# Abstract

Vladislav Novakovsky

This whitepaper proposes common protocol “One way pegged sidechains” that allows to abstract crypto tokens (and other ledger assets) to be transferred between blockchains. Assets may be abstracted from their “physical carriers” – peer-to-peer networks/blockchains – lifetime of abstract tokens may be longer than lifetime of networks of their emission.

There is demand to increase interoperability between blockchains. Some efforts are made but they rely on trusted party. This document proposes censorship resistant solution using a group signature – the same consensus mechanism Bitcoin uses.

The goal is to develop protocol (stack) general enough to increase the number of blockchain projects that can implement it for standardized way of transferring tokens between blockchains. Open system interconnection Model is used as the guideline. This investigation is in Analysis phase yet – gathering requirements. It establishes the directions for definition of protocol (API) but the protocol itself should be defined in Design phase when requirements are clear and stable. 2023-May-03

# Scope and Classification.

**Cryptocurrency** – name most widely used in mass media. That is digital currency supported by cryptographic methods.

**E-cash** – this name was often used in whitepapers published 10-30 years ago [[E-Cash Payment Protocols]](https://www.researchgate.net/publication/266279383_E-Cash_Payment_Protocols):

*“****E-cash*** *is a payment system designed and implemented for making purchases over open networks such as the Internet.”*

“*A purely peer-to-peer version of electronic cash would allow online*

*payments to be sent directly from one party to another…”* [Nak09]

*"Bitcoin is a collection of concepts and technologies that form the basis of a digital* ***money*** *ecosystem.* ***Units of currency*** *called bitcoin are used to store and transmit value among participants in the Bitcoin network. ...*

*...Users can transfer bitcoin over the network to do just about anything that can be done with* ***conventional currencies****, including buying and selling goods, sending money to people or organizations, or extending credit. Bitcoin can be purchased, sold, and exchanged for other currencies at specialized currency exchanges. Bitcoin is arguably the perfect* ***form of money*** *for the internet because it is fast, secure, and borderless... Unlike traditional currencies, the bitcoin* ***currency*** *is entirely virtual. There are no physical coins or even individual digital coins. The* ***coins*** *are implied in transactions that transfer value from spender to receiver."*[MASTBIT2023]

What is the proper name for assets used to make purchases in such systems as Bitcoin? How to classify such financial assets using traditional financial terminology? Since we are going to separate economical/intellectual property aspect of a currency from technological aspects/form of representation we should investigate abstract characteristics of the underling object.

“*To understand the effects of money on the economy, we must understand exactly what money is. In this chapter, we develop precise definitions by exploring the functions of money, looking at why and how it promotes economic efficiency, tracing how its forms have evolved over time….To avoid confusion,* ***we must clarify*** *how*

*economists’ use of the word money differs from conventional usage…*

*…Economists define money (also referred to as the money supply) as anything that is generally accepted in payment for goods or services or in the repayment of debts. Currency, consisting of dollar bills and coins, clearly fits this definition and is one type of money. When most people talk about money, they’re talking about* ***currency*** *(paper money and coins)*.*”* [Mishkin2004]

Following are the characteristics/functions of **currency**:

* 1. Medium of exchange;
  2. A Unit of account;
  3. Divisible;
  4. Durable;
  5. Fungible;
  6. Portable.

**Money** includes all mentioned characteristics plus “Storage of value”. Nowadays cryptocurrencies are considered more as speculative asset then storage of value. High correlation with Nasdaq shows that it’s difficult to consider cryptocurrencies as safe heaven/money/storage of value. At least today. Some economists do not agree with Mishkin[Mishkin2004]: they claim that paper currency is not money because they cannot be considered as “Storage of value” in the long term. Therefor we will use term “currency”. “*As*

*you can see, there is no single, precise definition of money or the money supply, even for economists*.” [Mishkin2004] – and here we agree with Mishkin.

Some economists claim that cryptocurrencies can’t be used even as currencies because of their rate volatility – it’s difficult to trade using currency that doesn’t support stable prices and has significant rate fluctuations. But nowadays we can observe that many fiat currencies are impacted by inflation introducing difficulties in trade so we will stick with the term “currency”.

## The scope.

Blockchain allows controlling ownership of different “smart assets” but here we will focus on currency. Because cryptocurrency transfer is the predominant role of the most blockchains today.

“*Bitcoin’s objective is relatively simple: it is a* ***blockchain supporting the transfer of a single native digital asset****, which is not redeemable for anything else. This allowed many simplifications in the implementation, but real-world demands are now challenging those simplifications… There are many more trade-offs for blockchain features. For example, Bitcoin’s script could be more powerful to enable succinct and useful* ***contracts***”[BackSidachains].

And some blockchains increased their market share because they introduced smart contracts. And group signature that this paper proposes as mechanism for interoperability between blockchains likely can support transferring of other “smart assets”. Mechanism may be broad but the protocol should be precise. Therefore we restrict our focus on cryptocurrency only, at least for first versions of the protocol.

## Classification.

Proper classification using traditional financial terminology allows to predict further trends in cryptocurrency’s evolution based on the trajectory passed by traditional financial systems/notions.

* 1. **Medium of exchange**: some cryptocurrencies with low transaction fee are used as medium of exchange (for example: EOS, Tron, Matic…). But cryptocurrency hasn’t taken off the same way in every country, with its adoption and use being sporadic across the world[. These are the 10 countries](https://www.euronews.com/next/2022/08/16/ukraine-now-ranks-second-in-the-world-for-crypto-use-which-other-countries-have-embraced-i) with the highest number of businesses accepting payments in [crypto](https://www.euronews.com/next/2022/08/16/ukraine-now-ranks-second-in-the-world-for-crypto-use-which-other-countries-have-embraced-i): Colombia , Venezuela, South Korea, France, Greece, Thailand , Germany, Spain, Austria, The Netherlands.
  2. **A Unit of account**: cryptocurrency is measured in numbers, so it’s obvious a unit of account.
  3. **Divisible**: yes.
  4. **Durable**:

- The intensive circulation of the token can reduce its value in some blockchains significantly because of high transaction fee. So to be durable transaction fee should be low.

- Most of the tokens have a lifetime equal to lifetime of their network. The history of Bitcoin-like blockchains is short. One of the goals of this paper is to make lifetime of the token independent of lifetime of its origin network.

1.5 **Fungible**: the token in my pocket(wallet) is the same as token in other pocket. *Almost* the same: tokens in cold wallet are more secure then tokens in the exchange hot wallet. But traditional currencies have the same security issues: cash is more secure then amount on bank account from credit risk perspective and less secure in other aspects.

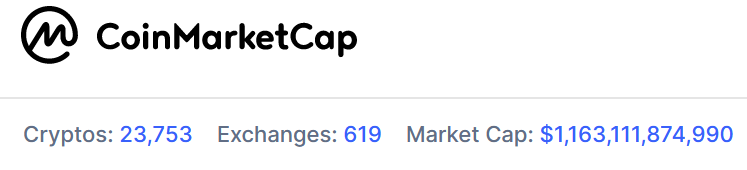
Parallels with [ERC-20](https://eips.ethereum.org/EIPS/eip-20):

“The ERC-20 introduces a standard for **Fungible** Tokens, in other words, they have a property that makes each Token be exactly the same (in type and value) as another Token. For example, an ERC-20 Token acts just like the ETH, meaning that 1 Token is and will always be equal to all the other Tokens[[1]](#footnote-1)”

1.6 **Portable**. This is the main characteristics discussed in this paper. It depends. It’s easy to transfer an amount of some cryptocurrency between persons that are located in different places all over the world. But within the same blockchain. If two counterparties use different cryptocurrencies/blockchains then sender should convert token issued in his blockchain into token circulated in receiver’s blockchain. Often using exchanges. And this often requires some efforts: sender may have to transfer amount from cold storage into hot wallet of exchange (paying transaction fee). Then sender pays commission for conversion one token to another. And he pays commission for withdrawal when sending from exchange wallet to wallet of receiver. The steps may be different but often transferring is not a trivial task and often it has significant overheads. Evolution of traditional currencies shows the tendency of simplification of transfers between different geographical regions. But legislations of different countries introduce berries. So difficulties in transferring are comparable between traditional and cryptocurrencies. Nowadays it’s harder for legislation of different countries to control free movement of cryptocurrency. This property is often related with so called censorship-resistance.

