alice_and_Bob) - accessed June 28th 2014.

<sup>5</sup>see section 6.2 to find the definition of this term and why it is used here instead of the more general term **money**.

equivalent.<sup>6</sup> **Cryptocurrencies** on the other hand are a kind of digital cash scheme that claim to have resolved the crucial issue of **double-spending** completely without the need for a trusted central authority or even a central party of any kind.

## 2.2 What is a cryptocurrency?

In this work we recognize a **cryptocurrency** as a **digital cash** scheme that does not need the support of a central server or other trusted central party to prevent its users from ‘**double-spending**’, but instead relies on a specific **cryptographic protocol** to do so. Furthermore **cryptocurrencies** operate on a *peer-to-peer* basis with **nodes** running **client** software that is typically *open source*. Hereafter we introduce the **Bitcoin** project by depicting some key features. The detailed description and explanation of the underlying **cryptographic protocol** however follows further below, as it is an essential part of section 7.

**Bitcoin - the inception of cryptocurrencies** The first and therefore oldest **cryptocurrency** project started in January of 2009 with the public release of the standard **client** source code and the initial calculation of the **genesis block**.<sup>7</sup> Its inventor Satoshi Nakamoto<sup>8</sup> gave the project the name ‘**Bitcoin**’ in a white paper that was released in late 2008 (see **Nakamoto, 2008**). In this work it is the **cryptocurrency** project **Bitcoin** that is going to be used as the prime example for **cryptocurrencies** in general.

**Transactions** **Bitcoin** allows **payments** to be made within the **Network** by executing **transactions** that are fast, cheap and irreversible.<sup>9</sup> An average waiting time of ten minutes for the initial and about 60 minutes for the final confirmation<sup>10</sup> of an international payment, sounds very compelling and is indeed very fast compared to the alternative of a standard commercial bank money transfer that can take multiple days in international venues.<sup>11</sup> Once the **transaction** is finally confirmed it is irreversible. Additionally the transaction costs of a bitcoin **transaction** are - at least so far - nothing but minuscule compared to bank transfers or credit card payments. For more details on **transactions** see section 7.2.2.

**Bitcoins are traded in traditional currencies** On exchanges that formed within the **Bitcoin EcoSystem** traders can buy or sell **bitcoins** for traditional **currencies**, such as USD or EUR. The exchanges are so far the interface between the **Network** and the traditional monetary system. While the price of **bitcoin** in USD has been very volatile all along it has been rising from being essentially zero at inception to about \$ 600 per one **bitcoin** in June of 2014, reaching highs in late November of 2013 of more than \$ 1100.

<sup>6</sup>One example of a completely failed digital cash scheme of the token and central authority kind is beenz.com. See ‘The greatest defunct Web sites and dotcom disasters’ (June 5th 2008) - <http://web.archive.org/web/20080607211845/http://crave.cnet.co.uk/0,39029477,49296926-7,00.htm> - accessed June 26th 2014.

<sup>7</sup>This is a reference to the key innovation of Bitcoin: the public ledger called **blockchain** that started on Jan 3rd 2009 with the **genesis block** - [https://en.bitcoin.it/wiki/Genesis\\_block](https://en.bitcoin.it/wiki/Genesis_block) - accessed June 28th 2014.

<sup>8</sup>This name is probably a pseudonym, potentially for a group of people, however there is a claim by newsweek for it being the real name of one person only: <http://www.newsweek.com/2014/03/14/face-behind-bitcoin-247957.html> - accessed June 10th 2014.

<sup>9</sup>For a list of these properties, see <http://www.coindesk.com/information/why-use-bitcoin/> - accessed June 10th 2014.

<sup>10</sup>The first confirmation of a **transaction** happens after approximately 10 minutes, a chart that shows confirmation time can be accessed here: <http://blockchain.info/charts/avg-confirmation-time> - accessed June 28th 2014.

<sup>11</sup>While being limited in the transferable amounts, payments by credit cards are even faster than the confirmation of **Bitcoin transactions**, since the credit card company is paid by the payee for acting as a fiduciary middleman. Using fiduciary trust in this way is something that can be reproduced for **Bitcoin transactions**, making them potentially equally fast, however obviously introducing a trusted third party in doing so.

**No central authority** In early 2013 the price increase in USD for **bitcoins** coincided with problems in financial stability in Cyprus. Amongst other measures the government of Cyprus decided to lower the allowed cash (**money proper**) limit to € 1000 per traveler. By various means, including the levying of special taxes, the government of Cyprus also called on businesses and private households that owned savings accounts with more than a € 100.000 balance in the effort to reinstate financial stability in Cyprus (see [Stavárek, 2013](#), pg 316). Recognizing the decentralized cryptocurrency infrastructure, central governments are very limited in their abilities to levy special taxes of this kind on **cryptocurrencies**. In the context of central governments trying to levy special fees or taxes, albeit under the guise of general welfare, the conjuncture that there simply is no central point of reference in **cryptocurrencies** can be viewed as an immense advantage for individual users.

**Why are cryptocurrencies not prevalent?** Most of the above in this section could be construed as advertisement, even though **Bitcoin** is, after all, not run by a for profit corporation that has a marketing budget. Still, the properties of **Bitcoin** described so far could be perceived as almost too good to be true (to recap pointedly: “*open-source, peer-to-peer, cheap, fast, secure, irreversible, rising prices, safe from arbitrary intervention of governments et cetera*”). So the **payment system** powered by **cryptocurrency** technology should be already the most used payment system out there, shouldn't it? But since it is not, even though the properties sound so compelling, we are confronted with the question “why isn't it?” Why are private businesses and organizations at large not pricing their **assets**, goods and services in bitcoin? Why do private households still at large shy away from pricing their **properties** or other **assets** in some **cryptocurrency**?

Is it just technically too difficult to use for average households and businesses? Is it possible that the still rare usage is a mere usability issue? **Cryptocurrencies** would not be the first on-line technology that was invented years before broad or even mass adoption and usage started for mere improvements in usability.<sup>12</sup>

Maybe potential users just do not know enough about the new **cryptocurrency payment systems** and their capabilities. Are potential users just too uninformed about the possibilities that the new cryptocurrencies provide? If this was true, it would be a mere hermeneutical challenge and as soon as the education about cryptocurrencies is advanced and successful enough they would reign supreme in usage amongst the payment systems worldwide, considering the seemingly obvious advantages in velocity and cost of transactions to perform **payments**.

**The working hypothesis** Perhaps it is neither simply a usability or an educational issue nor just a combination of both. Potentially there is a deeper reason that has so far prevented the mass adoption of **cryptocurrencies** that still have not had their march of triumph as ‘a new kind of money’ yet. In fact, it is going to be the working hypothesis for this work that something indeed is prohibiting **bitcoin** from mass adoption that goes beyond usability or educational issues and it probably has to do with risk avoidance of potential users.<sup>13</sup> These considerations concerning risks could be revolving around the volatility in the price of **bitcoin** compared to other **currencies**. The potential users most likely have expenses that are nominated in those **currencies** and they need to cover these expenses when they are due by transferring the **money proper** of those **currencies**, they can not rely solely upon the **cryptocurrency** income.<sup>14</sup>

<sup>12</sup>This is an idea from [Falkvinge \(2011\)](#).

<sup>13</sup>Refer to footnote 31 for [expedia.com](#) on one hand announcing its willingness to ‘accept’ **bitcoins** as **payment** but at the same time strictly limiting the acceptance to certain use cases and announcing the unwillingness to holding **bitcoins**, due to risk avoidance.

<sup>14</sup>This issue is going to occupy us throughout this work, but for reflections on risk reduction specifically, see section 6.6 on the **Money Meta Infrastructure (MMI)** that **Bitcoin**, seen as a **Cryptocurrency Payment System**, shows significant differences in compared to the **MMI** that the **Banking Payment System** is embedded in.

**Security** **Bitcoin** is designed to prevent the tampering with information, impersonation and outright stealing of others. In this sense **Bitcoin** is all about making sure the integrity of transactions is always guaranteed. This is a challenge that is generally well known to the physical world, where it is dealt with by signatures, bank vaults, safes and locks of various kind. In the world of bits and bytes this challenge is tackled by employing **cryptography** and this is exactly what **Bitcoin** does and this is why it can also be viewed as being a **cryptographic protocol**.



## 3 Cryptography

On the one hand it is not necessary, on the other it is way beyond the scope of this work to go into detail about **cryptology**. However to comprehend the fundamental principles that **cryptocurrencies** work on in some detail, it is helpful to understand the basic ideas of certain cryptographic principles. The principles that are specifically relevant for the **cryptographic protocol** that is implemented by the cryptocurrency project **Bitcoin**, are going to be briefly explained in this section.

### 3.1 Symmetric cryptography

Cryptography is the study of technologies that allow secure communication that is the exchange of information unscathed by third party adversaries, who potentially want to intrude, listen to or change the communication. So, if Alice wants to send information to Bob securely, the basic idea is to not just send the information as **plaintext** - say by E-Mail - from Alice to Bob, but for Alice to use a key to encrypt the information and sending it to Bob as **ciphertext**. The same key can later be used by Bob to decrypt the information to make it **human readable** again. This way Alice and Bob can send each other messages that cannot be read by third parties as long as Alice and Bob are holding the key exclusively and the encryption cannot be broken.

Alice and Bob obviously have to exchange the key before being able to communicate securely. This can be a problem for multiple reasons. If Alice and Bob are communicating on-line the simple exchange of the key as **plaintext** undermines the intent to encrypt information at all, since a key that was sent as plaintext via internet communication can be considered the release of this key to the whole world. Using this key still with the intent to communicate securely, even though it was transferred in plaintext between Alice and Bob beforehand, is blatantly imprudent. So for symmetric cryptography to work in the intended way, there needs to be an initial key exchange of at least one secret key. This key exchange needs to be *secure* in the sense that after the exchange the keys are known by the intended parties (here: Alice and Bob) and stay secret for everybody else.

One possible way to exchange keys securely is to arrange a physical meeting and exchange the keys without third party interference. However the physical meeting to do a key exchange can be problematic in itself and is especially impractical for fast-paced on-line technologies, like **cryptocurrencies**.

There is a way to overcome this key exchange problem, by using another cryptographic algorithm that involves not only a single key, but a key-pair, where one key is released to the public and the other key stays private, the encryption is done with one key, the decryption with the other. This class of algorithms is called ‘public key cryptography’ or ‘asymmetric cryptography’.

### 3.2 Asymmetric cryptography - public key cryptography

Public key cryptography is a class of cryptographic algorithms that involve two separate keys. The two keys are mathematically linked in such a way that allows the encryption of data with one key and the decryption with the other. The user now has the option to make one of the keys public (so called ‘public key’) and to keep the other one secret (so called ‘private key’). For the sending of encrypted text the *encryption* is done with the public key and the *decryption* with the private key. The algorithm makes use of the fact that mathematical problems exist that currently cannot be solved efficiently. While it is necessary for the user to be able to calculate a private and a public key in reasonable time, the calculation of the private key from knowing the public key has to be mathematically infeasible to make the algorithm work (see [Katz & Lindell, 2007](#), pg. 333ff.).



Figuratively speaking we could think of public key cryptography as if it was a system of boxes and padlocks. The public key in this metaphor would consist of a box with an open padlock that could be closed by anyone who is in front of the padlock. To open the box that was locked with the padlock and find out what contents were put in you need the key to the padlock. So if Bob wants to communicate securely with Alice, he needs to create a key-pair and publish his public key. Alice then can use Bob's public key to encrypt plaintext she wants to send to Bob. In the metaphor this is putting something into the box and locking it with Bob's padlock. If we assume that the box is unbreakable, the padlock cannot be picked and that only Alice holds a key to her padlock, then indeed only Alice has access to whatever is in the box.

Notably there is no need for an initial secure exchange of a private key, since Alice doesn't need access to Bob's private key. She can use his public key to encrypt whatever plaintext she chooses, yet only Bob is going to be able to decrypt the ciphertext back to plaintext using his private key.

**Diffie-Hellman handshake** One possible use of these types of algorithms is the exchange of a secret key between two parties that initially do not know anything about each other. After the establishment of a common secret key, the two parties can start to communicate securely using symmetric cryptography. This is generally done by the calculation of a shared secret, using mathematical problems that - at current understanding - do not allow for an efficient solution. This process was first described and published by Diffie and Hellman, using a fundamental group in number theory ('multiplicative group of integers modulo  $n$ ') as source for the mathematical problem, and is therefore often called 'Diffie-Hellman key-exchange' or 'Diffie-Hellman handshake' (Diffie & Hellman, 1976).

**Most relevant insight** Generally speaking the most relevant insight about asymmetric cryptography is the realization that it is possible for two initially unknown parties who communicate over an unsecured channel to generate a shared secret key that can be subsequently used for secure communication.

Even though the word 'cryptocurrency' does signal the use of cryptography, neither symmetric nor asymmetric cryptography is used in the most basic **cryptocurrency protocol**. For this works purpose asymmetric cryptography is mentioned to facilitate the understanding of cryptographic hashing and digital signature algorithms, which are both heavily used in cryptocurrencies and are explained hereafter.

### 3.3 Cryptographic hashing

If Alice sends encrypted messages using Bob's public key for encryption, then Bob can read the message by applying his private key to receive **plaintext** again. However Bob cannot be sure that the message was not altered on the way to him. To alleviate this problem Alice can use a **cryptographic hash function** to create a **message digest** (also called 'hash value' or 'digest') and append it to the message. Bob can then himself use the cryptographic hash function, create a digest himself and compare it with the one he received by Alice attached to the original message. Cryptographic hash functions are designed in a way that they produce fixed length outputs (**message digests**) off of inputs of variable length. For these functions to be useful they are practically impossible to invert, meaning that someone receiving the **hash value** alone cannot create the original message from it (see Katz & Lindell, 2007, pg. 278ff.). So if Bob finds the two hash values are the same, he can be (reasonably) sure the message he received remained intact on the way to him.

However there is another challenge for Alice to make sure the message digest got sent to Bob intact, because if the message digest is altered on the way to Bob, then he cannot successfully

compare the digest he received with the one he created himself. One way to do this is the inclusion of the message digest in a digital signature.

**Most relevant insight** Cryptographic hashing is the creation of a *message digest* (also called *hash value* or just *hash* and sometimes *digest*) of fixed length by application of a **cryptographic hash function** on a message of variable length. It is used on multiple occasions in **cryptocurrencies**, this includes the hashing of the header of a **block**, which is part of **mining** and the hashing of **Bitcoin addresses**. It is also applied in what is called a Merkle tree, which is a tree structure that hashes the hashes of data until there is only one root hash left (see [Nakamoto, 2008](#), pg. 4).

### 3.4 Digital signature algorithms

Digital signatures are a class of public key algorithms, and therefore use a mathematically linked key-pair as all public key algorithms do. In the asymmetric cryptography algorithm depicted above however, the encryption was done using the public key and the decryption was done using the private key. Digital signatures are carried out vice versa. Within digital signature algorithms the *encryption* is done using the private key and the *decryption* using the public key (see [Katz & Lindell, 2007](#), pg. 421ff.).

If Bob was Alice's banker it could be necessary to prevent third party adversaries to impersonate Alice and perform actions on her account that she never approved off. So Bob wants to make sure that messages he supposedly got from Alice were actually sent by her. To accomplish this **requirement**, Alice could include a digital signature with her message.

For a digital signature to be created the first step is to create a **digest** by using a **cryptographic hash function**, as was depicted above in section 3.3. The second step is the encryption of the digest using the private key. Finally this digital signature is attached to the message.

1. Create a message digest
2. Encrypt the digest using the private key
3. Attach encrypted digest to the message

With digital signatures it becomes possible for Bob who has just received a message from Alice to verify that the message indeed was sent by Alice ('authentication') that it has not been altered on the way ('integrity') and if Alice still maintains that the private key that was used to digitally sign the message, is still private, then Alice cannot shy away from the responsibility that comes with having actually sent the message ('non-repudiation').

**Most relevant insight** Digital signatures are pretty much used like signatures in the physical world. Sophisticated mathematics can make digital signatures potentially as forgery-proof as real world signatures or make them even tougher to forge and therefore more secure. In **Bitcoin** an **Elliptic Curve Digital Signature Algorithm (ECDSA)** is used for the signing of **transactions**. The current owner of a sum of **bitcoins** signs the transfer to the new owner, by including the new owner's public key as an output in the **transaction** and then signing using his private key. This is going to be depicted in much more detail in section 7.

## 4 Software engineering

It is way beyond the scope of this work to give a comprehensive introduction to software engineering or even a detailed analysis of the multiple different practices, methods and principles of software engineering that reach from rigorously formalized approaches to much more informal and agile methods.

It is the aim of this section to introduce the sub-disciplines of software engineering that include practices that were used to approach the central issue of this work, which is the question if a recently invented open-source peer-to-peer software system, we call the **Cryptocurrency Payment System (CPS)**, can in any way change or even replace the already long existing **Banking Payment System (BPS)**. To do this the attempt is to understand these two **payment systems** as profoundly as possible and to achieve this, we employ practices out of software engineering. Essentially we are taking a look at existing **systems** from a software engineering perspective. The **BPS** is a complex system that is partly a software system, too, but cannot be reduced to being just that. Therefore the initial starting point of this section could actually even be set to ‘systems engineering’ that software engineering is a part of. But we will start at this point with software engineering, more specifically with the sub-disciplines of it relevant for this work: *Requirements*, *Software design* and *Software engineering models and methods*. The whole section is oriented on the ‘Software Engineering Body of Knowledge V3.0’ ([Bourque & Fairley, 2014](#)) and was inspired by [Sommerville \(2011\)](#).

### 4.1 Requirements and Design

In software engineering the **requirements** can be thought of as the bridge between the so called real world and the world of software-intensive systems. They are essentially the translation of real world needs into a form that can be realized as a software artefact. In this sense requirements are a reflection of real world problems that can be read and understood by software engineers, who then again translate requirements into actual software systems.

Typically requirements are defined early on in the software development process. Starting with an analysis of the real world problems and the environment that the system will be later used in, requirements are defined as a specification of what the system is supposed to be doing. Requirements can also put constraints on systems. Either as specification or as constraints, requirements define for the software developer what needs to be implemented and also how the implementation should be executed (see [Bourque & Fairley, 2014](#), p. 1-1).

**Definition** Requirements define what the **system** should or should not do. This is a very general definition that does not take into account that there are specific types of requirements. An example for a requirement is:

“The system shall provide a service that allows a user (payer) to safely **pay** another user (payee), within one business day, thereby conforming to all laws applicable.”

**The requirements process** The requirements process is conceptually split into the activities of *elicitation*, *analysis*, *specification* and *verification* (see [Bourque & Fairley, 2014](#), pg. 1-3). Even though this may sound like a linear process, and it is indeed initiated at the very beginning of a software development endeavour, it is neither linear nor finished up front. The requirements process is a continually running process throughout the software life cycle, while the requirements are constantly analysed, specified and refined.

**Product and process requirements** A requirement may concern the product, meaning the **system** that is to be developed, or the process of development. An example for a product requirement is:

“The system shall yield a **money proper** that allows the users to perform **payments**.”

An example for a process requirement is:

“The system shall be developed according to the **Rational Unified Process (RUP)**.”

**Functional requirement** Functional requirements describe the features of the system, for example the ability to execute **payments** within a certain time. They also may state what the system should *not* do. It is possible to test for the fulfilment of a functional requirement by a finite set of test steps. In any way functional requirements describe how the system is supposed to react to particular inputs and in particular situations. As functional *user* requirements, they are very high-level statements, just as:

“The **money of account** of the system shall have properties that lead to the voluntary pricing of assets, goods and services by the users.”

As functional *system* requirements however they should describe the system services in detail.

**Non-functional requirement** Non-Functional requirements are the constraints for the system and are sometimes also called *quality requirements* and apply therefore often to the system as a whole rather than to a specific feature or service. They can be classified further into requirements concerning security, reliability, safety, interoperability, performance or any other type of quality characteristic. An example of a non-functional product requirement

**Emergent property** An emergent property is a type of requirement that is dependent on the collaboration of many components of the system. The above mentioned requirement of pricing **assets** with a certain **money of account** depends on how all the system components interact under real-life operating conditions. This requirement clearly is an emergent property for a **payment system**, which is going to be further explored in section 6.3.

**Stakeholders** It is an important issue for a requirements engineer to appreciate that not only the potential users of a system can and do provide requirements. Various kinds of direct and indirect **stakeholders** are also able to contribute requirements or facilitate the understanding of requirements that are already elicited. To put it in another way: **stakeholders** are prone to lose something if the system does not conform to their **requirements**.

**Requirements document** The requirements document is a desired product of the requirements process. The continuous refinement of requirements through analysis, specification and verification iteratively improves the requirements document, which therefore is never finished at any one point during the software life cycle. The changes in the requirements document are often so frequent and quick that many agile software engineering methods, consider even creating a requirements document a waste of time. They are instead handling requirements in what is called user stories (see [Sommerville, 2011, 3.3](#)).

As a side note at this point: In this work there is not going to be an exhaustive requirements document produced for either one of the evaluated payment systems (**BPS**, **CPS** and a non-existent hypothetical payment system). The actual creation of a comprehensive requirements document for **payment systems** could be a field of potential further enquiry. However this could be more difficult than initially thought of. The specifics of the payment systems **domain**

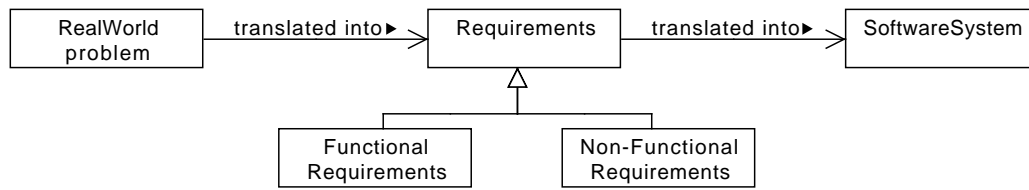


Figure 1: Object diagram on requirements, using concepts of the **UML**.

might cause certain stakeholders that are currently influential in the **BPS** (potentially particular nation states and other important stakeholders), to refrain from disclosure of all of their actual requirements. This is an example for special challenges that requirements engineers can face during the requirements engineering process. In this case actual **requirements** are simply not mentioned during requirements elicitation for lack of interest in disclosure of these **requirements** by the respective stakeholders.

**Conceptual Modeling** It is part of the requirements analysis activity within the requirements process to create **models** of real-world problems.

“[C]onceptual models comprise models of entities from the problem domain, configured to reflect their real-world relationships and dependencies.” (Bourque & Fairley, 2014, pg. 1-8)

We will apply modeling technology during the requirements analysis of **payment systems** in section 6 and for the understanding of **Bitcoin** and the analysis and evaluation of both the **CPS** and **BPS**. A very simple example for a graphical model of some of the contents of this section on requirements is shown in figure 1, using most basic modeling concepts (including generalization). This structural model is conforming to the general-purpose modeling language **UML** that was defined by Booch *et al.* (1998).

**Design transforms requirements** Software design transforms **requirements** into a description of how to solve the real-world problem at hand and includes the planning of a software solution for this problem. Design encompasses both low-level component and even algorithm design as well as high-level, architecture design (see Bourque & Fairley, 2014, pg. 2-1).

**Reverse engineering existing designs in this work** As we are involved in design activities in this work, we are reverse engineering already existing **systems** that realize a certain design. It is the task to reveal first the architectural design of the existing systems. We will go into reverse engineering of algorithm design in section 7, when we discuss the inner workings of the **Bitcoin protocol**.

## 4.2 Software engineering models and methods

The aim of scientific modeling in general is to make the comprehension of a particular part of the world easier. The intention for the use of modeling technology is the very same in software engineering. One key intention in using graphical modeling languages is to facilitate the communication process between the software developers and the other, potentially non-technical, stakeholders of the project.

### 4.2.1 Modeling

Modeling technology today is a key activity in virtually all information technology projects. In the software development process **modeling** is used to separate the complexity that is implementation specific from inevitable business logic complexity. By being able to abstract from complexity that arises during implementation, modeling helps to focus on and intellectually penetrate the business logic at hand. This is precisely the intention in this work, in which the graphical general-purpose modeling language **Unified Modeling Language (UML)** was used, to help understand the **domain** of this work, **cryptocurrencies**.

### 4.2.2 Software engineering methods

Facing a real world problem that is supposed to be solved by a software-intensive system and then just staring to write code is a strategy that might be successful for a real world problem that can be tackled by software solutions of limited complexity, but big software-intensive systems demand for another method. This is the reason why concise software engineering methods have been defined, whereby such a method is considered being first and foremost an “organized and systematic approach to developing software [...]” (quoted from **Bourque & Fairley, 2014**, pg. 9-7).

**Method applied in this work** There is no specific software development method rigorously applied to the real-world problem that is at the centre of this work. However the heuristic method applied is oriented on object-oriented analysis and design methods and was adapted to domain specific needs. The models created are based on concepts of the **UML** the intent always being to provide as much understanding of the **domain** as possible using these types of diagrams. So, sometimes the models are not conforming to UML meta-models in their entirety (see for example the nesting of systems in the diagram in figure 28 on page 63).

### 4.2.3 Modeling tools used in this work

For the creation of the **UML** models that are shown in this work the free and open-source software UMLet was used.<sup>15</sup>

---

<sup>15</sup>see <http://www.umlet.com/> - accessed July 1st 2014.



## 5 Currency, money, payment systems

As mentioned in 1.2 the chief aim for this work is to find out if the **cryptocurrency** community has indeed created a new kind of **money**. It seems obvious that certain basics on **cryptography** need to be discussed in an introductory section in this work (see section 3). The same is true for **cryptocurrencies**, employing **cryptographic** technologies in a new and potentially sweepingly innovative way (see section 2). Also a very short introduction to software engineering, respectively specific sub-areas that provide practices and concepts employed in this work, is given in section 4. But also the terms **currency**, **money** and **payment systems** as concepts are suited to get a section that sets the stage. Not because **money** as a technology is so new, but because we want to start off this investigation as little as presumptuous as possible. First we want to inquire modern macroeconomics what it has to say about **money** and the functions or **requirements** it has to fulfil to be used as such.

### 5.1 Money

In mainstream macroeconomic textbooks, an example is Mankiw (2007) or Mankiw (2014), another is Hicks (1967), money is explained by the *functions* it fulfils (see Mankiw, 2007, pg. 22f.). The functions referred to are in brief:

1. medium of exchange
2. unit of account
3. store of value

Having the research objective of this work in mind and looking at these functions it is tempting to merely check if the **Bitcoin** system does indeed create something (**bitcoins**) that is used as a ‘medium of exchange’, as a ‘unit of account’ and as a ‘store of value’. Of course we would have to elaborate on these three functions - we do this extensively in section 6 - but let’s assume for sake of the current section that these functions were self explanatory. Our investigation could already end right here and we would just state: “Well, if it was used in a way to fulfil these functions, then it must be money.” Astonishingly, in a way, this is exactly the perspective that is taken on **bitcoin** by some economists:

“Currency is any agreed upon means of exchanges of goods and services, so you could have some small stones, as used in history, and if it’s accepted by a sufficiently large population, then that’s enough [...]”<sup>16</sup>

The logic in this reasoning is simple and at first glance seems to be convincing: if it is “accepted by a sufficiently large population”, then it is money. End of story. How a **payment system** would have to be designed however for it to be used “by a sufficiently large population” is exactly what is of interest to us in this work, the assumption here being that **Bitcoin** is currently not used “by a sufficiently large population” as their **money**.

### 5.2 Currency

Since it is our goal to enquire if *cryptocurrencies* are “a new kind of money” we need to establish an understanding of what a *currency* is. The term **currency** is defined in Black’s Law Dictionary as follows:

---

<sup>16</sup>Paul Ehling, associate professor in the department of financial economics at the BI Norwegian Business School - cited in ‘Bitcoins Fail Currency Test in Scandinavia’s Richest Nation’ - <http://www.bloomberg.com/news/2013-12-12/bitcoins-fail-real-money-test-in-scandinavia-s-wealthiest-nation.html> - accessed May 13th 2014.

