



**BRNO UNIVERSITY OF TECHNOLOGY**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF INFORMATION SYSTEMS**

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

**PROCESSING OF THE BLOCKCHAIN EMPLOYING IPFS**

VYUŽITÍ IPFS PRO ZPRACOVÁNÍ BLOCKCHAINU

**MASTER'S THESIS**

DIPLOMOVÁ PRÁCE

**AUTHOR**

AUTOR PRÁCE

**Bc. MATÚŠ MÚČKA**

**SUPERVISOR**

VEDOUCÍ PRÁCE

**Ing. VLADIMÍR VESELÝ, Ph.D.**

**BRNO 2020**

## **Abstract**

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

## **Abstrakt**

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

## **Keywords**

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

## **Klíčová slova**

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

## **Reference**

MÚČKA, Matúš. *Processing of the Blockchain Employing IPFS*. Brno, 2020. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Vladimír Veselý, Ph.D.

# Processing of the Blockchain Employing IPFS

## Declaration

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Matúš Múčka  
January 10, 2020

## Acknowledgements

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant apod.).

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Cryptocurrencies</b>	<b>3</b>
2.1	Ethereum . . . . .	4
2.2	Bitcoin . . . . .	4
2.3	DigiByte . . . . .	4
2.4	Decred . . . . .	4
2.5	Monero . . . . .	4
<b>3</b>	<b>Decentralization (Web3.0)</b>	<b>5</b>
<b>4</b>	<b>IPFS</b>	<b>6</b>
4.0.1	Cluster . . . . .	7
4.0.2	Node . . . . .	7
4.0.3	CID . . . . .	7
4.1	IPLD . . . . .	7
4.1.1	Formats . . . . .	7
4.1.2	Routing . . . . .	7
4.1.3	Exchange . . . . .	7
4.1.4	Objects . . . . .	7
4.1.5	Files . . . . .	7
4.1.6	Naming (IPNM) . . . . .	7
<b>5</b>	<b>Design</b>	<b>8</b>
5.0.1	Blockbook . . . . .	8
5.1	Feeder . . . . .	9
5.2	Explorer . . . . .	9
<b>6</b>	<b>Implementation</b>	<b>10</b>
<b>7</b>	<b>Conclusion</b>	<b>11</b>
	<b>Bibliography</b>	<b>12</b>

# Chapter 1

## Introduction

HTTP is „good enough“ for the most use cases of distributing files over the network (like webpages, etc.). But when we want to stream lots of data to multiple connected clients at once, we are starting to hit its limits. When two clients are requesting the same data, there is no mechanism in HTTP that would allow sending the data only once. Sending duplicate data has become a problem in large companies because of bandwidth capacity. Blizzard <sup>1</sup> started to distribute video game content by distributed solution because it was cheaper for the company and faster for players [2]. Linux distributions use BitTorrent to transmit disk images <sup>2</sup>.

The bitcoin blockchain has now 242 gigabytes <sup>3</sup>. When blockchain is processed (parsed address, created search indexes), the size on a disk can double. If there are multiple blockchains, then data can have few terabytes. When we are sharing blockchains data from the server for several clients, there is a big chance that multiple clients want the same data. They may be working on the same case and investigating the same wallets. So in standard solution with relational database and some HTTP server, for every request server has to search in all data (that can have a size of few terabytes) and transmit selected data to the client. This happens even if a different client asks for the same data in a few minutes ago. Behaviour mentioned above dramatically limits the scalability of the server.

Services that are using HTTP, often have client-server architecture, so there is also a problem with one point of failure. If the server for some reason stops working, the client can not receive data. In distributed file system such as IPFS, there is no such problem as one point of failure, because all data are duplicated on multiple clients.

---

<sup>1</sup>Game company

<sup>2</sup>Image of Debian downloadable by BitTorrent <https://www.debian.org/CD/torrent-cd/>

<sup>3</sup>Current size of bitcoin blockchain can be seen at <https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/>

## Chapter 2

# Cryptocurrencies

There were hundreds of failed attempts of creating cryptographic payment systems before cryptocurrencies like Bitcoin and Ethereum come into existence. Some of these systems are listed in figure 2.2. All of them were created before Bitcoin, and despite that, some of these attempts were only academic proposals, others were actually deployed and tested systems, only a few of them survived to these days. One of the survival is PayPal, but only because it quickly give up its original idea of hand-held devices for cryptographic payments.[4]

So there is a question, what makes cryptocurrencies successful nowadays? It may be it's easy to use principle and no need for external hardware. Another critical component of cryptocurrencies discussed in this work is Blockchain. Simply, it is a ledger in which are all transactions securely stored. The idea behind blockchains is pretty old, and it was originally used for timestamping digital documents.[1]

