UNIVERSITY of NICOSIA

Session 6

Lightning Network & Ethereum Scalability

BLOC 514: Emerging Topics in Blockchains and Digital Currency

Session Objectives

- To discuss how a network of micropayment channels can solve the scalability problem of Bitcoin.
- To present how the 2-of-2 multi-signatures can be utilized in micropayment channels.
- To discuss the scalability issue for the case of Ethereum.
- To present the idea of sharding, which was proposed for enhancing Ethereum's scalability.

Different approaches have been proposed for alleviating the scalability issues of blockchains. We will present two representative approaches for the case of Bitcoin and Ethereum. We will see that different techniques are employed.

Agenda Slide

- Lightning Network (LN): micropayment channels can solve Bitcoin's scalability
- Benefits of the Lightning Network
- Ethereum at a glance
- Sharding in general
- Sharding for solving Ethereum's scalability
- Conclusions
- Bibliography

Lightning Network: micropayment channels can solve Bitcoin's scalability

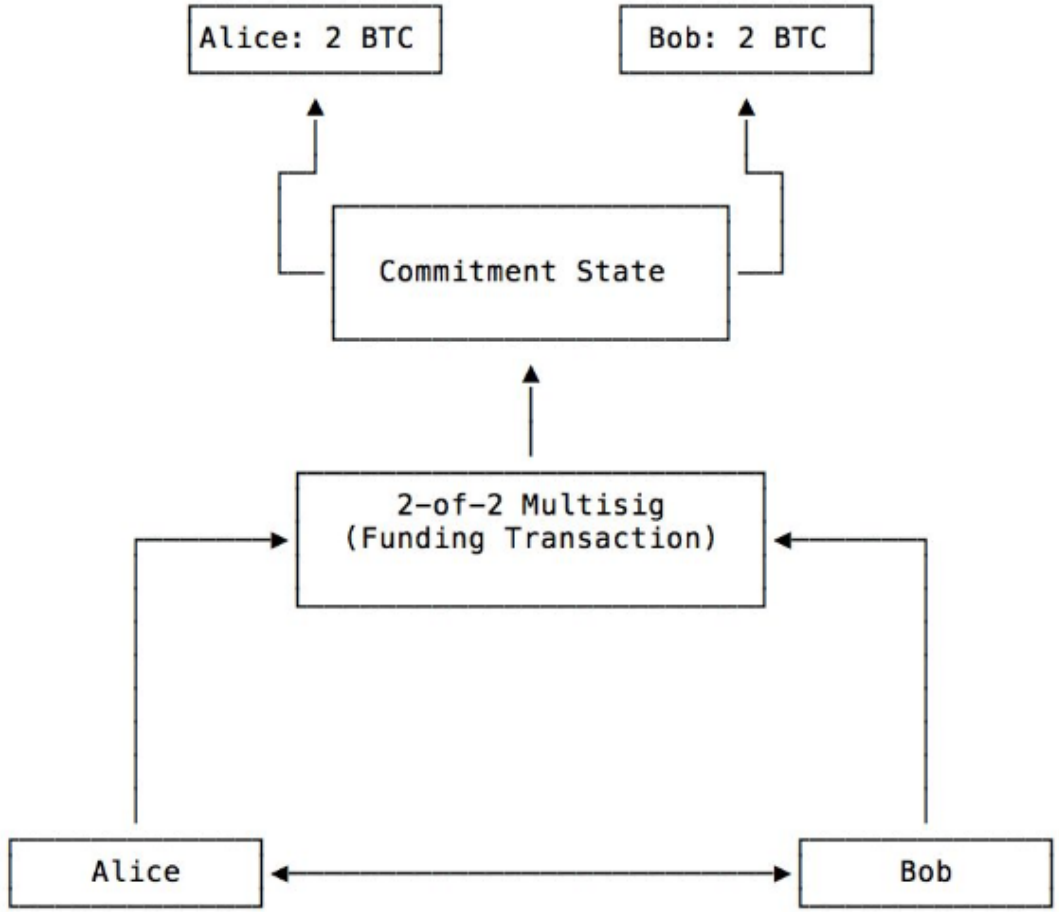
Lightning Network

- New Idea 1: The everyday transaction does not store on the main blockchain
- We can increase the 7-transactions limit with this off-chain methodology
- New idea 2: Open a payment channel
- The channel can stay open for any time
- So, we can have a network of payment channels
- Payment channels between many parties in a multi-hop hub and spoke model (similar to internet routing)
- See
  - Technical paper
  - Related article

Lightning Network at a glance

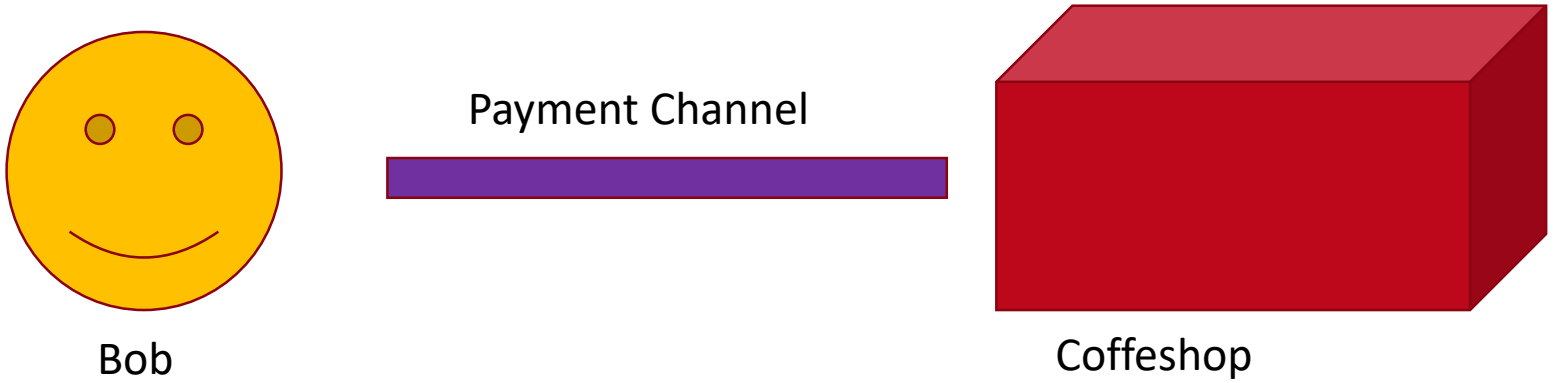
- Enter Lightning: **Off-Chain** Bitcoin payments