“Coined money and such banknotes or other paper money as are authorized by law and do in fact circulate from hand to hand as the medium of exchange.”<sup>17</sup>

According to this definition there are two requirements to a currency.

1. Being authorized by law
2. Circulate from hand to hand as the medium of exchange

**1. Legal tender** The first requirement we consider to be fulfilled for what is called **legal tender**. Legal tender laws define what has to be accepted in final **settlement of debt**, if it is offered as **payment**.<sup>18</sup> Legal tender laws do not force private businesses to accept the specified **currency**.<sup>19</sup> To be precise and stick to the terms used in his work: legal tender laws do not force any private party to offer their **assets** or services at a price that is specified in a certain **money of account**.

**2. Being in circulation** The second requirement might be reflected in the etymological origins of the term ‘currency’ that roots in the Latin word ‘currere’ that is a reflection of ‘currere’, which means ‘to run’.<sup>20</sup> A currency is therefore characterized by the virtue of being ‘on the run’ in the sense that it has the ability to change hands quickly - for purposes of **payment** - and does so without discount. If we interpret this requirement as a shall requirement, then a currency that - for whatever reason - loses its acceptance as a current means of payment without discount stops being a **currency**.<sup>21</sup>

**Summarizing definition** If something is ‘currently’ used and accepted as **money**, it is a **currency** within a jurisdiction, if this jurisdiction specified it as **legal tender**. This definition allows for other **monies** to exist within the jurisdiction, but if only one **money** is defined as **legal tender** within one jurisdiction, then there is only one **currency**.

**‘Currency’ versus ‘money’** Following the definition above, **money** and **currency** are no synonyms. An outright synonymous use therefore is strictly avoided in this work.<sup>22</sup>

**Implications for the term ‘cryptocurrency’** Considering the definition of **currency** above, the term **cryptocurrency** is interesting, since currently there are no central authorities declaring it as **legal tender** in any jurisdiction. As long as no central government does declare a **cryptocurrency** as legal tender, the term *cryptocurrency* might be considered misleading. Without being **legal tender** a **cryptocurrency** can by definition be no **currency**. On the other hand it might be a notion on the power of **cryptography** that might make it a **currency** in just another sense, which

<sup>17</sup>See: *Griswold v. Hepburn*, 2 Duv. (Ky.) 33; *Leonard v. State*, 115 Ala. SO, 22 South. 504; *Insurance Co. v. Keirou*, 27 111. 505; *Insurance Co. v. Ivupfer*, 2S 111. 332, 81 Am. Dec. 284; *Lackey v. Miller*, 01 N. O. 20. - <http://thelawdictionary.org/letter/c/page/234/> - accessed July 4th 2014.

<sup>18</sup>An example for a legal tender law is to be found in the US Coinage Act of 1965, Section 31 U.S.C. 5103: “United States coins and currency (including Federal reserve notes and circulating notes of Federal reserve banks and national banks) are legal tender for all debts, public charges, taxes, and dues. Foreign gold or silver coins are not legal tender for debts.” - <http://www.law.cornell.edu/uscode/text/31/5103> - accessed July 5th 2014.

<sup>19</sup>“There is [...] no Federal statute mandating that a private business, a person or an organization must accept currency or coins as for payment for goods and/or services.” - <http://www.treasury.gov/resource-center/faqs/currency/pages/legal-tender.aspx> - accessed July 5th 2014.

<sup>20</sup><http://en.wiktionary.org/wiki/currency> - accessed July 4th 2014.

<sup>21</sup>This has happened in the past to **currencies** of certain nation states that had their currencies devalued or essentially destroyed, leading to restrictions in the use of foreign **currencies** by ‘their’ citizens; lately seen in Argentina: <http://www.theguardian.com/world/2014/jan/24/peso-collapse-argentina-economic-crisis-fernandez-de-kirchner> - accessed July 8th 2014.

<sup>22</sup>As an example for synonymous use of these terms: the economist cited in 5.1, footnote 16 used currency and money synonymously.



brings us right back to the research question that motivated this work: are cryptocurrencies “a new kind of money?”

### 5.3 Payment Systems

So far we have used the term **payment system** without a concise definition. This term is going to be much used in this work, therefore we want to find a definition that is as unambiguous as possible and suitable to the needs of this work. To find out what a **payment system** is, we first need to establish an understanding of the process of **payment**.

**A payment is the “delivery of money”** As we did with the term **currency**, we want to consult Black’s Law Dictionary to start our inquiry for a definition of the term **payment**:

“The performance of a duty, promise, or obligation, or discharge of a debt or liability, by the delivery of money or other value. Also the money or other thing so delivered.”<sup>23</sup>

A “performance of a duty” or the “discharge of a debt” clearly implies (at least) two parties, a creditor, who has a claim on a debtor, and a debtor, who is obliged to perform the duty towards the creditor. Such a creditor-debtor relationship<sup>24</sup> can, for example, be established by a loan payment from a Creditor to a Debtor as is shown in figure 2.

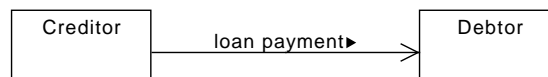


Figure 2: Creditor pays out loan to Debtor, thereby establishing a creditor-debtor relationship that is shown in figure 3.

This is a good example to show that a payment can initiate a creditor-debtor relationship, in this case, because by agreeing upon the terms of a credit agreement, the Creditor was obliged to make the initial principal payment to the Debtor. So, again, the payment does discharge the debt, even though after the initial payment from Creditor to Debtor, the creditor-debtor relationship is just established (shown in figure 3). We are going to expand on credit agreements in section 6.6, where we are showing the course of a credit agreement as an activity diagram in figure 8 on page 32.

Figure 3 shows the ongoing relationships between Creditor and Debtor after they are established, here, after the loan payment from Creditor to Debtor, but before the final payment from Debtor to Creditor has happened.

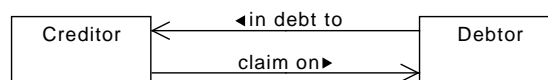


Figure 3: Established creditor-debtor relationship before payment.

<sup>23</sup>See: Brady v. Wasson, 6 Ileisk. (Tenn.) 135; Bloodworth v. Jacobs, 2 La. Ann. 24; Root v. Kelley, 39 Misc. Rep. 530, 80 N. Y. Supp. 4S2 [...] - <http://thelawdictionary.org/payment/> - accessed July 4th 2014.

<sup>24</sup>this is a term borrowed from terminology used by Heinsohn & Steiger (1996).

The payment is the “delivery of money”, thereby discharging the debt. The circumstance that the debt can be discharged by the payment, implies that the debt was established before the payment is made. ‘Before’ can mean any time-frame including ‘immediately preceding’. The “delivery of money” implies that some form of **money** is available. Figure 4 shows the **payment** of principal + interest by the Debtor to the Creditor by transferring **money**. The **payment** is the discharge of the debt, thereby ending this creditor-debtor relationship that is shown in figure 3.

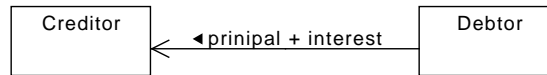


Figure 4: Debtor pays principal + interest to Creditor using **money**, thereby discharging the debt.

**Money allows for payments to be made** Following this understanding of a **payment**, it is clear that there can be no “delivery of *money*” if there is not some form of **money** available. Money, obviously plays a central role for the performance of **payments**.

**A payment system facilitates payments** A **payment system**, in this work, is any **system** that facilitates **payments** for its users. As a first analysis, the users of a **payment system** can be of two different types:

- Payer - the debtor uses the **payment system** as a payer to pay his creditor
- Payee - the creditor is receiving **payments** by the debtor through the **payment system**

We will expand upon the different types of users of a **payment system** in section 6.1, where an object diagram on user types, exceeding payer and payee, is shown in figure 6.

Obviously single entity users can be creditors in one ongoing relationship and debtors in others and even within one creditor-debtor relationship, the roles of payer and payee change back and forth as is shown in figures 2 (Creditor is the payer, Debtor is the payee) and 4 (Debtor is the payer and Creditor the payee). Concerning one single **payment** however the users are always engaged as exactly one of the two types depicted: they are either the *payer* or they are the *payee* in one single **payment**.

**A payment system has to provide for a money** To facilitate payments, a payment system needs to provide its users (*payers* and *payees*) some form of **money** that allows them to perform their **payment** transactions. **Money** is therefore a crucial issue for any **payment system** and any model about payments or payment systems needs to find a systematic place for the **money** that is delivered to perform **payments**. At this point we are not going into any more detail about the specificity of the **money** that is being used, whether the **payment system** itself provides for the money (endogenous money), or it uses some exogenous form of money, that is subsidized by some other form of settlement, e.g. *netting contracts* (see Emmons, 1995, pg.13f.). But we have to state that we need a theory that has a systematic place for **money**.

## 5.4 Synopsis on ‘money’

So far our understanding of **money**, following the views portrayed in 5.1 by mainstream economists, is that **money** is whatever is used and accepted as such. To define **money** like this makes a lot of sense in one way, but does not help us in another. In this work, we are

interested in finding an answer to the question under which *circumstances* something would be used and accepted as **money** or how a **payment system** would have to be designed to spawn “a new kind of money”. To be able to do this, we want to try to further expand our toolbox of monetary terms.

**Currency and money** The term **currency** as it is defined in 5.2 is different from **money**, in the sense that a **currency** is backed by a central government through legal tender laws. This definition allows for the existence and usage of **monies** other than what is defined as **currency** by legal tender laws. The realization that **money** and **currency** are not the same, takes the opportunity away to constrain **money** to only that what the government by legal tender laws declared to be **money**.

**Money that is not legal tender** An example of a **money** that is widely used and accepted as such are short term liabilities of commercial banks, we call **bankmoney** in this work. **Bankmoney** is specifically not included in **legal tender** laws of central governments and are therefore not **currency**, but they are used and accepted as **money** by virtually all economic entities, including - at least in part - central governments themselves for example for the purpose of non-cash tax **payments**.<sup>25</sup> The way **bankmoney** is created is briefly discussed in section 8.

**Non-acceptance of some currencies by their own population** Furthermore if **currency** and **money** were used synonymously, there was no way to explain how in certain cases the acceptance and usage of the currency, i.e. what the government declared to be money, was rejected by the population. In these cases the population insists on other **monies**, mostly foreign **currencies**. We expand on this point in section 6.3.3.

**Money at the heart of every payment system** Concerning **money** - that is at the heart of any **payment system** - it seems necessary to attempt a more thorough analysis that we pursue throughout the ensuing section 6 by employing a requirements analysis approach for a **payment system** and thereby introducing the concept of **money of account** and of **money proper**.

---

<sup>25</sup>Cf. e.g. “Pay by Check or Money Order” - <http://www.irs.gov/Individuals/Pay-by-Check-or-Money-Order> - accessed July 5th 2014.

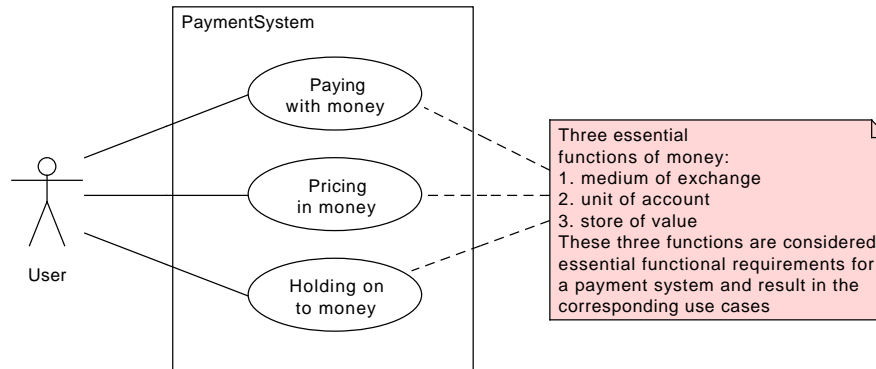


Figure 5: Use case diagram on the functional requirements for a **payment system** concerning **money**.

## 6 Requirements analysis

In this section we do a requirements analysis for a yet to design - totally hypothetical and software-intensive - **payment system**. It is a mock requirements analysis of course, since there are no actual customers or **stakeholders** to be asked or interviewed in this work. We will be focused here on functional requirements, even though usability, reliability and performance issues certainly play a role for any **payment system**.

We do this functional requirements analysis keeping in mind that there is the predominantly and currently used national and international **payment system**, based on nation states, central and commercial banks on one hand, we call it the **Banking Payment System (BPS)**, and on the other hand is the newly created **Cryptocurrency Payment System (CPS)**, that is the **Bitcoin** project. The latter claiming to have created “a new kind of money” - ‘new’ of course relating to the older, currently used **BPS** and its **money**.

We are doing a requirements analysis for a hypothetical **payment system**, with the aim to clarify our view on **money**, **money proper** and **money of account**. The **payment system** we think of is a software-intensive **system**, even though a **payment system** of a certain kind could provide for banknotes to be passed on, which theoretically allows for **payments** to be performed without any software support.

What **requirements** do exist for a successful **payment system**? A payment system facilitates **payments** by providing its users with a **money** that can be delivered to perform payments. We can therefore restate this question as: what is required to create something that is actually used and accepted as **money**?

### 6.1 Basic functions of a payment system

We take the three functions of money that were mentioned in section 5.1 (*medium of exchange, unit of account and store of value*) as a starting point for a requirements analysis. We consider these three basic functions as requirements that *have* to be fulfilled for the **payment system** to be applied as such by its users. If these three functional requirements are not fulfilled the **payment system** would not have anything that could be called ‘**money**’ and it would not be working properly for lack of usage and acceptance of its ‘**money**’. So these three basic functions can be thought of as what would be *shall requirements* in a software project.

Figure 5 shows these coherences in a use case diagram for a **payment system**. For reasons of simplicity the different types of users of a payment system are not shown in this diagram, they are going to be covered in section 6.1.1.

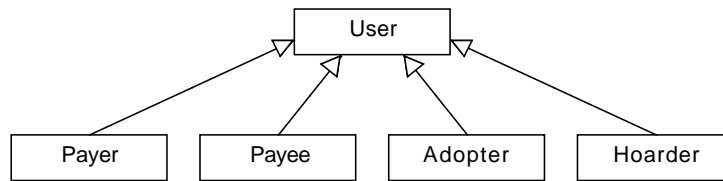


Figure 6: Object diagram on the different user types of a **payment system**.

### 6.1.1 Users of a payment system

While we have mentioned two types of users of a **payment system** in section 5.3, we want to explore the types in a bit more detail here. The four different types of users of a **payment system** that are identified here, are shown in figure 6.

**Payer** A payer is involved in the use case of *paying with money* (6.1.2) and therefore produces a set of requirements concerning the **money** of a **payment system**, since it is this **money** that is being transferred from payer to payee by the **payment system** to discharge debt for the payer.

**Payee** A payee is involved in the use case of *paying with money* (6.1.2) and does produce a set of requirements for the **money** of the hypothetical **payment system** we are analysing here.

**Adopter** An adopter is a term applied to the type of user of the **payment system** that adopts the **money** to price his assets, goods and services in it. The adopter therefore is involved in the use case of *pricing in money* (6.1.3).

**Hoarder** A hoarder wants to hold on to money now to spend it at a later point in time. The hoarder is involved in the use case of *holding on to money* (6.1.4) and has therefore certain demands to the **payment system's money** to enable this use case.

**'Pricing in' and 'paying with'** It is worthwhile to note already at this point that **money** is used here with different prepositions that are indicating distinct functions of **money**. While 'paying *with*' and 'holding *on to*' indicates an instrumental use of something that can be transferred or that can be held on to. 'Pricing *in*' on the other hand certainly does not refer to a *thing*. In section 6.2, about **money** as the 'medium of exchange', we will develop the idea of a **money proper**, which is the form of **money** that can be thought of as something that is changing hands, even though the actual 'thing' that is changing hands can be immaterial, like an entry in bookkeeping records. The notion of a **money of account** that reliably allows the nomination of contracts and to set prices for **asset**, goods and services, is described in section 6.3, where we analyse the money function of being a 'unit of account'.

### 6.1.2 Use case: Paying with money

To add more substance to the use cases we depicted in the use case diagram in figure 5 we want to write out fully dressed use cases based on the suggestions made by Cockburn (2004) concerning the writing of effective use cases. What we are calling 'Extensions' here, certainly could be expanded upon by 'Exceptions' and 'Variation Scenarios' as is suggested by Alexander & Beus-Dukic (2009) for systems in general. We will stick to the software modeling view here.

**Scope** Payment System

**Level** User Goal

**Primary Actor** Payer and Payee

### Stakeholders and Interests

1. Payer
  - a) wants his funds to be secure
  - b) wants to be able to safely and quickly transfer his funds anytime, anywhere in any amount available
2. Payee
  - a) wants his funds to be secure
  - b) wants to be able to safely and quickly receive funds anytime, anywhere in any amount sent to him

### Preconditions

\* Payment System is available and provides for a **money** that can be sent to fulfil payments

1. Payer
  - a) has sufficient funds to make the payment
  - b) has sufficient information about Payee
2. Payee
  - a) is reachable for Payer by means of the payment system

### Postconditions

1. Payer
  - a) has transferred funds removed from his account or transferred funds are not accessible any more
2. Payee
  - a) received transferred funds on his account or is able to access transferred funds

### Main Success Scenario

1. Payer inputs all data necessary into the payment system interface and confirms entry
2. Payment system processes order
3. Payee has transferred funds at his disposal

**Extensions**

1. Payer
  - a) does not have all data necessary about payee
    - i. acquire data about payee and retry payment
    - or
    - ii. end payment
  - b) does not have sufficient funds
    - i. Payment is declined for lack of funds
    - ii. end payment
2. Payment System
  - a) is not available
  - b) send status message about unavailability to maintenance service
  - c) maintenance brings payment system back on-line
  - d) publish payment system availability to users
3. Payee
  - a) did not receive the funds or does not have access to them
  - b) induces correction measures towards the payment system

**Paying with money** In section 6.2, we analyse in more depth what type of money is needed for the design of a payment system that makes this use case available for its users.

**6.1.3 Use case: Pricing in money**

**Scope** Payment System

**Level** User Goal

**Primary Actor** Adopter

**Stakeholders and Interests**

1. Adopter

**Preconditions**

1. Payment System provides a unit

**Postconditions****Main Success Scenario**

1. Adopter uses unit of the payment system
2. Pricing of assets, goods and services, as well as nomination of contracts in unit
3. contracts are successfully fulfilled

## Extensions

3. a) Contractual nuisances happen, e.g. payment does not happen afterwards
- b) Adopter needs to seek remedy, e.g. legal remedies with third party intervener (i.e. governmental agency)

**Pricing in money** The use case just outlined might look trivial at very first glance, however **Extension 3.** makes it clear that there is a third party intervener needed, or rather an infrastructure that the **payment system** is embedded in, to ensure adopters of the **money** that their contracts nominated in the **money** (actually it is the **money of account**, as will be explained in **6.3**) are going to be fulfilled by the second party to the contract. This infrastructure is going to be named the **Money Meta Infrastructure** and is going to be explored in section **6.6**.

### 6.1.4 Use case: Holding on to money

**Scope** Payment System

**Level** User Goal

**Primary Actor** Hoarder

#### Stakeholders and Interests

1. Hoarder
  - a) wants to hold on to money
  - b) does not want his money to lose value significantly
  - c) wants to spend money at a later point in time or stash it away for deliberations on liquidity and security

#### Preconditions

1. hoardable **money** is provided by the **payment system**

#### Postconditions

1. Hoarder has **money** at his disposal that he can make **payments** with at any time

#### Main Success Scenario

1. Hoarder holds on to **money** for as long as he wants
2. Hoarder spends **money** at a later point in time

**Holding on to money** This use case is going to be further explored in section **6.4**, where we inquire the function of money called ‘Store of value’. It is going to be revealed that this use case cannot be available if the requirements for the other two use cases, ‘paying with money’ and ‘pricing in money’ are not fulfilled. This is going to be explained in section **6.5** on the hierarchy of requirements.



## 6.2 Medium of exchange

As one of the three essential functions of money, as for example proposed by (Mankiw, 2007, pg. 22f.), money needs to serve as a ‘medium of exchange’. If we think of **money** just as a medium of exchange, it is tempting to view money as a mere transitory item, in the sense that it is not the earnings of money that is the target of a sale, but the real target is the next item’s purchase that is afterwards paid with the money acquired by the previous sale. We want to use a very simple model to first depict a so called barter trade and then adding more detail to this bartering process by including money, thereby making **money** conceptually accessible to our inquiry.

### 6.2.1 Barter exchange

Karl Menger<sup>26</sup> views man as trying to improve his current situation by way of exchanging his assets or services for others he prefers. Menger writes:

“[E]ach man is intent to get by way of exchange just such goods as he directly needs, and to reject those of which he has no need at all, or with which he is already sufficiently provided.” (Menger, 1892, pg. 242)

With  $A$  representing an **asset**, a barter exchange of two **assets** could be symbolized as

$$A_1 - A_2. \tag{1}$$

In this barter exchange there is - at first - no need for any **money**.

### 6.2.2 Money - facilitating barter trades

The idea of man trying to improve his situation by way of exchange is an old one, but is still considered relevant for monetary theory, because an actual barter trade is considered a highly unlikely transaction for the reason of “the double coincidence necessary to an act of barter” (see Jevons, 1875, Chap I par 6).<sup>27</sup> Or as Menger puts it:

“Consider how seldom it is the case that a commodity owned by somebody is of less value in use than another commodity owned by somebody else! And for the latter just the opposite relation is the case. But how much more seldom does it happen that these two bodies meet!” (Menger, 1892, pg. 242)

Money alleviates this double coincidence, therefore money is considered a facilitator of these kinds of barter trades by reducing transaction costs. (see Brunner & Meltzer, 1971, pg. 786ff.)

Money in this view is merely seen as a means to an end, the end being the sale of the asset  $A_1$  and finally the acquisition of another asset  $A_2$ . **Money** however does not play a role itself in this view, it is merely facilitating a barter trade. The bartering being the exchange of  $A_2$  for  $A_1$  and vice versa. If there was no money at all, then it would be an actual barter trade.

<sup>26</sup>Menger is mentioned here as just one of many classical, neoclassical or Austrian - referring to the Austrian school of economic thought - economists that could be cited here, including Adam Smith, Leon Walras, F. A. von Hayek et al.

<sup>27</sup>There is no need to have two distinct goods to show the lacking double coincidence of wants. Having one good to barter between two parties is sufficient to show that there is a problem with bartering in itself that can only be overcome by a recognized and used medium of bookkeeping (i.e. **money of account** and credit nominated therein), as is shown by Ostroy & Starr (1988).

### 6.2.3 Lifting the veil of barter

If we model the process that was shown in model 1 with a little more detail, we insert the so called ‘medium of exchange’ money, symbolized by  $M$ .

$$A_1 - M - A_2 \quad (2)$$

For being able to look at the actual process in even more detail, we want to split this process in the subprocess of ‘Paying’

$$M - A_2 \quad (3)$$

in which the holder of  $M$  transfers  $M$  to the owner of  $A_2$  and therefore becomes the new owner of  $A_2$ , and the subprocess of ‘Being paid’

$$A_1 - M \quad (4)$$

in which the initial owner of  $A_1$  releases  $A_1$  to the holder of  $M$  and in turn gets  $M$ . Obviously these two subprocesses are effectively identical and could be viewed vice versa as well. We want to emphasize however that there is an actual process of buying and paying happening that sometimes does get lost, if we think of money as a mere facilitator of barter trades. Since we are interested in this work in **payment systems** we have to look at these processes in detail, thereby making **money** visible. We cannot afford to gloss over these processes by the ‘veil of barter’ (see [Heinsohn & Steiger, 1989](#)), because we would be losing track of the **money** in doing so.

### 6.2.4 Final settlement of debt

While the notion of money being a ‘**medium of exchange**’ seems to be predominant in the more recent literature, we prefer the term ‘**means of payment**’ by Hicks, since this term accounts for the process of **payment** that is behind every perceived ‘barter’ transaction (see [Hicks, 1967](#), pg. 1). In section 5.3 we concluded that a **payment** is the “delivery of **money**” thereby “discharging the debt”. In other words **money** has to fulfil the requirement of being the means of final **settlement of debt**.

### 6.2.5 Money proper

In conclusion of this section on the requirement called ‘medium of exchange’, we want to emphasize here that ultimately **money** is not only a medium of exchange, but it is the means of final **settlement of debts**. This may sound trivial, yet it is important enough to mention specifically: it is the ability to provide final **settlement of debts** that a **payment system** has to be able to provide for being a payment system after all. This aspect of **money** that is being a means of final **settlement of debt** without discount is what we call ‘**money proper**’ in this work.<sup>28</sup> We consider the yielding of a **money proper** an *emergent property* of a well functioning **payment system**.

## 6.3 Unit of account

The function of money as a unit of account comprises the usage as a unit to price **assets** and services, to define contracts and to make calculations (see [Mankiw, 2014](#), pg. 323). In this section we are going to explore under what circumstances a unit of account is actually used and therefore becomes a **money of account**, thereby fulfilling this functional requirement for **money**.

<sup>28</sup>While ‘money proper’ was used by J.M. Keynes in another context - he used the term to differentiate government backed ‘proper money’ from ‘bank money’ - we are using the term **money proper** for the purpose of this work synonymous to what is accepted without discount as the actual means of payment.

### 6.3.1 Pricing is voluntary

In short one could argue: money is used as a unit of account if households, organizations and businesses typically have the prices of **assets** and for goods and services denominated in this unit. As was mentioned in footnote 19 even for **currencies** that are **monies** backed by legal tender laws of central governments, it is not mandatory for private households or businesses to price their **assets** in the corresponding **money of account** of the **currency**. If the right amount of **money proper** of a **currency** - that by definition is declared as legal tender - is offered to a creditor as a means of **payment**, the debt is considered to be discharged. This assertion by a central government does mean that debts, already established *and* nominated in the **money of account** of the currency, are legally discharged if the **money proper** of the **currency** was offered as **payment**. The assertion does *not* mean that private households, organisations or businesses are *obliged* to offer their assets in prices nominated in the **money of account** or use the **money proper** of the currency. Choosing a certain **money of account** for the pricing of an **asset** is a *voluntarily* made decision, albeit a momentous one, as we shall see.

### 6.3.2 Choosing a unit of account

The circumstances that are necessary to motivate economic units to actually do positively decide to use a certain unit as their unit of account are not obvious and have to be investigated. We could cut this inspection short, by simply stating: if the government says it's money, it has to be used as such. But we now know this isn't even true for **currencies**, since legal tender laws only require acceptance of a certain **money proper** as means of final settlement of debt, if a debt is due that was established using the corresponding **money of account**. The use of the **money of account** however is voluntary, as we have just seen in 6.3.1. This limited view is especially useless for our discussion revolving around **cryptocurrencies** that - at least so far - have not been declared **legal tender** anywhere by any government. In the following we are going to explore potential motivations for users to voluntarily use a certain unit of account as their **money of account**.

### 6.3.3 Potential motivations to choose a unit

It seems reasonable to assume that a certain stability in the price level is necessary for a business to be motivated to price its assets in a certain unit of account for it is not only laborious to reprice its **assets** continually but it is also a price risk that the business has to account for. Especially if it has negative cash flow from costs that are to be paid in a different unit of account, every argument concerning the hedging of foreign currency risk for businesses applies here as well.

