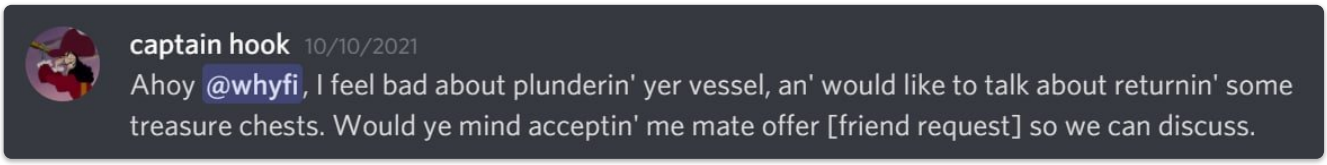
Lesson 15

“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.
The first method is far more difficult.” -C.A.R. Hoare

Exploited by a pirate

https://twitter.com/bertcmiller/status/1505698980067434541
An exploit in an MEV bot allowed an attacker to take $1m in a single transaction
The attacker felt bad so they revealed themselves in the Flashbots Discord and offered to give the victim their money back... all while roleplaying as a pirate



Attack at LiFi

https://blog.li.finance/20th-march-the-exploit-e9e1c5c03eb9
$600K have been stolen from 29 wallets.

The hack took advantage of our pre-bridge swap feature.
The contract allows a caller to pass an array of multiple swaps using any address with arbitrary calldata.

The attacker started by passing a legitimate swap of a small amount followed by multiple calls directly to various token contracts. Specifically, they called the `transferFrom` method which allowed the attacker to transfer funds from users’ wallets that had previously given infinite approval to our contract for that specific token.

Fears about complexity

https://twitter.com/peter_szilagyi/status/1504887160842903552

"As good as it feels that we're approaching The Merge, I must emphasize that Ethereum is not going in a clean direction. Tangentially it's achieving results, but it's also piling complexity like there's no tomorrow. If the protocol doesn't get slimmer, it's not going to make it."

Security Issues around NFTs

Paper

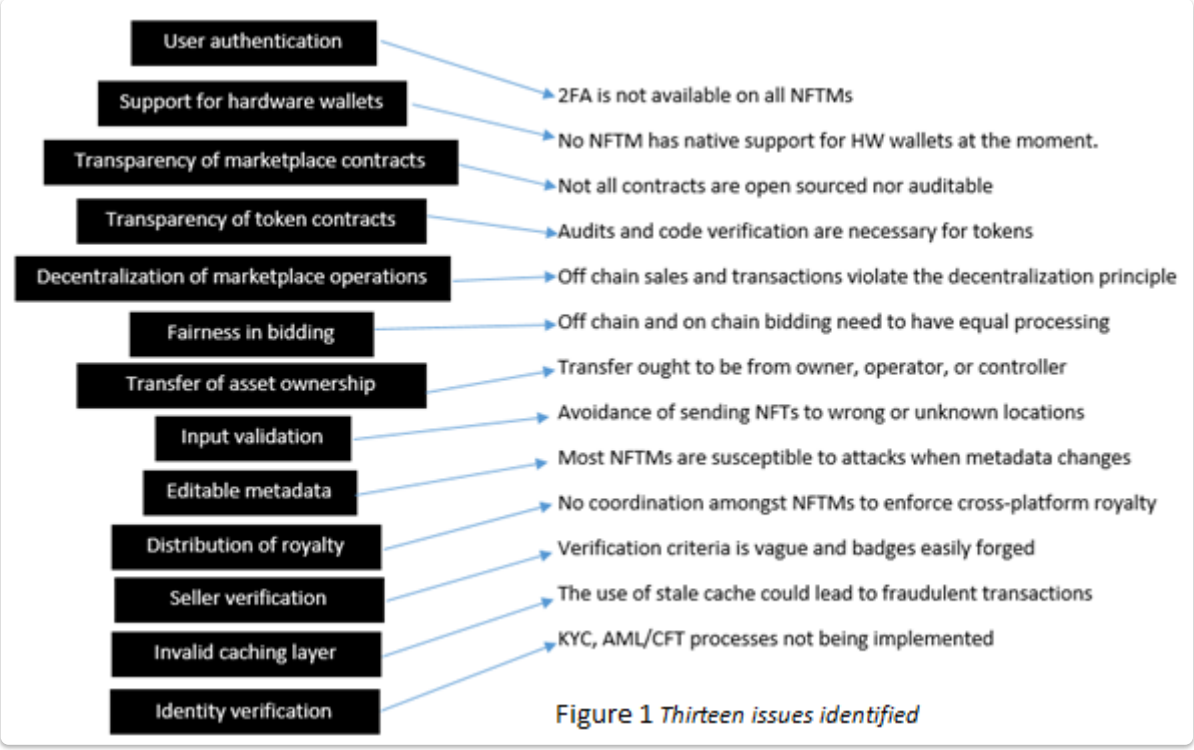


Figure 1 *Thirteen issues identified*

- The study identifies 5 security bugs in 3 of the largest NFTMs (OpenSea, Rarible, and Sorare), three of which the identified parties had identified and remedied. The remedied security bugs remain undisclosed due to non-disclosure agreements signed by the authors.
- The analysis of images and metadata reveals that many old tokens are invalid and do not contain images; consequently, a high number of NFTs have broken chains. This conclusion was reached after reviewing 12,215,650 assets from OpenSea, which returned only 4,393,566 assets with a valid metadata URL.
- The findings reveal that 98.14% of wash-trade transactions reported point to Rarible, which the authors attribute to malicious users attempting to capture the platform's $RARI tokens. OpenSea represented 1.71% of its transactions, with Sorare making up the rest of the total. OpenSea and Sorare showed 3,395 instances of shill bidding, while 492 instances of bid shielding involved 745 users across 113 collections on OpenSea.