- Alice and Bob enter into a **contract**
- Contract creation:
  - Funds put into 2-of-2 multi-sig
  - **Before broadcast** transaction to *deliver* funds is signed
    - Requires malleability fix
  - Funding transaction broadcast
- Off-chain payments (sub-contract):
  - **HTLC**: Hash-Time-Lock-Contract
- Contract completion:
  - Closing transaction broadcast, final balance delivered
- On-chain footprint:
  - **2** transactions
  - All updates **point-to-point**
  - **Predictable** fees



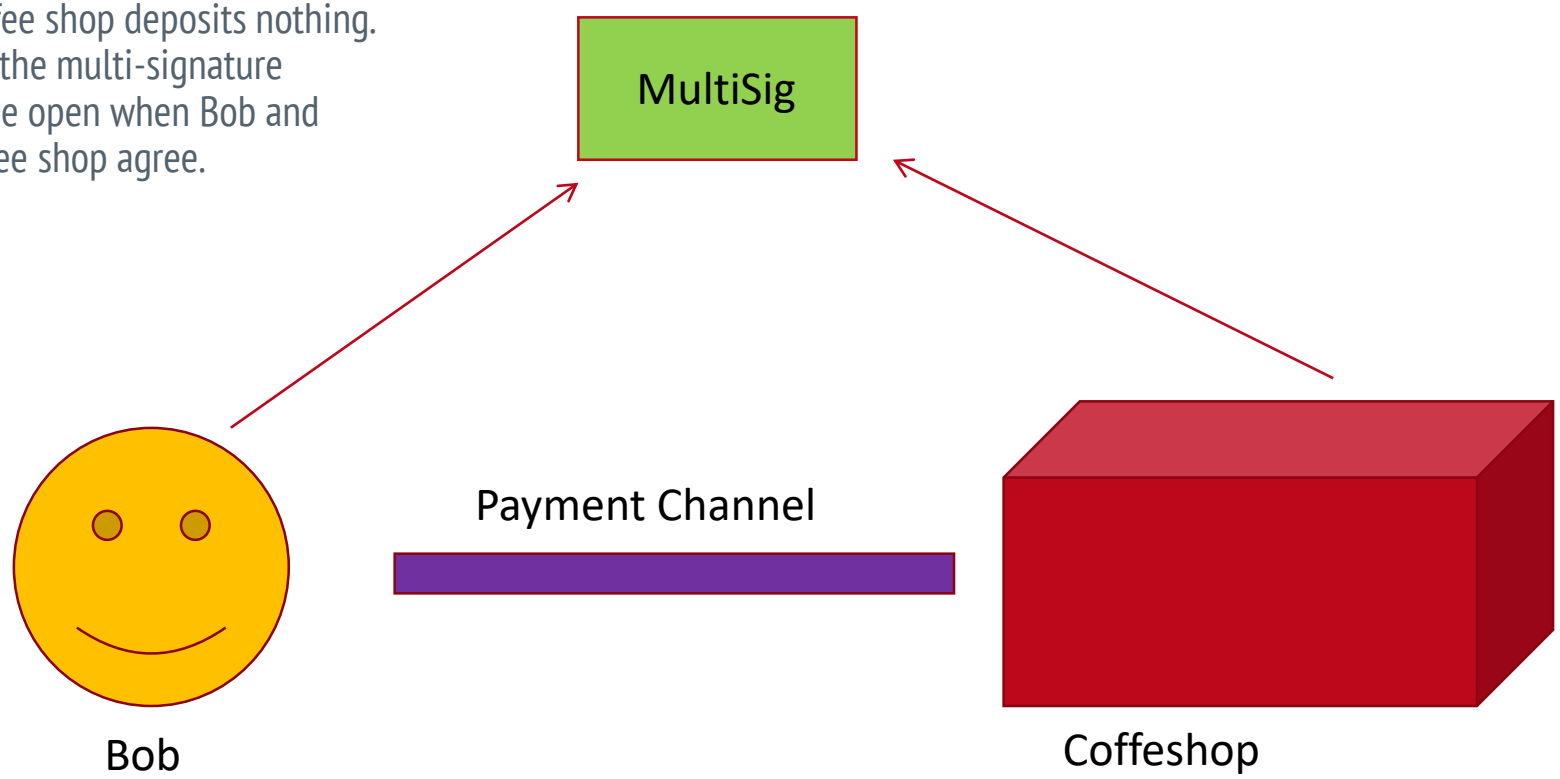
Bitcoin's Lightning Network - Explained

- Assume that Bob wants to pay for a coffee via Bitcoin: more fees compared to the price of the coffee. If Bob use the Lightning Network, he will be able to set up a payment channel with the coffee shop.



Bitcoin's Lightning Network - Explained

- Bob and the owner of the coffee shop deposit an amount of BTC in a multi-signature address.
  - Assume that Bob's deposit is 0.1 BTC.
  - The owner of the coffee shop deposits nothing. As you already know, the multi-signature is safe and can only be open when Bob and the owner of the coffee shop agree.



Bitcoin's Lightning Network - Explained

- When the two entities open a payment channel, they also create a balance sheet that says how the funds in the address should be distributed. So right now it says, “Bob will get 0.1 BTC and the owner of the coffee shop will get 0 BTC.” This way, full transparency is provided. The coffee shop owner can see that Bob has deposited 0.1 BTC and he can safely assume that he will get his money once the channel closes. Now Bob can order his morning coffee.
- Assume that the coffee costs 0.001 BTC. To pay for it, Bob simply changes the balance sheet. Specifically, he subtracts the cost of coffee from his balance and adds it to the coffee shop’s balance. After that the balance sheets says “Bob gets 0.099 BTC and the coffee shop gets 0.001BTC.”

Bitcoin's Lightning Network - Explained

- Bob and the owner of the coffee shop now sign the updated balance sheet with their private keys. Also, they can keep a copy of it.
- Bob can continue ordering coffees as long as he has a balance in the payment channel. Both of them can make hundreds of thousands of transactions between them. There is really no limit because this happens away from the main blockchain.
- The payment channel can be closed at any time by either Bob or the coffee shop. All they have to do is take the latest balance sheet.
- Finally, miners will validate the signatures on the balance sheet and release the funds according to it.