An example would be a business that has to make the decision to price its assets in **bitcoin** or not. If the business does have costs that are to be paid in some other currency, e.g. EUR or USD, a high volatility in **bitcoin** prices is a foreign currency risk for this business and can therefore inhibit

It further seems to be a great motivation for a business to use a certain unit of account, if their own costs, at least significant parts, are to be covered with **money proper** of said unit. This is an important issue for us in this work, since **bitcoin** is still rarely used as the unit of account by businesses and it is also true that there is not a lot of costs to be paid that are nominated in **bitcoin** as the **money of account**. For example neither electricity costs can be covered by transferring **bitcoins** to a utility company nor are any salaries paid in **bitcoins**.<sup>29</sup>

A much cited mainstream economist is Nicholas Gregory Mankiw, who wrote in 2014 about the acceptance of **fiat money**:

<sup>29</sup>Notably **bitcoins** are created as **outside money** without the need for a debtor to pay back principal and interest from day one. Like electricity costs and salaries, interest and principal payments are very relevant costs for businesses. This might be a mere correlation, but it is a notable one that demands further investigation.

“To a large extent, the acceptance of fiat money depends as much on expectations and social convention as on government decree. The Soviet government in the 1980s never abandoned the ruble as the official currency. Yet the people of Moscow preferred to accept cigarettes (or even American dollars) in exchange for goods and services because they were more confident that these alternative monies would be accepted by others in the future.” (Mankiw, 2014, pg. 324)

An explaining example for **fiat money** that Mankiw talks about here, is the proverbial paper note that has no value as a physical thing, except that of a piece of paper. In the way that it has no intrinsic value per se the **money proper** of **Bitcoin** is certainly **fiat money**, too. However it is a term used to indicate that the **money** is backed by government decree, meaning legal tender laws, and therefore stands in this context for what we call **currency** in this work. While it may have no value as a mere physical thing, **fiat money** certainly does have value as **money** if it is used and accepted as such. Again the issue arises under what circumstances a **money** is used and accepted.

Concerning this matter Mankiw mentions ‘expectations’ and ‘social conventions’ that usually impart ‘value’<sup>30</sup> to a **money**. According to Mankiw these mechanisms seem to have failed for the example that is given in the Russian ruble. The Soviet government may have issued legal tender laws at the time, but people did not want to use the ruble as their unit of account any more for lack of ‘expectations’ and ‘social conventions’, if we follow Mankiw’s terminology here. We want to look at these ‘expectations’ and ‘social conventions’ in more detail to further comprehend this functional requirement for **money** called ‘unit of account’.

#### 6.3.4 Specification of contracts

The tagging of **assets**, goods or services with a price in a certain unit of account is nothing else than an offer to sell the **asset** to a willing buyer. It is one side of a purchase agreement that already stands there for any willing buyer to complete it. A purchase agreement is a *contract* that involves a unit of account that is necessary for the contract’s specification. This is also true for credit agreements.

#### 6.3.5 Credit agreements

Credit agreements are another type of contract that require a unit of account to enable the contracts definition. Or looking at it from the other angle: if a unit of account is proposed and it is not used in credit contracts, it does not work properly as a unit of account and therefore is no **money of account**. Certain requirements have to be fulfilled for a unit of account to be used in credit contracts.

Creditors engage in credit agreements only under certain premises regarding the infrastructure that is set up around the unit of account. To delineate these requirements in detail would certainly exceed the scope of this work. However it is worthwhile to note that without certain specifics in the legal, civil and fiscal infrastructure the use of a unit of account in credit agreements would have to be declined by creditors due to risk considerations. We want to give one example that is very significant for **cryptocurrencies**: if a creditor has no means to effectively insist on contract settlement, by means of enforcement measures including eventual debt collection, then he just simply cannot engage in any credit agreement with any debtor for the reason that the debtor could potentially just refuse to pay back the loan and the creditor would be powerless to do anything about that.

<sup>30</sup>In the **money** context having value essentially means for the **money** to be used and accepted as such. To be precise: it means the usage of the **money of account** to specify contracts and the acceptance of the **money proper** as a means of payment without discount.

### 6.3.6 Discovering the Money Meta Infrastructure

For a business to be able to enforce any contract (including credit agreements) or to initiate debt collection against non-performing debtors, it has to rely on a legal, civil and fiscal infrastructure that we will call the **Money Meta Infrastructure (MMI)**. Without a properly working **MMI**, creditors have no means to enforce contracts and are therefore powerless against non-performing debtors. It is this powerlessness that would have to lead to creditors not want to become creditors in the first place, thereby totally inhibiting any financing activities. Without credit contracts, there can be no financing of anything and arguably, without financing there is not going to be much of an economy. Since without a proper **MMI** not many creditors are going to be willing to engage in credit contracts, a proper **MMI** can be considered a prerequisite for any economic activity. It certainly is a requirement for a functioning **money of account**. Even though default on contracts, as it was described here, is assumed to never occur in most formal macroeconomic models, by way of the implicit assumption of ‘transversality’, according to Goodhart, for the assessment we are trying to do in this work we do not have this luxury (see Goodhart, 2008, pg. 213). Defaults on contracts can and do happen in the real world, and will happen in the world of **cryptocurrencies**. This simple fact needs to be taken into account in an assessment of **cryptocurrencies** as a **money of account**, which is what we try to do by taking the **MMI** into consideration here.

### 6.3.7 Money of account

The requirement for **money** to be used as a unit of account depends on certain prerequisites that evolve around contracts and their enforcement or finally even the initiation of debt-collection against non-performing debtors. A **payment system** that is embedded within an infrastructure that provides for all the necessary requirements and therefore allows for the effective use of its **money** as a unit of account, such a **payment system** we consider to provide its users with a **money of account**. We use the term **money of account** if we are talking about a unit of account that fulfils all the requirements to be used as such, especially the embeddedness within a **MMI**.

## 6.4 Store of value

The type of users that want to store value in the **money proper** of a payment system, is called *hoarder*. They are not engaged in performing **payments** right now, but they expect to be able to fulfil payments with the money at a later stage and are therefore holding on to or *hoarding* it.

This implies that the two preceding functions of money - serving as **money proper** (explained in 6.2) and as **money of account** (explained in 6.3) - are expected by the hoarder to be fulfilled at the future point in time. This is effectively an expectation about future liquidity of the money that is stored. To clarify this function of money we want to take a look at **assets** in general that can potentially serve as a store of value.

### 6.4.1 Funding liquidity

Having the sufficient amount of **money proper** at the needed point in time is being *liquid*. Mehrling calls this type of liquidity **funding liquidity** (see Mehrling, 2010, pg. 68). It is the ability to make **payments**, when they are due, or to be able to pay somebody else to make them. The latter is also called the *rolling over* of debt (cf. Abel, 1992, pg. 4).

### 6.4.2 Market liquidity

Any **asset** can be used as a store of value, but assets are different in their **market liquidity** and therefore differ in their function as a store of value. The market liquidity of an asset is high,

if the holder of the asset is able to sell it quickly and thereby does not cause any substantial change in the price of the asset.

The asset with the highest imaginable market liquidity can be sold immediately at (virtually) infinitely high amounts, without changing the price of the asset. This asset is called **money**.

### 6.4.3 No new requirements

Important to note here is: if a **money** does fulfil all the requirements for being used as a **money of account** and being accepted as a **money proper**, there is little worry that the **money** will not also be used as a store of value for lack of liquidity or any other reason by hoarders that want to do so. This insight has the effect that no additional requirements for **money** originate solely from the functional requirement posed to **money** to have to serve as a ‘store of value’ for its users. This realization brings us to the hierarchy of functional requirements for **money**.

## 6.5 Hierarchy of requirements

We want to make the argument here that the three basic *functions of money* or *requirements* have hierarchical relationships. These relationships further explain the **requirements** for a smoothly running **payment system**, with the requirement that demands yielding of a used **money of account** being the most important requirement.

### 6.5.1 Pricing first, then payments

As explained in 6.3 the requirement for a **money** to work properly as a **money of account** leads to private households, organizations and businesses to *voluntarily* price their **assets** and services in said **money of account**. This pricing clearly has to happen before anyone can **pay** for **assets** and services with the corresponding **money proper**. This perspective emphasizes the importance to model the process of money as a ‘medium of exchange’ in enough detail to comprehend that what looks like a simple exchange of **assets** is a more complex process that involves **payment**. The **payment** in turn requires a pricing a priori. If there is no price nominated in a certain **money of account**, there can be no **payment** with the appropriate **money proper**. Or to put it in another way: if a seller does not offer his product at a price nominated in, say US dollars, then a purchaser will not be able to purchase the product using US dollars. It’s as simple as that, but with non-trivial consequences for the hierarchy of requirements for a **payment system**, making the general acceptance of a **money proper** dependent upon the widespread, voluntary use of the **money of account**.

### 6.5.2 Hoarding the money proper

The function to store value means effectively the withholding of payments now, to rather make payments later. We called this process the ‘hoarding’ of **money proper** in 6.4. However, if nobody would ever price any **assets** and services in a certain **money of account**, then saving or hoarding the money for later spending becomes irrelevant, since **payments** are impossible for lack of a price on any **asset** or service in said **money of account**. As an example may serve any bill, denominated in a today’s defunct **currency** (e.g. bills of the Rentenmark of Germany) that was hoarded at the time with the intent to spend later. For lack of prices in the defunct currency the attempt to spend the bill has to fail.

### 6.5.3 Conclusion on the hierarchy

If there are no prices in the **money of account** first, then there will be nothing to pay for with the **money proper**, neither now nor later. If there is no **money proper** to hold on to, then the



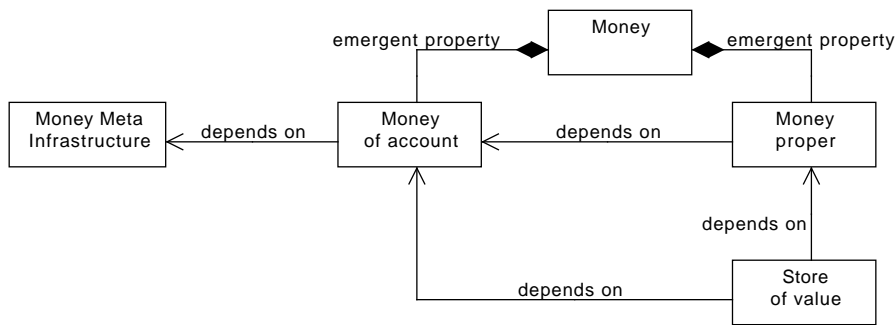


Figure 7: Domain model on the hierarchy of functional requirements concerning **money**.

function to be a store of value cannot be fulfilled. It is therefore fair to say the requirement that leads to a used **money of account** is hierarchically superior to the other two basic requirements for a **payment system** of yielding a **money proper** and a store of value. Figure 7 is representing these relationships in a domain model. It is also shown that we consider the **money of account** and the **money proper** as not only as some functional requirement but as emergent properties for the **money** of a **payment system**. A very important role plays the so far just very briefly described **MMI** that we want to enquire further in 6.6.

#### 6.5.4 Relevance for Bitcoin

This is highly relevant for the evaluation of **Bitcoin** as a **payment system** - generically we call this a **CPS** - since right now the pricing of **assets** and services by private households and businesses alike using **bitcoin** as their sole **money of account**, is not common. Even if a business starts to price their services in **bitcoin** - as lately did Expedia, Inc. - it does so in a limited fashion and is emphasizing its unwillingness to hold **bitcoin** for issues of price risk that they cannot afford to (or are unwilling to) bear.<sup>31</sup> This is an issue that needs further inquiry. Could it be possible that **Bitcoin** has a - potentially - ideal **money proper** and store of value, but lacks the requirement of having a used **money of account** for the reason of deficiencies in the **MMI**?

## 6.6 Money Meta Infrastructure

**Money Meta Infrastructure** (**MMI**) is a term given to all institutions, agencies and organizations that, as an entirety, provide processes that are supporting all kinds of services concerning **property**, contracts and **money**, including the potential enforcement of voluntarily engaged in contracts of various kinds against non-performing parties to these contracts.

### 6.6.1 Central hypothesis

The central hypothesis of this section is that without a properly working **MMI** creditors would not engage in credit agreements or similar contracts for reasons of risk avoidance. A potential creditor that has no means to insist on the fulfilment of obligations by potential debtors is not going to engage in the business of credit agreements in the first place. Ideally the means are so powerful and prevalent that creditors and debtors are totally aware of those means in a way that does not even make the deployment of those means necessary in a majority of cases. Within

<sup>31</sup>'Expedia to accept Bitcoin payments for hotel bookings' - <http://www.bbc.com/news/technology-27810008> - accessed June 27th 2014.

nation states the **MMI** is realized by agencies like the land register and executive institutions like bailiffs or the police.

### 6.6.2 Credit agreements

To substantiate the central hypothesis of this section confer figure 8, which contains an activity diagram showing the activities within a creditor-debtor relationship during the course of a credit agreement. In this simple case the relationship starts with the credit agreement as a contract that is agreed upon by two independent parties (Creditor and Debtor) voluntarily. A possible main success scenario for the activity of lending is listed below.

#### Main Success Scenario

1. Creditor and Debtor agree on terms of a credit agreement
2. Creditor pays out principal
3. Debtor pays principal and interest while these payments are due
4. Creditor received all principal and interest payments
5. Credit agreement is terminated

However we cannot just assume for every credit agreement to always be running smoothly and be terminated successfully every single time. For a multitude of reasons - that cannot be elicited exhaustively at this point - credit agreements can be willingly or unwillingly breached just as any contract can be. A contractual relationship may be breached in the sense that at least one party is unhappy with the performance of another party of the contract, because the party did not stick to the agreements or at least it is perceived as such by the opposing party. This is accounted for in figure 8 by displaying the option for the Creditor to initiate remedies, if the Debtor fails to perform according to his obligations.

The main success scenario for a credit agreement that is depicted above, does have extensions that have to be accounted for by the system. While figure 8 does not account for the Debtor being unhappy with the performance of the Creditor. This case is also accounted for in the list of extensions below.

#### Extensions

- \* at any time
  - a) Creditor is unhappy with performance of Debtor and seeks remedies as listed under 3.
  - b) Debtor is unhappy with performance of Creditor and seeks remedies as listed under 2.
- 2. Creditor is unwilling or incapable of paying out the full principal amount
  - a) Debtor is unhappy with performance of Creditor
  - b) Debtor seeks remedy for his situation
    - i. Debtor may inform regulatory agencies about Creditor performance
      - A. Regulatory agencies decide to intervene
      - B. Creditor performs as was agreed upon
      - C. Credit agreement continues



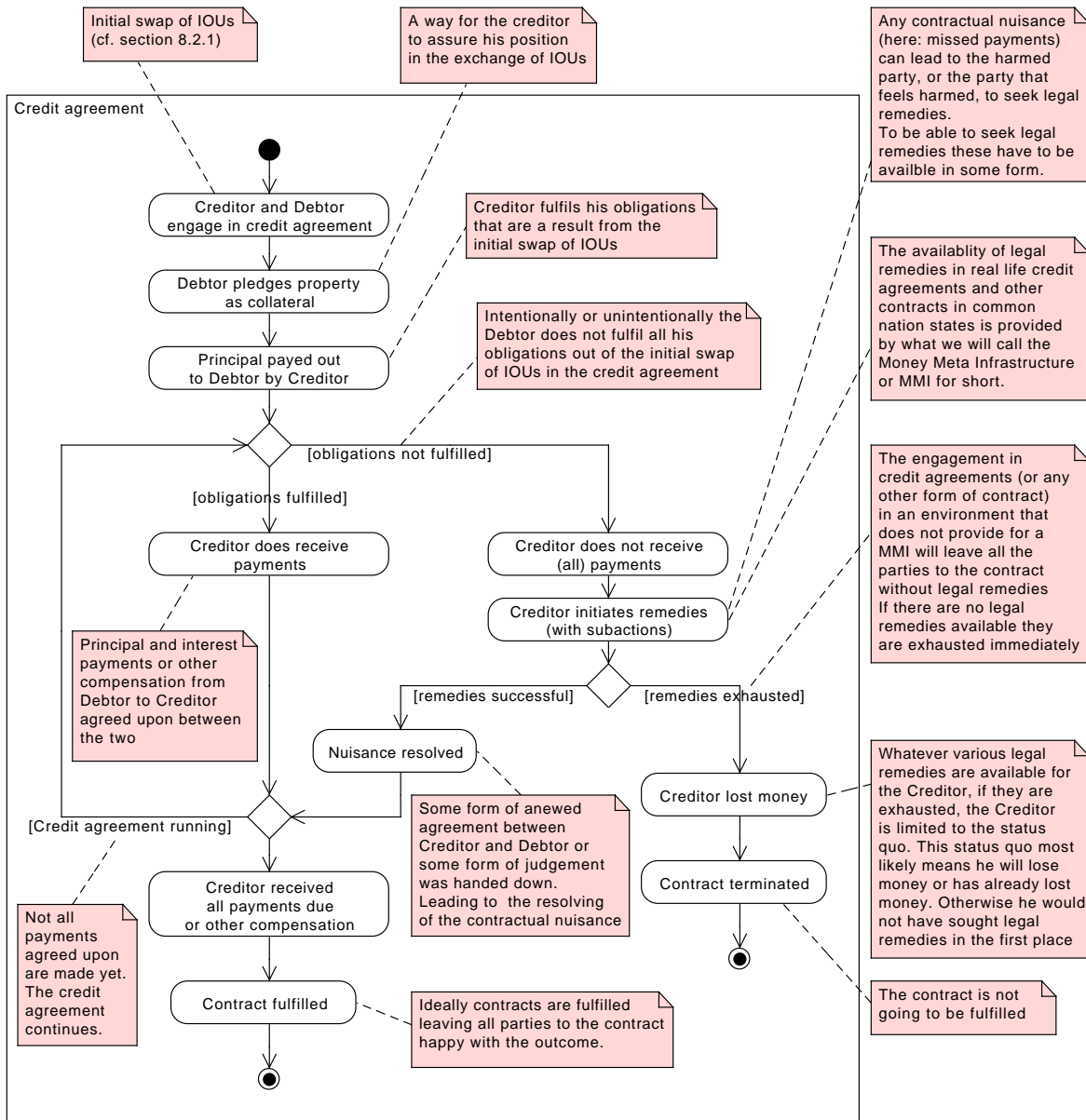


Figure 8: Activity diagram on a real world credit agreement.

- or
  - A. Regulatory agencies decide not to intervene
  - B. Creditor does not perform as was agreed upon
  - C. Debtor might withhold own payments of principal and interest and is going to seek further remedy, e.g. by suing the Creditor which will lead to further extensions
- ii. Remedies for Debtor are exhausted
- iii. Creditor and Debtor come to new agreement
  - A. Credit agreement continues
  - or
  - B. Credit agreement is terminated and Debtor is compensated
- 3. Debtor is unwilling or incapable of performing principal and interest payments when they are due
  - a) Creditor is unhappy with performance of Debtor
  - b) Creditor seeks remedy for his situation by renegotiation
    - i. Creditor and Debtor successfully renegotiate
    - ii. Credit agreement continues under new terms and conditions
  - or
  - i. Creditor and Debtor do not successfully renegotiate
  - ii. Creditor seeks further remedies
    - A. Creditor initiates foreclosure measures
    - B. Debtor is foreclosed upon by executive agencies
    - C. Credit Agreement is terminated
    - D. Creditor is compensated by funds out of foreclosure measures (e.g. sale of pledged **property**)
  - c) Remedies are exhausted for Creditor

The studying of the extensions listed under **2.** and **3.** makes clear that in case of any breach of contract the party that is harmed by the infringement does eventually have to look to third party intervention for remedy. Especially in the case of the non-performing Debtor under **3.** the Creditor has to rely upon third party intervention in the form of executive agencies that perform the foreclosure measures and an eventual sale of the **property** that was initially pledged by the Debtor as part of the credit agreement.

### 6.6.3 Interim Conclusion

The third party intervention in case of breach of contract and the pledging of **property** as collateral by a debtor is considered as being part of what we call the **MMI** in this work. Without this **MMI** a creditor is put at high risk to lose his stake, if the debtor is not willing or unable to perform his duties according to the terms of the credit agreement. A properly working and available **MMI** is a key requirement for a **payment system** to be able to yield a widely used **money of account**.

### 6.6.4 Relevance for Bitcoin

We deem these reflections on the **MMI** as highly relevant for the research objective of this work, which is to find out if **Bitcoin** created “a new kind of money”. If **Bitcoin**, as a **payment system**, is not embedded in some form of **MMI** (or does provide for the requirements in some form), its users are probably mostly unwilling to use its unit of account as a **money of account** to price their **assets**, goods and services in it, for reasons of risk considerations. To evaluate, if **Bitcoin** does account for these requirements that are posed to a **payment system** we now have to look at **Bitcoin** and its proposed **money of account** and **money proper** called ‘**bitcoin**’ in more detail.

## 6.7 Synopsis

Since this section is on one hand rather expansive on the other potentially not even detailed enough we want to try to recap the key insights of it here by showing the progression of understanding as the progression of the design of a use case diagram. The use case diagram we started this section on is shown again in figure 9, stripped of any comments.

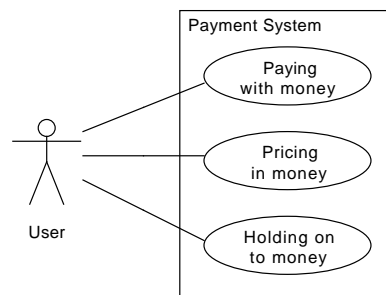


Figure 9: This is the use case diagram on the functional requirements for a **payment system** that we started this section with.

**Including understanding of money** If we add the understanding about **money**, the **money proper** and the **money of account** gained in this section to the detail of the use case diagram in figure 9, we receive the rather still unspectacular use case diagram shown in figure 10 below.

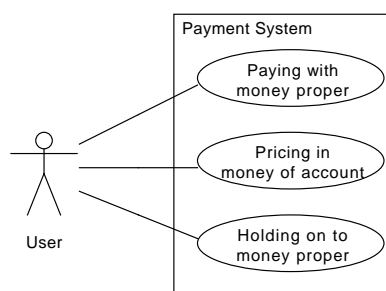


Figure 10: This is the use case diagram on the functional requirements for a **payment system** that is depicted with additional precision on **money** that was gained in this section

It seems like a negligible detail, but it will render very useful for our further inquiry of **Bitcoin**, to carefully distinguish the **money proper**, which is the means to effectuate the final **settlement of debt** by transferring it, from the **money of account**, which is the unit that contracts and prices are nominated in. This differentiation will allow for a much more precise analysis as just simply

talking about **money** without clearly noting if the **money proper** or the **money of account** is actually meant.

**Including the hierarchy** The next step was to get an understanding of the hierarchy on the functional requirements. If we take the findings on the hierarchy on requirements from section 6.5 into account, the use case diagram in figure 11 is the result.

**‘Holding on to money proper’ includes ‘Paying with money proper’** As it is shown by the dependency relationships marked with *«include»*, ‘*Holding on to money proper*’ is possible only, if a **money proper** does exist for a user to hold on to. A potential medium of exchange in turn is a **money proper** only, if it is actually accepted as the means of final **settlement of debt** without discount, which means nothing else but users being able to successfully ‘*pay with money proper*’.

**‘Paying with money proper’ includes ‘Pricing in money of account’** If a user of any **payment system** does have a proposed **money proper** in his hands, but there is nowhere any price tag to be found that is nominated in the corresponding **money of account**, the **money proper** he holds in his hands will be useless.

This is very blatantly visible in the case of a **CPS**. If a **Bitcoin** user holds on to the **money proper** called **bitcoin**, but there is nowhere anybody to be found that has put a price tag nominated in the **money of account** (called **bitcoin**, too!<sup>32</sup>), then this user will have to hold on to his **bitcoins** for he will not be able to perform any **payments**.

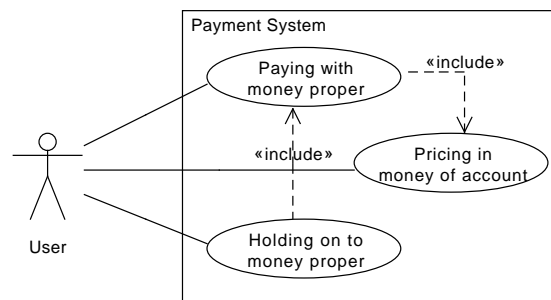


Figure 11: This is the use case diagram on the functional requirements for a **payment system** with additional detail on the hierarchy.

**Including the Money Meta Infrastructure** Finally, following advice from **Cockburn (2004)**, to try to identify actual supporting users from **stakeholders**, the **MMI** is added to the use case diagram as a supporting (or secondary) user, providing services to the **payment system** as described in section 6.6.

Again a rather insignificantly looking change in detail, depicted in figure 12 on page 36. Yet the identification and inclusion of the **MMI** as a significant supporting agent of a **payment system** will render useful in the analysis of the **Bitcoin** project as a **CPS**, for it is not self-evident that the **Bitcoin EcoSystem** does provide for a **MMI** that offers all the services that the **MMI** of the **BPS** does (and does so mostly unnoticed by users of the **BPS** and economists alike).<sup>33</sup>

<sup>32</sup>Eventually a useful distinction of **money proper** and **money of account** in the world of **Bitcoin** should be established. For example by agreeing to a convention in calling the **money proper** ‘**bitcoin**’ and the corresponding **money of account** either ‘**BTC**’ or maybe even ‘**XBT**’.

<sup>33</sup>The author of the work at hand owes a great amount of insights revolving around the **MMI**, without having it identified as such at the time, to **Theil (2000)** and **Theil (2001)** and ensuing discussions with Wolfgang Theil.

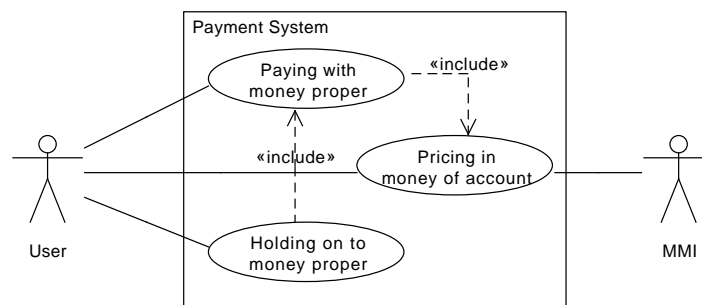


Figure 12: Use case diagram on the functional requirements for a **payment system** that has the **MMI** included as a supporting user.

## 7 Bitcoin

We will start this section with an attempted answer to the question what a **bitcoin** actually *is*, thereby introducing certain fundamental principles of **cryptocurrencies** that we will subsequently expand upon in section 7.2. The intent is to lay a thorough foundation of our understanding of **cryptocurrencies** in general.

In section 7.3 we will focus on the actual process of using **Bitcoin** as a **payment system**, looking in detail into the processing of **payments** within the **Network**. In this way **Bitcoin** is going to serve as an example for a **CPS**.

We are also going to explore other potential capabilities of the **Bitcoin Network** that exceed those of a mere **payment system**. Being embedded in what we call a **MMI** has been deemed an indispensable requirement for a **payment system** (see section 6.6). By decentralized measures **Bitcoin** might potentially provide for these requirements - or at least parts of them - that are accounted for by infrastructure provided by agencies of central governments for the competing **BPS**. This analysis is done in section 7.5.

### 7.1 Introducing the ‘coin’

Right upfront we want to attempt to answer the question what a **bitcoin** is. In this attempt we will have to introduce certain key concepts concerning **cryptocurrencies** that we will subsequently expand upon.

**A chain of digital signatures** Simply going by the name - *bitcoins* - one might tend to expect there to be some sort of coin, a thing or a token that is one bitcoin. This imagination can be very misleading, however if we follow the inventor of **Bitcoin**, there is indeed a sort-of ‘coin’ identifiable.

**A ‘coin’ defined** Nakamoto defines a ‘coin’ as “a chain of digital signatures” (see [Nakamoto, 2008](#), pg. 2). The ‘coin’ therefore changing hands by being digitally signed over from one owner to the next by including the next owner’s public key into the signed **transaction**. This process makes use of a digital signature algorithm, which were conceptually described in section 3.4.