Maths

https://www.smartcontractresearch.org/t/deep-diving-into-prbmath-a-library-for-advanced-fixed-point-math/686
You get these functions

- Absolute
- Arithmetic and geometric average
- Exponentials (binary and natural)
- Floor and ceil
- Fractional
- Inverse
- Logarithms (binary, common and natural)

- Powers (fractional number and basic integers as exponents)
- Multiplication and division
- Square root

In addition, there are getters for mathematical constants:

- Euler's number
- Pi
- Scale (1e18, which is 1 in fixed-point representation)

Ethereum Upgrades

Rollup centric roadmap :https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698
Upgrade Guide

BEACON CHAIN

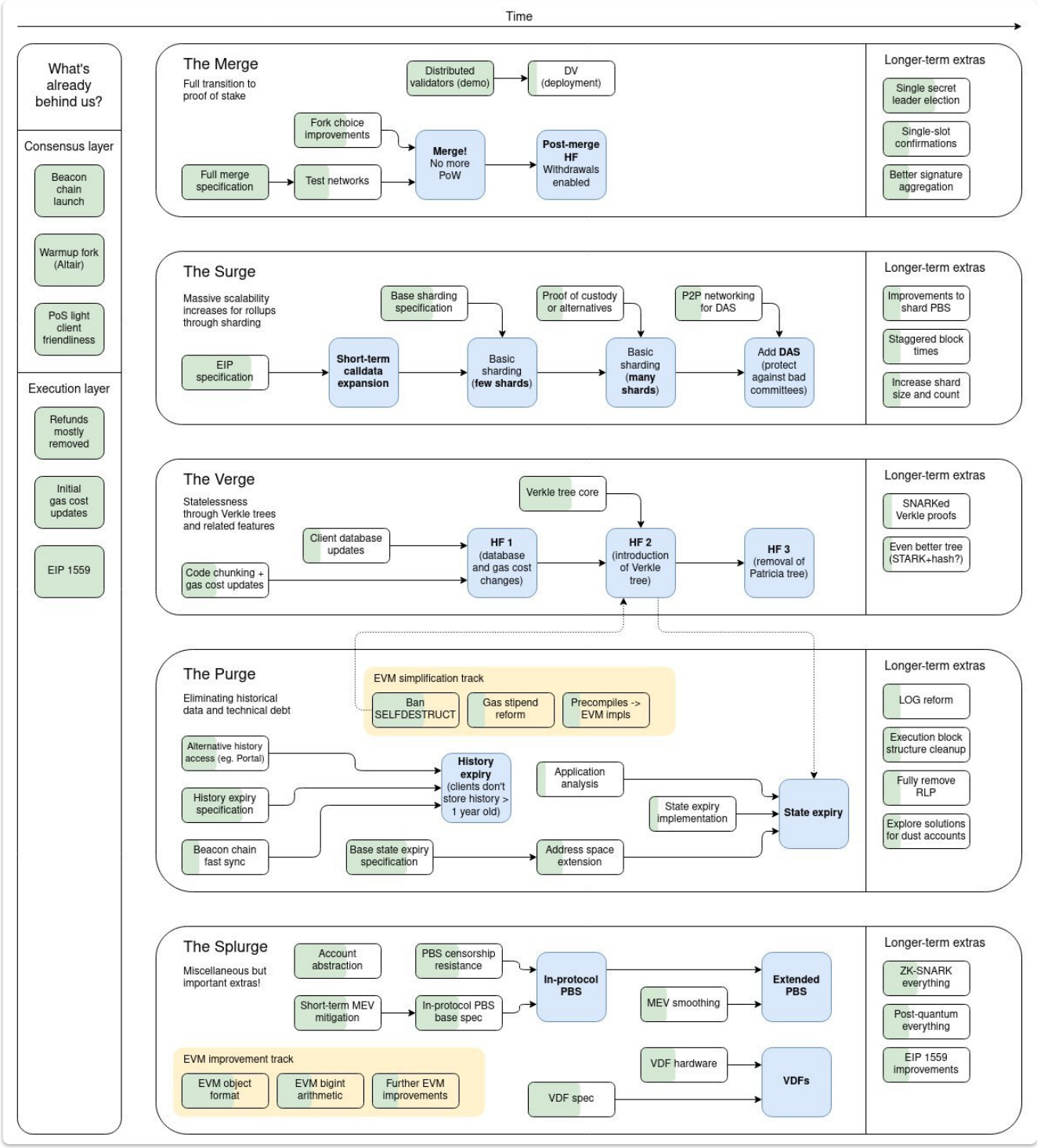
Does not handle accounts or contracts
Introduces PoS
Went live on Dec 01 2020

THE MERGE

- Aiming for Q2 2022
- Brings mainnet and the beacon chain together
- Mainnet will be a shard on the beacon chain

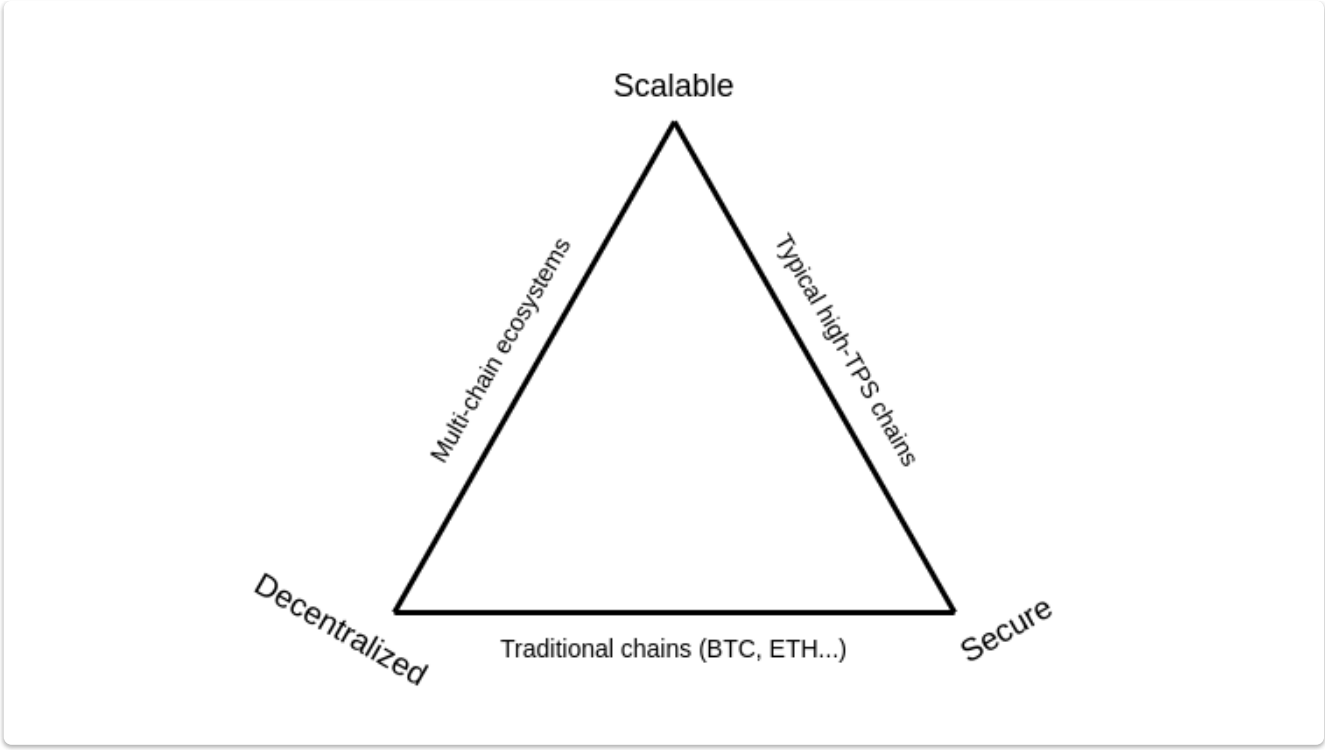
SHARD CHAINS

- Aiming for 2023
- Sharding is a good way to scale if you want to keep things decentralized as the alternative is to scale by increasing the size of the existing database.
- Will lower hardware requirements
- Initially shards will be there to provide data, they wont run contracts



Rollups

Why do we need rollups ?