## Existing attempts to integrate blockchains using trusted party.

In 2023 we have following picture: hundreds of tokens are listed in exchanges (more than 23 thousands are registered in [CoinMarketCap](https://coinmarketcap.com/)).



Most often there is one to one relationship between token and blockchain/network. Usually P2P network supports one blockchain – distributed ledger/database. Most blockchains has their “native” currency/tokens: say, Ethereum network has its native token *Ether* for paying transaction fee - gas. Some blockchains support their native token plus wrapped tokens issued by other blockchains. For example [article from [CoinMarketCap](file:///D:\Books\Blockchain\Blockchains_Integration\WrappedTokens\What%20Is%20Wrapped%20Ethereum%20(WETH)_%20_%20CoinMarketCap%20(4_22_2023%208_17_14%20PM).html)]:

“[WETH](https://coinmarketcap.com/currencies/weth/) is the wrapped version of [Ether](https://coinmarketcap.com/currencies/ethereum/). Wrapped tokens, like WETH or [Wrapped Bitcoin](https://coinmarketcap.com/currencies/wrapped-bitcoin/), are tokenized versions of cryptocurrencies that are [pegged](https://coinmarketcap.com/alexandria/glossary/peg) to the value of the original coin and can be unwrapped at any point. Almost every major blockchain has a wrapped version of its native cryptocurrency like [Wrapped BNB](https://coinmarketcap.com/currencies/wbnb/), [Wrapped AVAX](https://coinmarketcap.com/currencies/wavax/), or [Wrapped Fantom](https://coinmarketcap.com/currencies/wrapped-fantom/). The mechanism of such coins is similar to that of [stablecoins](https://coinmarketcap.com/alexandria/glossary/stablecoin). **Stablecoins are essentially “wrapped USD” in the sense that dollar-pegged stablecoins can be redeemed for**[**FIAT**](https://coinmarketcap.com/alexandria/glossary/fiat)**dollars at any point.** In a similar fashion, WBTC, WETH, and all other wrapped coins can be redeemed for the original asset at any time.

Wrapped coins solve a particular problem: because of the low interoperability of blockchains, native coins of one chain cannot be used on another chain. For instance, you cannot use Bitcoin on the Ethereum blockchain and you cannot use Ether on Bitcoin or [Avalanche](https://coinmarketcap.com/alexandria/article/a-dive-into-avalanche). Wrapping coins solves this problem by tokenizing them and applying the blockchain’s token standard to the tokenized version of the original cryptocurrency.

On Ethereum, **almost all fungible tokens follow the**[**ERC-20**](https://coinmarketcap.com/alexandria/glossary/erc-20)**standard developed in 2015**. This token standard was developed to have a standardized set of rules for tokens on Ethereum, which simplified new token launches and made all tokens on the blockchain comparable to each other. Mandatory rules all ERC-20 tokens have to follow are totalSupply, balanceOf, transfer, transferFrom, approve, and allowance. Unfortunately, **Ether itself does not comply with the ERC-20 standard. Wrapped Ethereum was developed to increase**[**interoperability**](https://coinmarketcap.com/alexandria/glossary/interoperability)**between blockchains and make Ether usable in decentralized applications ([dApps](https://coinmarketcap.com/alexandria/glossary/decentralized-applications-dapps" \t "_blank)).”**

Thus, “low interoperability of blockchains” reduces portability of tokens.

Attempts to create digital cash were made 3 decades before appearance of Bitcoin[BackSidechains]:

“David Chaum introduced digital cash as a research topic in 1983, in a setting with a central server

that is trusted to prevent double-spending[Cha83]. To mitigate the privacy risk to individuals from

this central trusted party, and to enforce fungibility, Chaum introduced the blind signature, which he

used to provide a cryptographic means to prevent linking of the central server’s signatures (which

represent coins), while still allowing the central server to perform double-spend prevention. The

requirement for a central server became the Achilles’ heel of digital cash[Gri99]. While it is

possible to distribute this single point of failure by replacing the central server’s signature with

a threshold signature of several signers, it is important for auditability that the signers be distinct

and identifiable. This still leaves the system vulnerable to failure, since each signer can fail, or be

made to fail, one by one.

In January of 2009, Satoshi Nakamoto released the first widely used implementation of peer-to-

peer trustless electronic cash[Nak09], replacing the central server’s signature with a consensus

mechanism based on proof of work[Bac02], with economic incentives to act cooperatively.”

So, Bitcoin eliminated single point of failure replacing it with a decentralized consensus within P2P network. Decentralization made Bitcoin a robust system with absence of trust to centralized 3-rd party. And mechanism used for existence of such wrapped tokens as WETH is based on trust to 3-rd party [CMCWETH]:

“Wrapped tokens require [custodians](https://coinmarketcap.com/alexandria/glossary/custodian) to hold the collateral.For instance, if you want to wrap Ethereum, a custodian will hold your Ether and give you Wrapped Ethereum in return. Custodians can be **merchants,**[**multi-signature**](https://coinmarketcap.com/alexandria/glossary/multisignature)**wallets, or simply a**[**smart contract**](https://coinmarketcap.com/alexandria/glossary/smart-contract)**.”**

Approach with Custodians works but introduces single point of failure – further decentralization is desirable for interoperability of blockchains. The goal of this paper is to find out the way to increase interoperability of blockchains/portability of cryptocurrency using decentralized mechanisms.

Thus, we can see that cryptocurrencies mostly support characteristics of traditional fiat currencies. And trends of fiat currencies may influence trends of cryptocurrencies – cryptocurrencies may face the same requirements. Fiat currencies are issued by national Central Banks and interoperability between them (and also between commercial banks) is essential for international trade and financial system. Interoperability of blockchains is actual direction of cryptocurrency evolution. We can see that some steps were made to introduce interoperability (using custodians) but further elimination of central trust is required to comply with decentralized nature of cryptocurrency. According to weakest link principle of IT Security: Security is Only as Strong as the Weakest Link.

## ERC-20: Ethereum Token Standard

In 2015 Ethereum made an attempt to standardize transferring of tokens and introduced the interface (API) for token transfer with following functions:

function name() public view returns (string) // *Returns the name of the token - e.g. "MyToken"*

function symbol() public view returns (string) //*Returns the symbol of the token. E.g. “HIX”.*

function decimals() public view returns (uint8)// *number of decimals the token uses*

function totalSupply() public view returns (uint256)// *Returns the total token supply*

function balanceOf(address \_owner) public view returns (uint256 balance) //*Returns the account balance of another account with address \_owner*

function transfer(address \_to, uint256 \_value) public returns (bool success) //*Transfers \_value amount of tokens to address \_to*

function transferFrom(address \_from, address \_to, uint256 \_value) public returns (bool success) //*Transfers \_value amount of tokens from address \_from to address \_to*

function approve(address \_spender, uint256 \_value) public returns (bool success)// *Allows \_spender to withdraw from your account multiple times, up to the \_value amount.*

function allowance(address \_owner, address \_spender) public view returns (uint256 remaining)// *Returns the amount which \_spender is still allowed to withdraw from \_owner*

Ethereum standard was created with following **Motivation**:

“*A standard interface allows any tokens on Ethereum to be re-used by other applications: from wallets to decentralized exchanges.*”

“…*Although Ethereum allows developers to create absolutely any kind of application without restriction to specific feature types, and prides itself on its "lack of features",* ***there is nevertheless a need to standardize certain very common use cases in order to allow users and applications to more easily interact with each other. This includes sending currency units, registering names, making offers on exchanges, and other similar functions***…”[[2]](#footnote-2)

Often new standards are based on existing de facto standards in an industry.