**Transferring a ‘coin’ by example** Let’s assume Alice wants to hand over a ‘coin’ of this fashion to Bob. She can do so by digitally signing a section of data that we will call a **transaction** in this work. Into the digital signature on **transaction** *TA\_2* Alice will include a reference to **transaction** *TA\_1* that transferred the ‘coin’ to her beforehand, in the form of a **hash** of *TA\_1*, thereby creating a chain of **transactions**. Bob then passes the ‘coin’ on to Carol, by signing **transaction** *TA\_3*, again including a public key of Carol and the reference to *TA\_2* as a **hash** of *TA\_2*. Bob can only validly sign the ‘coin’ over to Carol (or anyone for that matter), because he controls the private key that matches the public key included by Alice into her signature in the **transaction** *TA\_2* from Alice to Bob.

We are illustrating this process in figure 13, which is entirely based on an illustration of Nakamoto (see [Nakamoto, 2008](#), pg. 2) with explaining comments added in.

**Double-spending suspicions** Carol, who received the ‘coin’ from Bob, by *TA\_3*, can so far only be sure that Bob was indeed allowed to pass the ‘coin’ on. He must have been entitled to pass on the ‘coin’, because his public key was included by Alice in *TA\_2* from her to Bob. But Carol cannot be sure if Bob did not also sign the ‘coin’ over to Dave by creating another **transaction** *TA\_3’* and including Dave’s public key into this transaction and thereby **double-spending** the ‘coin’. We will expand on **double-spending** later, when we are talking about the intentional ‘**forking of the blockchain**’ on page 44.

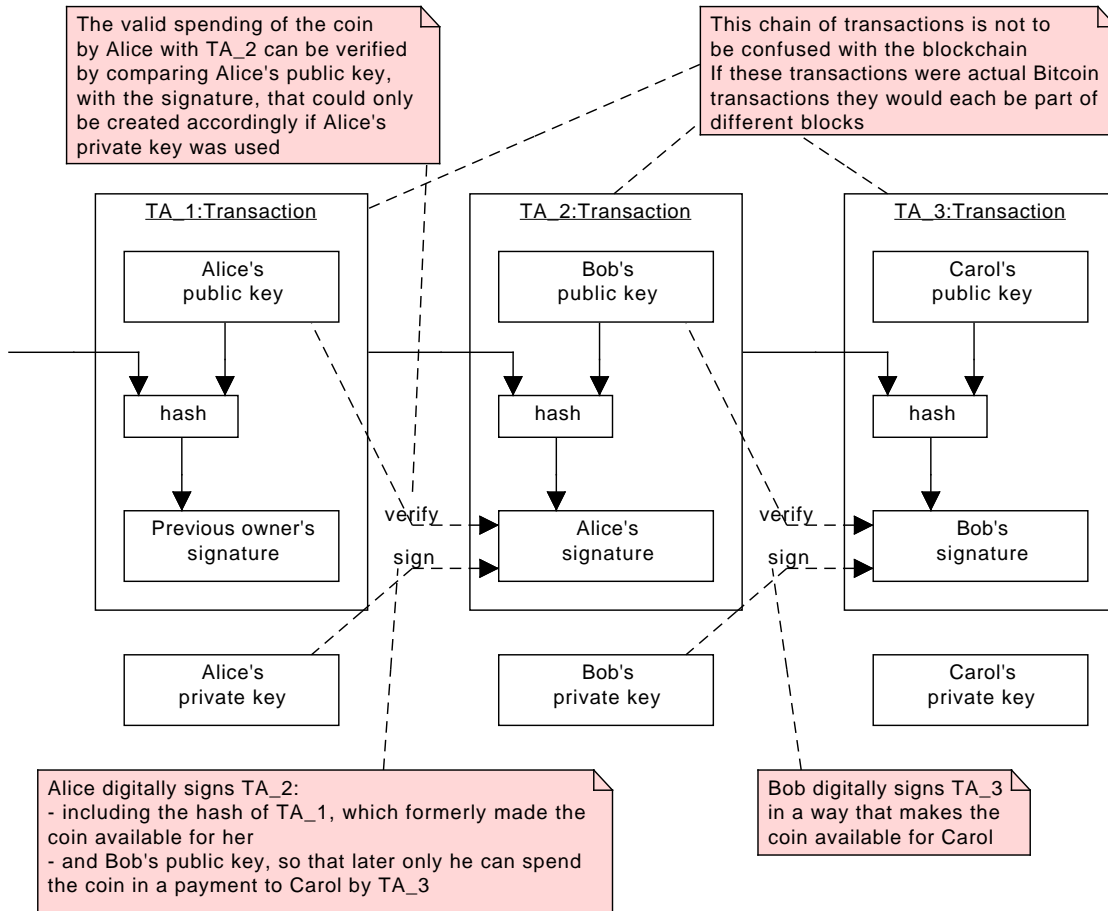


Figure 13: Diagram showing a coin, as a chain of digital signatures, passing on from Alice to Bob and from Bob to Carol.

**No central authority** One possible solution for Carol’s dilemma of her **double-spending** suspicions of Bob, is to insist that a trusted central authority validates all **transactions**. Only if the trusted third party confirms that Bob did not **double spend** the ‘coin’ and the **transaction** was valid, Carol would accept to be the real and confirmed owner of the ‘coin’.

The solution of one central third party that is trusted by all participants - often called a ‘mint’ - has been tried before, but Nakamoto justifiably notes on the merit of this proposal:

“The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.” (Nakamoto, 2008, pg. 2)

The central question for any digital cash scheme using ‘coins’ of any kind is:

How can **double-spending** be prevented?

It is this central question that **cryptocurrencies** for the first time try to answer differently than preceding digital cash schemes did in the past, for **cryptocurrencies** do *not* answer it by introducing a centrally trusted third party, i.e. a ‘mint’. There is no need for a ‘mint’ of this kind in **cryptocurrencies**, because they introduce a specific concept to validate the chronological order of **transactions** in a distributed fashion in a peer-to-peer **network**. We want to introduce this concept to prevent **double-spending** that **cryptocurrencies** utilize in the following paragraphs.

**Time-stamping transactions** Going back to our example, Carol can be sure that Bob did not **double-spend** the ‘coin’, if there was a time-stamp included in the **transaction** in a way that it can not be altered later on. If Carol can now see that there is no **transaction** by Bob concerning this ‘coin’ that is older than the **transaction** to her, she knows Bob did not double-spend the coin, or - if he tried to - Carol could prove that she was the legitimate owner of the ‘coin’ because the **transaction** granting the ‘coin’ to her is the oldest.

**Centrally authorized time-stamping** Obviously the type of time-stamping just described and its validation could be done via a central time-stamping server. However this - yet again - would provide a target for attack and a central point of failure for the whole system. And again **cryptocurrencies** of the **Bitcoin** kind provide a way to accomplish the time-stamping in decentralized fashion.

**A public ledger** The solution to decentralized time-stamping is the publishing of a **transaction**, including the time-stamp, within a ledger, we will call a **blockchain**. The public **blockchain** is going to be accessible to every potentially interested party and by not only containing all **transactions** but also the time-stamp of when they were created, this public ledger will contain all the information necessary to determine who currently is the legitimate owner of the ‘coin’.

The question remains how this public ledger can be protected against attempts to maliciously change it.

**Proof of work** To protect the public ledger against malicious change, it will be stored in a distributed fashion with every participant - we will call them **nodes** from here on - and a mechanism will be established that makes any change to this public ledger difficult.

1. There will only be one way to legitimately change the ledger (**blockchain**) and that is to expand it, the **blockchain** can never be shrunk
2. Any legitimate change, in the sense that the ledger grows, can not be made for single **transactions** but must be made in bundles of **transactions**, we will call **blocks**
3. The attachment of **blocks** to the **blockchain** can only be made after a ‘proof-of-work’ is carried out for each block (see [Nakamoto, 2008](#), pg. 3)

The ‘proof-of-work’ is going to involve a mathematical quest that will not be trivial to solve. The quest will consist of the **hashing** of the **block** in a way that the **hash** created of the **block** is smaller than a predefined number, we will call the **target**.<sup>34</sup> This action is going to be done by **nodes** that decide to participate in it. We will call these participating **nodes** from now on **miners** and the action they participate in **mining**. If any **node** ex-post facto wants to change the **block** in any way, has to redo the proof-of-word.

The successful **mining** of one **block** takes a specific amount of time by given computing power, so if the majority of **miners** are honest, the **blockchain** that is honest should grow the fastest and should be always the longest. Honest **miners** will always try to enlarge the currently longest **blockchain**.

**Separability of a ‘coin’** The ‘coin’ as defined by Nakamoto and used accordingly in this section is what is always ‘spent’ in total as described in the example above and illustrated in figure 13. The type of **transaction** used in our example here has exactly one input and one output. Either the ‘coin’ is passed on entirely from Alice to Bob or it stays with Alice: one input (coin is entirely owned by Alice), one output (coin is entirely owned by Bob). This principle would make a ‘coin’ the smallest denomination transferable.

<sup>34</sup>In **Bitcoin** not the entire **block** but the header of the block is hashed. See section 7.2.4.



**The bitcoin value** Now there shall be a certain value included into the ‘coin’ that could be distributed partially to Bob and the remainder to Alice. We will call this value the amount of **bitcoins**. We want to establish a type of **transaction** that allows for a transfer of a certain amount of **bitcoins** from Alice to Bob, without being limited to the nomination of one entire ‘coin’. This is what makes multi-input-multi-output **transactions** necessary.

**Expanding on the fundamental principles** We now want to expand on the fundamental principles of **cryptocurrencies** that were so far just touched upon in this section and thereby start our inquiry of the actual **Bitcoin** project.

## 7.2 Fundamentals

We are commencing the inquiry into **Bitcoin** as a specific **cryptocurrency** by explaining certain fundamental terms, principles and concepts concerning the **Bitcoin Network** that the **nodes** are building. These terms, principles and concepts were so far only briefly mentioned in the preceding section, if at all. We start this inquiry from the view of a prospective user.

### 7.2.1 Clients, nodes and wallets

The very first step for any user of the peer-to-peer **Bitcoin Network** is to start participating in it by becoming a part of it. This is done by installing a piece of software to a computer (or a similar device) that is called a **client**.

**Clients make computers into nodes** The entry ticket to the **Bitcoin Network** for any user is the open source **client** software that can be downloaded on [bitcoin.org](http://bitcoin.org).<sup>35</sup> It is by this software that a computer can become a **node** within the peer-to-peer **Bitcoin Network**. According to the peer-to-peer principle the **nodes** of the **cryptocurrency Network** are all equal in rights and duties.<sup>36</sup> This is a point that clearly differentiates **cryptocurrencies** from hierarchically designed **currencies**. The **client** does provide for a user interface that allows users of the **Network** to interact with it, for example initiate a **bitcoin transaction** from one **node** to another.

**Wallet** The **client** does create a file, which is called **wallet** that colloquially speaking ‘contains’ the **bitcoins** and holds them ready for the user to spend. However **wallets** do not literally contain **bitcoins**, rather they contain private-/public-key-pairs we call **transaction points** that in turn provide access to the **bitcoins**, as described in the following section 7.2.2.<sup>37</sup>

### 7.2.2 Transactions, addresses and transaction points

In the most general view a **transaction** in the **cryptocurrency domain** is a form of *communication* between **nodes** and is first and foremost simply “a signed section of data that is broadcast to the **network**”.<sup>38</sup> Users of the **Network** can initiate **transactions** with their **clients**, typically to transfer **bitcoins**.

<sup>35</sup>Various clients exist by various vendors and also for different platforms, including desktop wallets for computers, mobile wallets for smartphones and other devices and also web wallets that can be accessed using any web browser on any device. The standard client, which is the **client** that was published first and is since then developed by the original development team of the **Bitcoin** project, can be accessed through <https://bitcoin.org/en/getting-started> - accessed May 14th 2014.

<sup>36</sup>The nodes are all equal, provided they use the standard **client** or a client that implement every feature of the standard client.

<sup>37</sup>Sometimes the whole client software is confusingly named ‘wallet’ even though the **wallet** is just a file that, most importantly, contains the private-/public-key-pairs, we call **transaction points** - see <https://en.bitcoin.it/wiki/Wallet> - accessed July 18th 2014.

<sup>38</sup>see <https://en.bitcoin.it/wiki/Transactions> - accessed July 17th 2014.

The use of a **transaction** is to make **bitcoins** transferable from one **transaction point** to another. To do this, every **transaction** has at least one input and one output. The input can be seen as the source of **bitcoins**, it references the **transaction point** that already holds **bitcoins**, the output can be seen as the target for the flow of funds and is a **transaction point** itself. The input of a given **transaction** does always reference an output of a preceding **transaction**.

**Addresses and transaction points** On a keystroke of a user the **client** provides a public-/private-key-pair that we call a **transaction point** in this work, incurring a term here that was coined by Cap and used as a concept to explain the verification of **transactions** (see Cap, 2012, pg. 86). As an **Elliptic Curve Digital Signature Algorithm (ECDSA)**-key-pair, **transaction points** are conforming to digital signature algorithms, as they were described in section 3.4. The public-key is called **address**, the private-key of a **transaction point** does not have a specific name in the **cryptocurrency** context.<sup>39</sup> While being feasible, calling the public-key of the key-pair the **address** is not entirely technically correct, as the actual **address** is not plainly the public-key, but it is the **hash** of a multi-step **hashing** of the public key.<sup>40</sup>

In the most simple form imaginable a **transaction point** can be viewed as the combination of a bank account number, resembling the public key, and an according pin, which stands for the private key. The address, much like a newly created bank account, does initially not ‘contain’<sup>41</sup> any **bitcoins**, the ‘account’ (**address**) is empty upon creation by a **client**. It can be ‘filled’ by corresponding **transactions**.

The private key of the key-pair is used to digitally sign the transfer of **bitcoins** from a payer to a payee by including the payee’s public key into the signature, which is created using the private key of the payer. We will look at this process in greater detail, since it is an important part of the verification of **transactions** within the **Network**, which itself is at the very heart of it.

**Simple example** Going back to the example of one ‘coin’ passing on from Alice to Bob and from Bob to Carol, now just redone with a fixed amount of **bitcoins**, instead of one ‘coin’, and depicted with the inclusion of the simplifying concept of **transaction points**. The communications diagram in figure 14 is the illustration of the two **transactions**  $TA_1$  and  $TA_2$ , transferring a fixed amount of **bitcoins** first from Alice to Bob and then the same amount from Bob to Carol. Both **transactions** having one input and one output each. The inputs and outputs are, what we call **transaction points**.

In the beginning of this example Alice has already access to the **transaction point**  $TP_1$ , which means she is equipped with the private key of  $TP_1$ , i.e. by having it stored in her **wallet**. Alice now transfers all the **bitcoins** located at **transaction point**  $TP_1$  to Bob, by initiating **transaction**  $TA_1$ . She does so by including a **hash** of the previous **transaction**  $TA_0$  (not shown in the illustration in figure 14) and the public key of  $TP_2$ , which Bob has control over, into  $TA_1$  and then digitally signing it using her private key that granted her access to  $TP_1$ . Now Alice and Bob have to wait until  $TA_1$  is verified by the **Network** (as described below).

As soon as  $TA_1$  is verified, Bob commands the **bitcoins** located at  $TP_2$ . He decides to send them to Carol. To do so, Bob initiates a **transaction**  $TA_2$ . He includes into  $TA_2$  a hash of  $TA_1$  that provided him access to the **bitcoins** that are currently at  $TP_2$  and the public key of **transaction point**  $TP_3$  that only Carol has at her command. Carol must have sent the information about her public key to Bob or made it otherwise available to him. Consequently

<sup>39</sup>see <https://en.bitcoin.it/wiki/Address> - accessed July 17th 2014.

<sup>40</sup>The **hashing** of the public key, thereby making it an **address**, is done for multiple reasons, including the reduction of typing errors when inputting **addresses**. For more information about the technical background of **addresses** see [https://en.bitcoin.it/wiki/Technical\\_background\\_of\\_Bitcoin\\_addresses](https://en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses) - accessed July 18th 2014.

<sup>41</sup>While ‘contain’ may match the metaphor of a bank account, the **address** itself is just the account number and therefore does not ‘contain’ anything, except for the 27-34 alphanumeric characters it consists of.

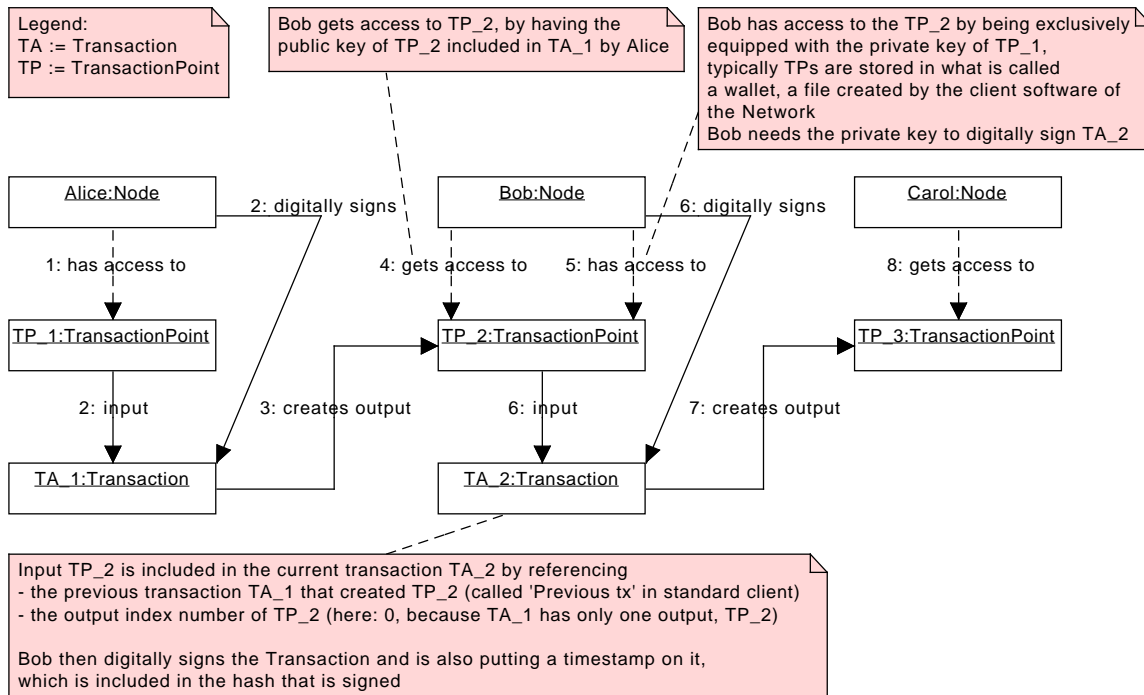


Figure 14: Communications diagram showing two single-input/single-output transactions  $TA_1$  and  $TA_2$  passing on a fixed amount of bitcoins first from Alice to Bob (steps 1. - 4.) and subsequently from Bob to Carol (steps 5. - 8.).

Bob signs  $TA_2$  with the private key of  $TP_2$  that only he has in his wallet, thereby making him the only node with the capability to create a valid transaction  $TA_2$  that attempts to send the bitcoins at  $TP_2$ . Now Bob broadcasts  $TA_2$  to the whole Network and Carol has to wait for the verification of  $TA_2$ , for her to have the bitcoins now at her disposal at transaction point  $TP_3$ .

**Verification of transactions** The verification of transactions is a key ingredient of the Bitcoin Network, especially if it is used as a payment system. Transactions can be employed to perform payments within the Network.

Transactions are verified in multiple steps:

1. The transaction is broadcast to the Network by the node that created it
2. Following the publication the transaction is added to the current block by all miners
3. By successfully hashing the current block that contains the transaction, it is added to the blockchain within the most recent block and thereby verified for the first time
4. Further confirmation happens with every additional block that is added to the blockchain after the block containing the transaction in question. This process is described in more detail in paragraph 'Forking on purpose' on page 44.

The following section 7.2.3 explains the central data type for storing information in the Network, called 'blocks' and the public data structure that stores generated blocks in a sequence, called the 'blockchain'. It is followed by section 7.2.4 that goes into details on bitcoin mining.

## 7.2.3 Blocks and the blockchain

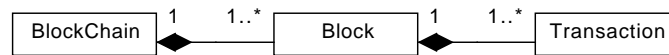


Figure 15: Conceptual class diagram that shows composition relationships between **transactions** and a **block** and between **blocks** and the **blockchain**.

We have so far used the concepts of **blocks** and the **blockchain** in our explanation, without providing any significant details about them. This shall be done as follows.

**Blocks** **Blocks** are data types that are employed to permanently store data within the **Bitcoin Network**.<sup>42</sup> The main reason to want to permanently store data with the **Network** is found in the way that **transactions** are verified by the **Network**, by permanently keeping all the information about them publicly accessible.

As is shown in figure 15 a **block** consists of multiple **transactions**, which must not be contained within any other **block**. Every single **block** does always reference exactly one preceding **block** by containing a **hash** of the referenced **block**. This feature is what creates the **blockchain** that therefore consists of multiple **blocks**.

The most recent **block** does contain some or (ideally) all **transactions** that have been broadcast to the **Network** but are so far not stored in any preceding **blocks** which are already part of the **blockchain**. To become part of the **blockchain** a **block** needs to be generated by a **mining node**. This process of **mining** is explained in section 7.2.4.

**Blockchain** A **blockchain** is a specific path within a tree data structure that consists of **generated blocks**, each referencing exactly one previously **generated block** as is shown in figure 16.

Successful **mining** adds the current **block** as the most recent to the **blockchain** making it a **generated block**, thereby rewarding the successful **miner**. The dominating **blockchain** is the longest one (determined by difficulty, not by mere number of **blocks**) that is relevant for the current distribution of all **bitcoins**. As mentioned in 7.2.2 all **transactions** that are stored within the **blocks** in the longest **blockchain** are considered verified. **Transaction points** are the *outputs* of **transactions**. Therefore the longest **blockchain** determines how many **bitcoins** currently are accessible at which specific **transaction point**. This is the motivation for **nodes** to add the **block** they are currently working on to the longest **blockchain**. Honest **miners** will always try to add the **block** that they are currently working on to the longest **blockchain**, making this **blockchain** the dominant one (see section 7.2.4 on **mining** for more details).

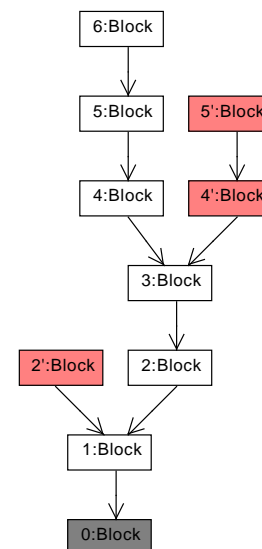


Figure 16: **Blockchain**, starting with the **genesis block 0** going to the most recently **generated block 6** on top. **Orphans 2', 4' and 5'** are red.

<sup>42</sup>see <https://en.bitcoin.it/wiki/Block> - accessed July 19th 2014.

Since the **genesis block** is the very first **block** that was ever generated it is the only **block** that contains no reference to a previous **block**. It is contained in every **blockchain** and therefore is always part of the longest **blockchain**, too. The **genesis block** is special in the way that it consists of only one **coinbase transaction** that has one input of 50 brand new **bitcoins** and one output (see 7.2.4 for more details).<sup>43</sup> The **blockchain** is being shared amongst all **nodes** participating in the **Network**, therefore all **nodes** do always know how many **bitcoins** are accessible from which **transaction point** or to translate this into terms that are well known from banking: every single **node** knows exactly how many **bitcoins** are on what ‘bank account’ by consultation of the completely public ledger that contains every relevant **transaction**. This public ledger is known as the **blockchain**.

**Forking the blockchain** It is worthwhile to mention that under certain circumstances it is possible for the **blockchain** to (temporarily) fork. One such possibility is the simultaneous or almost simultaneous successful **hashing** of the current **block** by two different **miners**, both referencing the same previous **block**. This is shown in figure 16, where blocks 2 and 2’ are both referencing the same **block 1**. This can occur theoretically completely unintentionally for latency issues within the peer-to-peer **Network**. The very next **generated block 3** however has to reference again exactly one previous **block**, 3 does reference 2 here, making the one specific **blockchain** containing the most recent block longer again (except the same circumstances happen again and the successful mining happens again virtually simultaneously), thereby **orphaning block 2’**. The **transactions** that are contained in 2’ but are not contained within 2 have to be rebroadcast to the **Network** and included again in the then current **block** for verification.

**Forking on purpose** Forking of the **blockchain** can be done on purpose by an attacking **node**, with the aim to maliciously revise the **transaction** history. The goal of intentional forking is to try to **double-spend bitcoins** by creating an alternate **blockchain** that eventually becomes the dominant one. We will illustrate this potential attack here using figure 16 on page 43.

Let’s assume Mallory and Alice are in a business transaction, where Mallory wants an **asset**, good or service from Alice.

1. Mallory offers to **pay** Alice using the **Bitcoin Network** as a **CPS**.<sup>44</sup>
2. Mallory first broadcasts an honest **transaction  $TA_h$**  to the **Network** that is sending **bitcoins** from a **transaction point  $TP_M$** , controlled by Mallory, to a **transaction point  $TP_A$**  that is controlled by Alice.
3. As  **$TA_h$**  gets verified the first time by being included in **block 4** that is added to the **blockchain**, Alice may already feel paid and therefore perform what was requested by Mallory. For example, Alice could send some physical good to Mallory as soon as she sees that **transaction  $TA_h$**  was verified the very first time.
4. Now Mallory, being the attacker, tries to successfully **mine block 4’** that contains virtually all the same **transactions** that **block 4** contains, except for **transaction  $TA_h$** , which is replaced by  **$TA_m$**  that sends **bitcoins** not from  **$TP_M$**  to  **$TP_A$**  but to another  **$TP'_M$**  that also Mallory controls.
5. At this point Mallory tries to add another **block 5’** to the **blockchain** before **block 5** is added. If she is successful, she has effectively created the longest **blockchain** that now contains ‘her’ malicious  **$TA_m$** .

<sup>43</sup>The actual **genesis block** of **Bitcoin** can be accessed as **block 0** of the **Bitcoin blockchain** at <http://blockexplorer.com/block/00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f> - accessed July 20th 2014.

<sup>44</sup>We will go into the details of using the **Bitcoin Network** as a **payment system** in section 7.3 on page 50.

This just briefly described attack to maliciously fork the **blockchain** on purpose is not very likely accomplished successfully. Other honest **miners** are trying to add **block 5** already, while Mallory is still working on **block 4** to be successfully **hashed** and added to the **blockchain**. So if Mallory is not lucky right at the beginning her chances of succeeding with this kind of attack are small. Mallory needs to convince a the receiving **node** (Alice) that she has sent **bitcoins** to her, by allowing at the very least one **block** containing the first **transaction** to be added to the still honest **blockchain**.

This fact of an attacker having to catch up with the honest **blockchain** makes this kind of attack very unlikely as is shown by Nakamoto (see [Nakamoto, 2008](#), pg. 6f.). Still a potential attacker could get lucky and it is therefore recommended to wait more than just one confirmation, to make the number of additional honest **blocks** a potential attacker has to jump almost impossible. The probability  $P$  for an attacker to be successful in the way described decreases exponentially with the number of honest **blocks**  $z$  he has to overcome. For example waiting not only for the first confirmation but for six additional **generated blocks**, making the total waiting time roughly 70 minutes, the probability for the attacker to be successful drops to  $P = 0.0000647$  with  $z = 7$  (see [Nakamoto, 2008](#), pg. 8).