ACC	CyberCents	iKP	MPTP	Proton
Agora	CyberCoin	IMB-MP	Net900	Redi-Charge
AIMP	CyberGold	InterCoin	NetBill	S/PAY
Allopass	DigiGold	Ipin	NetCard	Sandia Lab E-Cash
b-money	Digital Silk Road	Javien	NetCash	Secure Courier
BankNet	e-Comm	Karma	NetCheque	Semopo
Bitbit	E-Gold	LotteryTickets	NetFare	SET
Bitgold	Ecash	Lucre	No3rd	SET2Go
Bitpass	eCharge	MagicMoney	One Click Charge	SubScrip
C-SET	eCoin	Mandate	PayMe	Trivnet
CAFÉ	Edd	MicroMint	PayNet	TUB
CheckFree	eVend	Micromoney	PayPal	Twitpay
ClickandBuy	First Virtual	MilliCent	PaySafeCard	VeriFone
ClickShare	FSTC Electronic Check	Mini-Pay	PayTrust	VisaCash
CommerceNet	Geldkarte	Minitix	PayWord	Wallie
CommercePOINT	Globe Left	MobileMoney	Peppercoin	Way2Pay
CommerceSTAGE	Hashcash	Mojo	PhoneTicks	WorldPay
Cybank	HINDE	Mollie	Playspan	X-Pay
CyberCash	iBill	Mondex	Polling	

Figure 2.1: Electronic payment systems before cryptocurrencies [3]

## 2.1 Ethereum

## 2.2 Bitcoin

>You will not find a solution to political problems in cryptography.

Yes, but we can win a major battle in the arms race and gain a new territory of freedom for several years.

Governments are good at cutting off the heads of a centrally controlled networks like Napster, but pure P2P networks like Gnutella and Tor seem to be holding their own.

Satoshi

Figure 2.2: Satoshi Nakamoto at vistomail.com *Thu Nov 6 15:15:40 EST 2008* <sup>1</sup>

## 2.3 DigiByte

## 2.4 Decred

## 2.5 Monero

## Chapter 3

# Decentralization (Web3.0)



## Chapter 4

# IPFS

IPFS stands for InterPlanetary File System and is a peer-to-peer distributed filesystem designed to make the Web faster, safer, and more open. In contrast with standard filesystems, objects in IPFS are content-addressed, by the cryptographic hash of their contents. In the case of the standard Web, when user wants some file, he needs to know on which server is a file located and the full path to the file. In IPFS user needs only to know the hash of the requested file. He does not care about the location of the file.

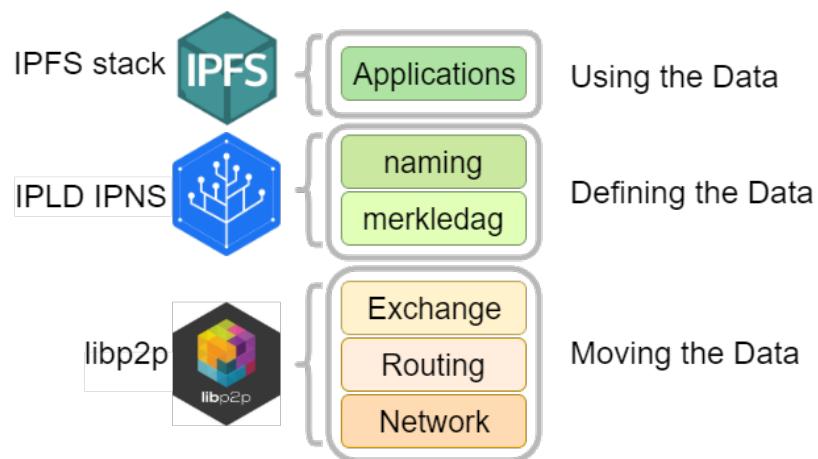


Figure 4.1: IPFS stack

**4.0.1 Cluster**

**4.0.2 Node**

**4.0.3 CID**

**4.1 IPLD**

**4.1.1 Formats**

**4.1.2 Routing**

**4.1.3 Exchange**

**4.1.4 Objects**

**4.1.5 Files**

**4.1.6 Naming (IPNM)**

# Chapter 5

## Design

The system consists of one or more Feeders and Explorers. Feeders are connected to external API and provide synchronization between cryptocurrency and IPFS data. Explorer can request data from the system network and display them to user.