Lightning Network Benefits

- Privacy. Lightning Network payments are much more private than payments on the bitcoin blockchain, as they are not public.
- Fungibility. A Lightning Network makes it much more difficult to apply surveillance on bitcoin, increasing the fungibility.
- Speed. Bitcoin transactions using Lightning Network are settled in milliseconds, rather than minutes.
- Granularity. A Lightning Network can enable payments at least as small as the bitcoin "dust" limit, perhaps even smaller. Some proposals allow for subsatoshi increments.
- Capacity. A Lightning Network increases the capacity of the bitcoin system by several orders of magnitude. There is no practical upper bound to the number of payments per second that can be routed over a Lightning Network, as it depends only on the capacity and speed of each node.
- Trustless Operation. A Lightning Network uses bitcoin transactions between nodes that operate as peers without trusting each other. Thus, a Lightning Network preserves the principles of the bitcoin system, while expanding its operating parameters significantly

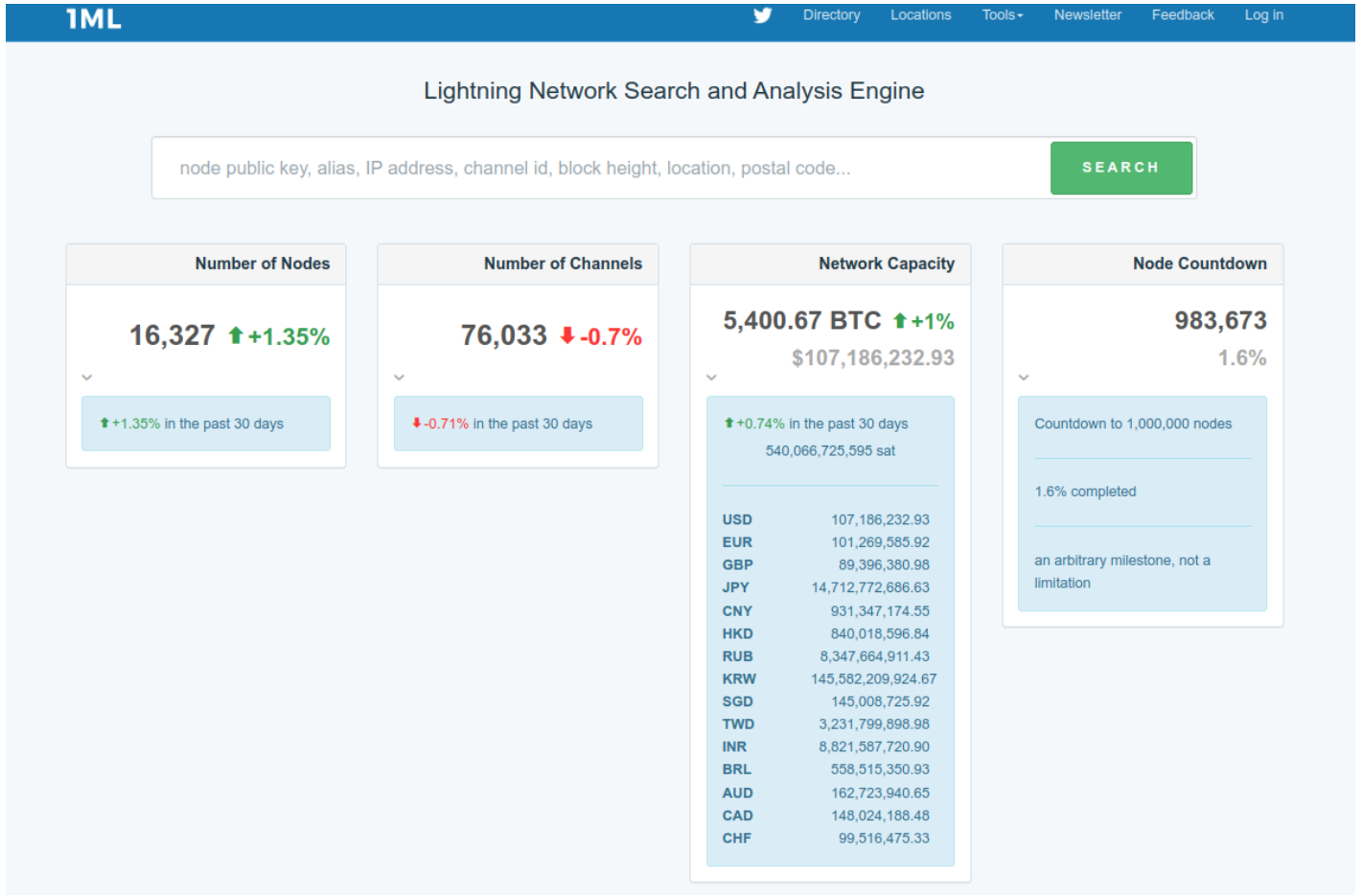
Lightning Network reviewed

According to this article there is a number of potential problems

- Bitcoin's fees (Layer 1 fees) are not fully eliminated
- LN as a Layer 2 solution comes with specific types of costs:
  - Cost for opening and closing the secondary channels
  - Cost for the routing operations in the secondary channels
- If the Layer 2 fees are quite low, then the respective incentive weakens
- Who provides the Layer 2 solution really matters
  - What if this is done by a 3rd party having the control of Layer 2 costs?
- Problems related to price volatility cannot be addressed
  - What if an invoice in a Layer 2 channel is allowed a 30-day period to be paid?
- In summary:
  - Not all Bitcoin-specific challenges/problems can be solved by LN
  - LN constitutes a product under development