The **blockchain** as it is described here and the corresponding algorithm creating it and making it work is considered the main innovation of **Bitcoin**.<sup>45</sup>

### 7.2.4 Mining

So far we have just stated that successful **mining** creates a **generated block** and that it is then added to the **blockchain**. We have not yet explained what **mining** is, why **miners** are keen to do it and how it can be done successfully. This explanation is going follow here.

**Block hashing algorithm** If a node decides to start **mining bitcoins** it executes the following algorithm:<sup>46</sup>

1. Harvest from the **Network** as many **transactions** as possible that are so far not part of a **generated block** in the longest **blockchain** and are therefore not verified yet.
2. Validate newly harvested **transactions**. Check for inconsistencies in signatures, script and hashes and for **double-spending** of **bitcoins**, especially by comparing the timestamps of the inputs of the broadcast transaction with those in the locally stored **blockchain**
3. Put new validated **transactions** into the current **block** (and forward them to the peer-to-peer **Network**) and discard invalid transactions
4. Create new hash of all validated transactions<sup>47</sup>
5. Put the hash of the previous **block** (hashPrevBlock), the hash of all current transactions (hashMerkleRoot)<sup>48</sup>, a timestamp, the current **target** and a random number called **nonce** into what is called the block's header
6. **Hash** the current block's header
7. Compare the just created **hash** to the **target**

<sup>45</sup>see <https://en.bitcoin.it/wiki/Blockchain> - accessed July 19th 2014.

<sup>46</sup>see [https://en.bitcoin.it/wiki/Block\\_hashing\\_algorithm](https://en.bitcoin.it/wiki/Block_hashing_algorithm) - accessed July 20th 2014.

<sup>47</sup>this hash is called hashMerkleRoot, because it is the root of a Merkle hash tree. Merkle trees were mentioned as an application of cryptographic hashing in 3.3 and are briefly depicted by Nakamoto (see [Nakamoto, 2008](#), pg. 4).

<sup>48</sup>By putting just a hash into the block's header and not all transactions, the time needed for hashing the header is constant, whether the block contains 1 or 1000 transactions.



- a) If the hash is greater than the **target**, then start again at step 1., making sure that at least the **nonce** is incremented and thereby changed to receive a totally different hash for the current block's header.
  - b) If the hash is equal to or smaller than the target, proceed to step 8.
8. The **miner** was successful.
  9. The successful **miner** broadcasts the complete block to the **Network** including the successful nonce.
  10. Other **miners** in the **Network** check the broadcast **block** by rehashing it with the given values. The approval of a once successfully **generated block** is simple compared to the **mining**.
    - a) If all **transactions** are valid and do not **double-spend** and the **hashing** with the proposed nonce yields a hash that is smaller than the target proceed to step 11.
    - b) If there is any problem with the proposed **block**, disregard it and continue with step 1.
  11. Other miners express their acceptance of the newly **generated block** that was just broadcast to the **Network** by adding it as the most recent **block** in their **blockchain**, too. They are referencing now this newly **generated block** in the hashPrevBlock field of their current block and start again from step 1 completely updating their current block.
  12. The successful miner receives newly created **bitcoins** in a **coinbase transaction** that now is part of the **blockchain** of all approving **miners**.

The block hashing algorithm depicted here is based on Nakamoto's proposition (see [Nakamoto, 2008](#), pg. 4) and on analysis from Hobson (see [Hobson, 2013](#), pg. 42) and [Nielsen \(2013\)](#). To further the understanding of entities involved in mining we created a simple object model that is show in figure 17.

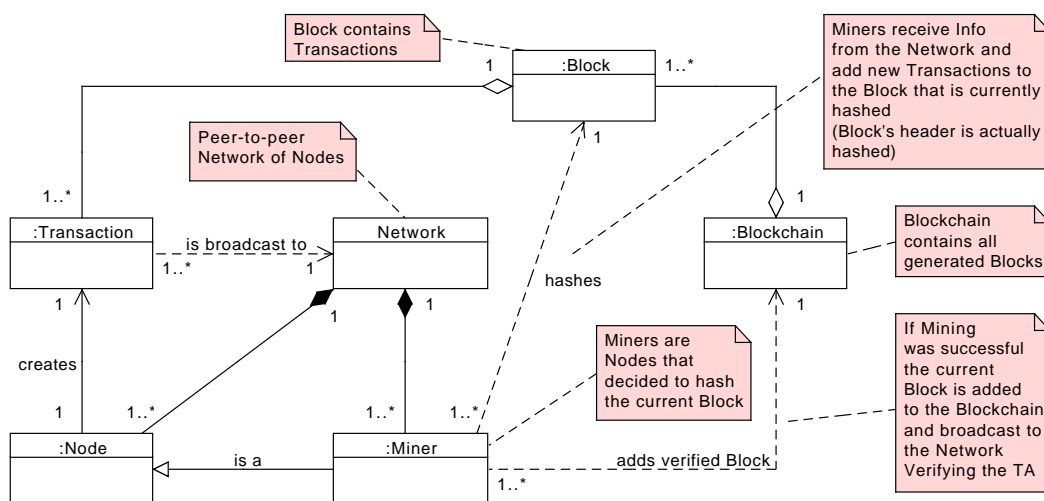


Figure 17: Object model showing the relevant objects and their relations involved in the **hashing** of a **block** (**bitcoin mining**).

**Creating bitcoins** New **bitcoins** are created as a so called **coinbase transaction** in every **generated block**. Within this **transaction** the **node** that generated the **block** first receives newly created **bitcoins**. In **Bitcoin** terminology these additional **bitcoins** are ‘**mined**’. The initial reward was 50 **XBT**. Currently (July 2014) it has dropped to 25 **XBT** per **coinbase transaction** as can be seen in figure 18 that contains a screen shot of the **coinbase** of an actual **block**, block #312286.

Transactions				
Transaction <sup>2</sup>	Fee <sup>2</sup>	Size (kB) <sup>2</sup>	From (amount) <sup>2</sup>	To (amount) <sup>2</sup>
<a href="#">55a2e17c65...</a>	0	0.122	Generation: 25 + 0.01541 total fees	<a href="#">JH1sq6Mst9HjRrBuz8ieZdThzWXS6oPVA</a> : 25.01541

Figure 18: Screenshot of a **coinbase transaction**, created with a **block explorer**.<sup>49</sup>

The **coinbase transaction**, rewarding brand new **bitcoins** to the **miner** is the incentive to provide computational power, which is obviously required to engage in **mining**. It is this incentive that should keep malicious **miners** or attackers from defrauding honest **nodes**, because instead of cheating they might as well choose to gain from being honest, thereby strengthening the overall health of the **blockchain** and thereby the **Network** as a total.

Nakamoto puts it this way:

“If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.” (see [Nakamoto, 2008](#), pg. 4)

So, to offer an incentive to **miners** by awarding them new **bitcoins** is an important feature of **Bitcoin**. We might assume that a drop in incentive might result in a drop in the soundness of the **Network**.

**Constant rate of generated blocks** We have already mentioned the **target** as a property of a **block**. A successful **miner** needs to create a hash of the current **block** that is equal to or lower than the **target**. This **target** (and thereby also the **difficulty**) is adjusted by a so called retarget every 2016<sup>50</sup> **generated blocks** in such a way that the **generation of a block** happens at a *constant rate* of roughly one **block** every 10 minutes. This is also the average time it takes for a **transaction** to be verified once.

**Total bitcoin supply** The total amount of **bitcoins** that will ever be **mined** was determined by Nakamoto and subsequently by the software development team that currently maintains the standard **client** to 21 Million **bitcoins**. This effectively means that at some point no new coins are going to be rewarded to successful **miners** any more, thereby obviously changing the incentive for **miners** at the latest at that point in time.

The date of this happening is more than a hundred years away, since the **coinbase** rewards are halved every 210.000 **generated blocks**, which at the current rate of one **generated block** every ten minutes corresponds to roughly 4 years for each bisection of rewards. This leads to an asymptotic approach of the targeted 21 Million **bitcoins** that will be never reached. Ceteris paribus **mining** will stop to create new **bitcoins** right around 2140, for the very last bisection

<sup>50</sup>see <https://en.bitcoin.it/wiki/Difficulty> - accessed July 22nd 2014.



from 0.00000001 XBT, or 1 satoshi will drop to 0 satoshi with the completion of generated block #6,929,999.<sup>51</sup>

**Transaction fees** What is also visible in figure 18 is that additionally to the reward that created 25 brand new bitcoins in the coinbase transaction the miner is rewarded - in this case - additional 0.0194147 bitcoin in transaction fees. This is equalling roughly \$ 12 at a price of \$ 625 per bitcoin, which was the price of the latest trade on <https://www.bitstamp.net/> on July 20th 2014. The fees are paid by the creators of the 132 other transactions that are contained in this very block.<sup>52</sup>

**Incentive for mining after coinbase transactions stop** As explained above in the paragraph on creating bitcoins, mining is critical to confirming transactions and therefore to the overall health of the blockchain and the Network, it cannot just stop or even fall dramatically however far away this point in time is, if the Network is not supposed to stop at that time, too.

The answer to this pending question are transaction fees that will have to be high enough to keep mining profitable, but low enough to not take away the feature of Bitcoin of being able to offer transactions at competitive transaction costs for its users. Therefore the sufficient motivation of miners especially after coinbase transactions stop, is a non trivial issue that, if not successfully provided, also might make Bitcoin potentially vulnerable to a revision of the entire blockchain, called a history revision attack (see Barber *et al.* , 2012, pg. 405). This type of attack on the soundness of the blockchain was in principle already described in this work as the intentional forking of the blockchain. If the incentive for miners however becomes so low to allow a successful ‘doomsday’ history revision attack, then the intentional forking does not only allow for the double-spending of a limited number of bitcoin, but the manipulation of big parts of the entire blockchain is meant. A successful history revision attack of this kind might not only harm single users but could potentially endanger the whole Bitcoin project .

Ideas on how to potentially prevent this type of attack even though mining will at some point stop to create new bitcoins as incentive are given by Barber et al. (see Barber *et al.* , 2012, pg. 406f.).

### 7.2.5 What a bitcoin is eventually

All of the above put together means that our current understanding of a bitcoin is that it is not a token or coin and certainly not a physical thing. A bitcoin is, above all, a real number with eight decimal places located at a specific transaction point.<sup>53</sup> The tamper-proof public ledger called blockchain that contains these numbers in blocks determines how many bitcoins are accessible at each transaction point.<sup>54</sup>

Superficially treated this seems to very much resemble the type of money that is created and managed by commercial banks. They create bankmoney in their ledgers, by an exchange of IOUs with their debtors. This type of money, also seems to be ‘just’ a number that can not be tampered with easily, albeit not prevented by the public blockchain and the corresponding cryptographic protocol, but by a central authority called a bank.

<sup>51</sup>see [https://en.bitcoin.it/wiki/FAQ#How\\_long\\_will\\_it\\_take\\_to\\_generate\\_all\\_the\\_coins.3F](https://en.bitcoin.it/wiki/FAQ#How_long_will_it_take_to_generate_all_the_coins.3F) - accessed July 22nd 2014.

<sup>52</sup>see information on this block at <http://blockexplorer.com/block/...> - accessed July 20th 2014.

<sup>53</sup>In fact, the bitcoins at transaction points are not stored as a real, but as an 8 Byte non-negative integer, since they are stored as satothis - see ‘value’ field in Txout here: [https://en.bitcoin.it/wiki/Transactions#general\\_format\\_.28inside\\_a\\_block.29\\_of\\_each\\_output\\_of\\_a\\_transaction\\_-\\_Txout](https://en.bitcoin.it/wiki/Transactions#general_format_.28inside_a_block.29_of_each_output_of_a_transaction_-_Txout) - accessed July 20th 2014.

<sup>54</sup>Transaction point are used here as simplifying abstract entities to explain the transfer of a value of bitcoins.

Obviously this naive view does neither hold for banks nor for **Bitcoin**, which is why we want to look at the actual use of both, the **CPS** and the **BPS**. The **CPS** is analysed extensively in section 7.3 and the **BPS** in section 8.3.

## 7.3 Cryptocurrency Payment System

We now want to look at **Bitcoin** as a **payment system** specifically, basing our analysis on the findings in section 6, where we did a requirements analysis for a hypothetical **payment system**, thereby building a terminology concerning **money** we want to employ here on **Bitcoin** as a **CPS**.

**Motivation for this section** Since **Bitcoin** was specifically planned and designed by one entity, it seems plausible to assume that this entity had either explicit or implicit assumptions about the requirements that **Bitcoin** as a **payment system** would have to fulfil for it to be used as such. We want to find out what these assumed requirements were and if they match with those we found to be essential for a **payment system** in section 6. What requirements did the designers of the Bitcoin payment system - probably - have in mind? What requirements that are a result of our analysis does the **Bitcoin Network** fulfil?

**How the Network may serve as a CPS** It is not trivial to understand how the **Network** may serve as a **payment system** at all. The name of the project is **Bit<sup>coin</sup>** and its proposed **money** is called **bit<sup>coin</sup>**. Contrary to the intuition that the name provides however, there are no **bit<sup>coins</sup>** actually passed on from hand to hand. So far, our understanding in this work of **bit<sup>coins</sup>** is emphasizing that **bit<sup>coins</sup>** are first and foremost (near) tamper-proof real numbers with eight decimal places, as we have just seen in section 7.2.5. But how exactly does a tamper-proof number serve as **money** that can be used for **payments**?

**Approach - looking at the CPS from the view of a user** We are discussing the functional principles of **cryptocurrencies** in detail in this section by going through fully-dressed use cases based on suggestions by **Cockburn (2004)**. We find the view of an actual user as a starting point for the analysis of the **CPS** most valuable. The aim is to explore the capabilities of **Bitcoin** as a **CPS** and then compare these findings to the necessary functions of a (hypothetical) **payment system** we found in section 6.

### 7.3.1 Use case diagrams

We want to first generate an overview on the use of **Bitcoin** as a **CPS**. We hope to achieve this by depicting use case diagrams that show our progression of understanding **payment systems** in general, as it is shown as a synopsis in section 6.7, transferred to **Bitcoin** that shall be evaluated as a **CPS**. Furthermore we want to include additional detail that is required to understand the **Bitcoin Network** from a user perspective, thereby integrating what is called **mining** into our understanding.

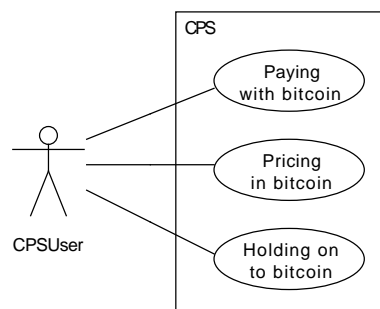


Figure 19: Initial use case diagram on **Bitcoin** used as a **CPS**.

Figure 19 shows three use cases for a CPS that are already well known in principle from the use cases that were identified for a hypothetical payment system in sections 6 and specifically 6.1. The difference is that the payment system is a CPS and that therefore the notion of money in general is replaced by the money of the CPS called bitcoin. The three use cases yielded are as follows:

- **Paying with bitcoin** - bitcoin is used by the Payer and accepted by the Payee as the money proper to fulfil debts nominated in bitcoin as the money of account. This is expanded on in section 7.3.3
- **Pricing in bitcoin** - bitcoin is used as the money of account for the pricing of assets, goods and services and for the nomination of other types of contracts. This is expanded on in section 7.3.4
- **Holding on to bitcoin** - bitcoin is used by a Hoarder, whose aim is to hold on to bitcoins. This is expanded on in section 7.3.5

If we include the findings on payment systems and money in general into a use case diagram for the CPS, we receive the diagram shown in figure 20.

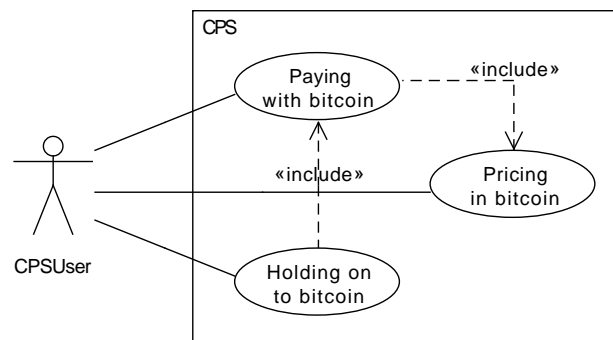


Figure 20: Use case diagram on Bitcoin used as a payment system that takes the hierarchy of functional requirements on money into account.

Again, making visible that there is a hierarchy on the functional requirements of a payment system that holds true for the CPS as well.

**‘Holding on to bitcoin’ includes ‘Paying with bitcoin’** If a hoarder is holding on to bitcoins, we must assume he does so, because he expects to be able to make payments with the hoarded bitcoins at a later stage. The argument is the same as it is for a hypothetical payment system. It just does not make sense to hold on to any bitcoin if it is not possible to make payments with the hoarded bitcoins at a later stage. Even if payments right now were considered impossible, we must consider it at least an unconscious assumption by the bitcoin hoarder to expect to be able to make payments in the future. If the assumption of the hoarder was to never be able to make any payments with hoarded bitcoins, then a holding on to them still could hardly be explained by rational motives.

**‘Paying with bitcoin’ includes ‘Pricing in bitcoin’** The pricing in bitcoin means the offering of assets, goods and services at a price nominated in bitcoin. Paying with bitcoin is the fulfilment of dues payable in bitcoin. Consequently, if there are no price tags to be found that are nominated in bitcoin, then there will be no payments in bitcoin. To clarify the meaning of this point: imagine if the CPS was designed to perfectly process payments in a way that they were fast,

secure and reliable but it was not designed to motivate users to use the **money** of the **CPS** as **money of account**, then what good is the perfect processing of (potential) **payments**, if there are none?

However we want to state that the relationship is similarly close if viewed vice versa. It hardly makes any sense to price **assets**, goods and services in **bitcoin** and specify contracts using it as **money of account** if there isn't a means of final **settlement of debt**, i.e. a **money proper**, available. What a precarious situation would any debtor be in, if he was in debt in **money of account** that does not have a corresponding **money proper**? In **Bitcoin** terms, this would mean actors could get into debt, owing a certain amount of **bitcoins**, but there was no means of **payment**, no **money proper** available. Obviously debtors would have to avoid making any purchases that create a debt they can never pay down for lack of a **money proper**. This is why we consider both, the **money proper** and the **money of account** as emergent properties of a **payment system**, as we have already termed them in section 6.5 and shown in figure 7 on page 30.

We have renounced to show the reciprocity of relationships 'Paying with bitcoin' and 'Pricing in bitcoin' in the use case diagram, for we deem the implicit assumption to have to have a **money proper** available (= the ability to successfully make **payments**) if a corresponding **money of account** is used as rather common, than the relationship vice versa. That is why we want to emphasize the *«include»*-relationship from 'Paying with bitcoin' to 'Mining' by including it explicitly in the use case diagram.

**Including the Network** We want to abstract for a minute from the hierarchy of the functional requirements for money and quickly go back to figure 19 (page 50) and expand our understanding from there.

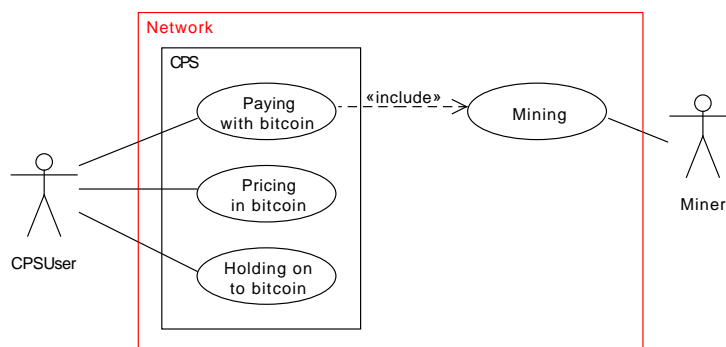


Figure 21: Nested systems diagram on **Bitcoin** used as a **payment system** that is enclosed by the **Network**, which provides an additional use case called 'Mining'.

Figure 21 is the attempt to broaden the view portrayed in figure 19 to now include the **Bitcoin Network**, which does serve as a **CPS** here and therefore contains all the use cases of the **CPS** itself. We are showing the **CPS** with all its use cases as being contained as a whole within the **Network**, in what we want to call a nested systems diagram, for lack of a better term.<sup>55</sup> The purpose is to show that the additional use case called **Mining** is specifically part of the **Network**. It is not at the user goal level of the **CPS** as it is nothing that the actual **CPSUsers** are interested in. It is a sort of back-office use case that is included by other use cases, as is shown in figure 21, but this inclusion does not mean **Miners** are **CPSUsers**. **Miners** are not primarily interested in the **payment** functions of the **CPS**, they are interested in the incentives that are related to

<sup>55</sup>Without going into the details here, a meta-model for the 'nested systems diagram' we use here would essentially match the meta-model for use case diagrams in the **Unified Modeling Language (UML)** today, except for the permission of a 'nesting' of system boundaries

mining.<sup>56</sup> However, it must be noted here that the incentive for mining is certainly closely connected to the performance of Bitcoin as a payment system, for if it was entirely useless, what good would it be to mine additional bitcoins? We will see in section 7.4 that successful miners can sell their bitcoins to other traders by means of what we call the EcoSystem.

**Including the MMI** During the analysis of a hypothetical payment system in section 6 the role that the MMI plays was discovered, thereby creating the term in the process. The MMI was explained further in section 6.6. The analysis in section 6 was done having the BPS in mind (as the status quo) that is embedded in a MMI that consists of various governmental agencies, institutions as well as certain civil and legal norms.

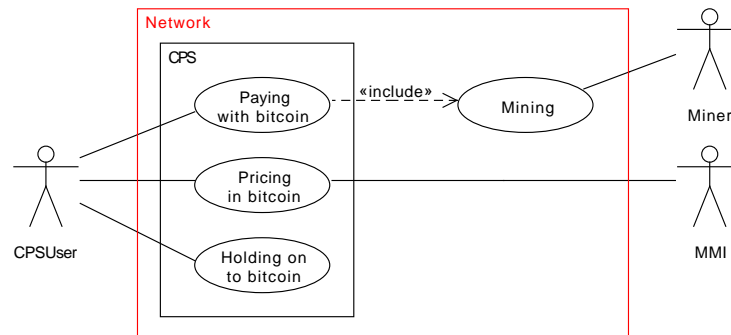


Figure 22: Including the MMI into the diagram that shows the CPS, the Network and the Miner.

The MMI that the CPS could potentially be embedded in seems to be rather difficult to determine right away. We can assume however that potential users of the CPS will call for the possibility of legal remedies the same way users of the BPS do. Not regularly, but as soon as contractual nuisances or other problems occur. We include the MMI for the CPS as a supporting actor in the use case diagram as shown in 22.

**Assembling hierarchy, Network and MMI** If we put the findings on the hierarchy of functional requirements, the Network and the MMI into one use case diagram we receive the one shown in figure 23.

### 7.3.2 Users

The users of Bitcoin as a CPS are naturally synonymous to the users identified for a hypothetical payment system in section 6.1.1 on page 20. We list them here again in brief, modified however for the use of Bitcoin as a specific CPS:

- **Payer**  
A payer is an actor that has to perform payments using bitcoin, the money proper of the Bitcoin project.
- **Payee**  
A payee is an actor that receives payments in bitcoin.
- **Adopter**  
An Adopter is pricing assets, goods or services in bitcoin, using it as a money of account.

<sup>56</sup>The incentives for miners and the important role these incentives play in the overall health of the Network are described in section 7.2.4.

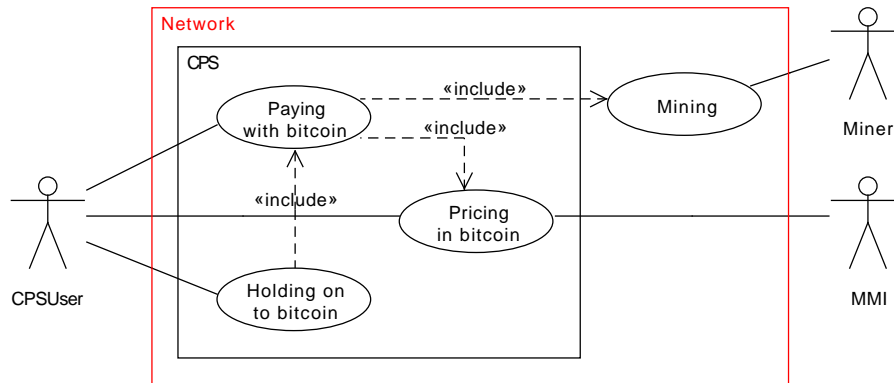


Figure 23: Nested systems diagram showing the enclosing **Network**, the supporting agent **MMI** and does include the hierarchy of **payment system** requirements as *«include»*-relations.

- **Hoarder**

Hoarders intend to stash away **bitcoins** for later use. Potentially exploiting the proposed lack of governmental influence in **Bitcoin**.

These 4 types of users have been aggregated as CPSUsers. If we expand our view to include the **Network**, as is shown in figure 21 on page 52, additional users that are identifiable are **Miners**.

- **Miner**

A **Miner** is a primary and arguably a secondary user, too. On the one hand the **miner** is engaged in **mining** for the incentives it provides by itself. On the other hand without the **miners** a successful verification of **payment transactions** by including them in the public **blockchain** is impossible. In this sense **Miners** are supporting actors for the **CPS** to be able to successfully process **payments**

### 7.3.3 Use Case: Paying with bitcoin

For use cases be rather incomplete without being written out, we want to include at this point a written out form of the use case ‘Paying with bitcoin’ that is oriented on the fully-dressed use case format proposed by **Cockburn (2004)**.

**Scope** Cryptocurrency Payment System (**Bitcoin**)

**Level** User Goal

**Primary Actor** Payer and Payee

#### Stakeholders and Interests

1. Payer

- a) has **transaction points** (**addresses** (public key) with **bitcoins** stored that he has access to by means of the private key) in his **wallet** to transfer **bitcoins** from
- b) wants to make **payment** to Payee
- c) wants to be able to pay in various amounts

- d) wants to receive change from **transactions**
2. Payee
  - a) has an **address**
  - b) wants to be paid on time
  - c) does not want to have his **bitcoin payment** compromised by **double-spending**

### Preconditions

1. Payer and Payee use the **client** software to participate in the **Network**
2. Payer is in debt to Payee that is to be settled in **money proper** of the **CPS** (i.e. **bitcoin**)
3. Payer controls **transaction points** that grant access to enough **bitcoin** to meet the amount that is due

### Postconditions

1. Payee is paid successfully, by now holding **transaction points** in his **wallet** that grant access to **bitcoins** in the amount due

### Main Success Scenario

1. Payer logs in to his Node
2. Payee provides Payer with his receiving **Address** (col. „account number“)
3. Payer provides Node with Address of Payee and the Amount (= value) to be sent
4. Node of Payer initiates **transaction TA** and broadcasts **TA** to **Network**
5. **Network** verifies **TA**
6. Payee logs in to his Node that becomes part of **Network**
7. Payee recognizes funds as received

**Creating an object model** If we identify the relevant nouns in the written out use case above, we find:

- Payer
- Node
- Payee
- Address
- Amount (value)
- Transaction
- Network.

Putting these nouns in a domain model for a **payment transaction** is the next step in our analysis of using the **CPS** for the purpose of performing **payment**. The domain model is shown in figure 24.