Interface for interoperability of blockchains may be influenced by ERC-20. It can use **name**, **symbol**, **decimals**, **balanceOf**, **transfer**, **transferFrom** functions with additional specification of ID of source or destination blockchain (where appropriate).

# Comparison with evolution of other systems

Many blockchains appeared as fork of Bitcoin codebase. Some blockchains added new outstanding properties to Bitcoin. For example, Monero added anonymity. Ethereum is not just platform for particular token – Ethereum pretends to be a distributed world computer that runs turing complete code. Some projects appeared because it’s difficult to introduce changes to Bitcoin. And they try to improve Bitcoin. But most of projects didn’t introduce something new.

## 2.1. Banking sector

Creating a new cryptocurrency project sometimes is similar to currency emission. Fiat currency is not created only by emission of Central Bank – 10% reservation gives opportunity for bank to issue new amount of currency giving a credit. In exUSSR countries in 90-s hundreds of banks were founded. It was profitable to create banks. But financial system doesn’t require such amount of similar banks. Most of them were merged or ended its life. The most robust (about 10-20) survived. Some cryptocurrency analytics predict the reduction of blockchain projects (that have no outstanding features) by similar reasons.

## 2.2. There can be only one?

Can some particular blockchain merge other blockchains covering all the market? It seems no. Because unlikely one particular blockchain is capable to support all required characteristics. “Ratio of Ether’s market cap doubled Bitcoin in the last [year](https://cointelegraph.com/news/ratio-of-ether-s-market-cap-doubled-bitcoin-in-the-last-year-pantera-capital)”. But Bitcoin’s difficulty of introducing changes is itself useful property: external parties have a lot less control making it censorship resistant. Ethereum is more centralized is the sense that its founders can introduce changes on their discretion easily. Ethereum positions itself as “moving fast” but this reduces its censorship resistance. Cryptocurrencies like Monero are not listed in some exchanges because their anonymity doesn’t comply with legislation restrictions of some countries. And therefore some projects support anonymity but allow to reveal private information. So, there is no universal thing and diversity of projects will exist. Reduction of similar projects are expected.

We can observe the same trends for example in market share of **operating systems**: Android dominates in smartphones market; Windows is often used by Desktops; BSD, Linux runs on servers. Even within Linux distributions there is spectrum of systems where each proposes its specifics. There is no universal thing – in different areas we can observe diversity of solutions. Each solution proposes better characteristics for particular needs and has its drawbacks in certain applications.

## 2.3. TCP/IP stack. [Protocol Wars](https://en.wikipedia.org/wiki/Protocol_Wars).

On the other hand communication stack TCP/IP is predominantly used all over the world.

“A long-running debate in [computer science](https://en.wikipedia.org/wiki/Computer_science) known as the **Protocol Wars** occurred from the 1970s to the 1990s when engineers, organizations and nations became polarized over the issue of which [communication protocol](https://en.wikipedia.org/wiki/Communication_protocol) would result in the best and most robust [computer networks](https://en.wikipedia.org/wiki/Computer_network). This culminated in the **Internet–OSI Standards War** in the late 1980s and early 1990s, which was ultimately "won" by the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite) ("TCP/IP") by the mid-1990s and has since resulted in most other protocols disappearing.

The pioneers of [packet switching](https://en.wikipedia.org/wiki/Packet_switching) technology built computer networks to research [data communications](https://en.wikipedia.org/wiki/Data_transmission) in the early 1970s. As [public data networks](https://en.wikipedia.org/wiki/Public_data_network) emerged in the mid to late 1970s, the debate about interface [standards](https://en.wikipedia.org/wiki/Standardization) was described as a "battle for access standards". An international collaboration between several national [postal, telegraph and telephone](https://en.wikipedia.org/wiki/Postal,_telegraph_and_telephone_service) ("PTT") providers and commercial operators developed the [X.25](https://en.wikipedia.org/wiki/X.25) standard in 1976, which was adopted on public networks providing global coverage. Several proprietary standards also emerged, most notably IBM's [Systems Network Architecture](https://en.wikipedia.org/wiki/Systems_Network_Architecture).

The [United States Department of Defense](https://en.wikipedia.org/wiki/United_States_Department_of_Defense) developed and tested TCP/IP during the 1970s in collaboration with universities and researchers in the United States, United Kingdom and France. [IPv4](https://en.wikipedia.org/wiki/IPv4) was released in 1981 and the DoD made it standard for all military computer networking. By 1984, an international reference model known as the [OSI model](https://en.wikipedia.org/wiki/OSI_model) had been agreed on, with which TCP/IP was not compatible. Many governments in Europe – particularly France, West Germany, the United Kingdom and the European Economic Community – and also the [United States Department of Commerce](https://en.wikipedia.org/wiki/United_States_Department_of_Commerce) mandated compliance with the OSI model and the US Department of Defense planned to transition away from TCP/IP to OSI

Meanwhile, the development of a complete Internet protocol suite by 1989, and [partnerships with the telecommunication and computer industry](https://en.wikipedia.org/wiki/Internet_protocol_suite#Adoption) to **incorporate TCP/IP software into various** **operating systems** laid the foundation for the **widespread adoption** **of TCP/IP** as a comprehensive protocol suite. While OSI developed its networking standards in the late 1980s, **TCP/IP came into widespread** **use** on multi-vendor networks for [internetworking](https://en.wikipedia.org/wiki/Internetworking) and **as the core component of the emerging** [**Internet**](https://en.wikipedia.org/wiki/Internet)”

## 2.4. Summary of comparison.

Why after long evolution operating systems segment preserves the spectrum of different solutions and TCP/IP "won" the Protocol Wars? Because different operation systems can operate on different machines in parallel. Servers run Linux or Windows operating systems, embedded devices run their specialized OS, smartphones their OS, very restricted in recourses devices such as smartcards powered by JavaCard platforms or MultOS. From the other hand each computer needs to communicate with another computer and this is not possible to achieve if they would talk on different languages – using different protocols. Internet consists of subnetworks – LAN. LAN can operate using its particular channel protocol (like Ethernet or TokenRings in the past) but on network layer subnets communicates using common language - TCP/IP stack – for routing and data transfer.

We can consider blockchain networks as equivalent of subnetworks (LANs) and interoperability of blockchain networks as equivalent of internetwork communication protocol - network (and above) layer of TCP/IP or ISO Open System Interconnection Model (OSI Model).

We see or expect some trends in blockchain technology. Thousands of blockchain projects often do not add distinct features and we can expect reduction in the set of existing projects. Likely spectrum of projects will consist from specialized blockchains that achieves certain goals better than other projects. Some properties contradicts each other. For example, high level of control under project by developers accelerates improvements but may introduce centralization and reduce censorship resistance property. Intentions to increase market share requires compliance with legislation and legislation often restricts anonymity. So, strictly anonymous blockchains cannot pretend on widespread adoption (or they need to propose the ways to reveal private information).

Because different blockchains will still exist in the future there is emerging necessity for interoperability. And such attempts have been taken already using trusted custodians (see WETH example above). The demand for interoperability of blockchains defiantly exists.

Considering blockchain networks as subnetworks (managed by their own rules) we can use existent experience from computer networks and introduce common protocol for interoperability between blockchains.