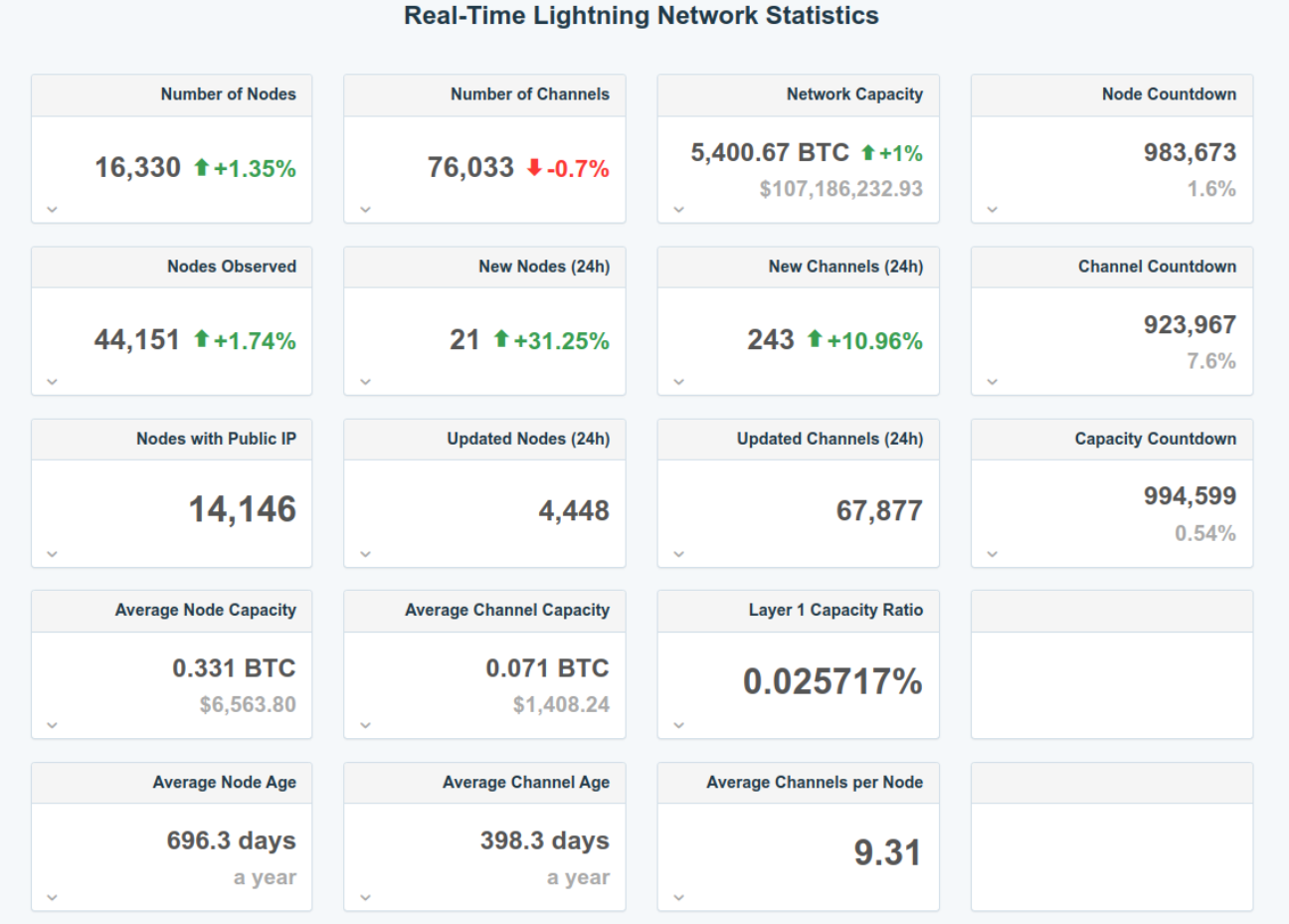
Lightning Network: Explorer

https://1ml.com/



Lightning Network: Explorer - Statistics

https://1ml.com/statistics



Lightning Network: Explorer - Statistics

- In https://1ml.com/directory organization of use cases into the following categories:
  - Utilities
  - Financial
  - Software & Hardaware
  - Business
  - Media
  - Games
  - Marketplaces
  - Education
  - Lifestyle
- Indicative examples
  - Send/receive tips (and, in general, micropayments): https://tippin.me/
  - Giftcards. In general, enable retailers to use their existing gift card infrastructure: https://coincards.com/
  - More use cases mentioned in a recent study: Bitcoin Lighting Network: The Future of Payments?" by Betz et al. (2023)

Lightning Network

Lightning labs: https://lightning.engineering/

Design and development of most widely-used LN software

- LN Daemon
  - Implementation of the LN
- Pool
  - Marketplace dealing with channel liquitidy
- Loop
  - Service enabling transactions between on-chain addresses and LN channels
- Neutrino
  - Application enabling the operation of an LN node in devices like phones (connection to remote a Bitcoin node)

Lightning Network

Lightning labs - deamon: https://github.com/lightningnetwork/lnd

Lightning Network Daemon

build failing license MIT chat on libera reference

The Lightning Network Daemon (`lnd`) - is a complete implementation of a Lightning Network node. `lnd` has several pluggable back-end chain services including `btcd` (a full-node), `bitcoind` , and `neutrino` (a new experimental light client). The project's codebase uses the btcsuite set of Bitcoin libraries, and also exports a large set of isolated re-usable Lightning Network related libraries within it. In the current state `lnd` is capable of:

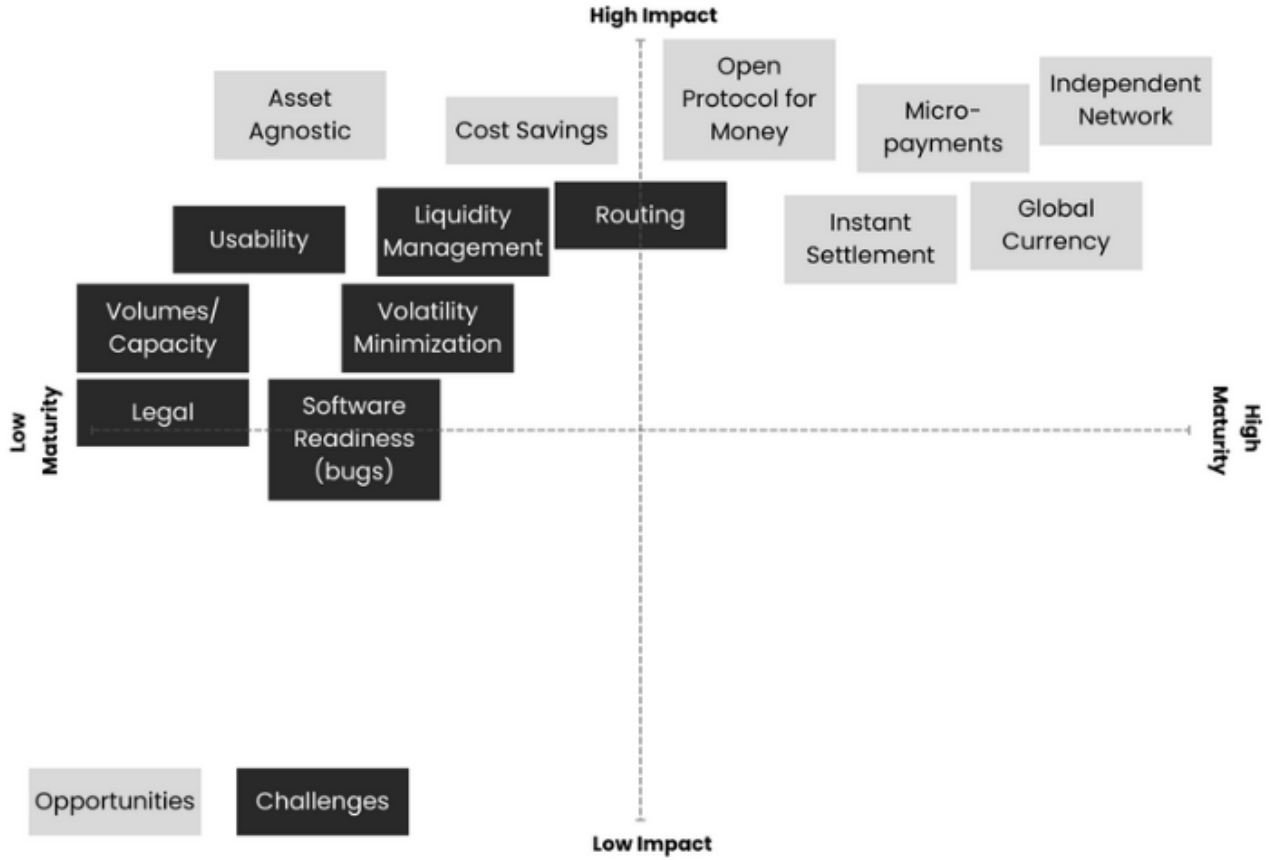
Lightning Network

Map of LN channels

Source: https://mempool.space/graphs/lightning/nodes-channels-map

Lightning Network

Challenges and opportunities



Source: "Bitcoin Lighting Network: The Future of Payments?" by Betz et al. (2023)