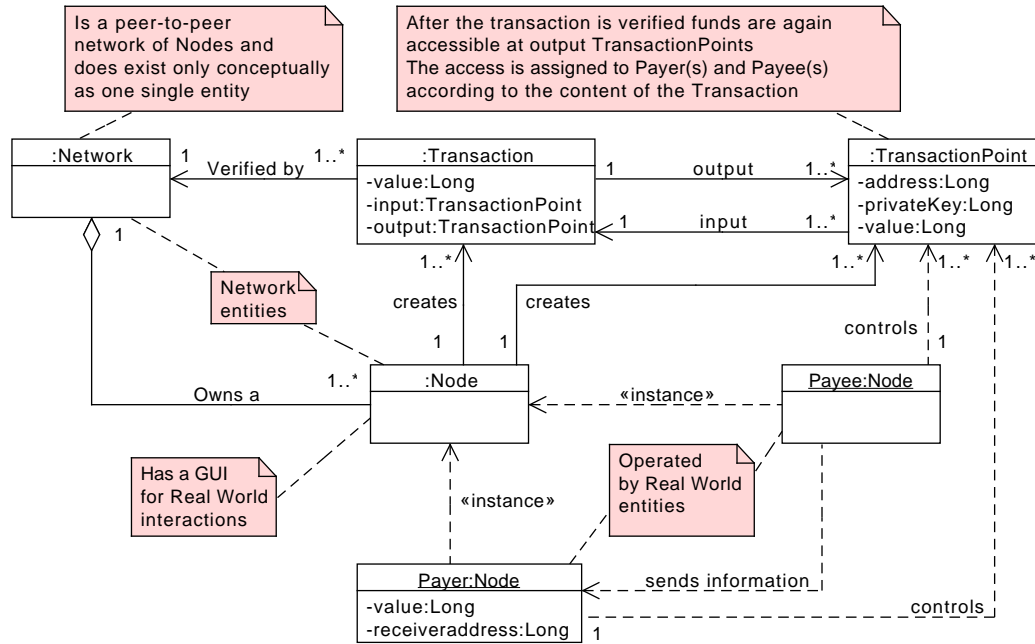


Figure 24: Object model showing the relevant objects and their relations involved in the completion of a **payment transaction** in a **CPS**.

**Bitcoin as a money proper** Being able to receive change in a **Bitcoin transaction** is a requirement deducted from the written out use case ‘Paying with bitcoin’. Providing the ability to receive change in a **CPS payment transaction** involves the creation of a single-input/multi-output **transaction**. We want to cover this type of **transaction** by going through an example at this point that involves a multi-input/multi-output **transaction**, thereby including single-input/multi-output **transactions**.

**A multi-input/multi-output example** We assume Alice wants to make a **payment** to Bob in the amount of 10 **bitcoin**. If Alice was equipped with a **transaction point** that holds precisely 10 **bitcoin** the example would be done at this point, since we could simply refer to the example involving single-input/single-output **transactions** in **the simple example** on page 41. While in this example here Alice does have the funds to pay 10 **bitcoin** to Bob, her funds are not located at one **transaction point**, but distributed at three different **transaction points**  $TP_1$ ,  $TP_2$  and  $TP_3$ .<sup>57</sup>

- $TP_1$  holds 5 **bitcoin**
- $TP_2$  holds 4 **bitcoin**
- $TP_3$  holds 3 **bitcoin**

All of these **transaction points** are controlled by Alice. To make the **payment** of 10 **bitcoin** to Bob, she creates **transaction**  $TA$  that will include:

- the **message digests** of all three previous **transactions** that created  $TP_1$ ,  $TP_2$  and  $TP_3$
- the public key of  $TP_4$  that will grant Bob access to his 10 **bitcoins**

<sup>57</sup>This is based on an example given by Cap (Cap, 2012, pg. 86).

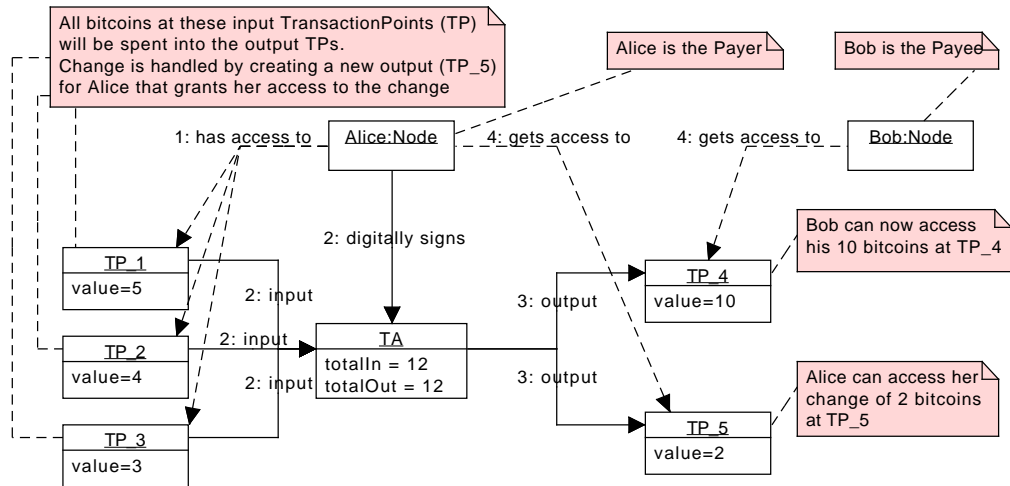


Figure 25: Communications diagram showing an example of a multi-input/multi-output **transaction**.

- the public key of *TP\_5* that will grant her access to the 2 remaining **bitcoins** in change

Alice will subsequently digitally sign the **transaction**, by including all private keys of *TP\_1*, *TP\_2* and *TP\_3*. They will be verified by a scripting system that is involved in the verification of all **transactions**, we have just so far seen no need in mentioning it. This scripting system allows for the creation of a valid multi-input **transaction** of this kind only if all input **transaction points** are authorized by providing each and every private key, otherwise the funds at the inputs cannot be transferred. The script compares every one of private keys individually for verification purposes.<sup>58</sup>

*TA* is then broadcast by Alice to the **Network** and Bob just needs to wait for verification, by seeing *TA* included in a **block** that is part of the longest **blockchain**. Ideally Bob will wait until a few **blocks** are attached after the **block** containing *TA*. Figure 25 is illustrating the effect of a verified multi-input/multi-output **transaction**.

### 7.3.4 Use Case: Pricing in bitcoin

**Scope** Cryptocurrency Payment System

**Level** User Goal

**Primary Actor** Adopter

#### Stakeholders and Interests

1. Adopter

#### Preconditions

1. Cryptocurrency Payment System provides a unit

#### Postconditions

<sup>58</sup>see <https://en.bitcoin.it/wiki/Transactions#Verification> - accessed July 26th 2014.

**Main Success Scenario**

1. Adopter uses **bitcoin** as **money of account**
2. Adopter is pricing assets, goods or services, as well as nomination of contracts in **bitcoin**
3. All contracts are successfully fulfilled

**Extensions**

3. a) Contractual nuisances happen, e.g. payment does not happen afterwards
- b) Adopter needs to seek remedy

**Bitcoin as a money of account** The **Bitcoin** community itself claims that

“Bitcoins have value because they are useful and because they are scarce. As they are accepted by more merchants, their value will stabilize.”<sup>59</sup>

However the notion of scarcity as the source of ‘value’ is justifiably discarded in the same paragraph:

“If confidence in Bitcoins (sic) is lost then it will not matter that the supply can no longer be increased, the demand will fall off with all holders trying to get rid of their coins.”<sup>60</sup>

While we do agree to the content of this quotation in principle, we want to emphasize that the notion of ‘confidence’ in a **money** is first and foremost expressed by the use of it as the **money of account**. For tagging an **asset** with a price nominated in this **money of account** is an offer to sell it. The act to choose the tagging is the expression of confidence in a **money**. While it is clear that no one today is forced to tag their **assets** with prices nominated in **bitcoin**, this is also not true for conventional **currencies** in nation states with **legal tender** laws as described in section 6.3.1. The *pricing* of **assets**, goods and services always happens voluntarily, but once a debt is created (e.g. by initiation of a purchase) the acceptance of the tendering of **currency** as **payment** is not voluntary. So, for **Bitcoin** and other **cryptocurrencies** there probably lies much potential in convincing merchants and other users to price their **assets**, goods and services in **bitcoin**. How to design a **cryptocurrency** precisely in a way that is attractive for users to use the **cryptocurrency** as their **money of account** is beyond the scope of this work, but we hope to at least lay it open and approach this issue with the work at hand. Simply assuming that **cryptocurrencies** already are designed in this way might be considered bold, but certainly will not help in the advancement of **cryptocurrencies** if it was ever not the case.

**7.3.5 Use Case: Holding on to bitcoin**

**Scope** Cryptocurrency Payment System

**Level** User Goal

**Primary Actor** Hoarder

<sup>59</sup>see [https://en.bitcoin.it/wiki/FAQ#Where\\_does\\_the\\_value\\_of\\_Bitcoin\\_stem\\_from.3F\\_What\\_backs\\_up\\_Bitcoin.3F](https://en.bitcoin.it/wiki/FAQ#Where_does_the_value_of_Bitcoin_stem_from.3F_What_backs_up_Bitcoin.3F) - accessed Jul 23rd 2014.

<sup>60</sup>ibid.

### Stakeholders and Interests

1. Hoarder
  - a) wants to hold on to **bitcoin**
  - b) does not want his **bitcoin** to lose value significantly
  - c) wants to spend his **bitcoins** at a later point in time
  - d) wants to stash **bitcoins** away for deliberations on security, especially considering macro risk on governments cracking down on hoarders of their **money**

### Main Success Scenario

1. Hoarder holds on to **bitcoin** for as long as he wants
2. Hoarder spends **bitcoin** at a later point in time

### Extensions

1.
  - a) Stashed away **bitcoins** are not used as **money of account** and therefore hardly as **money proper**
  - b) The price in **currency** of stored **bitcoins** falls significantly
  - c) Purpose of hoarding failed

**Bitcoin as a store of value** Holding on to **bitcoins** means that **transaction points** are stored in the **wallet** that allow access to **bitcoin**. This almost sounds like there was some thing to hold on to. Rather, a **transaction point** equipped with enough access to **bitcoins** allows for the creation of a valid **transaction** that will be accepted by **miners** and therefore eventually put into the **blockchain**. Thereby putting a time-stamp on it, which allows for the checking against **double-spending**.

If we insist on thinking of a bitcoin as a thing, we might want to remember the explanation given in the original white paper on **Bitcoin** (see **Nakamoto, 2008**, pg. 2). A ‘coin’ - as defined by Nakamoto and laid out extensively in section 7.1 - is a chain of digitally signed sections of data.<sup>61</sup> One section of data does not have a lot of effect on anything, so the ‘coin’ is a whole chain of digitally signed sections of data, the ‘coin’ ‘changing hands’, so to speak, by the intent of the current owner, which is expressed by his digital signature on the data, thereby including the subsequent owners **address**. Now, only the subsequent owner can again produce a valid signature on the ‘coin’ to spend it. All this is not new to us at this point.

Holding on to a coin in this sense then means not to sign the ‘coin’ again, thereby not passing it on. Alternatively an owner can hold on to his wealth stored in the ‘coin’ by signing the ‘coin’, but to an **address** only he himself has access to.

## 7.4 Bitcoin EcoSystem

At this point we want to broaden our view on **Bitcoin** even further and include the **EcoSystem** that has sprung up around the **Bitcoin Network**. This section will try to give answers to the question what the **EcoSystem** is and what it does provide from a users perspective.

### 7.4.1 Including the EcoSystem

With the **Bitcoin EcoSystem** we extend our assessment to all supporting The nested systems diagram in figure 26 shows the **EcoSystem** enclosing the entire **Network**.

<sup>61</sup>Again: The way digital signatures work is briefly described in section 3.4, which contains a description of the principle digital signature algorithms work on.

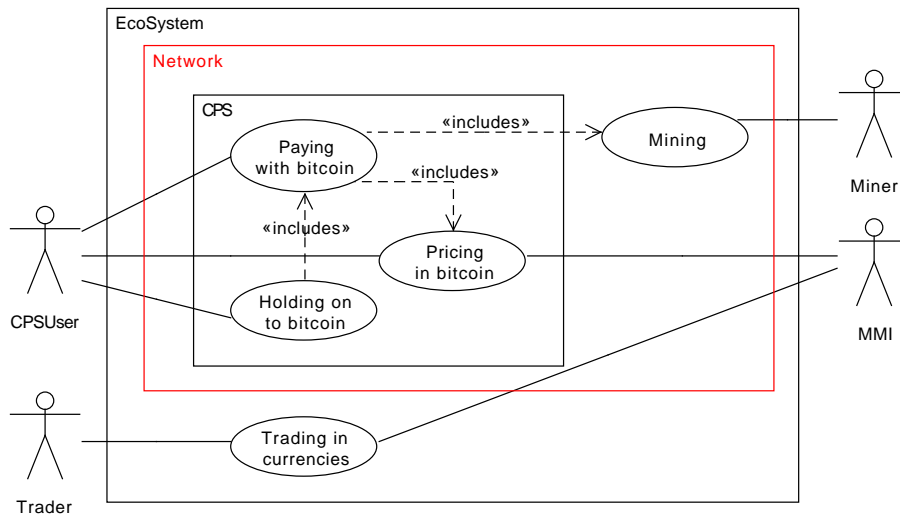


Figure 26: Nested systems diagram that includes the **Network** and the **EcoSystem**.

**Getting a hold of bitcoins** There are essentially two ways to get a hold of **bitcoins**:

- Creating new **bitcoins**, by successfully **mining** a **block**
- By ‘trading in **currencies**’ as is shown in figure 26, here is specifically just one side of the trade meant, namely the buying of **bitcoins** with **currency** (or **bankmoney**) to get a hold of them

**Bitcoins** can naturally be bought peer-to-peer by meeting people physically and handing over **currency** in cash. However we want to allude to specific marketplaces, called ‘exchanges’, which allow trading in **bitcoins**.<sup>62</sup>

#### 7.4.2 Use case: ‘Trading in currencies’

When there are buyers of something anywhere, then there must be sellers there of the same thing accordingly. The same is naturally true for **bitcoins**, too. Every time an amount of **bitcoin** is bought by a **buyer** on an exchange (or wherever else) the same amount was sold by a **seller** on the same exchange (or the same place).<sup>63</sup>

By ‘trading in currencies’ we mean the trading that is done by **buyers** and **sellers** of **bitcoin** into and out of **currencies**. This can be seen very much like the trading on foreign exchange markets.<sup>64</sup> In an attempt to be precise in our language, throughout this work we specifically never speak of ‘buying’ when a purchase is made of an **asset**, a good or a service that had a price tag nominated in **bitcoin**. The just described action is the **payment** with **bitcoin** of a purchase. Buying (or selling) of **bitcoin** stands for the exchange of them in and out of **currencies** like the Euro or the US Dollar.

<sup>62</sup>While there are many - and many have also failed already (cf. Moore & Christin, 2013) - one example for an exchange within the **Bitcoin EcoSystem** is ‘Bitstamp’ - see <https://www.bitstamp.net/> - accessed July 27th 2014.

<sup>63</sup>Assuming that the exchange is a pure broker of **bitcoins** only (and not a dealer that does not have a matched-book 100 per cent of the time).

<sup>64</sup>In fact, **bitcoin** is foreign to every **currency** at every place on this world.

**Miners as regular sellers** Most likely regular **sellers** of **bitcoins** are to be found among (professional) **miners**, which often use specific and therefore expensive hardware (so called **application-specific integrated circuits (ASICs)**) to increase their chances in winning the **mining** lottery, so to speak, by successfully finding a **hash** of a **block's** header. By selling the awarded **bitcoins** on exchanges (professional) **miners** can obtain Euro's or \$'s or other **currency** (or much rather according **bankmoney**).

**The EcoSystem and the MMI** In figure 26 we have included a supporting agent we call the **MMI**.<sup>65</sup> Its importance for the use case 'Pricing in bitcoin' was explained in section 7.3.4, however it is also shown as being relevant for the use case 'Trading in currencies'. With the relation of the **MMI** to this use case we want to account for the fact that users of the **CPS** currently are using the **BPS** and therefore the **monies** it provides for.<sup>66</sup>

### 7.4.3 Users

The aim for this short subsection is to provide an overview on all the different types of primary users of the entire **EcoSystem** we have identified by now. We start our summary analysis of primary users with a simple object diagram shown in figure 27.

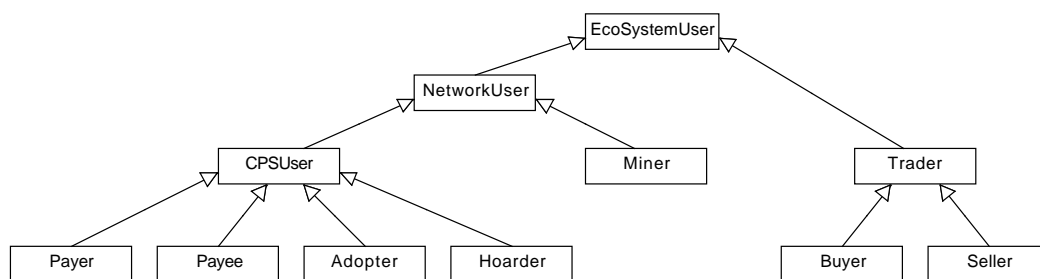


Figure 27: Object diagram on the different potential user types of the **Bitcoin EcoSystem**.

In the object diagram in figure 27, two primary user types have so far not been explained completely:

- **Buyer**

A **buyer** is one of two types of **traders**. A **buyer** wants to change **currency** he holds on to into **bitcoin** (e.g. Euro or US Dollar into **bitcoin**).

- **Seller**

A **seller** is the other type of a **trader**. A **seller** wants to sell **bitcoins** for **currency** (e.g. selling **bitcoin** for Euro or US Dollar).

### 7.4.4 One nested use case diagram to include it all

If we put all our insights on **Bitcoin**, the **Network** and the enclosing **EcoSystem** together at the user level, we receive a nested systems diagram that includes:

- the basic functional requirements of any **payment system**, including a **CPS** that were found in section 6

<sup>65</sup>The idea of a **MMI** was described in section 6.6.

<sup>66</sup>Namely **currencies** - like the Euro or US Dollars - and **bankmonies**.

- all the CPSUsers expressly:
  - Payer
  - Payee
  - Adopter
  - Hoarder
- the hierarchy of the basic functional requirements and of the emergent properties (the **money of account** and the **money proper**) that were explained for a **CPS** in section 7.3.1.
- the encompassing system, we call the **Network**
- the additional NetworkUser that is at the same time a primary and supporting user we call **Miners**.
- the all encompassing system we call the **EcoSystem** that includes all the services offered by companies and other organizations that revolve around the **Network**. Expressly included is the use case ‘Trading in currencies’ that is provided by the **EcoSystem** in the form of exchanges, which we include into what we call the **MMI** which is the last item we want to mention in this list
- the secondary, supporting actor we call the **Money Meta Infrastructure (MMI)**

This summarizing and with the explaining comments rather large, yet still not highly complex, nested systems diagram is to be found in figure 28 on page 63.

#### 7.4.5 Specific risks

Having described the use of the **Bitcoin Network** as a **CPS** and its embeddedness into the **EcoSystem** we now want to touch upon risks that concern **cryptocurrencies** in general and **Bitcoin** specifically.

**Loss of access to bitcoins** The access to **bitcoins** can be lost, if the private keys of the **transaction points** are not available any more, for example by loss or destruction of the **wallet** file. This is hardly any different than losing the physical wallet with banknotes in it.

**Bitcoin wallets**, being simple files, deserve all protection a physical wallet deserves, for it is not in itself encrypted. All computational measures feasible to protecting the most sensible data is to be used to protect the **wallet**. If users do not choose to outsource this responsibility (e.g. by using web-wallets), the responsibility to protect the **wallets** is entirely up to the users.

**Rising transaction fees** Looking at the chart of transaction fees on [blockchain.info](http://blockchain.info), we find that overall fees for **transactions** have declined to the lowest levels in 12 months.<sup>67</sup> Without going into the details, we can state that right now, at least on the surface, transaction fees are almost non-existent, if we compared them to fees for international remittances using the **BPS**. But this notion might be misleading, since the creation of new **bitcoins** through **mining** is what currently accounts for the investments **miners** have made in mining hardware. As noted previously, the creation of **bitcoin** through **mining** will decline over time, but we can not assume that relative cost of mining hardware will in the same fashion. In fact it can't, for **bitcoin** creation will at some point come to an end entirely. There are basically two possibilities:

- Mining declines

<sup>67</sup>For a chart of transaction fees see <http://blockchain.info/charts/transaction-fees> - accessed July 28th 2014.

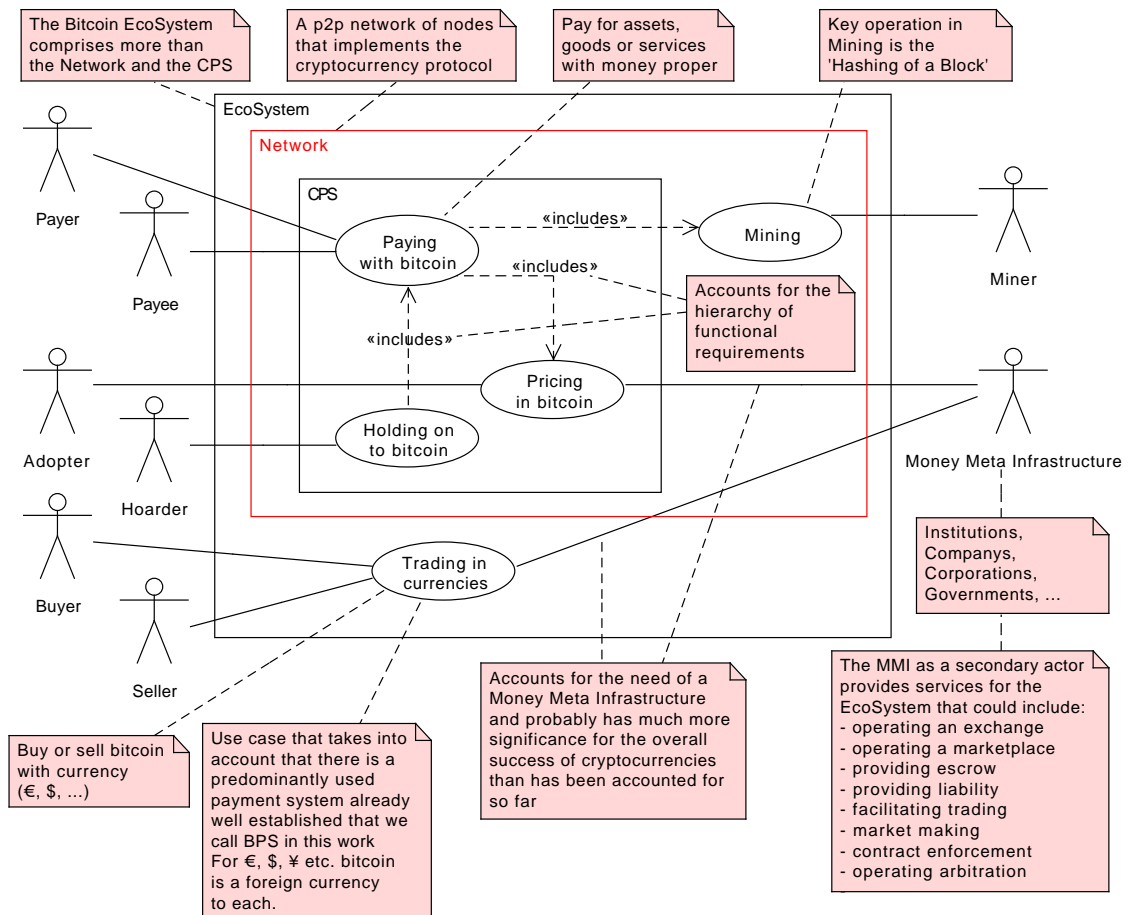


Figure 28: Nested systems diagram, putting it all together.



- Transaction fees will rise to make up for lesser revenues from **bitcoin** creation

A decline in **mining** might make **cryptocurrencies** more vulnerable to history revision attacks, as noted in ‘**Incentive for mining**’ on page 48. If transaction fees are significantly rising for users this might lower their willingness to use the **CPS** as their **payment system** of choice.

**Cryptocurrency competition** **Bitcoin** certainly has the advantage of being the first-mover in the field of **cryptocurrencies**, bringing innovations to public understanding and use that has basically founded this **domain** or at least set it on a completely new footing. However, there are no patents in place that would fortify this first-mover position per se. In fact, multiple other **cryptocurrencies** have sprung up, mainly as ‘forks’ of the open-source code of **Bitcoin** itself, which is thereby furthering **cryptocurrency** competition one might say. So one potential risk for **Bitcoin** is to lose its standing of being recognized as a mathematically and cryptographically guaranteed **cryptocurrency** with an ever limited total supply. But if the findings of this work hold any merit, then the competition among **cryptocurrency** is probably not just fought on issues of implementation and the realization of functioning as a **CPS**, but of how the respective **EcoSystem** of the **cryptocurrency** is responding to user needs.

**Technical risk** Of course there is the risk that at some point a technical flaw might be discovered - say in the **cryptographic protocol**<sup>68</sup> - that is not to easily fixed and could therefore destroy trust in the **CPS**. So far this hasn’t happened in a way to destroy **Bitcoin** altogether, but certain movements in market prices can certainly be attributed to technical problems that made headlines, even though most of the time these issues did not concern the **Network** but parts of the **EcoSystem**.<sup>69</sup>

**High fluctuations in currency prices** **Bitcoin** prices have been volatile all along even in US Dollars, which is the **bitcoin** market with the highest **market liquidity**. This is a problem for merchants that have costs in US Dollars (or any other **currency** for that matter) and would have revenue in **bitcoin**, if they’d tag their products with prices in **bitcoin**, thereby using it as their **money of account**. If they are paid in **bitcoin** for their products, they still have to cover their costs in US Dollars. With high volatility in prices the likeliness of **bitcoin** being used as **money of account** decreases. Currently merchants tackle this problem by using supporting **MMI** actors that are part of the **EcoSystem** and that offer the taking of this form of foreign currency risk at premium that is - compared to foreign currency risk - easier calculable for merchants.

This is showing an interesting interconnection of the nation states that provide the **MMI** for the **BPS** and the **cryptocurrency EcoSystem**. This becomes visible by looking at the concrete example of a company that is involved in this business of risk taking, called ‘Bitpay’.<sup>70</sup> This is a company located in the jurisdiction of the United States of America and is therefore trusted by its customers to be able to bear the risk they claim to be able to bear, for they do business according to US legal standards. If anything went wrong in contracts with Bitpay its users were able to hold this company accountable within the US jurisdiction.

So, even right now, for **bitcoin** to be used as a **money of account**, the adopters depend on the **MMI** that the nation state provides, which brings us back to the question of how much of the **MMI** requirements currently provided for by nation states for the **BPS**, the **Network** might be able to provide for the **CPS**.

<sup>68</sup>For an assessment on vulnerabilities of elliptic curve cryptography - as used in **Bitcoin**, see *Bos et al. (2013)*.

<sup>69</sup>One example are the problems the largest **bitcoin** exchange at the time, called ‘Mt. Gox’, had, before eventually filing for bankruptcy. For the problems with these type of middlemen in general (and in great anticipation!) see *Moore & Christin (2013)*.

<sup>70</sup>see <https://bitpay.com/> - accessed July 28th 2014.

## 7.5 Beyond a payment system

What we have almost entirely omitted to mention so far, is the actual process of how exactly in **payment transactions** within the **Network** the correctness of digital signatures is established by validating **miners**. This is done using an intentionally non-touring complete script language<sup>71</sup> that provides the possibility to create multiple different conditions that have to be met, until the output **transaction points** can be used as inputs of subsequent **transactions** or, more colloquially speaking, until the **bitcoins** can be spent again.<sup>72</sup> Out of the possibilities that the script offers can be created what is called a 'contract'.<sup>73</sup> These contracts allow - so far mostly theoretically - for additional use cases that go far beyond the use of **Bitcoin** and its core innovation, the **blockchain**, as a mere **payment system** and might potentially relief some of the hindrances that currently inhibit further use and acceptance of **bitcoin** as **money**.