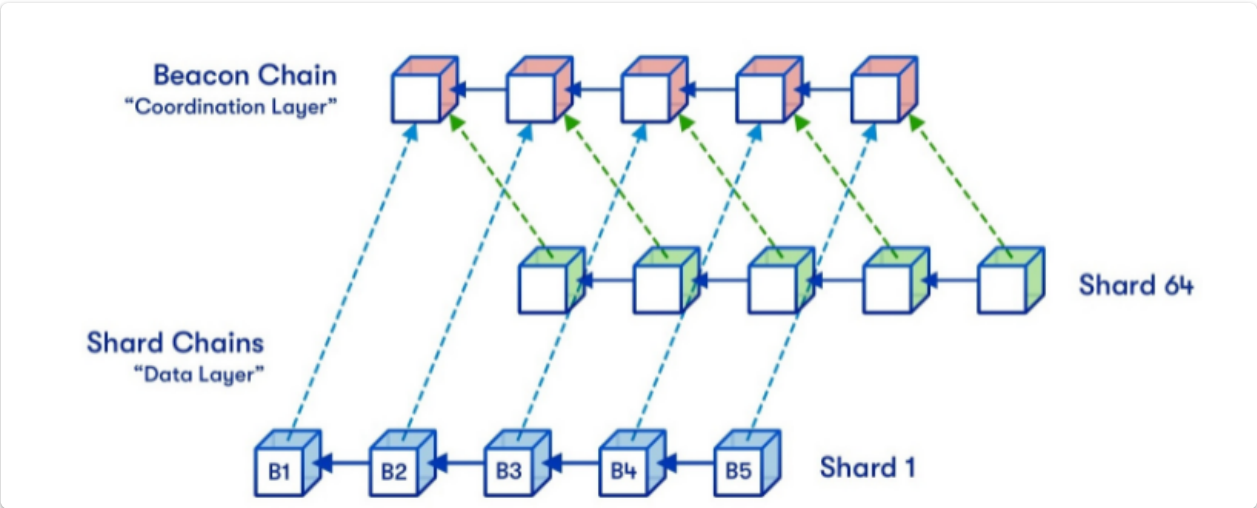


Approaches to increase scalability

ON CHAIN (L1) SOLUTIONS

- Consensus mechanism
  Using DPoS - EOS
  PoH - Solana
  Snow etc. - Avalanche
  For example moving from Proof of Work to Proof of Stake - Ethereum
  The Beacon chain is already live, in 2022 there will be the merge of main net and the beacon chain.