Sharding for solving Ethereum's scalability

Ethereum at a glance

- A blockchain equipped with programming language(s)
  - Add expressive power to the intended applications
  - Suitable for the development of smart contracts
- Comparison against public blockchains
  - Most public blockchains are designed for the transfer of cryptocurrencies between users
  - Smart contracts enable the development of more complex (and interesting) applications
- High-level definition of a smart contract
  - A program implemented in a programming language the enables the direct control of digital assets
  - Need to be securely executed
- Digital assets: more than cryptocurrencies
  - Examples: precious metals, stock shares, domain names, intellectual property, etc
  - Even computational resources such as processing power, bandwidth

Ethereum at a glance

There are two types of accounts

- Normal accounts
  - Similar to Bitcoin account: address and balance
- Smart contract accounts: Represented as an object (i.e., an entity – concept use in object-oriented programming languages) consisting of the following
  - Code: implementation of the contract logic using the respective programming language
  - Private storage: data consisting of key-value pairs
- Functionality of the code:
  - Transfer digital assets to accounts
  - Access (read/write) to the private storage
  - Execute other contracts

Bitcoin vs. Ethereum states

- A Bitcoin's state consist of
  - Key-value pairs that map addresses to balance
- An Ethereum's state consist of
  - Key-value pairs that map addresses to objects
  - Balance is part of the object

Bitcoin's address	Bitcoin's balance	Ethereum's address	Ethereum's object
0xb1a33f	5.2	0xb1a33f	X
…	…	…	…
0xek365f	2.1	0xek365f	Y

Ethereum transaction chart (Mar 2023): scalability issue

- Every Ethereum node that participates in the network needs to process every transaction

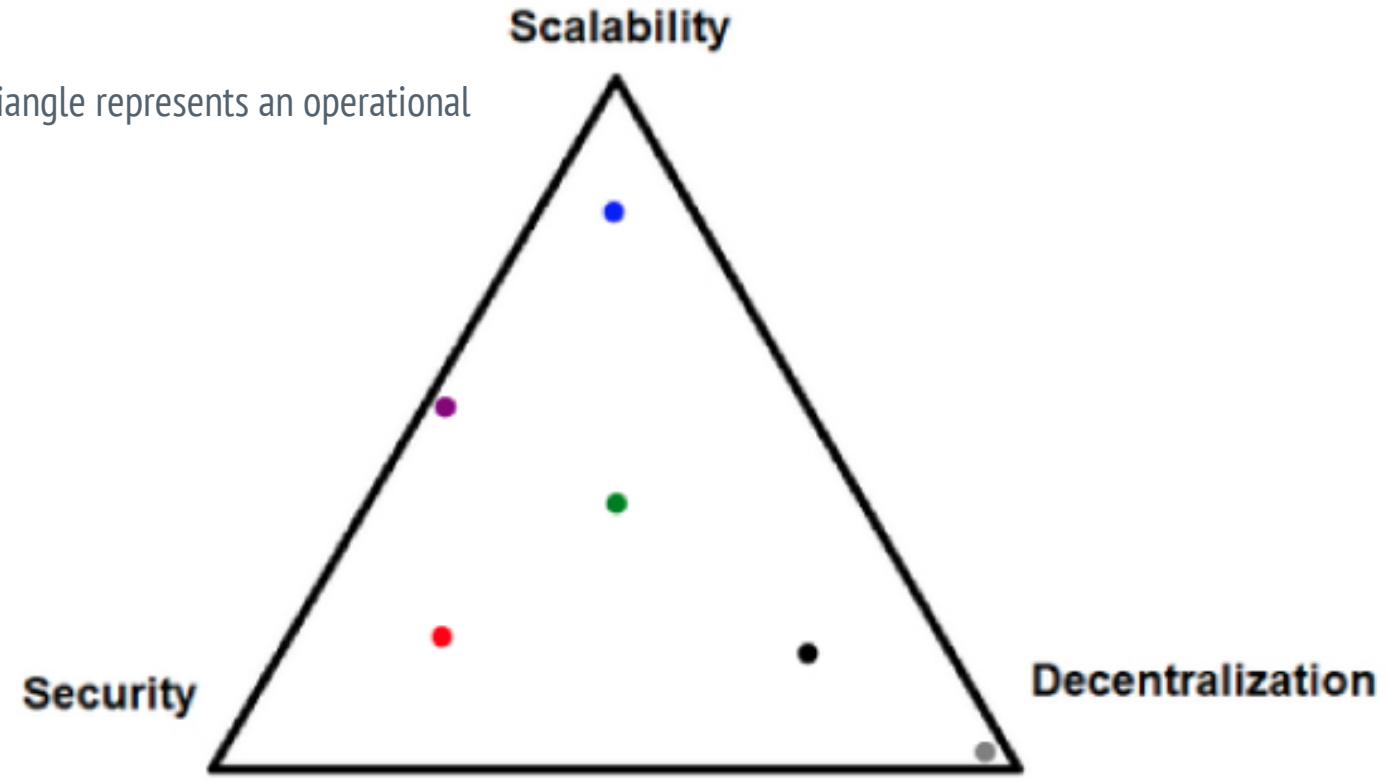
Source: https://etherscan.io/chart/tx

Ethereum scalability

- Current situation: Every Ethereum node that participates in the network needs to process every transaction
- Advantage of current situation
  - High security degree: the amount of validation that is devoted to each block is proportional to the number of nodes
- Disadvantage of current situation
  - The speed of the network is bounded by the fastest individual node (unlike the degree of security)
- Currently
  - Transactions can not be parallelized, i.e., a sequential mode is used for their processing
- Two general (and not trivial) approaches for tackling the scalability issue
  - Employ off-blockchain techniques: a number of transactions are processed in an external environment, while the interaction with the blockchain takes place occasionally
  - Adapt the internal protocol enabling the parallelization of transactions