### 7.5.1 Some preliminary use cases

This is the section that is briefly touching on some use cases that are mostly not yet available for **Bitcoin** users, but go beyond the **payments** and payment related use cases we have covered in this section so far. The analysis is going to comprise use cases that might potentially be covering parts of what we described in section 6.6 as being the **MMI**.

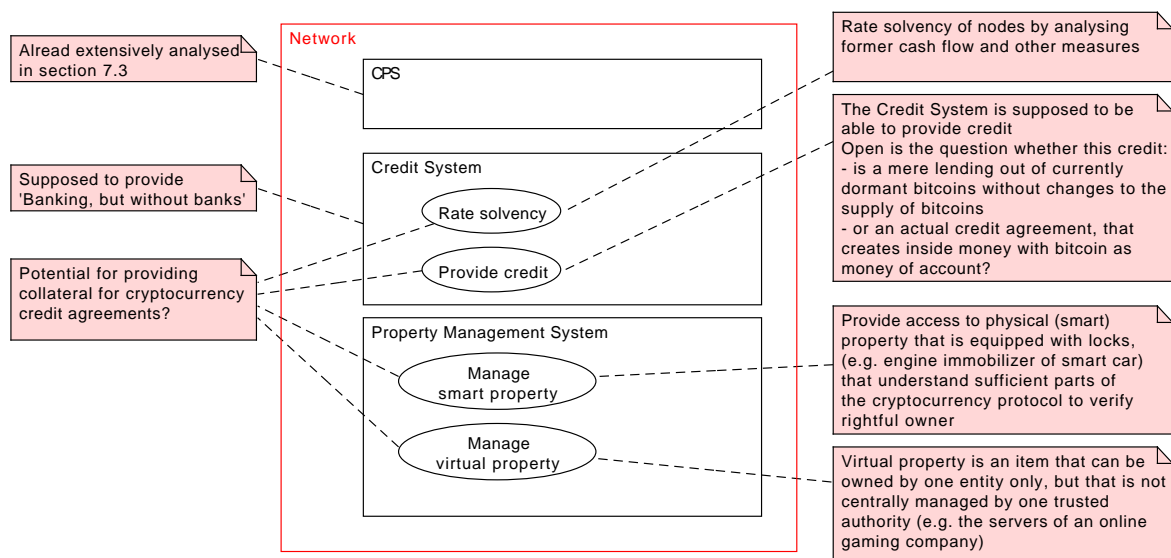


Figure 29: Diagram showing just some use cases that go beyond **Bitcoin** as a mere **CPS**.

**Dispute Mediation** The script in **transactions** can be designed in way that allows for rudimentary dispute mediation. The form of dispute mediation that is currently available for **Bitcoin** users is what are called 'multisignature transactions'.<sup>74</sup> The 2-of-3 multisignature transaction is designed in a way that the third party to the **transaction**, called the mediator, can decide who gets the money. Whatever happens, the mediator can never get access to the **bitcoins** himself.

<sup>71</sup>The imperative Script is similar to FORTH and is stack based, see <https://en.bitcoin.it/wiki/Script> - accessed July 28th 2014.

<sup>72</sup>This subsection is greatly inspired by a talk of Mike Hearn at the Bitcoin conference 2012 in London. The slides of the presentation can be accessed at <https://docs.google.com> - accessed July 28th 2014.

<sup>73</sup>see <https://en.bitcoin.it/wiki/Contracts> - accessed July 28th 2014.

<sup>74</sup>see <https://en.bitcoin.it/wiki/Contracts#Theory> - accessed July 28th 2014.

This form of ‘dispute mediation’ must be considered still rather incomplete and will for example hardly replace what is currently achieved by employing arbitration courts in conflicts that can emerge from general business transactions within day-to-day life using the **BPS**.

In any case, dispute mediation should not be available only, if a very special type of transaction is chosen. Dispute mediation should be possible in any type of transaction for any party, as it arguably is in business transactions (at least in western countries) by employing procedures offered by arbitration courts or other legal remedies provided for by national governments. The lack of proper dispute mediation (as a part of what we call the **MMI**) is most likely one of the key issues that have so far greatly inhibited further seamless use and therefore dissemination of **cryptocurrency** technology.

**Assurance contracts** With assurance contracts a funding model can be realized that releases the funds pledged only if the targeted amount is hit in total. Otherwise all the funds are returned (at a predefined date). This resembles what is currently realized outside of the **cryptocurrency** world by projects like ‘Kickstarter’.<sup>75</sup>

A potential use case would be the translation of a website. Interested readers of an article in a foreign language could pledge a certain (rather small) amount each and the translator is paid in full after successful (had to be defined) publication of the translated article. If no translator would be interested, the funds pledged would return after a certain amount of time.

**Virtual property** The design of ‘contracts’ in the **cryptocurrency** sense might allow for what could be called ‘virtual property’. Now, **property**, as a right, is always ‘virtual’ in a sense. **Property** is never a physical thing, it is a right to own, operate, pledge etc. a thing, it is not the thing itself. ‘Virtual property’ in the **cryptocurrency** sense can be thought of virtual tokens, e.g. an item in an online game that is owned and controlled not by a central third party (like the servers of an online gaming website), but by the individual owner, managed by what could be called a Property Management System that is operated by the **Network**.

**Smart Property** The idea of having smart property in the future is based on the fact that more and more physical appliances will be equipped with programmable hardware in a way that allows for this hardware to understand and carry out sufficient parts of the **cryptographic protocol**. In this way physical ownership of physical things can be cryptographically controlled.

An example given by Hearn<sup>76</sup> is a car that is equipped with an engine immobilizer that allows for the engine to be started only by the rightful owner with a private key that fits the public key that the car was signed over to using the **cryptographic protocol**. The car could in this way be repossessed (not physically in the sense that it would be towed, but by access management).

### 7.5.2 Resembling (parts of) the MMI

If the implementation of so called ‘contracts’ within **cryptocurrencies** is very successful, maybe they are helpful in creating what has been termed by Hearn “banking, but no banks”. The most significant part of banking is probably the ability to create collateralized credit agreements.

**Collateralized credit agreements** If smart property and virtual property can be established it could potentially be used as collateral in credit agreements. As we have noted in section 6.6, credit agreements heavily depend on the ability of the debtor to produce enough collateral, and the ability for the creditor to get access to the collateral in the case of a breach of contract by the debtor.

<sup>75</sup>see <https://www.kickstarter.com/> - accessed July 28th 2014.

<sup>76</sup>see footnote 72.

**Automated repossession** The design of **Bitcoin transactions** could potentially be done in a way that can be understood by the hardware of the smart property, thereby granting access to it only to the rightful owner. For example if the debtor defaults on a collateralized credit agreement, the smart property might automatically change ownership to the creditor. The **transaction** designs implied here are hardly trivial and are subject to future research as the whole field of contracts is.

## 7.6 MMI for a CPS

As a thought experiment in an attempt to understand what scope the **MMI** of a **cryptocurrency** would have to have, we could experimentally look at **Bitcoin** as if it was a transnational economy on its own. If we think of **Bitcoin** as an economy we could try to assess its performance the way nation states are ranked as economies by their ability to enable business activity. One source for rankings of this kind and also for criteria of assessment is **Doing Business**.

If we did that how would Bitcoin perform on the ranking of [doingbusiness.org](http://doingbusiness.org/)?<sup>77</sup> ‘Doing Business’ is a project supported by the World Bank that is trying to measure business regulations by surveying business owners about the ease of doing business in a specific country.<sup>78</sup> The list of criteria surveyed can be viewed below:

1. Starting a Business
2. Dealing with Construction Permits
3. Getting Electricity
4. Registering Property
5. Getting Credit
6. Protecting Investors
7. Paying Taxes
8. Trading Across Borders
9. Enforcing Contracts
10. Resolving Insolvency

Even being experimentally viewed as a transnational economy on its own, **Bitcoin** will never be a physical place. Therefore it will never have to provide physical infrastructure, so many of the above mentioned criteria do not make a lot of sense to survey for **Bitcoin** at all. These physical infrastructure necessities - e.g. ‘Getting Electricity’ - will always have to be provided by the nation state that claims dominion<sup>79</sup> over the physical land, wherever the respective **Bitcoin** user is located. However the criteria of protecting investors and enforcing contracts is certainly an issue for **Bitcoin**, following our argument about an **MMI** in section 6.6 that any **payment system** needs to be embedded in, if the **money** the **payment system** does provide is sought to be used as such by the users of the payment system.

The example of the rankings at [doingbusiness.org](http://doingbusiness.org/) clearly shows that business operators do recognize that the environment for doing business is not the same at every place on this earth. Even with **Bitcoin** potentially levelling the playing field for making **payments**, the infrastructure differences that reside on this planet will remain highly relevant for all the infrastructure that is needed for **Bitcoin** to successfully operate as a **CPS**.

<sup>77</sup><http://www.doingbusiness.org/rankings> - accessed May 14th 2014.

<sup>78</sup><http://www.doingbusiness.org/> - accessed May 14th 2014.

<sup>79</sup>‘Ownership, or right to property. Title to an article of property which arises from the power of disposition and the right of claiming it’ - <http://thelawdictionary.org/letter/d/page/114/> - accessed May 15th 2014.

## 8 The money view on actuality

In this section we want to develop and apply a view on money we want to call the money view on actuality. First, we will delineate the currently still predominant view on money and banks and subsequently present a view on **money** that is rooted in the actuality of the **payment system** that is most used today we call the **BPS**. This ‘money view on actuality’ is based on the findings in monetary theory that not only has room for money, credit and banks, but is centred right around them. This view is based on the money view presented in **Mehrling (2010)**, **Mehrling (2012)** and **Mehrling (2013)**.<sup>80</sup>

**Motivation** Neoclassical, general-equilibrium macroeconomics does not have room for money. As Frank Hahn famously stated:

“The most serious challenge that the existence of money poses to the theorist is this: the best developed model of the economy cannot find room for it.” (see **Hahn, 1982**, pg. 1)

But why do we even care about this, if we just want to find out in this work if **Bitcoin** creates “a new kind of money”? There are two reasons for this:

1. To assess if **Bitcoin** is money and what kind of money, we want to know as much about **money** and the different types of it as possible, thereby further expanding our findings on **money** in sections 5 and 6.
2. As it turns out, commercial banks do create **bankmoney**, which is a widely accepted **money proper** that is nearly autonomously created by commercial banks and their debtors. This **money** however is not included at all in **legal tender** laws of governments, even if they for convenience’s sake use wire transfers themselves. Now, **bankmoney** is not supported by **legal tender** laws of governments, neither is **bitcoin**, the **money** created by the **Bitcoin** project. **Bankmoney** however is widely used as **money proper** in **payments**, but **bitcoin** is far from this level of usage and acceptance. So, by analysing **bankmoney** we hope to learn about potential ways to improve usage and acceptance of **bitcoin** and other **cryptocurrencies**.

We will now proceed by sketching what we perceive as conventional views on money and banks, to make it distinguishable of what we call the money view on actuality.

### 8.1 A brief distinction

In this section we are very briefly touching on a conventional view on **money** and how far the understanding that it provides brings us in an attempted assessment of **cryptocurrencies**.

#### 8.1.1 Conventional view on money

Most mainstream macroeconomic models do not contain money, banks or debt.<sup>81</sup> Furthermore, as Mehrling notes, liquidity in the conventional view, is a completely free good in general-equilibrium macroeconomic theory, which arguably still is the mainstream in macroeconomics (see **Mehrling, 2010**, pg. 5 and 65f.). These conditions make it difficult to find a place for **cryptocurrency** using the tools and models mainstream, general-equilibrium macroeconomic models deliver.

<sup>80</sup>The money view is comprehensibly developed in the **MOOC** of Prof. Dr. Perry Mehrling called ‘Money and Banking’ available at <http://www.coursera.org> - accessed June 26th 2014.

<sup>81</sup>**Campiglio (2012)** notes that the leading central banks, including the Federal Reserve and the European Central Bank, are using models that at least until 2012 did not contain private banks at all.

In the conventional view we describe here **money** is essentially regarded as if it was a token. One could say in conventional monetary theory **money** is a thing, albeit a special one. Since it is ‘just’ a thing, money is also regarded as being ‘neutral’. The conventional view has no place for money dealers that provide liquidity, since - in this view - markets are always just cleared at general equilibrium market prices. In other words, the underlying assumption in conventional macroeconomics is: liquidity is free, there will always be a market price for everything.

This view of money being neutral and nothing but a special ‘good’ does not help us in this work. We might say: **bitcoin** should also be considered a special good, since it has no weight, can be transferred very cheaply and securely and therefore looking at these convincing properties *should* be used as money. We would be finished at this point, but it does not help us answer the question if it indeed is a new kind of money, in the sense that it will gain widespread adoption. In the whole of this work we are essentially trying to find out what might stop **bitcoin** from generally being used and accepted as **money**. We therefore need a view on money that at least allows us to start this type of investigation. A view on money that regards the very issue at hand as being ‘neutral’ can therefore hardly be a helpful starting point.

### 8.1.2 Applicability of conventional view

Does the invention of **cryptocurrencies** in form of the **Bitcoin** project create a “new kind of money”?

“If it is used and accepted as money, then it is money. If it isn’t used and accepted, it can’t be money.”

This completely fictional short quotation would probably have great similarity in content to the answer we would have to give to the central research question of this work, if we used only the toolbox that the conventional view on money provides.

An assessment of **cryptocurrencies**, by trying to find an answer to the question if they indeed create “a new kind of money” essentially has to fail if we employed a theoretical toolbox that has no room for **money** at all. As a side note, if banks are viewed as mere transitory agents of pre-existing money tokens, or mere ‘intermediaries’ of money, we will not be able to learn much for the **CPS** from assessing the **BPS**. That is why we want to first sketch the ‘money view on actuality’ and then add it to our analysis.

## 8.2 Fundamentals of the money view

Here we will introduce some fundamental ideas to the money view on actuality. First we will describe a principle that is found all over the place in the actuality of money and banking, in what can be called the swapping of **IOUs**. Then we will briefly distinguish **inside money** from **outside money**. This will help in the further assessment of **bitcoin**, if it indeed is “a new kind of money”. We finally will briefly try to give an answer to a much asked question when it comes to the **money** created by commercial banks, we call **bankmoney**. The question is: is it **money** or just ‘credit’?

### 8.2.1 Swap of IOUs

The first fundamental concept to the money view sketched here is a so called swap of IOUs, which is key to understand money and banking. As Mehrling notes, the “essence of banking is a swap of **IOUs**” (see Mehrling, 2010, pg. 72).

### Double entry bookkeeping - A domain specific modeling language for financial relationships

In this section we make use of stylized balance sheets we want to give a short classification of



at this point, by putting the notation we use face-to-face to a class diagram of virtually the same force of expression. In this sense we could view the kind of stylized balance sheet we use as a domain specific modeling language for the modeling of financial relationships of economic entities.

In section 5.3 we introduced an essential type of financial relationship that is called ‘creditor-debtor relationship’. It is conceptually shown in figure 3 on page 16 in an object diagram. We want to model the creditor-debtor relationship again, using a UML class diagram. It is shown in figure 30, showing not only the class diagram itself but also an actual instance of it.

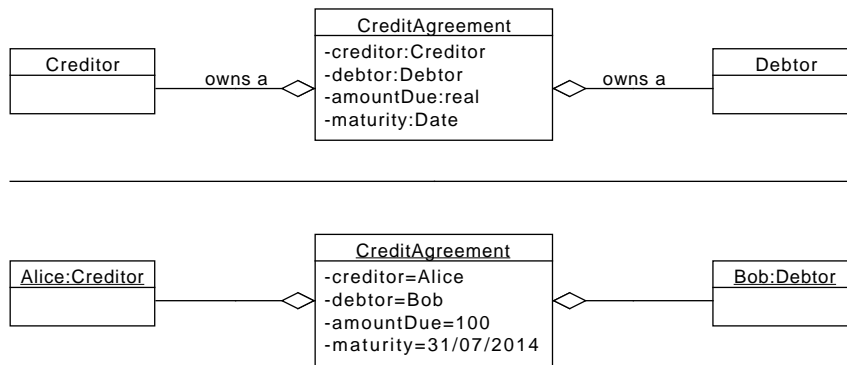


Figure 30: Class diagram that is showing a creditor-debtor relationship, including an instance of it for the example we use here right below the line.

The intention here is to show the two ways to model the creditor-debtor relationship, by bringing them face-to-face. We want to apply it to the most simple example of Alice being the creditor of Bob, who owes an amount of 100 to her, due on July 31st 2014.

Alice		Bob	
Creditclaim	100	Creditclaim	100

Table 1: Stylized balance sheets of Alice and Bob and showing how they can be used to illustrate a creditor-debtor relationship between Alice and Bob.

One way to illustrate the creditor-debtor relationship between Alice and Bob, is using an instance of the class diagram as is shown in figure 30. Another way is using the notation with stylized balance sheets that are contained in table 1. The stylized balance sheets used here do contain two sides for each economic entity. Showing **assets** on the left hand side and liabilities on the right hand side.

For reasons of domain usances and simplicity we will use the second style of notation as we proceed in this chapter, even though it is on one hand not showing ‘complete’ balance sheets that conform to accounting standards and on the other hand it does not show certain details as the class diagram does effortlessly (e.g. maturity date). As was already noted, this notation involving stylized balance sheets could be seen as being a domain specific modeling language for financial relationships, specifically useful for the depiction of creditor-debtor relationships.

**Example - introducing the swap of IOUs** Another simple example shall now illustrate the principle behind a swap of **IOUs**, which is a sort of double creditor-debtor relationship. The following assumptions are made for this illustrating example:

1. Bob needs a certain amount of **money** - we set it to 100 - for a project that he can start as soon as he has the full amount
2. The project will yield back his investment in 60 days
3. Bob wants to get the money from Alice
4. Alice is willing to lend the money to Bob
5. Bob wants to pay back to Alice as soon as his project yielded back the money
6. Alice does not have the money right now
7. Alice is sure that she will have the **money** necessary in 30 days
8. There will be no interest payments or discounts

Alice and Bob meet and they agree:

1. Alice will give the money to Bob as soon as she has it (in 30 days)
2. Bob will invest the money in his project right away
3. Bob will return the money to Alice as soon as his project has yielded it back

Bob and Alice have effectively agreed upon the swap of **IOUs**. Alice will pay Bob the amount of 100 in 30 days. Bob will pay back the full amount in 90 days from now.

If we put this swap of **IOUs** into a balance sheet that corresponds to double-entry bookkeeping, we receive the balance sheets in table 2. Alice now has an **asset** that is a claim on Bob to pay

Alice				Bob			
IOU <sub>90</sub>	+100	IOU <sub>30</sub>	+100	IOU <sub>30</sub>	+100	IOU <sub>90</sub>	+100

Table 2: How the balance sheets of Alice and Bob change, by a swap of **IOUs**.

100 in 90 days - IOU<sub>90</sub> as an **asset** (left hand side of Alice's balance sheet) - which corresponds to the liability of Bob to pay 100 to Alice in 90 days - IOU<sub>90</sub> as a liability (right hand side of Bob's balance sheet). The same double entry book keeping principle applies to the claim that now Bob has to receive 100 from Alice in 30 days, correspondingly Alice has to pay 100 in 30 days (right hand side of her balance sheet). This agreement is represented as IOU<sub>30</sub> in both stylized balance sheets.

Notably both balance sheets are enlarged by this operation. Both parties are creditors and debtors at the same time. It is a mutual *exchange* of **IOUs**. To understand this concept is a key step to understanding how banks can and do create what we call **bankmoney** in this work. **Bankmoney** being an **IOU** that is due right now (not in 30 or 90 days or any other time frame, but payable on demand), but is not paid out all the time immediately in full amount. Instead, bank customers use these, so called, 'deposits' that are payable on demand to make **payments** with the transfer of these 'deposits' (we rather use the term **bankmoney**). Why this works for the **BPS** is analysed in section 8.3.



### 8.2.2 Inside money and outside money

The idea to distinct **inside money** from **outside money** was probably first developed by Gurley *et al.* (1960) in an attempt to distinguish **bankmoney** (*deposits*) from **currency** (termed *fiat money*<sup>82</sup>). If we automatically think of a ‘thing’, when we are thinking of money, then most likely we assume - probably without consciously intending to do so - that *all money* is **outside money**. But that does not hold, if we take the actuality of money and banking into account.

**Outside money** **Outside money**, is an **asset** that is no ones liability on the ‘inside’ of the corresponding **payment system**. At the very least **outside money** is not expressly booked as a liability in any actors bookkeeping records. The argument could be made that any **outside money** needs to rely on a **MMI** that is not ‘just there’ but needs to be financed as well. This financing of the **MMI** creates a liability that is in nobodies bookkeeping records, but has to be paid eventually, too.<sup>83</sup>

**Inside money** **Inside money** does not just exist, it is being created. However this creation needs to be clearly understood as a dynamic process. It is the process of two (or more) contractors that do not have a contractual relationship at first, then they contract and thereby become each others creditors and debtors. At the end of the contractual relationship both are neither creditor nor debtor any more, when all obligations are fulfilled. So **inside money** creation is not only the once and for all creation of a token or thing that is bound to stay there forever. The creation of **inside money** does already include its ceasing to exist. **Inside money** is created with the initiation of the contract and destroyed in the fulfilment of the contract. During the course of the running contract, **inside money** can be treated as a ‘token’ in the time between initiation and fulfilment of the contract. In fact, all **money**, including **inside money**, is mistakenly treated as a ‘token’ in the conventional view on money, as it was described in section 8.1.1.

### 8.2.3 Money or credit

Whether **inside money** is **money** or mere credit depends on the viewpoint this question is asked from. **Bankmoney**, as an example, can be viewed as credit, because it can not be used as means of final **settlement of debt** among the **bankmoney** creating banks themselves. The same is true for debts that are due to the central bank, which is the bank of the commercial banks. Amounts payable to the central bank can never be paid by transferring **bankmoney**, these dues have to be settled by the central banks **currency**. From the viewpoint of the commercial banks among each other or even the central bank on top, **bankmoney** is mere credit. It can not be used as **money proper**, as means of effectuating final **settlement of debt** without discount.

This is different, if considered from the viewpoint of non-banks. Non-banks can indeed effectuate final **settlement of debt** without discount by **bankmoney**. Firstly, with their commercial banks and secondly, in wire transferring the **bankmoney** to other non-banks thereby effectuating **payments**.

The question whether a financial **asset** is **money** or credit cannot be finally answered once and for all, for each and every financial **asset**. The answer to this question changes depending on the

<sup>82</sup>The notion of fiat money implies that it is created from government decree. It is thought of as essentially falling out of the sky (or being dropped from a helicopter) as a thing or that it is simply created out of nothing and that it was created to stay. The notion of so called governmental ‘fiat money’ as being **outside money** is very misleading and does not correspond to the money view on actuality. In actuality central banks balance sheets are enlarged by the creation of so called ‘fiat money’. A better term for the misleading ‘fiat money’ is the term **currency** we use in this work. **Currencies** are **inside monies**.

<sup>83</sup>It will be interesting to see how much of a **MMI** can be resembled with the use of the **Network**, if it is used beyond a **CPS** and if eventually the cost for the financing of a **MMI** can be reduced in the future.

viewpoint that is taken within - what Mehrling calls - ‘the hierarchy of money’ (see [Mehrling, 2012](#), pg. 1).

**The hierarchy of monies** The hierarchy of money is built by the way contractual relationships are structured. So, for example, if on an international level contracts were nominated and payable in gold only, then gold would be the means of international payments (international **money proper**). If a central bank has to make an international payment of this kind (payable in gold), it can not use its own **money**, we call **currency**.

On a national level, commercial banks are customers of central banks. To make **payments** to the central bank, commercial banks can not employ their own **bankmoney**. The only means of payment (**money proper**) they are able to use is the one that is specified in the contract that created the debt. Normally, this is the **currency** that central banks create. Banks can not create **currency**, only central banks can do so. Central banks can not create (necessarily) the international **money proper**. This is what is creating the hierarchy.

On the bottom of the hierarchy, so to speak, are the non-banks. For non-banks **bankmoney** is **money (money proper)**, because non-banks can make **payments** to their bank creditors (commercial banks) and they can make **payments** to non-bank creditors using **bankmoney** (e.g. if an invoice contains a bank account number, this implies legally that the invoicing party will accept **payments** in **bankmoney**).

For commercial banks **bankmoney** is not **money**, it’s credit. They regard **currency** as being **money**, because they can effectuate **payments** to their creditors (other banks and the central bank) by using **currency**.

For central banks that have debts payable in a **currency** other than their own, to that extent their own **currency** is not **money**, even though it is **money** for the commercial banks and non-banks within its jurisdiction.

So, we see that this question whether it is **money** or mere credit can not be answered without the clear delimitation of the environment or the viewpoint this question is asked from. Once again, we need to be very precise in this **domain** of **money**.

**Relevance for cryptocurrencies** For any single **cryptocurrency** there is hardly any hierarchy to be observed yet, but this may change over time. The better the **MMI** for **cryptocurrencies** is, the more actors will nominate their contractual relationships in the **money of account** and will therefore potentially initiate a hierarchy of contractual relationships in the respective **cryptocurrency**.

### 8.3 Banking Payment System

The **payment system** that is provided by the cooperation of commercial banks of all kinds (private banks, credit unions, savings banks etc.), we call the **Banking Payment System (BPS)**. In this section we want to establish an understanding of the **BPS** to a degree that allows for the assessment of **Bitcoin** as a **Cryptocurrency Payment System** relative to the properties of the **BPS**.

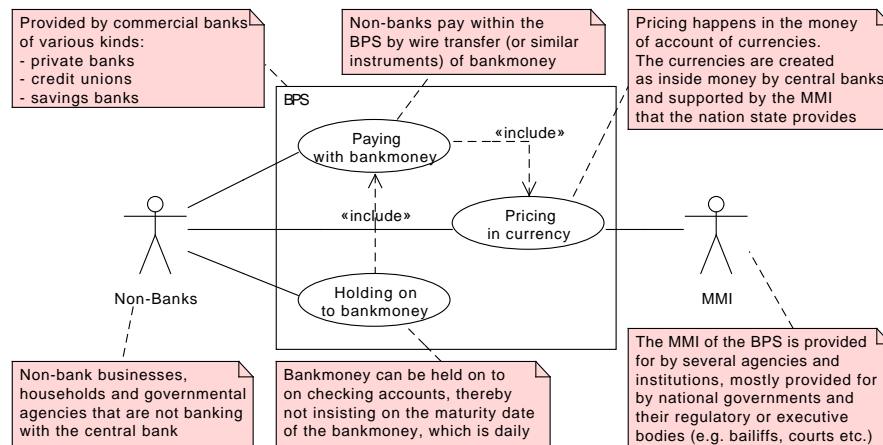


Figure 31: Use case diagram showing the **BPS** and its users, the so called non-banks.

#### Primary actors

- Commercial Banks, including private banks, credit unions and savings banks, as banks for the non-banks
- Non-Banks, i.e. private businesses, organizations and households

The users of the **BPS** are non-bank businesses and households that utilize it to make national (and potentially international) **payments**, e.g. by wire transfers, checks and electronic funds transfers.

#### Other key stakeholders

- National Governments (e.g. as participants in international contracts, national regulations, law enforcement, etc. - in short: as providers of the **MMI**)
- Central Banks as banks for the commercial banks

**What makes the BPS interesting for the assessment of CPS** The **BPS** does currently provide a type of **inside money** we call **bankmoney**, as described in section 8.3.1 below. Most notably this form of **money**, created by commercial banks, is not included in **legal tender** laws of governments, yet it *is* accepted in final settlement of debts in-between banks and non-banks but also in **payments** among non-banks. This is a very interesting condition that might bear great insights for the **CPS**. The reason being that the **CPS** - in the design provided for by the **Bitcoin** project (see section 7.3) - does also create a **money** that is not included in **legal tender** laws of governments. We want to look at differences in the **money proper** that the **BPS** creates and that is widely accepted, called **bankmoney**, and the proposed **money proper** of the **CPS** that hasn't yet reach such a level of acceptance, called 'bitcoin'.