Interconnection between systems is modeled by The Open Systems Interconnection model (OSI model). This successful model is widely recognized and exists for a long time (from the late 1970s) and orientation on such model will give the correct direction for the development of protocol.

Several networking models have sought to create an intellectual framework for clarifying networking concepts and activities,but none have been as successful as the OSI reference model in becoming the standard model for discussing, teaching, and learning for the networking procedures in the field of [information technology](https://en.wikipedia.org/wiki/Information_technology).

# Using Open Systems Interconnection model (OSI model) for blockchain interoperability.

## 3.1 Open Systems Interconnection model (OSI model).

The challenges we face now is similar to those IT industry had in 1970-s: the necessity to interconnect subnetworks from different vendors into Internet. Nowadays we need to interconnect blockchains developed by different vendors into integrated system with possibility to transfer messages/transactions between blockchains/subnetworks. We need Interblockchain.

“*During the last ten years, many computer networks have been designed, implemented, and put into service in the United States, Canada, Europe, Japan, and elsewhere. From the experience obtained with these networks, certain key design principles have begun to emerge, principles that can be used to design new computer networks in a more structured way than has traditionally been the case*.

*Chief among these principles is the notion of structuring a network as a hierarchy of layers, each one built upon the previous one. This paper is a tutorial about such network hierarchies, using the Reference Model of Open Systems Interconnection developed by the ISO as a guide…* *networks are almost always organized as a hierarchy of layers.”* [TanenbaumProto]

“To help deal with this complexity, network designers have developed

general blueprints—usually called network architectures—that guide the

design and implementation of networks…

…Abstractions naturally lead to layering, especially in network systems.

The general idea is that you start with the services offered by the underlying

hardware and then add a sequence of layers, each providing a higher

(more abstract) level of service. The services provided at the high layers

are implemented in terms of the services provided by the low layers.”

The **Open Systems Interconnection model** (**OSI model**) is a [conceptual model](https://en.wikipedia.org/wiki/Conceptual_model) that 'provides a common basis for the coordination of standards development for the purpose of systems interconnection' [ISOOSI]. In the OSI reference model, the communications between a computing system are split into seven different abstraction layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

The model partitions the flow of data in a communication system into seven [abstraction layers](https://en.wikipedia.org/wiki/Abstraction_layer) to describe networked communication from the physical implementation of transmitting [bits](https://en.wikipedia.org/wiki/Bit) across a [communications medium](https://en.wikipedia.org/wiki/Transmission_medium) to the highest-level representation of data of a [distributed application](https://en.wikipedia.org/wiki/Distributed_application). Each intermediate layer serves a class of functionality to the layer above it and is served by the layer below it. Classes of functionality are realized in all software development through all and any standardized [communication protocols](https://en.wikipedia.org/wiki/Communication_protocol).

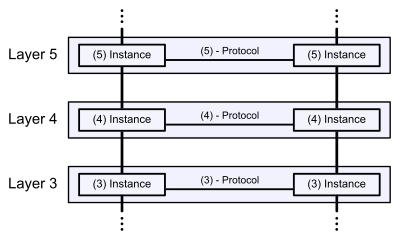
Each layer in the OSI model has its own well-defined functions, and the functions of each layer communicate and interact with the layers immediately above and below it, unless the layer does not have layers below or above.

The [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite) has a separate model, the layers of which are mentioned in [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1122](https://datatracker.ietf.org/doc/html/rfc1122) and [RFC](https://en.wikipedia.org/wiki/RFC_(identifier)) [1123](https://datatracker.ietf.org/doc/html/rfc1123). That model combines the physical and data link layers of the OSI model into a single link layer, and has a single application layer for all protocols above the transport layer, as opposed to the separate application, presentation and session layers of the OSI model. Additionally, the model allows transparent communication through equivalent exchange of [protocol data units](https://en.wikipedia.org/wiki/Protocol_data_unit) (PDUs) between two parties, through what is known as [peer-to-peer networking](https://en.wikipedia.org/wiki/Peer-to-peer_networking)[ISOOSIW].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OSI model** | | | | |
|
|  | **OSI Function** | | | **Blockchain function** |
| **Host** | 7 | [Application](https://en.wikipedia.org/wiki/Application_layer) | [High-level protocols such as for resource sharing or remote file access, e.g. HTTP.](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) | Validating transactions and inserting in block |
| **layers** | 6 | [Presentation](https://en.wikipedia.org/wiki/Presentation_layer) | Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption |  |
|  | 5 | [Session](https://en.wikipedia.org/wiki/Session_layer) | [Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes](https://en.wikipedia.org/wiki/Session_(computer_science)) | 5.1 Continuous update of blockchain headers  5.2 Tracking for transactions (in Buffer) with completed contest period |
|  | 4.2 | Transport of overlay |  | 4.2.1. Bootstrapping (initial addresses of nodes of parent blockchain).  4.2.2. Network churn: adding, removing nodes (structured networks)   4.2.3. DHT Routing (structured networks) |
|  | 4.1 | [Transport of underlay](https://en.wikipedia.org/wiki/Transport_layer) | **Reliable** transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing. facilitates error recovery. Reliability | Specific for particular blockchain/network |
| **Media** | 3 | [Network](https://en.wikipedia.org/wiki/Network_layer) | Structuring and managing a multi-node network, including addressing, routing and traffic control | Specific for particular blockchain/network |
| **layers** | 2 | [Data link](https://en.wikipedia.org/wiki/Data_link_layer) | Transmission of data frames between two nodes connected by a physical layer | Specific for particular blockchain/network |
|  | 1 | [Physical](https://en.wikipedia.org/wiki/Physical_layer) | Transmission and reception of raw bit streams over a physical medium | Specific for particular blockchain/network |

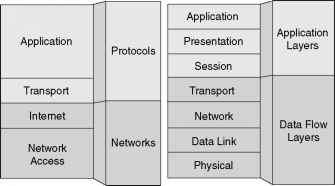
Here are some examples[[3]](#footnote-3) of well-known application layer protocols in use today:

* [**Bitcoin**](https://www.techtarget.com/whatis/definition/Bitcoin) for digital currency;
* [**Hypertext Transfer Protocol**](https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol) (**HTTP**) for message communications;
* **H.323** for packet-based communications, such as [voice over IP](https://www.techtarget.com/searchunifiedcommunications/definition/VoIP);
* [**Lightweight Directory Access Protocol**](https://www.techtarget.com/searchmobilecomputing/definition/LDAP) for queries of user information;
* [**Network File System**](https://www.techtarget.com/searchenterprisedesktop/definition/Network-File-System) for data storage and retrieval from various media;



4 Layers of TCP/IP stack also do not correspond to 7 Layers of OSI Model – these layers are too granular for some systems. The designers of TCP/IP felt that the application layer should include the OSI session and presentation layer details. They created an application layer that handles issues of representation, encoding, and dialog control.

But this document is focused on the simple blockchain that determines the history of asset control and provides a computationally unforgeable time ordering for transactions. Further evolution of ideas described in this work may extend focus towards decentralized applications (smart contracts). Then additional layers may be useful.



## 3.2 P2P networks, Overlay networks.

Here are two definitions of P2P Systems that cover the concepts of resource sharing, self-organization, decentralization, and interconnection:

“A distributed network architecture may be called a peer-to-peer network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration).

They are accessible by other peers.”