Trilemma

Each point within the triangle represents an operational mode/configuration



Source: https://www.coinreview.com/blockchain-trilemma/

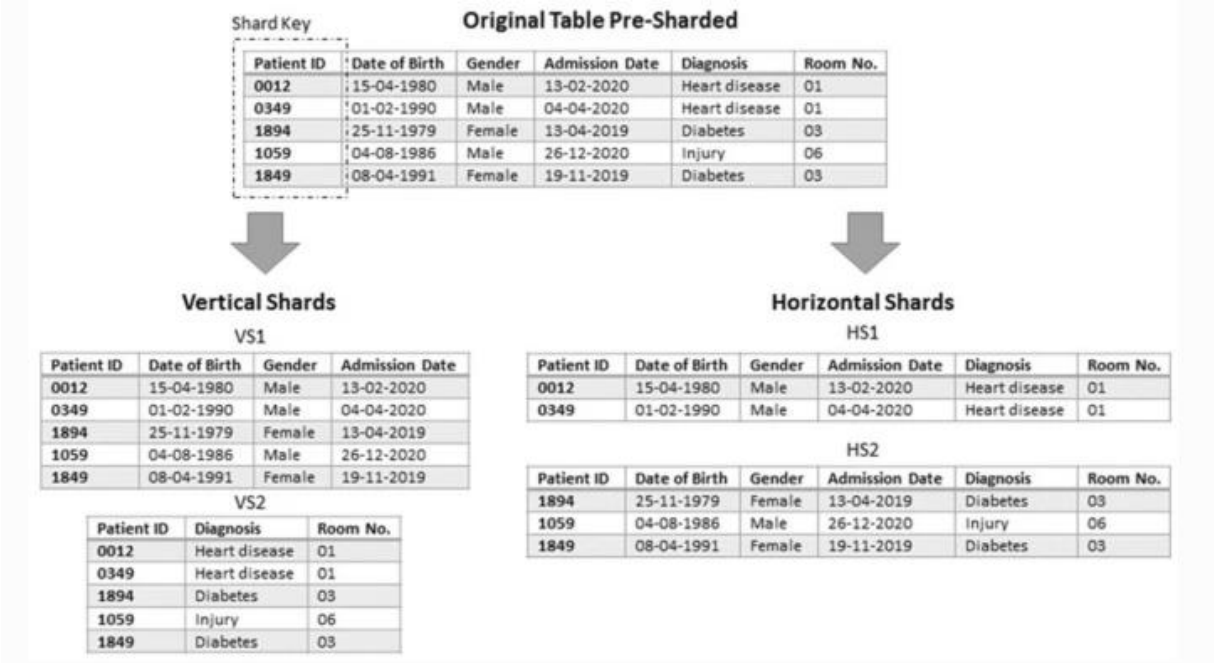
Ethereum scalability in the framework of a trilemma

The scalability issue can be also considered with respect to the following trilemma

- A blockchain can satisfy only two (at most) of the following three properties
  - Decentralization
  - Scalability
  - Security
- Example: assume increased scalability and security
  - This combination probably implies decreased degree of decentralization (why?)
- Other ways for enhancing throughput: increase the block size (in the case of Ethereum: GAS_LIMIT)
  - Sounds a reasonable approach, but …
  - It is likely to increase the mining centralization as computationally powerful nodes will gain an advantage compared to other nodes

Sharding in general

- In general utilized in relational databases (DB) for speeding up transactions
  - How: logical segmentation of respective tables ('tables' in the DB sense)
  - Two basic approaches according to the paper referenced below: vertical, horizontal



Original Table Pre-Sharded

Patient ID	Date of Birth	Gender	Admission Date	Diagnosis	Room No.
0012	15-04-1980	Male	13-02-2020	Heart disease	01
0349	01-02-1990	Male	04-04-2020	Heart disease	01
1894	25-11-1979	Female	13-04-2019	Diabetes	03
1059	04-08-1986	Male	26-12-2020	Injury	06
1849	08-04-1991	Female	19-11-2019	Diabetes	03

VS1

Patient ID	Date of Birth	Gender	Admission Date
0012	15-04-1980	Male	13-02-2020
0349	01-02-1990	Male	04-04-2020
1894	25-11-1979	Female	13-04-2019
1059	04-08-1986	Male	26-12-2020
1849	08-04-1991	Female	19-11-2019

VS2

Patient ID	Diagnosis	Room No.
0012	Heart disease	01
0349	Heart disease	01
1894	Diabetes	03
1059	Injury	06
1849	Diabetes	03

HS1

Patient ID	Date of Birth	Gender	Admission Date	Diagnosis	Room No.
0012	15-04-1980	Male	13-02-2020	Heart disease	01
0349	01-02-1990	Male	04-04-2020	Heart disease	01

HS2

Patient ID	Date of Birth	Gender	Admission Date	Diagnosis	Room No.
1894	25-11-1979	Female	13-04-2019	Diabetes	03
1059	04-08-1986	Male	26-12-2020	Injury	06
1849	08-04-1991	Female	19-11-2019	Diabetes	03

Source: "Sharding for Scalable Blockchain Networks" by Hashim et al..

Blockchain sharding

- Related terminology
  - **State:** Information that characterizes the blockchain at any moment (for Ethereum the state includes balances and smart contracts). Obviously, the successful execution of a transaction changes the blockchain state.
  - **Transaction:** An action that is committed by a user which results into a state change
  - **Receipt:** Every transaction generates an output (similar to the log messages of programs). The receipts are stored in a Merkle tree that does constitute part of the blockchain state. However, the nodes have access to the tree in order to verify the receipts.

Sharding in blockchains

3 types of sharding

- Sharding in blockchain enables parallelization
- Sharding: partitioning of the network into sub-networks
  - Shards can (also) be regarded as committees
- Three types of approaches:
  - Network sharding
    - The most basic approach
  - Transaction sharding
    - Based on network sharding
  - State sharding
    - The most challenging approach

Source: "Sharding for Scalable Blockchain Networks" by Hashim et al..