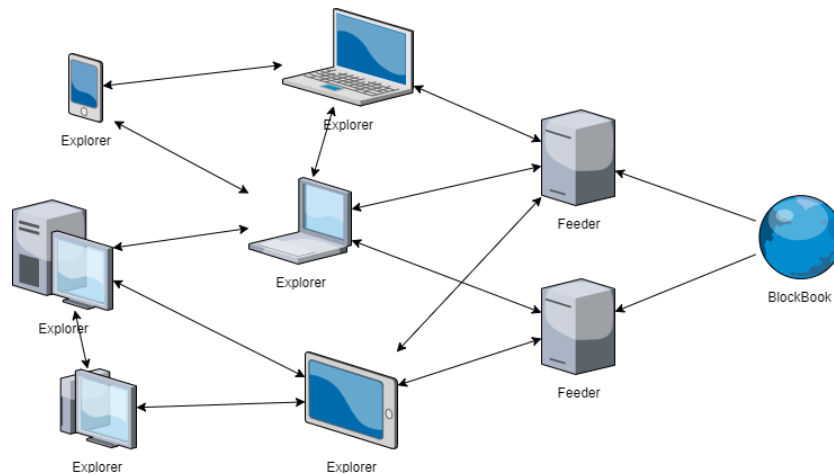


Figure 5.1: System design

### 5.0.1 Blockbook

Blockbook<sup>1</sup> is a blockchain indexer for Trezor Wallet<sup>2</sup>, developed by SatoshiLabs<sup>3</sup>. It currently supports more than 30 coins (and some others were implemented by community). For data storage Blockbook is using RocksDB<sup>4</sup> developed by Facebook which is NoSql database which stores only key-value pairs. Blockbook is providing fast api for accessing blocks, addresses and transactions. Main limitations of blockbook:

- **Not distributed** (client-server architecture) - problem with scalling for more users.
- **Not a SQL database** - it does not have a relational data model, it does not support SQL queries, and it has no support for indexes.

<sup>1</sup><https://github.com/trezor/blockbook>

<sup>2</sup><https://wallet.trezor.io/>

<sup>3</sup><https://satoshilabs.com/>

<sup>4</sup><https://github.com/facebook/rocksdb/wiki>

- **Single-Process** - only a single process (possibly multi-threaded) can access a particular database at a time.

## 5.1 Feeder

A Feeder is a command-line application that stores data in IPFS for all cryptocurrencies specified in the config file. For obtaining data it uses Blockbook API. Feeder stores data in structure such as in figure 5.2. Each block (except genesis and last block) has a link to the previous and next block. Also, it has links to transactions that had been processed in this block. A transaction has links to address and previous/spent transaction for every input/output. This scheme allows store blockchain data in IPFS in small objects with size less than 256kb (a limit for storing objects directly in DHT). Also, every logical link between objects is preserved.

The feeder should create indexes as it stores objects. These indexes will help Explorer perform queries.

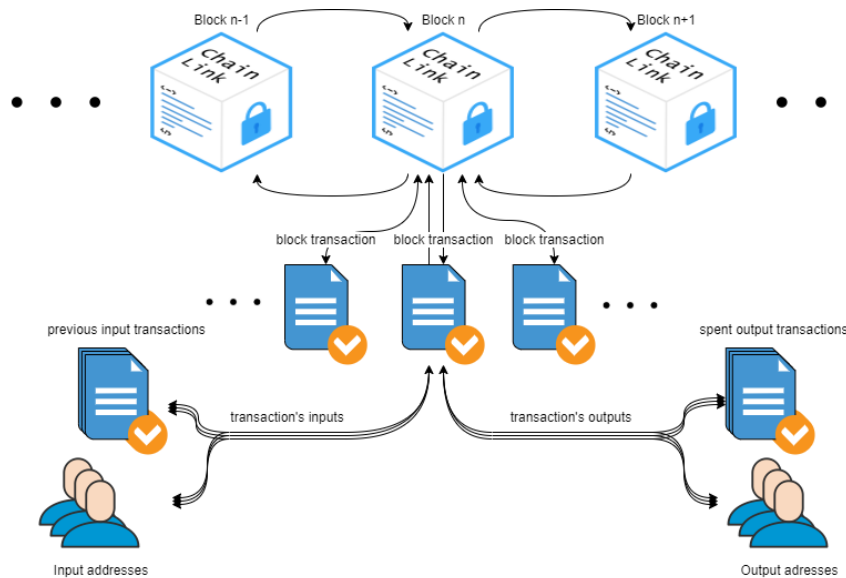


Figure 5.2: Feeder data storage structure

## 5.2 Explorer

Explorer is application providing simple gui and API. Application is runnable in browser or Node.js<sup>5</sup>. Explorer can perform basic queries like search for block by its height or hash and search address and transaction by hash. But depending on specified indexes in Feeder, Explorer can also make more complex queries as for example get 20 transaction where sum of inputs is more than 0.5, or get transactions between some time interval.

<sup>5</sup><https://nodejs.org/>

## Chapter 6

# Implementation

## Chapter 7

## Conclusion

# Bibliography

- [1] HABER, S. and STORNETTA, W. S. How to time-stamp a digital document. In: Springer. *Conference on the Theory and Application of Cryptography*. 1990, p. 437–455.
- [2] KALLE, K. *Big data in video games*. Lappeenranta, FI, 2017. Bakalářská práce. Lappeenranta University of Technology, School of Business and Management, Computer Science. Available at: <http://lutpub.lut.fi/handle/10024/147666>.
- [3] NARAYANAN, A., BONNEAU, J., FELTEN, E., MILLER, A. and GOLDFEDER, S. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.
- [4] WAYNER, P. *Digital cash: Commerce on the net*. Academic Press Professional, Inc., 1997.