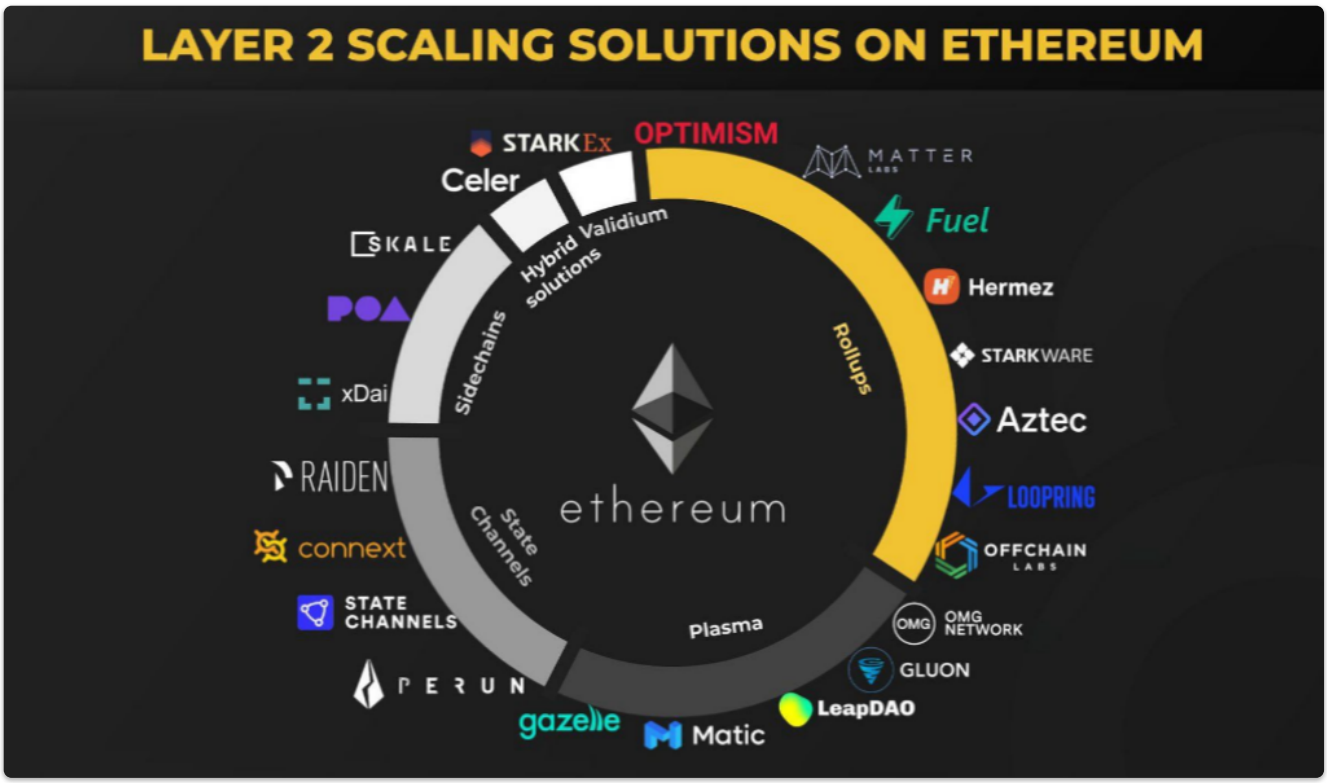
- Sharding

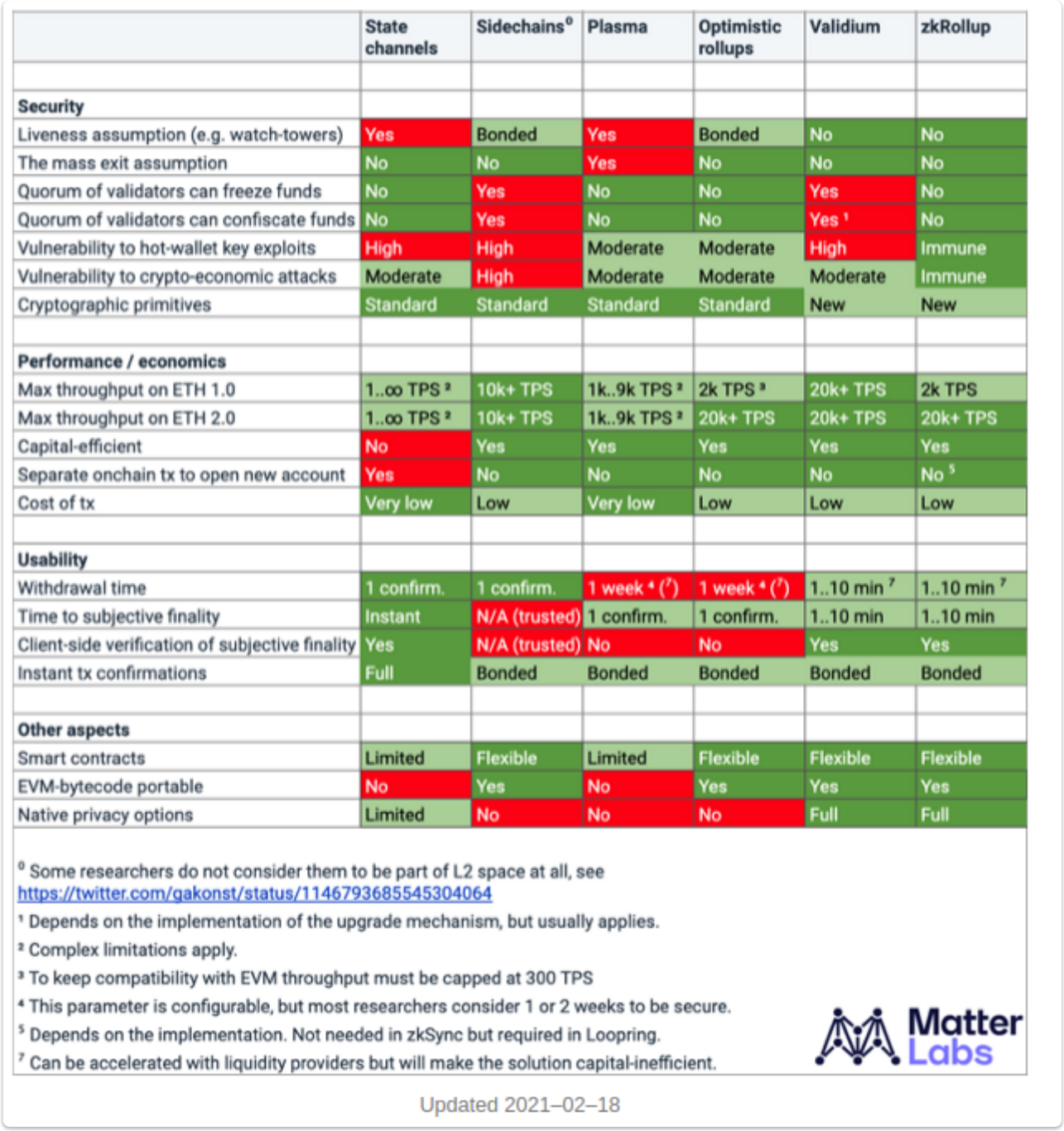


OFF CHAIN SCALING (LAYER 2)

In essence transactions are submitted to these layer 2 nodes instead of being submitted directly to layer 1 (Mainnet).

For some solutions the layer 2 instance then batches them into groups before anchoring them to layer 1, after which they are secured by layer 1 and cannot be altered.
A specific layer 2 instance may be open and shared by many applications, or may be deployed by one project and dedicated to supporting only their application.





	State channels	Sidechains[0]	Plasma	Optimistic rollups	Validium	zkRollup
Security						
Liveness assumption (e.g. watch-towers)	Yes	Bonded	Yes	Bonded	No	No
The mass exit assumption	No	No	Yes	No	No	No
Quorum of validators can freeze funds	No	Yes	No	No	Yes	No
Quorum of validators can confiscate funds	No	Yes	No	No	Yes [1]	No
Vulnerability to hot-wallet key exploits	High	High	Moderate	Moderate	High	Immune
Vulnerability to crypto-economic attacks	Moderate	High	Moderate	Moderate	Moderate	Immune
Cryptographic primitives	Standard	Standard	Standard	Standard	New	New
Performance / economics						
Max throughput on ETH 1.0	1..∞ TPS [2]	10k+ TPS	1k..9k TPS [2]	2k TPS [3]	20k+ TPS	2k TPS
Max throughput on ETH 2.0	1..∞ TPS [2]	10k+ TPS	1k..9k TPS [2]	20k+ TPS	20k+ TPS	20k+ TPS
Capital-efficient	No	Yes	Yes	Yes	Yes	Yes
Separate onchain tx to open new account	Yes	No	No	No	No	No [5]
Cost of tx	Very low	Low	Very low	Low	Low	Low
Usability						
Withdrawal time	1 confirm.	1 confirm.	1 week [4] ([7])	1 week [4] ([7])	1..10 min [7]	1..10 min [7]
Time to subjective finality	Instant	N/A (trusted)	1 confirm.	1 confirm.	1..10 min	1..10 min
Client-side verification of subjective finality	Yes	N/A (trusted)	No	No	Yes	Yes
Instant tx confirmations	Full	Bonded	Bonded	Bonded	Bonded	Bonded
Other aspects						
Smart contracts	Limited	Flexible	Limited	Flexible	Flexible	Flexible
EVM-bytecode portable	No	Yes	No	Yes	Yes	Yes
Native privacy options	Limited	No	No	No	Full	Full

[0] Some researchers do not consider them to be part of L2 space at all, see https://twitter.com/gakonst/status/1146793685545304064

[1] Depends on the implementation of the upgrade mechanism, but usually applies.

[2] Complex limitations apply.

[3] To keep compatibility with EVM throughput must be capped at 300 TPS

[4] This parameter is configurable, but most researchers consider 1 or 2 weeks to be secure.

[5] Depends on the implementation. Not needed in zkSync but required in Loopring.

[7] Can be accelerated with liquidity providers but will make the solution capital-inefficient.