Sharding in blockchains

Network sharding

- Network is partitioned into shards
- In each shard: each verification is done independently (according to the underlying consensus)
- This type of independence favors decentralization
- However, a basic assumption is adopted: given a shard, as respective nodes are nodes
- Shards are configured according to two schemes as follows
  - Statically
  - Dynamically
- Static configuration
  - The assignment of nodes to shards is fixed
  - This assignment follows a specific business logic or, in general, a criterion
  - Assumes trust, thus, it is not appropriate for permissionless networks
- Dynamic configuration
  - The assignment of nodes to shards is done periodically (unit of time: epoch)
  - The periodic assignment is conducted in a random way
  - This randomness reduces the possible harm by the (constant) presence of an adversary in a specific shard

Sharding in blockchains

Transaction sharding

- It is based on network sharding
- Given a set of shards, transactions are:
  - Classified, and
  - Assigned to the shards
- The specifics of the transaction processing depends heavily on the underlying model of the ledger
- Two types of assignments
  - Random
  - According to criteria
- Possible weak point of random assignment
  - Assume a "bad" node issuing multiple transactions
  - Those transactions are assigned to different shards
  - Possibility of double spending
- Indicative example of criteria-based assignment
  - Consider, simple transaction dealing with transfer of funds between sender and receiver
  - If multiple transactions from the (same) sender: assign all respective transactions to the same shard

Sharding in blockchains

State sharding

- Background: In general, given a network, the full ledger's state is stored and maintained in each node
- In state sharding, given a set of shards
  - The whole state is segmented
  - Each state segment is assigned to a shard
- This creates the following operational properties
  - Local (i.e., at the shard level) sub-states
  - Locality can be translated to independence
- Yielded benefits for nodes
  - Less computational demands
  - Less storage demands
- Challenges:
  - Cross-shard communication
    - How to model and implement it
    - Performance obstacle
  - In case of a shard is not available, the respective state-specific data is also not available.
    - Possible solution #1: special nodes maintaining the full state
    - Possible solution #2: maintainance of state archives

Blockchain sharding: collators

Cross-shard communication

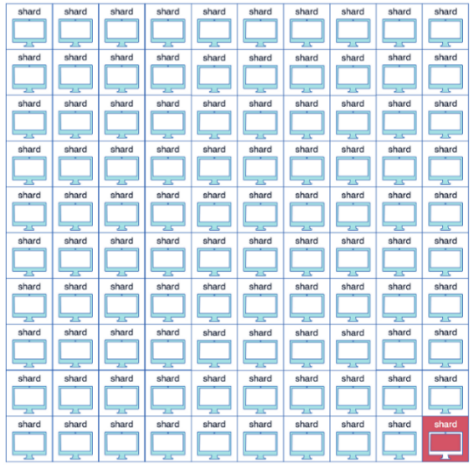
- Assume a certain shard
  - **Collators:** A set of nodes aimed to formulate a collation
  - **Collation:** A structure that encodes salient information that characterizes the respective shard
- The collations can be regarded as state/transactions descriptors of the corresponding shard
- Each collation includes a collation header that consists of the following fields:
  - The corresponding shard (e.g., Shard No. 12)
  - The shard's state before the execution of all transactions
  - The shard's state after the execution of all transactions
  - Digital signatures provided by the 67% (at least) of collators ensuring that the collation is valid
- Cross-shard transactions
  - Assume a transfer from an address in Shard No. 3 to an address in Shard No. 12
  - Clearly, a mechanism is needed for enabling such transactions
    - This is feasible through the incorporation of receipts

Transactions across shards: basic steps

- Assume that User 1 (owner of an address in Shard No. 3) wishes to transfer 1 ETH to User 2 (owner of an address in Shard No. 12)
  - Step 1: A transaction is transmitted to Shard No. 3 and the balance of User 1 is set accordingly (i.e., reduction by 1 ETH). Shard No. 3 waits for the finalization of the transaction.
  - Step 2: For this transaction a receipt is generated, which is stored in a Merkle tree. This receipt is verifiable.
  - Step 3: A transaction is transmitted to Shard No. 12 which is augmented by the aforementioned receipt. A check is conducted by Shard No. 12 in order to verify that the receipt has not been spent.
  - Step 4: If the previous verification is successful, Shard No. 12 allows the processing of the transaction and as a result the balance of User 2 is increased by 1 ETH. In addition, the corresponding receipt is updated (i.e., indicated as "spent").
  - Step 5: Finally, a new receipt is created by Shard No. 12. This receipt can be used in forthcoming transactions.

Single-shard takeover attack

- When the majority of collators in given a certain shard are controlled by the attacker
  - The affected shard can transmit invalid collations



1% Attack

> In 100 shards system, it takes only 1% of network hash rate to dominate the shard.

- Solution: the collators of each shard are selected via random sampling
  - Technical challenges: periodic synchronization with the randomly selected collators
  - Fortunately, this mechanism is implemented at the protocol layer and does not affect the development of applications

Ethereum 2 at a glance

- According to ethereum.org :

*"Eth2 refers to a set of interconnected upgrades that will make Ethereum **more scalable**, **more secure**, and **more sustainable**. These upgrades are being built by multiple teams from across the Ethereum ecosystem"*

More scalable

Ethereum needs to support 1000s of transactions per second, to make applications faster and cheaper to use.

More secure

Ethereum needs to be more secure. As the adoption of Ethereum grows, the protocol needs to become more secure against all forms of attack.