[R. Schollmeier. *A Definition of Peer-to-Peer Networking for the Classification of Peer-to-*

*Peer Architectures and Applications*. Peer-to-Peer Computing 2001]

“Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance,

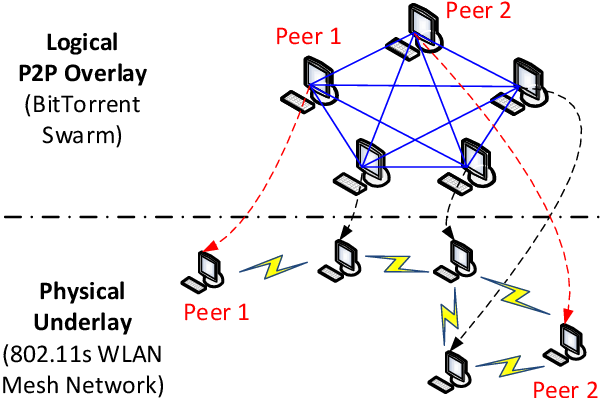
without requiring the intermediation or support of a global centralized server or authority.”[S. Androutsellis-Theotokis and D. Spinellis. *A Survey of Content Distribution Technologies.ACM Computing Surveys*, 36(4), December 2004]

"The following **properties** are characteristics found in most **P2P systems**.

* ***Resource sharing***each peer contributes system resources to the operation of the P2P system. Ideally this resource sharing is proportional to the peer’s use of the P2P system, but many systems suffer from the free rider problem;
* ***Networked***all nodes are interconnected with other nodes in the P2P system, and the full set of nodes are members of a connected graph. When the graph is no longer connected, the overlay is said to be partitioned.
* ***Decentralization***the behavior of the P2P system is determined by the collective actions of peer nodes, and there is no central control point. Some systems however secure the P2P system using a central login server. The ability to manage the overlay [24] and monetize its operation may require centralized elements.
* ***Symmetry***nodes assume equal roles in the operation of the P2P system. In many designs this property is relaxed by the use of special peer roles such as super peers or relay peers.
* ***Autonomy***participation of the peer in the P2P system is determined locally, and there is no single administrative context for the P2P system.
* ***Self-organization***the organization of the P2P system increases over time using local knowledge and local operations at each peer, and no peer dominates the system. Biskupski, Dowling, and Sacha argue that existing P2P systems do not exhibit most of the properties of self-organization....
* Self-organization is a process where the organization of a system spontaneously increases without being managed by an outside source....
* ***Scalable***This is a pre-requisite of operating P2P systems with millions of simultaneous nodes, and means that the resources used at each peer exhibit a growth rate as a function of overlay size that is less than linear. It also

means that the response time doesn’t grow more than linearly as a function of overlay size. *Stability* Within a maximum churn rate, the P2P system should be stable, i.e., it should maintain its connected graph and be able to route deterministically within a practical hop-count bounds."

**A P2P network is a logical overlay network over a physical infrastructure** - is a network that is built on top of an existing network. as illustrated following Figure, which provides a virtual environment for P2P developers to easily design and implement their own communication environment and protocols on the top of existing networks.



The overlay therefore relies on the so called underlay network for basic networking functions, namely routing and forwarding.

Some networks like **Bitcoin** have no routing: “The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They **do not need to be identified**, since **messages are** **not routed** to any particular place and only need to be delivered **on a best effort basis**. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone.

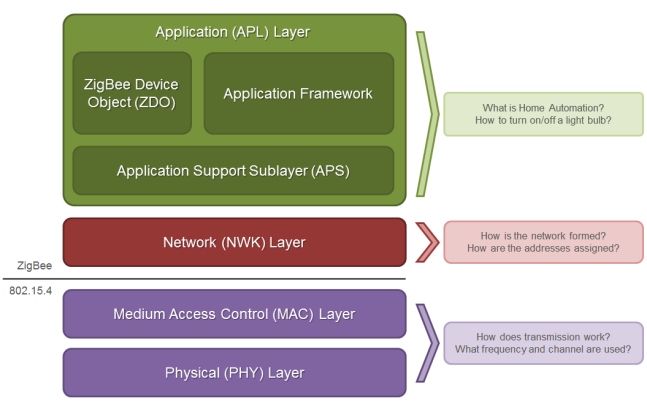
Other blockchains are based on Kademlia structured network and require overlay layer routing.

## 3.3 Example of OSI model application: Zigbee stack layers[[4]](#footnote-4)

Most network protocols use the concept of layers to separate different components and functions into independent modules that can be assembled in different ways.

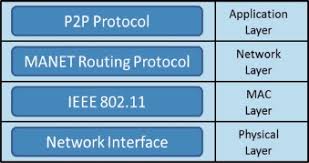
For example, Zigbee is built on the Physical (PHY) layer and Medium Access Control (MAC) sub-layer defined in the IEEE 802.15.4 standard. These layers handle low-level network operations such as addressing and message transmission/reception.

The Zigbee specification defines the Network (NWK) layer and the framework for the application layer. The Network layer takes care of the network structure, routing, and security. The application layer framework consists of the Application Support sub-layer (APS), the Zigbee Device Objects (ZDO) and user-defined applications that give the device its specific functionality.

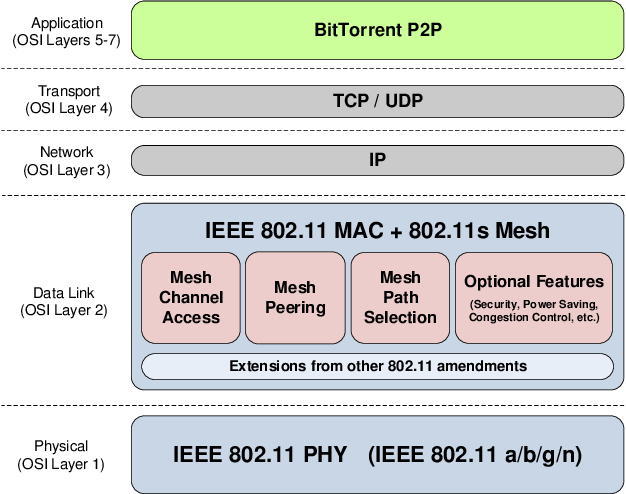


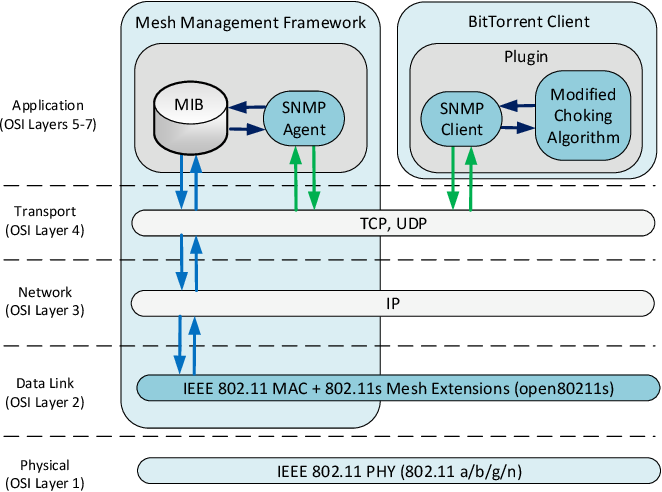
## 3.4 Example of OSI model application: Structured Peer-to-Peer overlay deployment on MANET[[5]](#footnote-5)

There are many common characteristics between Peer-to-Peer (P2P) [overlay networks](https://www.sciencedirect.com/topics/engineering/overlay-network) and Mobile Ad-hoc Networks (MANET). Self-organization, decentralization, dynamicity and changing topology are the most shared features. Furthermore, when used together, the two approaches complement each other. P2P overlays provide data storage/retrieval functionality, and their routing information can complement that of MANET. MANET provides wireless connectivity between clients without depending on any pre-existing infrastructure.