Matter Labs

Updated 2021–02–18

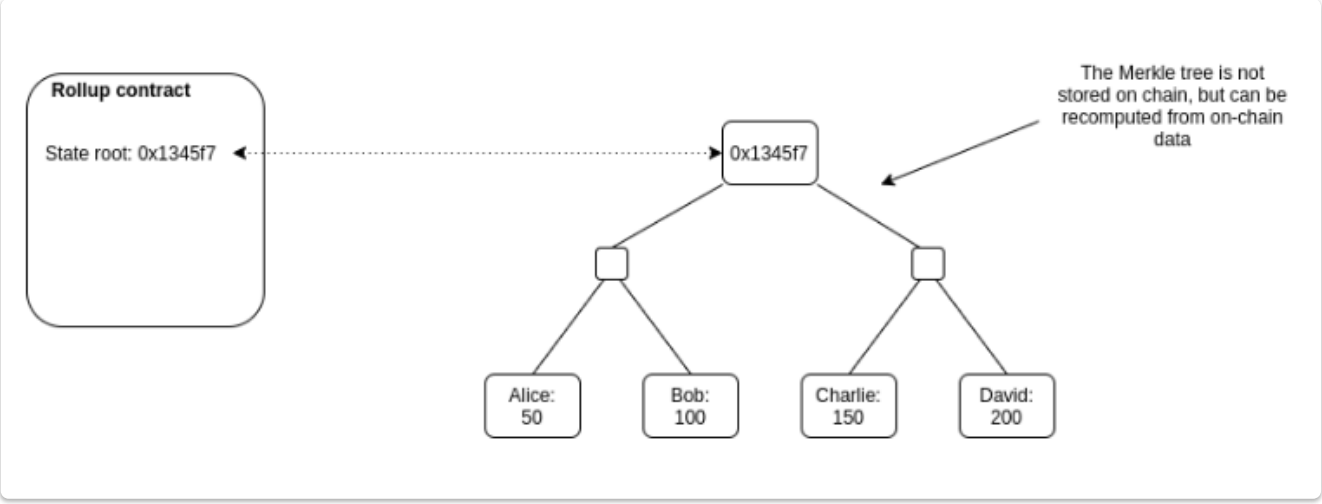
Rollups

Rollups are solutions that have

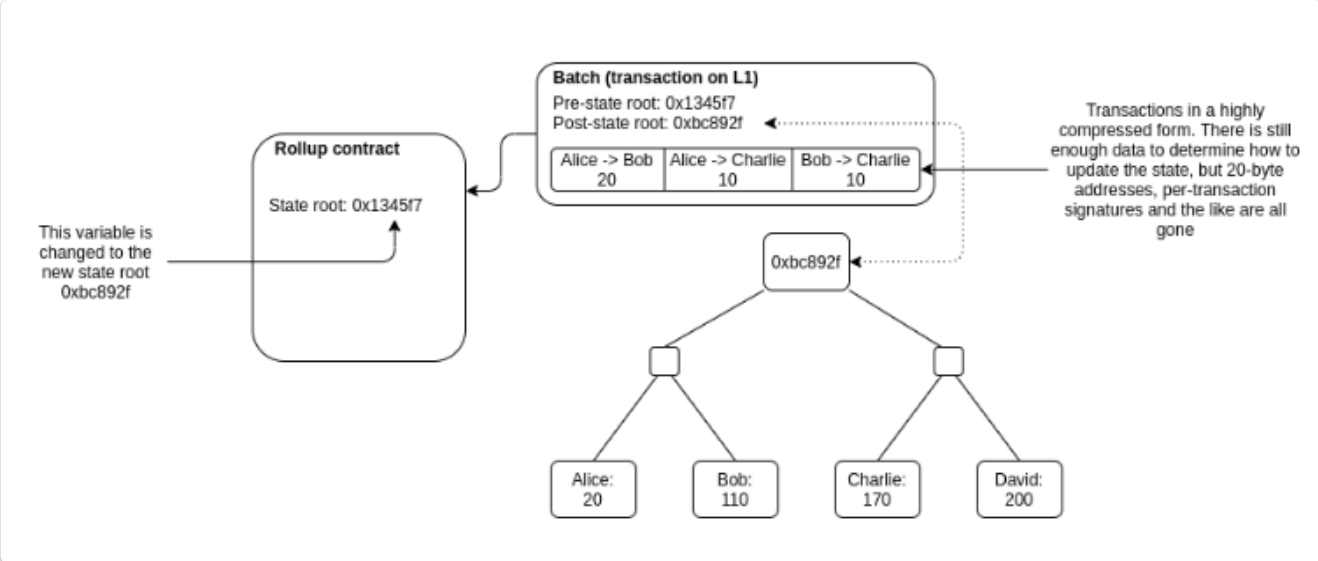
- transaction execution outside layer 1
- transaction data and proof of transactions is on layer 1
- a rollup smart contract in layer 1 that can enforce correct transaction execution on layer 2 by using the transaction data on layer 1

The main chain holds funds and commitments to the side chains
The side chain holds additional state and performs execution
There needs to be some proof, either a fraud proof (Optimistic) or a validity proof (zk)

Rollups require “operators” to stake a bond in the rollup contract. This incentivises operators to verify and execute transactions correctly.



Anyone can publish a batch, a collection of transactions in a highly compressed form together with the previous state root and the new state root (the Merkle root after processing the transactions). The contract checks that the previous state root in the batch matches its current state root; if it does, it switches the state root to the new state root.



There are currently 2 types of rollups

- Zero Knowledge Proof rollups
- Optimistic rollups

OPTIMISTIC ROLLUPS

The name Optimistic Rollups originates from how the solution works. 'Optimistic' is used because aggregators publish only the bare minimum information needed with no proofs, assuming the aggregators run without commiting frauds, and only providing proofs in case of fraud.
For more details see Arbitrum research