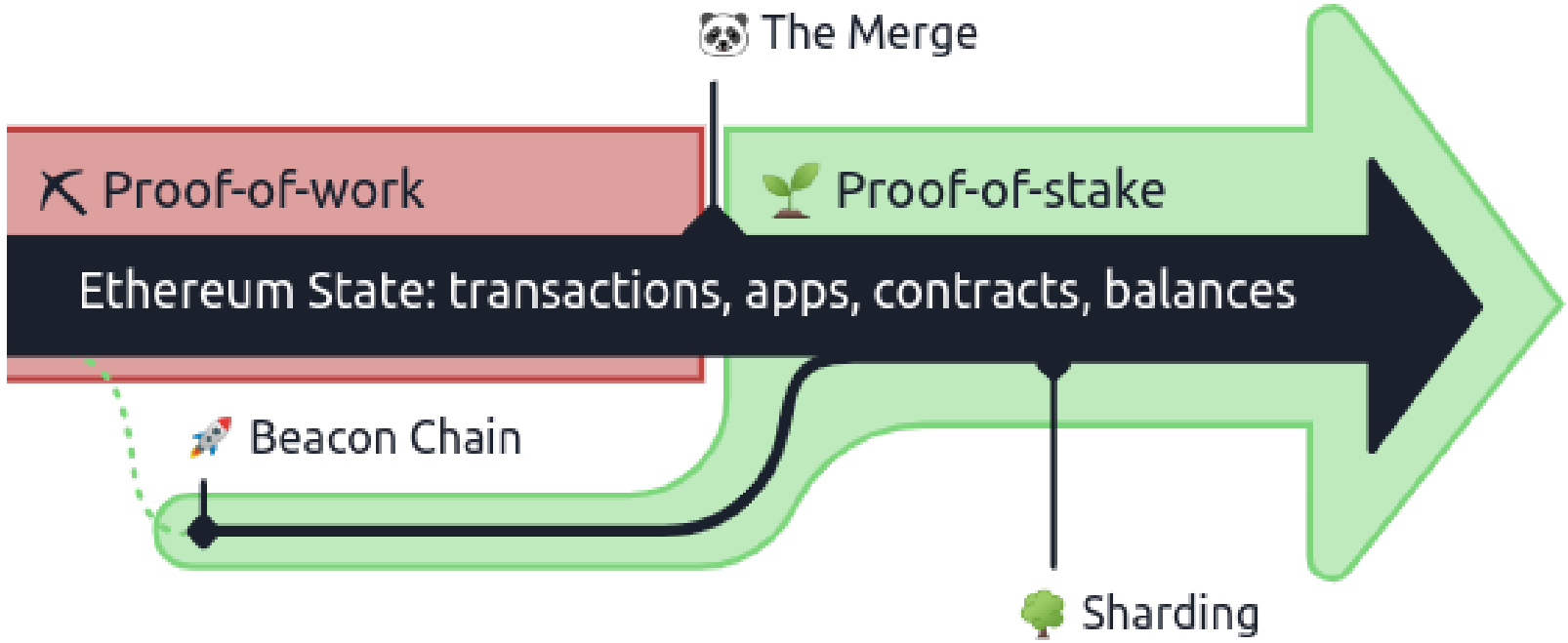
More sustainable

Ethereum was energy-intense until recently. The transition to proof-of-stake brought a network energy reduction of over 99.9%.

Source

Sharding at Ethereum 2

Merge: 15 Sep 2022 – Ethereum's transition to PoS



- According to ethereum.org, in the context of Merge, priority to PoS was given, as compared to sharding, due to the rise of Layer 2 DApps. At the same time, *"Plans for sharding are rapidly evolving, ..."*

"The Merge" by ethereum.org (last updated March 11, 2023)

Polygon

- Polygon, a scaling solution for Ethereum: https://polygon.technology/
  - Previously known as 'Matic Network'
  - Respective token: MATIC
  - Briefly: It constitutes both a protocol and framework for developing/integrating Ethereum-based blokchains
  - It is intended as a Layer 2 solution
  - Use of Proof-of-Chain

- High-level use:
  - Deposit funds into a Polygon-based smart contract
  - Conduct a series of transactions/operations within the Polygon network
  - Once "needed", withdraw funds to Ethereum

Solana

- Solana: https://solana.com/
  - Main features/innovation: high speed & low fees
  - Respective token: SOL
  - SOL can be also used as a governance token
  - Briefly: A framework in the sense of a platform where DApps can be developed through smart contracts programming
  - Use of Proof-of-Chain & Proof-of-History
    - An introductory short article about Solana's Proof-of-History
- More:
  - Unfortunately, manipulations/hacks are part of the game and Solanana is/was no exception, e.g., this case
    - Interestingly, as noted: *"Engineers across several networks have found that the bug isn't connected with Solana core code, but in software used by several software wallets ..."*
  - Also, the Mango Market case

Ethereum – Polygon - Solana

Comparative analysis [*] suggested by this article

Criteria	Ethereum	Solana	Polygon
Native Token	ETH	SOL	MATIC
Year of Foundation	2013	2017	2017
Programming Language	Solidity	Rust, C, C++	Golang, Solidity, Vyper
Transaction Speed	13-15	50,000-65,000	65,000
Consensus Mechanism	Proof of Work	Proof of Stake and Proof of History	Proof of Stake Plasma-based sidechain
Architecture	Stateful architecture	Stateless architecture	Multichain architecture
Scalability	Limited Scalability	High-Performance protocol for scalability	Multichain solutions offer better scalability

[*] Before Ethereum Merge

Conclusions

Conclusions

- Different approaches for solving scalability issues
  - Combination of off-blockchain and in-blockchain operations
  - Adaptation of the internal protocol for enabling parallelization transactions
- Examples
  - Bitcoin: Lightning Network
  - Ethereum: Sharding
- For every new approach:
  - New series of thorough tests are required
  - The introduction of changes should be hidden from the developers.
- New techniques are likely to be proposed based on variants of those approaches

Resources

References

- **References**
  - Andreas M. Antonopoulos (2017). **Mastering Bitcoin: Programming the Open Blockchain**. 2nd Edition. Sebastopol: O'Reilly Media
    - Chapter 12 - Section **"Routed Payment Channels (Lightning Network)"**
  - **The Decentralization-Security-Scalability trilemma** (here)

Bibliography (optional)

- Joseph Poon, Thaddeus Dryja **The Bitcoin Lightning Network** (link)
  The authors proposed a decentralized system (LN) whereby transactions are sent over a network of micropayment channels (a.k.a. payment channels or transaction channels) whose transfer of value occurs off-blockchain.
- Pavel Prihodko et al. **Flare: An Approach to Routing in Lightning Network** (link)
  The authors describe a hybrid routing algorithm Flare, which could be used for payment routing in Lightning Network. The design goal for the algorithm is to ensure that routes can be found as quickly as possible.
- Conrad Burchert, Christian Decker and Roger Wattenhofer **Scalable Funding of Bitcoin Micropayment Channel Networks** (link)
  The authors propose a new layer that sits in between the blockchain and the payment channels. The new layer addresses the scalability problem by enabling trust-less off-blockchain channel funding. It consists of shared accounts of groups of nodes that flexibly create one-to-one channels for the payment network.
- Christian Decker, Roger Wattenhofer **A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels** (link)
  The authors present a protocol for duplex micropayment channels, which guarantees end-to-end security and allow instant transfers, laying the foundation of the PSP network.
- Joseph Abadi, Markus Brunnermeier **Blockchain Economics** (link)

  Presentation of a model comparing the economic benefit of decentralized vs. centralized ledgers.

Bibliography (optional) LN book

“Mastering the Lightning Network”

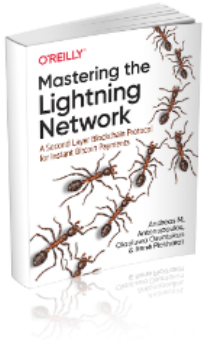
by Andreas M. Antonopoulos , Olaoluwa Osuntokun, Rene Pickhardt

See: https://github.com/lnbook/lnbook

Mastering the Lightning Network

build passing

STATUS: First Edition published on Dec 21, 2021



About

Mastering the Lightning Network is an O'Reilly Media book, by authors Andreas M. Antonopoulos (@aantonop), Olaoluwa Osuntokun (@roasbeef), Rene Pickhardt (@renepickhardt). It was published on Dec 21, 2021, in paperback and e-book, by O'Reilly Media. It is available everywhere that books are sold. This repository contains the manuscript of the book as published by O'Reilly Media, tagged as firstedition_firstprint.

UNIVERSITY *of* NICOSIA

Instructor’s Email:
iosif.e@unic.ac.cy