Often reduced number of layers is used as a single layer (as in TCP/IP) and P2P functionality occupies OSI **Layers 5-7**:





Today, most overlay networks are built in the application layer on top of the TCP/IP networking suite. Overlay technologies can be used to overcome some of the limitations of the underlay, and at the same time offering new routing and forwarding features without changing the routers. The nodes in an overlay network are connected via logical links that can span many physical links. A link between two overlay nodes may take several hops in the underlying network. [Tarkoma2013]

## 3.5 Blockchain as peer-to-peer network. Layering.

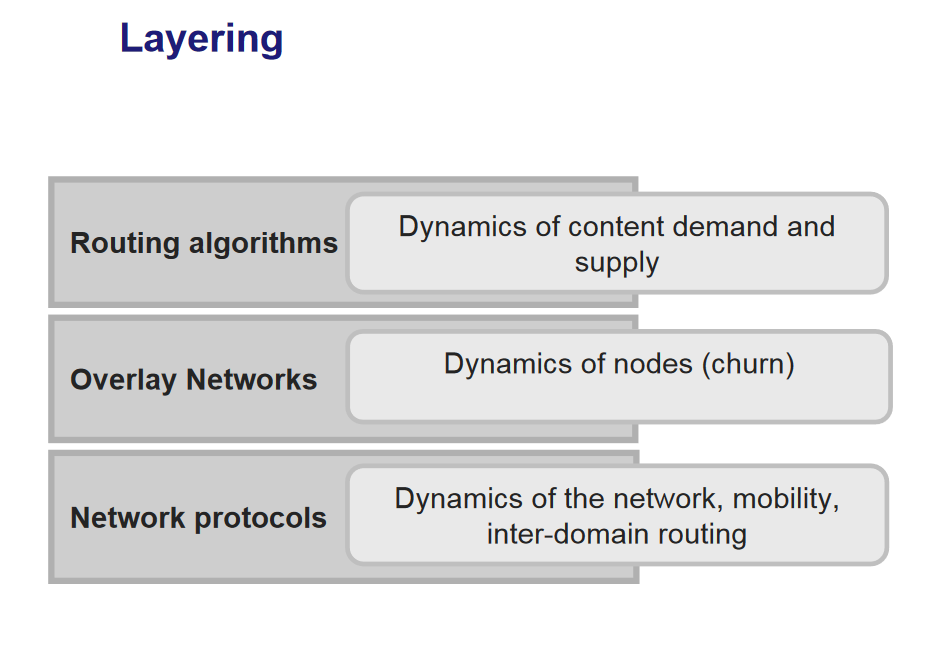
Blockchain networks are peer-to-peer networks because this allows to eliminate trusted party – consensus works by “voting” by CPU power of Miners[[6]](#footnote-6). And therefore the blockchain is designed to be decentralized.

No central control exists in a decentralized network, and each peer (or node)

is connected to one or more peers (or nodes) in the network. The peers cooperate to

send and forward messages in the network.

“*The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone.*” [Nak09]



# 4. One way pegged sidechains.

## 4.1 Rationale.

This document relies heavily on the whitepaper “*Enabling Blockchain Innovations with Pegged Sidechains,* 2014” [BackSidechains]. Some sections are the comments to this whitepaper. The two main ideas is shared between two documents:

“The core observation is that “Bitcoin” the blockchain is conceptually independent from “bitcoin” the asset: if we had technology to support the movement of assets between blockchains, new systems could be developed which users could adopt by simply reusing the existing bitcoin currency…

…We use bitcoin as an example because its strong network effects make it likely that users will prefer it over other, newer assets. However, any altcoin can be adapted to be usable with sidechains.” [BackSidechains].

That is cryptocurrency token is considered as abstract intellectual property. P2P network/overlay, blockchain/ledger/database – all these notions are only “physical” carriers. Like hardware and software, physical medium (wires or disks) used for storage and transfer of data.

The whitepaper “Enabling Blockchain Innovations with Pegged Sidechains, 2014” proposes to consider Dynamic membership multi-party signature (or DMMS) as a new type of group signature:

“A DMMS is a digital signature formed by a set of

signers which has no fixed size. Bitcoin’s blockheaders are DMMSes because their proof-of-work

has the property that anyone can contribute with no enrolment process. Further, contribution is

weighted by computational power rather than one threshold signature contribution per party, which

allows anonymous membership without risk of a Sybil attack (when one party joins many times and

has disproportionate input into the signature). For this reason, the DMMS has also been described

as a solution to the Byzantine Generals Problem[AJK05]…

Nakamoto’s key innovation is the aforementioned use of a DMMS as a signature of computational power rather than a signature of knowledge.”

DMMS (as **group signature**) is convenient interpretation of the Bitcoin consensus mechanism.

In turn, S. Nakamoto, in paper “*Bitcoin: A peer-to-peer electronic cash system*”, 2009 [Nak09] refers to Adam Back’s document "Hashcash - a denial of service counter-measure", 2002:

“*To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proofof-*

*work system similar to Adam Back's Hashcash*”.

Consensus mechanism of Bitcoin (or creation of group signature in other words) is following: when miner is creating the candidate block he should select parent block for adding its hash to candidate block header. By selecting the specific parent block miner is committing his CPU power to extending the chain that ends in that specific block. In virtue, this is how miners “vote” with their CPU power for the longest chain.

“*They vote with their CPU power, expressing their acceptance of*

*valid blocks by working on extending them and rejecting invalid blocks by refusing to work on*

*them.”* [Nak09]

It’s possible to use the same approach as Bitcoin protocol uses as its consensus mechanism for transferring crypto assets from parent chain to sidechain.

In short, Wallet from parent chain destroys amount in parent blockchain. After that Validators (or miners) from sidechain verify that amount was destroyed and insert new transaction in sidechain. In other words Validators should create the aforementioned group signature.

But approach proposed by this document is different in following: whitepaper “*Enabling Blockchain Innovations with Pegged Sidechains,* 2014” advocates usage of “**Two** way pegged sidechain”. And this document proposes to use “**One** way pegged sidechain” (in terms of [BackSidechains]).

Solution of this document is to “transfer” coins by destroying tokens in a publicly recognizable way, which would be detected by a new blockchain to allow creation of new coins.

This approach is barely mentioned in [BackSidechains] using term “One way pegged sidechain” – it doesn’t discuss this approach.

“Two way pegged sidechain” assumes that transferred tokens are **parked** in the parent blockchain (and later it’s possible to unfreeze parked amount returning the amount from some sidechain to the parent blockchain back). This means that lifetime of the physical carrier – parent blockchain is the same as lifetime of abstract asset (intellectual property). But this assumption is too restrictive. In reality lifetime of abstract asset may be longer. Physical network (its protocol) may be replaced by more advanced version (especially because it’s difficult to change the protocol of some blockchains; Bitcoin is an example):

“*Bitcoin’s rules must be determined at the start of its*

*history, and new valid transaction forms cannot be added except with the agreement of every*

*network participant. Even with such an agreement, changes are difficult to deploy because they*

*require all participants to implement and execute the new rules in exactly the same way, including*

*edge cases and unexpected interactions with other features*” [BackSidechains].

## 4.2 Transferring ownership.

Transferring an amount from parent blockchain to the sidechain is basically transferring ownership of an abstract token: tokens are destroyed in the parent chain in a publicly recognizable way, which would be detected by a sidechain to allow creation of new representation of abstract tokens. Only single blockchain may own the token in certain point in time. If some project limits the maximum number of tokens then there should be no way to increase such a number (say, no more than 21 million).

One way pegged sidechain allows to transfer from parent chain to blockchain A, then from A to B, from B to C,…, and so forth.

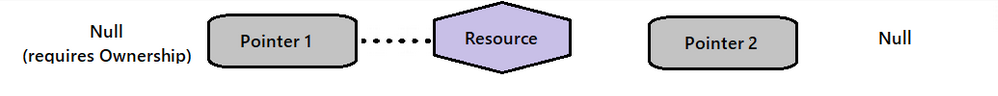
Parent blockchain can obtain the tokens back or may not – initial parent blockchain is not special (as for “Two way pegged blockchains”). Initial blockchain can obtain tokens back (if it supports corresponding rules).