ZKP Rollups

An excellent report

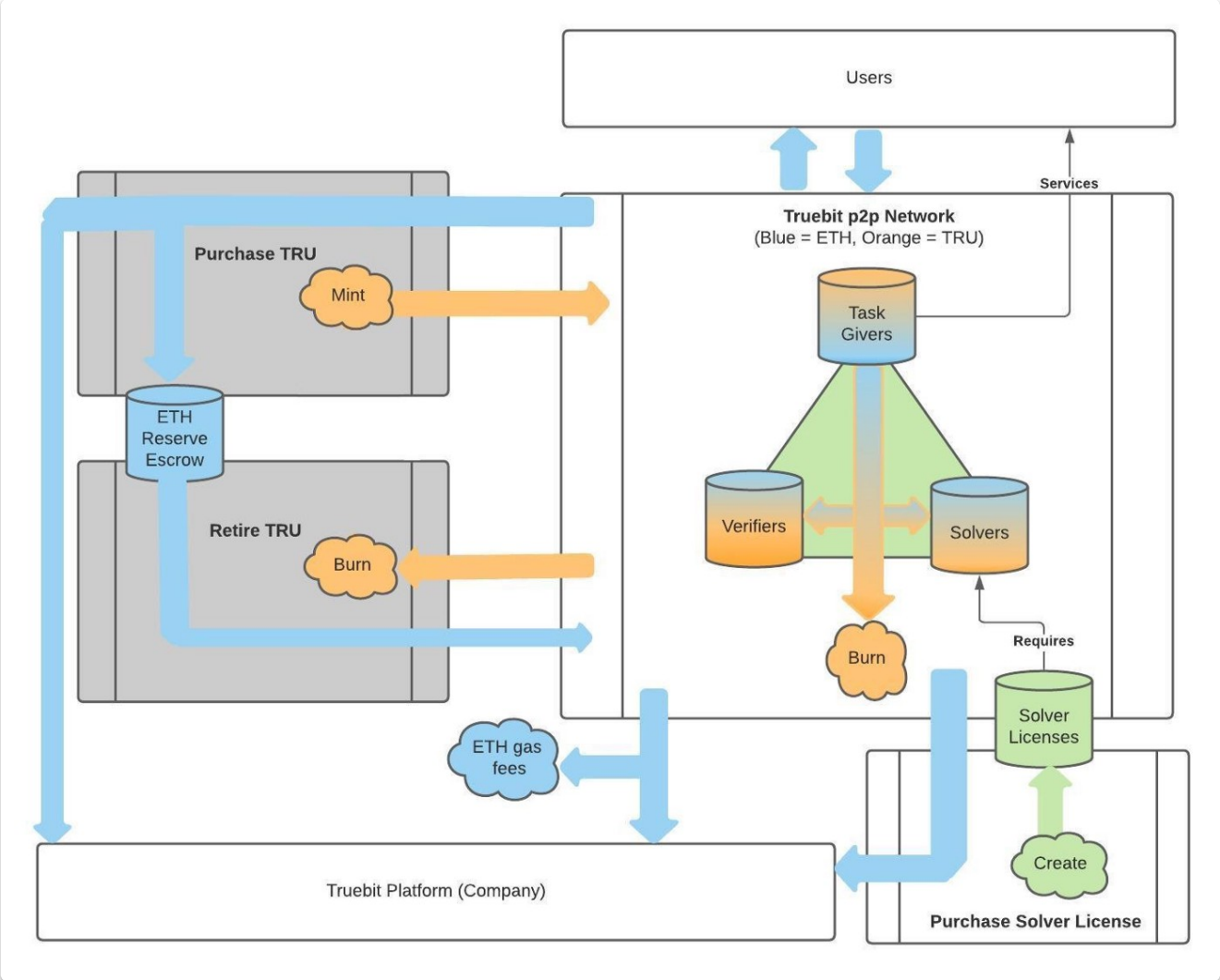
An overview from Ethereum
The ZK-Rollup scheme consists of two types of users: transactors and relayers.

- Transactors create their transfer and broadcast the transfer to the network. The transfer data consists of an indexed “to” and “from” address, a value to transact, the network fee, and nonce. A shortened 3 byte indexed version of the addresses reduces processing resource needs. The value of the transaction being greater than or less than zero creates a deposit or withdrawal respectively. The smart contract records the data in two Merkle Trees; addresses in one Merkle Tree and transfer amounts in another.
- Relayers collect a large amount of transfers to create a rollup. It is the relayers job to generate the SNARK proof. The SNARK proof is a hash that represents the delta of the blockchain state. State refers to “state of being.” SNARK proof compares a snapshot of the blockchain before the transfers to a snapshot of the blockchain after the transfers (i.e. wallet values) and reports only the changes in a verifiable hash to the mainnet.

It is worth noting that anyone can become a relayer so long as they have staked the required bond in the smart contract. This incentivises the relayer not to tamper with or withhold a rollup.

Truebit and Cartesi

Optimistic rollups with truebit
Truebit [Introduction] (https://medium.com/truebit/getting-started-with-truebit-on-ethereum-ac1c7cdb0907)
The Truebit microeconomy facilitates verifiable computations.



Also see https://cartesi.io/

	Ethereum [1]	Cartesi Rollups
Runtime environment	EVM	Linux VM
Languages	Solidity	All [2]
Development tools	Nascent / Immature toolchain	All [2]
Smart contact privacy	None (very expensive)	Possible (no extra cost)
Computational Throughput	x	> 10,000x (no fees)
Transaction Throughput	x	x; > 15x; > 40x [3]
Fee costs per tx	x	x; > x/15; x/40; [3 4]
Security	L1 consensus	Similar to L1 consensus

Danksharding

See article
A combination of Proposer / Builder Separation and crList
Rollup-centric projects in general would be set to benefit the most from the implementation of danksharding, with a potential ~1000x drop in fees.

PROPOSER / BUILDER SEPARATION

https://notes.ethereum.org/s3JToeApTx6CKLJt8AbhFQ#Hybrid-PBS-can-we-use-proposers-only-for-inclusion-of-last-resort

In the current transaction market, the **block proposer** (today: a miner, post-merge: a validator) directly chooses which transactions to include in the next block by looking at which transactions in the mempool pay the highest priority fee. This puts the block proposer in a position to use sophisticated strategies to choose which transactions to include, or even include their own, to take advantage of opportunities such as DEX arbitrage and liquidations (hereinafter just called "**MEV**" for simplicity) to maximize their profits. The complexity of these strategies creates a high fixed cost in running an effective miner or validator, and advantages centralized pools that take on this task on behalf of their participants.

Proposer/builder separation (PBS) fixes this by splitting the block construction role from the block proposal role. A separate class of actors called **builders** build **exec block bodies** (essentially an ordered list of transactions that becomes the main "payload" of the block), and submit bids. The proposer's job is only to accept the exec block body with the highest bid. Notably, the proposer (and everyone else) does not learn the contents of any exec block body until *after* they select the header (and hence the body) that wins the auction. This **pre-confirmation privacy** is needed to prevent "**MEV stealing**", where sophisticated proposers detect builders' MEV extraction strategies and copy them without compensating the builder.

PBS is an actively evolving research area. Mitigating validator centralization risks that come from MEV is very important for any blockchain protocol that seeks to retain its decentralization. However, preventing builder centralization from turning into a different type of censorship vulnerability is equally important.