### 8.3.1 How banks create inside money

Building on our understanding of a swap of IOUs (see section 8.2.1) we want to describe very briefly by means of an example how commercial banks create the **inside money** we call ‘**bankmoney**’.

The example is set up as follows:

- Two actors, a creditor and a debtor
- The creditor is called Commercial Bank, the debtor is called Customer
- The principal is 100, the maturity set to one year (360 days) and interest after one year is fixed at 5 per cent.
- The claim for the bank out of the credit agreement is named ‘Claim’ in the stylized balance sheet of the Commercial Bank
- The liability for the debtor out of the credit agreement is named ‘Claim’ in the stylized balance sheet as well, to indicate their mutual representation of one creditor-debtor relationship in the two balance sheet (confer the explanation on the use of stylized balance sheets in section 8.2.1)

Table 3 is depicting the change on both balance sheets after the credit agreement is made. This is showing the creation of **bankmoney** in this transaction, by increasing the Deposits.

Commercial Bank				Customer			
Claim <sub>360</sub>	+100	Deposits <sub>0</sub>	+100	Deposits <sub>0</sub>	+100	Claim <sub>360</sub>	+100

Table 3: How a commercial bank’s and its customer’s balance sheets are changing by an engagement in a credit agreement. The indices are indicating the time to maturity.

Table 4 shows the change on both balance sheets after the final payment is made, paying the amount of 100 for the principal of Claim<sub>0</sub> that is now due and the interest payment of 5, too, thereby reducing the equity of the Customer and increasing the equity of the Commercial Bank. This is the destruction of **bankmoney**, by reducing the overall Deposits by 100 (interest payments

Commercial Bank				Customer			
Claim <sub>0</sub>	-100	Deposits <sub>0</sub>	-105	Deposits <sub>0</sub>	-105	Claim <sub>0</sub>	-100
		Equity	+5			Equity	-5

Table 4: How a commercial bank’s and its customer’s balance sheets are changing by the final **payment** of principal (100) + interest (5).

ending up in bank’s equities are eventually either paid out in dividends or other consumption). This is very typical for **inside money**. It is being created in a creditor-debtor relationship and ceases to exist as soon as the creditor-debtor relationship ends.

### 8.3.2 Usage and acceptance of bankmoney

**Bankmoney**, being created in a mutual exchange of IOUs is **inside money**. We want to present an idea here why this form of money potentially is used and accepted the way it is.

**Creditor banks have to accept payments in their own bankmoney** The **bankmoney** creating commercial banks, do have to accept the tendering of **bankmoney** as **payment** by their own debtors. A bank creditor that is offered a payment in **bankmoney** from one of its debtor-customers has to accept this payment in final settlement of debt in the full amount of the offer. Declining their short term liabilities (called ‘deposits’ or better still **bankmoney**) as a means of payment or even just taking it only at a discount, is effectively declining their own promissory notes and banks doing this are thereby saying nothing less than that these notes are no good. This is equal to a default of the non-accepting bank. To avoid this type of default commercial banks have to accept their own short term-liabilities (maturity date ‘immediate’), we call **bankmoney**, in final **settlement of debt** without discount.

**Non-banks** We want to mention possible reasons why non-banks do voluntarily accept **bankmoney** as a means of final **settlement of debt** in their private non-bank to non-bank contracts, thereby possible reasons that revolve around mere usability and convenience issues.

Firstly, as a limitation, the non-banks use the **currency** of the respective nation state as **money of account**, as do all the banks. It is not the case that every single bank is introducing an individual **money of account**, however they arguable create individual **monies proper**. Again, it becomes clear how valuable it is to have a terminological toolbox rich enough to be able to distinct the **money of account** from the **money proper**. So, the non-banks and the banks rely on the **MMI** of their responsible and applicable jurisdiction.

Secondly, and being the key reason we suppose, non-banks probably accept **bankmoney** as **money proper** in final settlement of their debts, because they themselves (seen as an aggregate) are in debt to the banks and can settle those debts by means of the transfer of **bankmoney** to the bank. This is a point that has so far been mostly neglected in **cryptocurrency** research. For the potential users of **cryptocurrencies** are so far not at all in debt, nominated and payable in the **cryptocurrency**. There is - so far - no credit structure created that sits on top of any **cryptocurrency**. If a **cryptocurrency** is currently accepted as **money proper**, then this is most likely not due to debts that the accepting party (e.g. a merchant) needs to pay in the **cryptocurrency**.<sup>84</sup> This leads us to the conclusion on this work.

---

<sup>84</sup>On the contrary, most merchants accepting **bitcoin** today, do so only by using additional services the **EcoSystem** provides for, thereby relying on parts of the **MMI** that is supporting the **BPS**, see footnote 70.

## 9 Conclusion

This section is aiming to provide a compact conclusion on the entirety of this work, by first recapitulating the findings of this work, so far, and then subsequently drawing final conclusions out of these findings for the research question this work is based upon.

### 9.1 Recapitulation

We want to briefly recap this work's findings, by going through it step by step. This work was started by laying foundations on:

- Cryptography
- Cryptocurrencies
- Software Engineering

These foundations were laid in a compact way that allowed for the main part of the work to build upon them. These introductory chapters did not aim to gain any new insights whatsoever.

**Linguistic toolbox** We then developed a linguistic toolbox of terms in the domains of **money** and **cryptocurrencies**. This was done by performing a requirements analysis - focussing on functional requirements - for a hypothetical **payment system**. This analysis yielded the emergent properties of **payment systems**, the **money proper** and the **money of account** that were assigned into what is called the hierarchy of functional requirements for **payment systems** in this work.

**Assessing cryptocurrencies** Having this toolbox of precise terms, we started the main part of this work, which was the assessment of **cryptocurrencies** by reverse engineering parts of the **Bitcoin** project, especially the way it is used as a **payment system** by its users. This assessment was done through the view of a software engineer by means of modeling technology. The aim was never to produce an extensive class diagram that was only one step short of being executable code<sup>85</sup>, but rather by taking an approach from the macro view, having the user perspective in mind and the goal to get an understanding of functional principles of **cryptocurrencies**, executed by example of the **Bitcoin** project. The user perspective on **Bitcoin** seems important enough to take because it turns out that it is right now just not used as much as its proponents would like it to be used, in their hope it “will be bigger than Facebook”.<sup>86</sup>

**The MMI** Thanks to the toolbox developed by means of requirements engineering principles and to the understanding of the **Bitcoin** project through software modeling principles the role of the **MMI** for **payment systems** in general could be uncovered. It is the central insight of this work that the embeddedness of a **payment system** into a **Money Meta Infrastructure** is necessary, for it to yield a **money**, which is confidently used by users of the **payment system** as the **money of account** and accepted as the **money proper**.

The lacking of a **MMI** entirely for any **cryptocurrency** project or at least shortcomings within the **MMI** that are probably the main reason **cryptocurrencies** have not found more widespread use and acceptance. The lack of a proper **MMI** will be inhibiting further dissemination of **cryptocurrency** use and acceptance as long as these shortcomings are not overcome. Either by integrating **cryptocurrencies** into the already existing **MMI**, by means of regulation or other

---

<sup>85</sup>A class diagram of this kind can be generated any time, out of the already existing open source code of **Bitcoin** - see for example <https://github.com/bitcoin/bitcoin> - accessed July 29th 2014.

<sup>86</sup>see <http://www.cryptocoinsnews.com/news/winklevoss-twins-bitcoin-will-bigger-facebook/2014/05/20> accessed July 24th 2014.

measures, or by creating a totally new kind of **MMI** by cryptographic means, which - if done successfully - bears in it revolutionary potential not only for **currencies**, but for society in general. One possible idea, if a non-nation state **MMI** can be established by cryptographic or other means, is an institution we could best describe as a world credit union, powered by the (smart) property of millions of little shareholders and emitting an **inside money** that would be way more stable in prices, than the **outside money** that **bitcoin** is, is right now.<sup>87 88</sup>

## 9.2 On Bitcoin

In this subsection we provide our final thoughts on the specific **cryptocurrency** project we assessed in this work, called **Bitcoin** using all the tools and understanding portrayed up to this point.

**Agile view** Certainly, right now, **Bitcoin** does not utilize its full potential. Especially the lack of what we call the **MMI** and the high volatility in prices in **currencies** create challenges for merchants willing to use **bitcoin** as their **money of account**. Could Bitcoin - as a **CPS** - be considered like a first and early version of an already running software in an agile software development process? Could **Bitcoin** be considered a piece of running software that just does not yet cover all the extensions and exceptions? For example, if two parties are engaged in a contract that specifies **payment in bitcoin**, the **Bitcoin Network** is indeed able to provide final settlement of this debt that is specified in bitcoin, but does it - or rather the **MMI** it is embedded in - provide remedies for all the possible extensions and exceptions in this payment process? The answer currently has to be no, at least for the standard two party **transaction** that is currently predominantly used in the **Bitcoin CPS**.

**Bitcoin as money proper is outside money** Being able to settle debt that is specified in **bitcoin** means that the **Bitcoin Network** used as a **CPS** does provide for a working **money proper**, since the technology behind the **cryptocurrency** allows for effective **payments** within the **Network** as was shown in section 7.3.3. However the **money proper** created by the **Network** used as a **CPS** differs greatly from the **money proper** that the **BPS** offers. The **money proper** called **bitcoin** is **outside money**, for it is no-ones liability<sup>89</sup>, the **money proper** that banks create is **inside money**. If we wanted to show the creation of **bitcoin** after successful mining as **outside money**, using our notation of stylized balance sheets we would end up with the illustration in table 5 below.

	Miners
bitcoins	+25

Table 5: The creation of brand new **bitcoins** as **outside money** by a successful miner increases the bitcoin supply by (currently, July 2014) 25 bitcoins. There are no (obvious) corresponding liabilities anywhere, as there would be, if **bitcoins** were created as **inside money**.

In this sense, the **Bitcoin** project does indeed create “a new kind of money” it is an endogenous money that is created within the payment system itself (by the process of **mining**), but it is not

<sup>87</sup>Based on an idea of Prof. Dr. Dr. Gunnar Heinsohn, expressed in a weblog, see <https://blog.malik-management.com/gelddigitalisierung-und-eigentumssoekonomie/> - accessed July 30th 2014.

<sup>88</sup>This idea, as is a non central government based **MMI** in general, certainly is far out. But, whatever we can think of might be possible.

<sup>89</sup>As is totally correctly noted in the assessment of **cryptocurrencies** that the German **Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin)** published most recently (see [Münzer, 2014](#)). It is interesting to find that - unlike most economists - some regulatory bodies (like the **BaFin** in this case) seem to have a look on **monies** quite close to what we briefly described as the money view on actuality in section 8.



**inside money** as the endogenous **money** created in the **BPS**, called **bankmoney**. This suggests a careful distinction of endogenous money and **inside money** that was so far unnecessary for the **BPS**.

**Bitcoin Network does not have a fully functioning money of account** Currently **bitcoin** is not widely used as a **money of account** in purchase agreements by large sections of the population in any nation state. The pricing of **assets**, goods and services is still mainly done in **currencies**. So far, **bitcoin** is hardly used in credit agreements or any other type of long term contract anywhere. We suppose, this has to do with the lack of an effective **MMI** that allows for contract enforcement and other supporting services and measures, as was explained in section 6.6. A lack in the embedded **MMI** of this kind is what makes the proposed **money of account**, called ‘**bitcoin**’, unattractive to use in any contract.

Additionally the high volatility in prices in **currency** are problematic for merchants. This probably has to do with the way **bitcoins** are created as a form of **outside monies** and the **MMI** not yet providing for deep-pocket market makers, therefore lacking high **market liquidity**, which results in high volatility of prices. This is a field that almost screams for future research.

**More regulation** If the lacking of an effective **MMI** was indeed responsible for the still limited success of **cryptocurrencies**, the aim to improve the **MMI** suggests itself. The question is in what way this improvement can and will be made in the future. There are two possibilities:

1. Integrate **Bitcoin** into financial regulation that is provided by central governments, to profit from the **MMI** that is also used by the **BPS**
2. Further the possibilities that the **Network** itself may provide, by investigating and implementing **cryptocurrency** related ‘contracts’, as described in section 7.5, and other features of the **Network** that are yet laying dormant and are waiting for applicable implementation.

**1. Integrate into existing MMI** The call for a more effective and reliable **MMI** for **Bitcoin** might be construed as a call for an integration of **Bitcoin** into the **MMI** that currently powers the **BPS** by means of regulation and other measures. The regulation however, relying on central authorities enforcing them, is exactly what does not correspond to the notion that **Bitcoin** might remove any third party from financial transactions in general.

A steadfast decline of interconnecting **Bitcoin** with regulatory bodies of any nation state, by proper regulation, might potentially undermine further success for **Bitcoin** and other **cryptocurrencies** as widely used and accepted **monies**. Such a decline is rather unlikely in the long term, since even today if there are problems with trades in **bitcoin** and **currency**, the courts of nation states are used by litigant parties.<sup>90</sup>

On the other hand regulation might be considered taking away an essential use case for **Bitcoin** that is currently made use of by people that are critical of central governments and the **BPS** in general, which is stashing away monetary or near-money **assets** that are completely outside of the control of central governments.

**2. Create a totally new MMI** The currently most used type of two party **transaction**<sup>91</sup> will most likely have to loose importance in usage, to other types of **transactions**, because it does

<sup>90</sup>for a court ruling on failed **bitcoin** delivery, see <http://www.coindesk.com/dutch-court-declares-bitcoin-isnt-money-in-civil-trial/> - accessed July 29th 2014.

<sup>91</sup>The standard two party type of **transaction** is known as ‘Pay-to-PubkeyHash’, because the **payment** is made to the hash of the public key provide, as is described extensively in section 7.3 - see also <https://en.bitcoin.it/wiki/Transactions#Pay-to-PubkeyHash> - accessed July 30th 2014.



not provide for built-in remedies of any kind in case there was any nuisance in the underlying agreement that made the **payment** necessary in the first place.

Other forms of transactions will gain in importance immediately, if they can resemble parts of what we called the **MMI** in this work.<sup>92</sup>

However, it has to be noted that no matter how complex and sophisticated the ‘contracts’ that are possible to create with new types of **transaction**, there will always be a ‘physical’ component to contract enforcement and other **MMI** necessities. This will most likely leave **cryptocurrencies** with only one choice: find a way to integrate into the **MMI** of nation states, without **cryptocurrencies** losing every aspect of their power of fascination.

**Furthering monetary theory** Due to their current design, if the worst came to the worst for **cryptocurrencies**, they could at any time experience a massive blow to their acceptance and credibility. Such an event would not have to be highly likely, but certain technical occurrences, especially concerning the cryptography employed by the project, could potentially mean the sudden death of any **cryptocurrency**, if they ever occurred. However, even in this case, **cryptocurrencies** might still provide a significant contribution. If they effectively won’t do anything, but inspire monetary theorists to create an understanding of **money** that allows for a more complete assessment of cryptocurrencies, then this alone could be regarded as being a very valuable contribution to the ongoing discourse about money and banks. At least since the beginning of the financial crisis of 2007/2008, the better understanding of **money** and the inclusion of money and banks in macroeconomic theory and political practice is widely discussed, as it is now - probably more than ever since the Great Depression of the 20th century - perceived to be much needed.<sup>93</sup> We believe to have shown in this work that the specificity of the design of **cryptocurrency** calls for an understanding of **money** that goes way beyond the view of **money** being nothing more than a special kind of good that lowers cost of transactions in a barter economy.

---

<sup>92</sup>There is one other type of **transaction** available so far that is called a m-of-n multi-signature transaction which is a form of ‘Pay-to-Script-Hash’ transaction. The ‘Pay-to-Script-Hash’ transactions allow for what is called ‘contracts’, since they make it possible to create complex conditions for the transaction outputs to be redeemed.

<sup>93</sup>see ongoing discussion at the Institute for New Economic Thinking (INET) at <http://ineteconomics.org/> - accessed July 29th 2014.

## 10 Future research

In this closing section on future research we want to very briefly touch upon certain alleys that opened up as potential ways of further inquiry in the creation of this work, which is topically sitting on the intersections of multiple fields of research, as was mentioned in the preface of this work.

**Furthering the understanding of the MMI** Further research is needed on the scope of a **MMI** that actually is necessary to run a successful **cryptocurrencies**. A potential start would be to enquire banking practitioners about the properties they demanded for a **cryptocurrency** to be used as **money of account** in credit agreements.

**Clarifying legal remedies** To enhance the research on the **MMI** necessary for a **payment system** based on a **cryptocurrency**, a deep enquiry of currently available legal remedies might be promising. We have so far just stated their use, e.g. in the activity diagram in figure 8 on page 32, without describing them in detail. If this enquiry on legal remedies is done, a step-by-step comparison of the legal remedies on one hand and the current and potential capabilities of **cryptocurrencies** on the other. We have briefly touched upon potential capabilities of **cryptocurrencies** that go beyond the use of them as a mere **payment systems** in section 7.5.

**Contracts, trust and communication** Much more research is needed on contracts in digital environments in general. A potential starting point for such an inquiry could be Szabo (1997).

Maybe even going back another step and starting with communications in general, looking at the actual requirements of societies is a viable approach. Currently, in western societies there certainly is a requirement for being able to formally communicate by contracts. These contracts are currently formalized by civil legal procedures and secured by collateral that is **property** (cf. Heinsohn & Steiger, 1996). The contracts need to be enforceable, if necessary, by reasonable, effective means. Hence the need for a **MMI**. Contracts of this type are nothing but a form of communication.

The idea of money as a medium of communications is from Luhman, being another potential starting point for further inquiry. In his view this communication consists first and foremost of **payments**. He once deemed **payments** the ‘unit act’ of economics (see Luhmann, 1988, pg. 52). Already in 1973 Luhman wrote about trust:

“He who has money does not have to trust others. The general trust in the institution of money replaces the single, uncountable and difficult expressions of trust that were necessary to guarantee the fulfilment of human necessities in a cooperative society, by one single global act.”<sup>94</sup>

He clearly sees money as the central point of trust. **Cryptocurrencies** now try to dissolve this centrality. But do they dissolve the essence of **money** with that?

There is much more research needed in this field. The feeling remains that the inquiry has hardly even begun. It may start with an assessment of the potential of the bitmessage protocol.<sup>95</sup>

---

<sup>94</sup>This is not a direct quote but a translation by the author of this work at hand. The original quote by Luhman in German is: “Wer Geld hat, braucht insoweit anderen nicht zu vertrauen. Das generalisierte Vertrauen in die Institution des Geldes ersetzt dann jene unzähligen einzelnen und schwierigen Vertrauensweise, die nötig wären, um den Lebensbedarf in einer kooperativen Gesellschaft sicherzustellen, durch einen Globalakt.” (see Luhmann, 1973, pg. 55).

<sup>95</sup>see Warren (2012).

**Deep modeling** The conclusions on **money**, the hierarchy of money and other types of money, currencies and the emergent properties of payment systems in general (**money of account** and **money proper**) are potentially suitable for further inquiry using state of the art modeling technology that is termed ‘deep modeling’, see [Kennel \(2012\)](#). Initially the inclusion of deep modeling technology was thought of as potentially being already part of this work, but it turned out to be beyond this work. Nevertheless the feeling remains that this **domain** of **monies** does need further inquiry from software engineers that bring a totally different perspective to the picture than economists do. Deep modeling might be the next step of valuable tools to further the understanding of this domain with the software engineering perspective.

**Furthering systems engineering on financial systems** Not only software but systems engineering in general might be a promising path to an increased understanding of **money**, banks and the financial system in general and there is certainly to be found some low hanging fruits in systems engineering in the deeper evaluation of **cryptocurrencies**. For interesting approaches to modeling banks and the financial system with the help of systems engineering tools we refer to [Keen \(2004\)](#) and most recently [Keen \(2014\)](#).

**Triple entry bookkeeping** A potentially promising field of research right on this intersection of **cryptography** and **money** is so called triple entry bookkeeping. While we have mentioned double entry bookkeeping, when we were describing the creation of **bankmoney**, we could not touch on triple entry bookkeeping at all in this work. For more information on triple entry bookkeeping, we refer to [Grigg \(2004\)](#) and [Grigg \(2005\)](#).

**Disclaimer** With this work being ‘completed’ by now, I just barely understand enough to essentially realize that most of this work’s insights needed to be looked at from many additional angles to produce conclusions that could potentially render actual real-life use right away.

The hope remains that readers of this work gain insights on the issues covered nonetheless and can push the boundary of understanding much further in future research than was ever aimed for and possible with this work.

## Bibliography

- Abel, Andrew B. 1992. Can the government roll over its debt forever. *Business Review*, 3–18.
- Alexander, Ian F, & Beus-Dukic, Ljerka. 2009. *Discovering requirements: how to specify products and services*. John Wiley & Sons.
- Barber, Simon, Boyen, Xavier, Shi, Elaine, & Uzun, Ersin. 2012. Bitter to Better ,Â How to Make Bitcoin a Better Currency. *Pages 399–414 of: Keromytis, AngelosD. (ed), Financial Cryptography and Data Security. Lecture Notes in Computer Science, vol. 7397. Springer Berlin Heidelberg.*
- Booch, Grady, Rumbaugh, James, & Jacobson, Ivar. 1998. Unified Modeling Language (UML). *Rational Software Corporation, Santa Clara, CA, version, 1.*
- Bos, Joppe W, Halderman, J Alex, Heninger, Nadia, Moore, Jonathan, Naehrig, Michael, & Wustrow, Eric. 2013. Elliptic Curve Cryptography in Practice. *IACR Cryptology ePrint Archive, 2013, 734.*
- Bourque, Pierre, & Fairley, Richard E. 2014. Guide to the software engineering body of knowledge, Version 3.0.
- Brunner, Karl, & Meltzer, Allan H. 1971. The uses of money: money in the theory of an exchange economy. *The American Economic Review*, 784–805.
- Campiglio, Emmanuele. 2012 (August 8th). *Including banks in macroeconomic models - finally.* <http://www.neweconomics.org/blog/entry/including-banks-in-macroeconomic-models-finally> - accessed July 13th 2014.
- Cap, Clemens H. 2012. Bitcoin—das Open-Source-Geld. *HMD Praxis der Wirtschaftsinformatik, 49(1), 84–93.*
- Cockburn, Alistair. 2004. *Writing effective use cases*. 10 edn. The Agile software development series. Boston ; Munich [u.a.]: Addison-Wesley.
- Diffie, Whitfield, & Hellman, Martin E. 1976. New directions in cryptography. *Information Theory, IEEE Transactions on, 22(6), 644–654.*
- Emmons, William R. 1995. Interbank Netting Agreements and the Distribution of Bank Default Risk. *Federal Reserve Bank of St. Louis Working Paper Series.*
- Falkvinge, Rick. 2011 (accessed Nov 28th 2012). *Banks: The Fourth Victim of Citizen's empowerment.* <http://www.youtube.com/watch?v=mjmuPqkVwWc>.
- Goodhart, Charles AE. 2008. Money and default. *Chap. 13., pages 213–223 of: Forstater, Matthew, & Wray, L. Randall (eds), Keynes for the Twenty-First Century: The Continuing Relevance of the General Theory.* Palgrave Macmillan.
- Grigg, Ian. 2004. The ricardian contract. *Pages 25–31 of: Electronic Contracting, 2004. Proceedings. First IEEE International Workshop on. IEEE.*
- Grigg, Ian. 2005. Triple Entry Accounting. *Systemics Inc.*
- Gurley, John G, Shaw, Edward Stone, & Enthoven, Alain C. 1960. *Money in a Theory of Finance*. Brookings Institution Washington, DC.
- Hahn, Frank H. 1982. *Money and inflation*. Mitsui lectures in economics. Oxford: Blackwell.

- Heinsohn, Gunnar, & Steiger, Otto. 1989. The Veil of Barter: The Solution to The Task of Obtaining Representations of an Economy in which Money is Essential. *Inflation, Income Distribution and Capitalist Crisis*, London: Macmillan, 175–204.
- Heinsohn, Gunnar, & Steiger, Otto. 1996. *Eigentum, Zins und Geld : ungelöste Rätsel der Wirtschaftswissenschaft*. 1 edn. Reinbek bei Hamburg: Rowohlt.
- Hicks, John R. 1967. *Critical essays in monetary theory*. Oxford: Clarendon Pr.
- Hobson, Dominic. 2013. What is Bitcoin? *XRDS*, **20**(1), 40–44.
- Jevons, William Stanley. 1875. *Money and the mechanism of exchange*. The International scientific series, vol. 17. New York: Appleton.
- Katz, Jonathan, & Lindell, Yehuda. 2007. *Introduction to modern cryptography: principles and protocols*. CRC Press.
- Keen, Steve. 2004. *Using systems engineering software to build a model of the monetary circuit*. Tech. rept. Society for Computational Economics.
- Keen, Steve. 2014. *Modeling Financial Instability*. Tech. rept. Working Paper.
- Kennel, Bastian. 2012. A unified framework for multi-level modeling.
- Luhmann, N. 1973. Vertrauen: ein Mechanismus der Reduktion sozialer Komplexität.
- Luhmann, Niklas. 1988. *Die Wirtschaft der Gesellschaft*. 1 edn. Frankfurt am Main: Suhrkamp.
- Mankiw, Nicholas Gregory. 2007. *Macroeconomics*. 6 edn. New York, NY: Worth.
- Mankiw, Nicholas Gregory. 2014. *Principles of macroeconomics*. Cengage Learning.
- Mehrling, Perry. 2010. *The New Lombard Street*.
- Mehrling, Perry. 2012. The inherent hierarchy of money. *Festschrift for Duncan Foley, New York, NY*.
- Mehrling, Perry. 2013. Essential hybridity: A money view of FX. *Journal of Comparative Economics*, **41**(2), 355–363.
- Menger, Karl. 1892. On the origin of money. *The Economic Journal*, **2**(6), 239–255.
- Moore, Tyler, & Christin, Nicolas. 2013. Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk. *Pages 25–33 of: Sadeghi, Ahmad-Reza (ed), Financial Cryptography and Data Security*. Lecture Notes in Computer Science, vol. 7859. Springer Berlin Heidelberg.
- Münzer, Jens. 2014. Bitcoins: Supervisory assessment and risks to users. *BaFinJournal*, Feb 17.
- Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system.
- Nielsen, Michael. 2013 (Dec 6). *How the Bitcoin protocol actually works*. <http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/>.
- Ostroy, Joseph M, & Starr, Ross M. 1988. *The transactions role of money*. Department of Economics, University of California.
- Sommerville, Ian. 2011. *Software engineering*. 9 edn. Boston ; Munich [u.a.]: Pearson.

- Stavárek, Daniel. 2013. Lessons Learned from the 2013 Banking Crisis in Cyprus. *European Financial Systems 2013*, 312.
- Szabo, Nick. 1997. Formalizing and securing relationships on public networks. *First Monday*, **2**(9).
- Tapscott, Don. 1995. *The Digital Economy: Promise and Peril in the Age of Networked Intelligence*. McGraw-Hill.
- Theil, Wolfgang. 2000. *Bürgerliches Recht, Geld und zinsinduzierte Geldknappheit: ein Beitrag zur Heinsohn/Steiger-Riese-Kontroverse*. Inst. für Konjunktur-und Strukturforschung.
- Theil, Wolfgang. 2001. Eigentum und Verpflichtung. *Pages 175–200 of*: Stadermann, H.J., Steiger Otto (ed), *Verpflichtungsökonomik. Eigentum, Freiheit und Haftung in der Geldwirtschaft*. Metropolis.
- Warren, Jonathan. 2012. Bitmessage: A peer-to-peer message authentication and delivery system. *white paper*, <https://bitmessage.org/bitmessage.pdf>, (27 November 2012).

## **Ehrenwörtliche Erklärung**

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, 2. Oktober 2014

Unterschrift