For C++ programmers it looks like behavior of smart pointer std::auto\_ptr or std::unique\_ptr with move semantics for single ownership: there will be only one owner simultaneously.

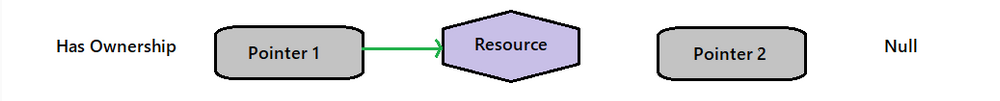
That is ownership of abstract token should belong to one blockchain only in each certain time.

There is no need to understand C++ - We just want to use the picture that is obvious. Imagine that blockchain has a pointer to Resource (token), consider “pointer” just as pointer in the street that points to some location. In this case it points to some Resource/token. Those who points to Resource - owns it:

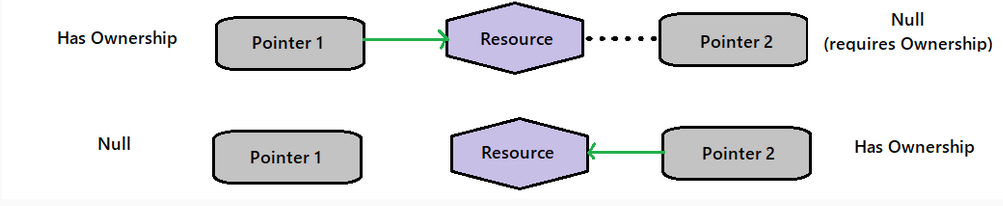
1. Initially token (Resource) doesn’t exist. Blockchain 1(left side) requires ownership of new token during mining process:



2. Than it has been mined in blockchain 1 and it owns token:



3. Than Resource/token is transferred from parent blockchain 1 to the sidechain 2(right side) together with the ownership to token:



|  |  |
| --- | --- |
| One Way Pegged Sidechain mechanism considers only two objects: parent blockchain (source) and sidechain (destination). It doesn’t require to save history – users of blockchain B3 should not track blockchain B1. When transferring from parent blockchain B2 to sidechain B3 validators track only blockchain B2 (no information about blockchain B1). When transfer from B3 to B4 is happening than validators of B4 do not track B2. |  |
| Abstract asset may return to the original blockchain B1 where it was issued. Blockchain B1 treats the asset (originally issued in B1) as any asset issued in other blockchain. Assets are not “parked” as in case of Two Way Pegged Sidechains.  Such circulation is safe only if mutually trusted blockchains participate in transfers – they all need to belong to the **association** of mutual trust. |  |

## 4. 3 Simplified SPV proofs (PoW)

Simplified SPV proofs reduces overheads but they rely on PoW.

If PoC or PoS project do not support similar simplified verification Validators can get and verify the whole database. And verify validity of transaction against the full blockchain using the same kind of group signature.



Because of confirmation period and contest period (defined below) we introduce intermediate blockchain - Buffer blockchain.

Sidechain (destination of transfer) may be fast – it may have restriction – short period between blocks. Parent blockchain may be reorganized. To avoid influence of reorganization we introduce Reorganization period. Also parent blockchain should not slow down processing of sidechain – sidechain validators should ask nodes of parent blockchain for proof. But nodes of parent blockchain may be not available or their responses may be slow. Often for interprocess communication queue is used: sender puts message in intermediate queue and continues its flow. Receiver reads this message as soon as it has possibility. The same idea brings us to Buffer Blockchain (part of sidechain). Sidechain may be destination for several parent blockchains. Some of them may be not sufficiently responsive.

Validators of sidechain are responsible for creating blocks in Buffer Blockchain. They receive messages of standard protocol from Wallets of parent blockchain. Then validators of sidechain create blocks in Buffer Blockchain. After contest period they migrate transactions from Buffer Blockchain to the Main Blockchain.

Buffer Blockchain is the implementation detail and not the part of the interoperability protocol. Some fast sidechains may go with such implementation and others may not. The interoperability protocol is implemented in code that is running on Validator’s nodes. Validators of parent blockchain or Wallets communicate with Validators of sidechain. Wallets send messages that initiate transfer. Full Nodes of Parent Blockchain send responses with block headers.

**Two** Way Sidechains define **confirmation** and **contest** **period** as below (for convenience definitions from [BackSidechains] are preserved as is with differences in red):

“*To synchronise the two chains, we need to define two waiting periods:*

*1. The* ***confirmation period*** *of a transfer between sidechains is a duration for which a coin must*

*be locked on the parent chain before it can be transferred to the sidechain. The purpose of this*

*confirmation period is to allow for sufficient work to be created such that a denial of service*

*attack in the next waiting period becomes more difficult. A typical confirmation period would*

*be on the order of a day or two.*

*After creating the special output on the parent chain, the user waits out the confirmation*

*period, then creates ~~a transaction on the sidechain~~ (1) referencing this output, providing an SPV*

*proof that it was created and buried under sufficient work on the the parent chain.*

*The confirmation period is a per-sidechain security parameter, which trades cross-chain*

*transfer speed for security.*

*2. The ~~user~~ (2)must then wait for the* ***contest period****. This is a duration in which a newly-transferred*

*coin may not be spent on the sidechain. The purpose of a contest period is to prevent double*

*spending by transferring ~~previously-locked~~ (3)coins during a reorganisation. If at any point*

*during this delay, a new proof is published containing a chain with more aggregate work*

*which does not include the block in which the lock output*” [BackSidechains]

Following are corrections for **One** Way Sidachains:

(1) User (Wallet) sends request to Validators of sidechain – wallet doesn’t create transactions in Buffer Blockchain directly. Code of Full Nodes is the Data Abstraction Layer where data is blockchain.

(2) Validators of sidechain.

(3) destroyed coins.

Let’s review how blockchain works in details using text from paper of S.Nakamoto [Nak09]. We can define steps based on slightly modified text from section 5:

After **confirmation** period:

{

1) New transactions are broadcast by **Wallet** to all nodes of **Sidechain**.

2) Each node in **Sidechain** collects new transactions into a block in **Buffer** blockchain.

3) Each node in **Sidechain** works on finding a difficult proof-of-work for its block.

4) When a node finds a proof-of-work, it broadcasts the block to all nodes in **Sidechain**.

5) Nodes accept the block only if all transactions in it are valid and amount is **Destroyed** in **Parent Network**.

6) Nodes express their acceptance of the block by working on creating the next block in the

chain, using the hash of the accepted block as the previous hash.

}

After **contest** period Validators of Sidechain migrate transactions from Buffer to Main blockchain:

{

1) New transactions in **Buffer** are tracked by **Validator** of **Sidechain**.

2) Each node in **Sidechain** collects new transactions into a block for **Main** blockchain.

3) Each node in **Sidechain** works on finding a difficult proof-of-work for its block.

4) When a node finds a proof-of-work, it broadcasts the block to all nodes in **Sidechain**.

5) Nodes accept the block only if all transactions in it are valid (not migrated from Buffer yet).

6) Nodes express their acceptance of the block by working on creating the next block in the

chain, using the hash of the accepted block as the previous hash.

}

## 4.4 Association of trust between blockchain

Sidechain D should trust sidechain S2 to allow S2 to transfer tokens into it because malicious validators of S2 may improperly validate transferring from Blockchain S1 (say Initial blockchain) creating tokens from air and then trying laundering those counterfeited assets.

Therefore Blockchains should found an association of mutual trust where each blockchain passes enrollment to this association. And only approved blockchain may be parent blockchain.