1. **Proposer broadcasts** `crList`, a list of transactions that the proposer sees that deserve to be included (ie. they have the correct nonce and sufficient balance in the state and sufficient priority fee and max-basefee to be included). The `crList` can only contain one transaction per `sender`. The proposer also signs and broadcasts a `crListSummary`, which includes the `tx.sender` and `tx.gaslimit` for every `tx` in the `crList`.
2. **Builder makes a proposed body**, which must include the `crListSummary`
3. **Proposer accepts winning header**
4. **Builder publishes body**. Verification of the body requires checking that for each `(sender, gaslimit)` in `crListSummary`, either `block.gasused + gaslimit > block.gaslimit` or `sender` is the sender of some transaction in the block

Furthermore
We are moving towards 3 layers

- Execution layer

- Settlement Layer
- Consensus and Data availability layer

Stateless Ethereum

The Ethereum world state contains all Ethereum accounts, their balances, deployed smart contracts, and associated storage, it grows without bound.
The aim of Stateless Ethereum is to mitigate unbounded state growth.
The way to achieve this, is to give the responsibility for storing state to other participanst on the network.
Block producers will in addition to the block provide a 'witness' that is the data required to execute the transactions in the block.
There could be policies for state storage

- Any new state trie object that gets created or touched gets by default stored by all full nodes for 3 months. This will likely be around 2.5 GB.
- Clients that wish to ensure availability of specific pieces of data much longer can do so with payments in state channels. A client can set up channels with paid archival nodes.

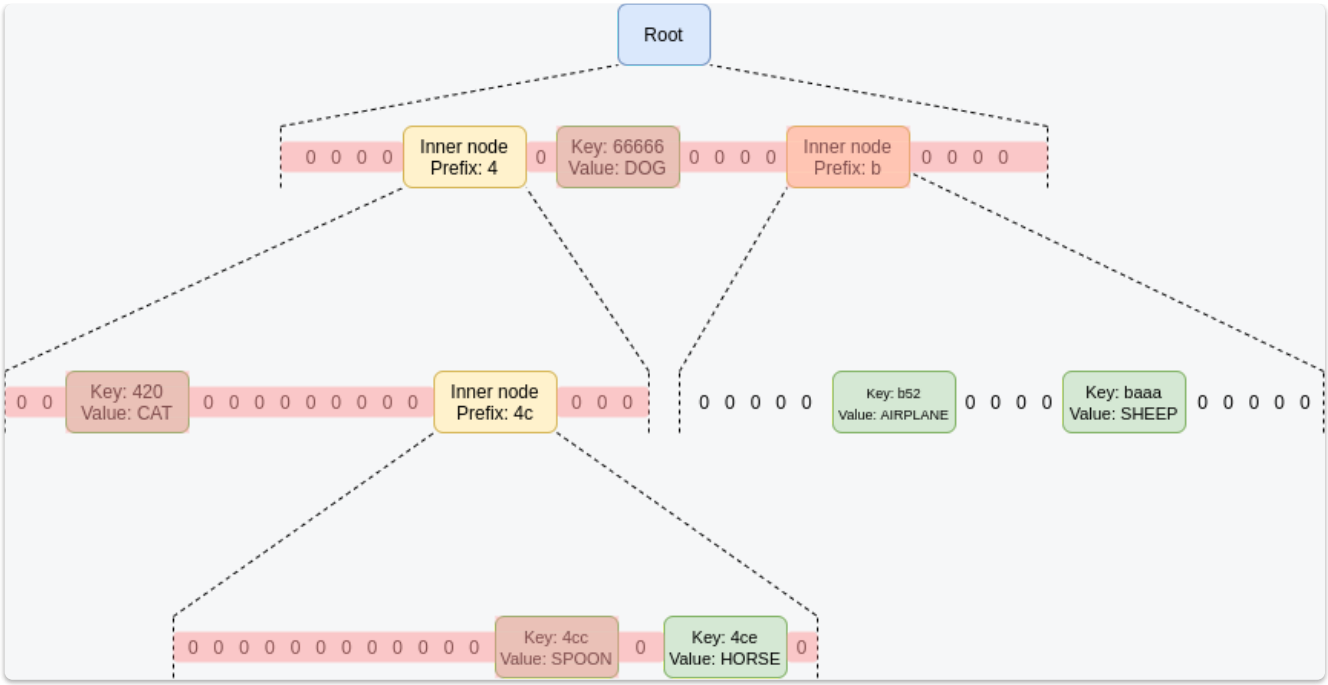
Verkle trees

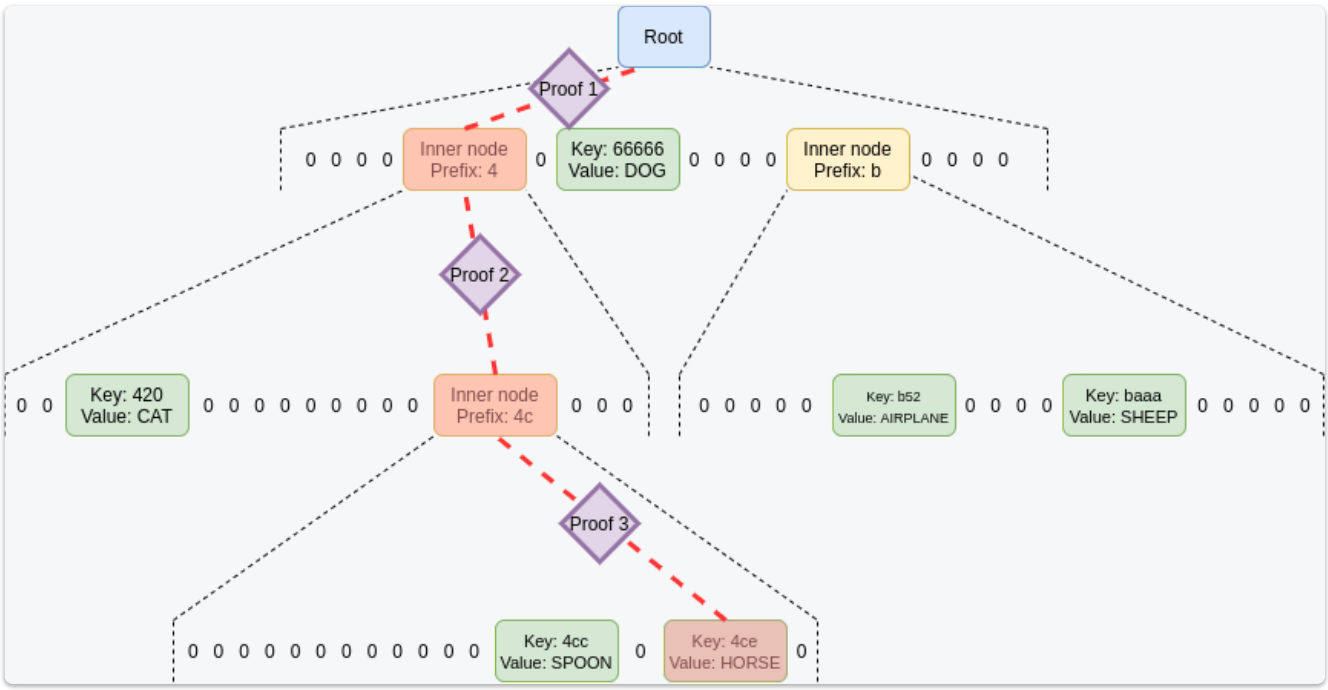
https://vitalik.ca/general/2021/06/18/verkle.html

See article
and Ethereum Cat Herders Videos

Like merkle trees, you can put a large amount of data into a Verkle tree, and make a short proof ("witness") of any single piece, or set of pieces, of that data that can be verified by someone who only has the root of the tree.
Verkle trees provide, however, is that they are much more efficient in proof size. If a tree contains a billion pieces of data, making a proof in a traditional binary Merkle tree would require about 1 kilobyte, but in a Verkle tree the proof would be less than 150 bytes.
Verkle trees replace hash commitments with vector commitments or better still a polynomial commitment.
Polynomial commitments give us more flexibility that lets us improve efficiency, and the simplest and most efficient vector commitments available are polynomial commitments.