## 4.5 Isolation between blockchains.

“a bug in one sidechain enabling creation (or theft) of assets

in that chain should not result in creation or theft of assets on any other chain.” [BackSidechains]

## 4.6 Unique direction specified in transaction during transfer to sidechain.

To avoid double spending parent blockchain should write ID of the sidechain during destruction of asset. Sidechain should accept only transfers addressed to this particular sidechain.

Maintaining the list of well-known IDs is not usually an issue: we have ID of networks now used by wallets, We have ticket of share/company in exchange and these Ids are well-known.

## 4.7 Incentive

Incentive for Validators in Sidechain may consist of payment for block creation and transaction fee.

## 4.8 Destroying bitcoin in a publicly recognizable way.

Sending bitcoins to a bogus address (an address with no known private key) is an easy way to burn bitcoins.

“Known examples of bogus addresses are:

* [1BitcoinEaterAddressDontSendf59kuE](http://blockchain.info/address/1BitcoinEaterAddressDontSendf59kuE) (2.10556692 BTC received),
* [1CounterpartyXXXXXXXXXXXXXXXUWLpVr](https://blockchain.info/address/1CounterpartyXXXXXXXXXXXXXXXUWLpVr) (2,130.84717717 BTC),
* [1111111111111111111114oLvT2](https://blockchain.info/address/1111111111111111111114oLvT2) (43.2884582 BTC), this address encodes the smallest possible hash160
* [1QLbz7JHiBTspS962RLKV8GndWFwi5j6Qr](https://blockchain.info/address/1QLbz7JHiBTspS962RLKV8GndWFwi5j6Qr) (0.01159201 BTC), and this one is the biggest possible hash160.

While sending bitcoins to such an address is a pretty sure way to burn them, it is not a provably definitive one as a private key for this address could exist (however [finding out if it exists would require way more energy than our solar system could produce during its lifetime](https://i.imgur.com/fYFBsqp.jpg)).”[[7]](#footnote-7)

“*The most secure way to destroy bitcoins is to send them to a script that can never return true, and which includes the reference code that confirms the goal of destroying these coins. I don't know of readily available software tools that can do that, and I'm not sure the network will accept these transactions*.”[[8]](#footnote-8)

## 4.9 Direction of destroying transaction

|  |  |
| --- | --- |
|  | To avoid double spending that may occur when Wallet from blockchain B1 tries to double spend amount transferring it to two different sidechains {B2 and B4} destroying transaction should specify single destination sidechain  For example destination specified in transaction is blockchain B2. Validators of B2 should process such transaction if other parameters are valid. Validators of B4 or other blockchains reject transactions with direction addressed not to them. |

## 4.10 SNARK as proof of destroying transaction.

[BackSidechains] proposes usage of SPV proofs to prove that the transferred amount was destroyed in the parent blockchain. But SPV proof is specific to the Proof of Work algorithm and may be not appropriate for blockchains that use other consensus mechanisms. To enlarge the scope of blockchains that can transfer the assets here we propose using more modern and universal way - using so called zkBridge. The sidechain should be convinced that the parent chain destroyed the asset. There is no need in zero knowledge here – we need only succinct proof of destroying transaction. Therefore we use term SNARK (not zkSNARK). We use SNARK to prove to the sidechain that the consensus protocol on the parent chain agrees to the fact that the asset has been destroyed.

Sequence diagram with SNARK introduced is following:

# References

|  |  |
| --- | --- |
| [BackSidechains] | Adam Back, Matt Corallo, Luke Dashjr,  Mark Friedenbach, Gregory Maxwell,  Andrew Miller, Andrew Poelstra,  Jorge Timón, and Pieter Wuille,  *Enabling Blockchain Innovations with Pegged Sidechains*, 2014,  [*https://blockstream.com/sidechains.pdf*](https://blockstream.com/sidechains.pdf) |
| [Bac02] | A. Back, *Hashcash — a denial of service counter-measure*, 2002, http://hashcash.org/papers/hashcash.pdf. |
| [Cha83] | D. Chaum, *Blind signatures for untraceable payments, Advances in Cryptology*  Proceedings of Crypto 82 (1983), no. 3, 199–203. |
| [CMCWETH] | CoinMarketCap, *What Is Wrapped Ethereum (WETH)?* ,  <https://coinmarketcap.com/alexandria/article/what-is-wrapped-ethereum-weth> |
| [Gri99] | Ian Grigg, Email correspondence, 1999, <http://cryptome.org/jya/digicrash.htm>. |
| [Tarkoma2013] | Prof. Sasu Tarkoma , “*Overlay and P2P Networks*”, 2013 ,<https://www.cs.helsinki.fi/webfm_send/1040> |
| [ISOOSI] | [*ISO/IEC 7498-1:1994 Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*](https://www.iso.org/standard/20269.html)*.* June 1999. Introduction*. Retrieved 26 August 2022*.  <https://www.iso.org/standard/20269.html> |
| [ISOOSIW] | <https://en.wikipedia.org/wiki/OSI_model> |
| [MASTBIT2023] | "*Mastering Bitcoin*", Andreas M. Antonopoulos, David A. Harding, 2023 |
| [Mishkin2004] | Frederic Mishkin, Columbia University - The Economics of Money, Banking and Financial Markets (7th Ed.), Addison-Wesley, 2004. |
| [Nak09] | S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2009, <https://www.bitcoin.org/bitcoin.pdf>. |
| [TanenbaumProto] | Andrew S. Tananbaum, “Network protocols”, 1981,  <https://core.ac.uk/download/pdf/15449668.pdf> |

1. # [ERC-20 Token Standard](https://ethereum.org/ru/developers/docs/standards/tokens/erc-20/) - The standard allows for the implementation of a standard API for tokens within smart contracts. This standard provides basic functionality to transfer tokens, as well as allow tokens to be approved so they can be spent by another on-chain third party. Fabian Vogelsteller <[fabian@ethereum.org](mailto:fabian@ethereum.org)>, Vitalik Buterin <[vitalik.buterin@ethereum.org](mailto:vitalik.buterin@ethereum.org)>, "ERC-20: Token Standard," *Ethereum Improvement Proposals*, no. 20, November 2015. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20

   [↑](#footnote-ref-1)
2. <https://github.com/ethereum/wiki/wiki/Standardized_Contract_APIs/499c882f3ec123537fc2fccd57eaa29e6032fe4a> [↑](#footnote-ref-2)
3. [<https://www.techtarget.com/searchnetworking/definition/Application-layer>] [↑](#footnote-ref-3)
4. [Zigbee](https://www.digi.com/resources/documentation/Digidocs/90001942-13/containers/cont_zigbee_communication_in_depth.htm) is a global wireless standard that enables simple and smart objects (IoT) to work together. This interoperability—multiple devices from different vendors working together to achieve a common goal—is one of the biggest advantages of using the Zigbee protocol. [↑](#footnote-ref-4)
5. # [“Structured Peer-to-Peer overlay deployment on MANET: A survey

   ”Mohammad al mojamed, [Mario Kolberg](https://www.researchgate.net/profile/Mario-Kolberg), 2016, <https://www.sciencedirect.com/science/article/abs/pii/S1389128615004831>] [↑](#footnote-ref-5)
6. Proof-of-Work(**PoW**) is currently **the most common consensus mechanism** for blockchain

   technologies. The miner builds a candidate block filled with transactions. Then the

   miner calculates the hash of his block header and see if it fits the current target. If the

   hash does not fit, it will modify(often increment) the nonce, and then try again. [↑](#footnote-ref-6)
7. <https://medium.com/@alcio/how-to-destroy-bitcoins-255bb6f2142e> [↑](#footnote-ref-7)
8. <https://bitcoin.stackexchange.com/a/7443> [↑](#footnote-ref-8)