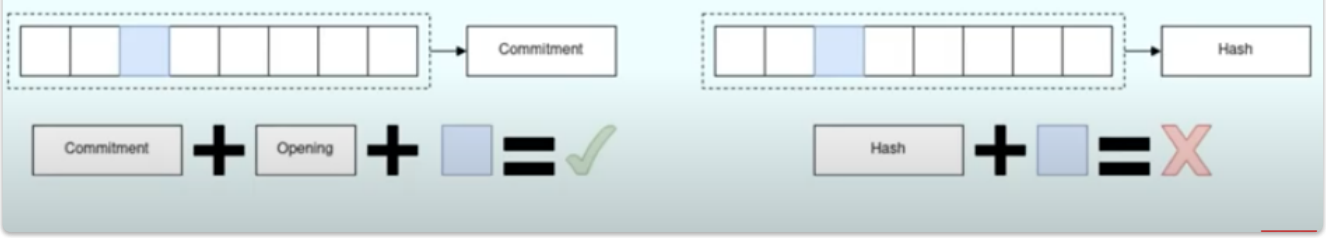
The number of nodes needed in a merkle proof is much greater than in a verkle proof

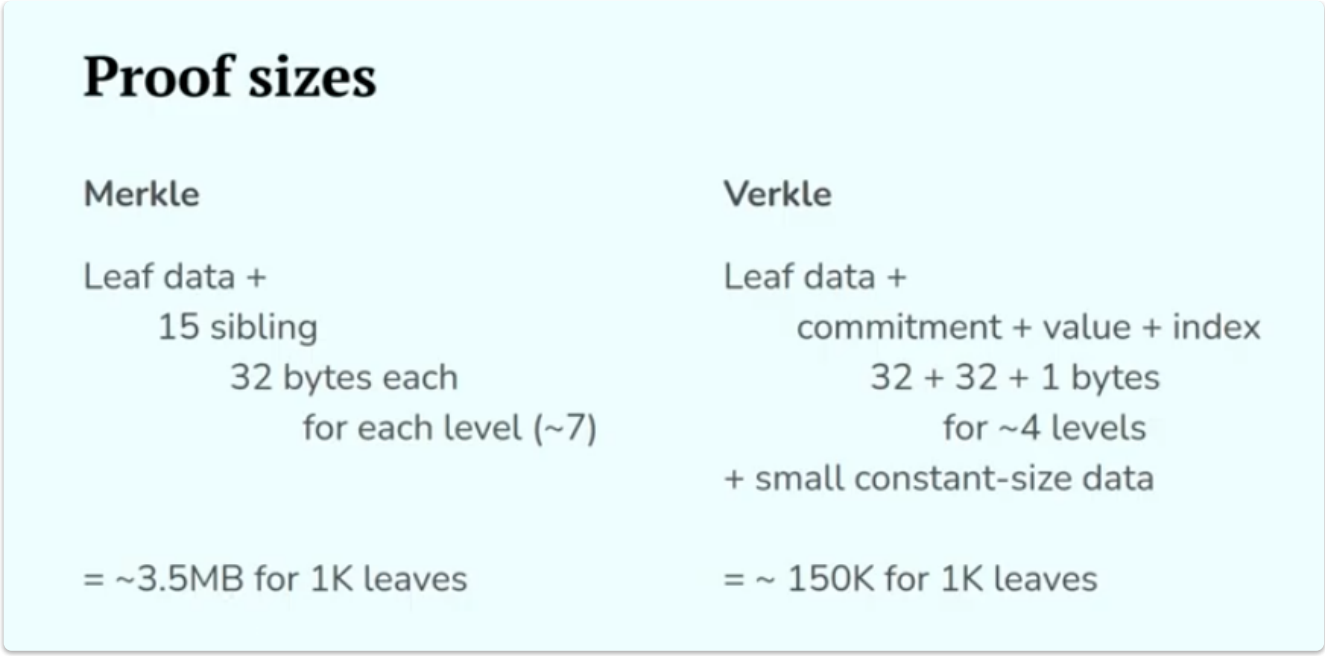




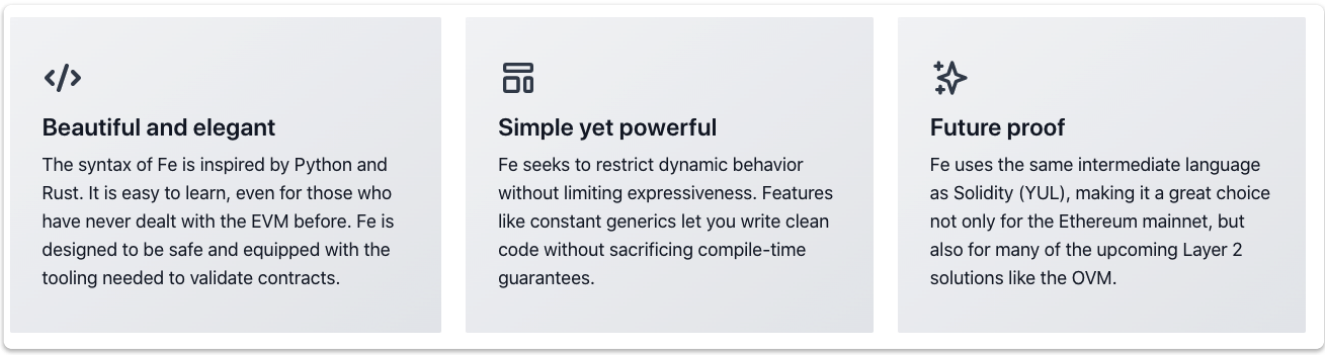
Vector commitments vs. Hash

- Vector commitments: existence of an “opening”, a small payload that allow for the verification of a portion of the source data without revealing it all.
- Hash : verifying a portion of the data = revealing the whole data.





Fe Language



ERC20 Example contract

Upcoming iterms in Solidity

- Event definition at file level
- Modifiers jump rather than in lining
- Optimisations
- Try catch for custom errors
- Have selectors for events and errors
- Immutable reference types
- Generics