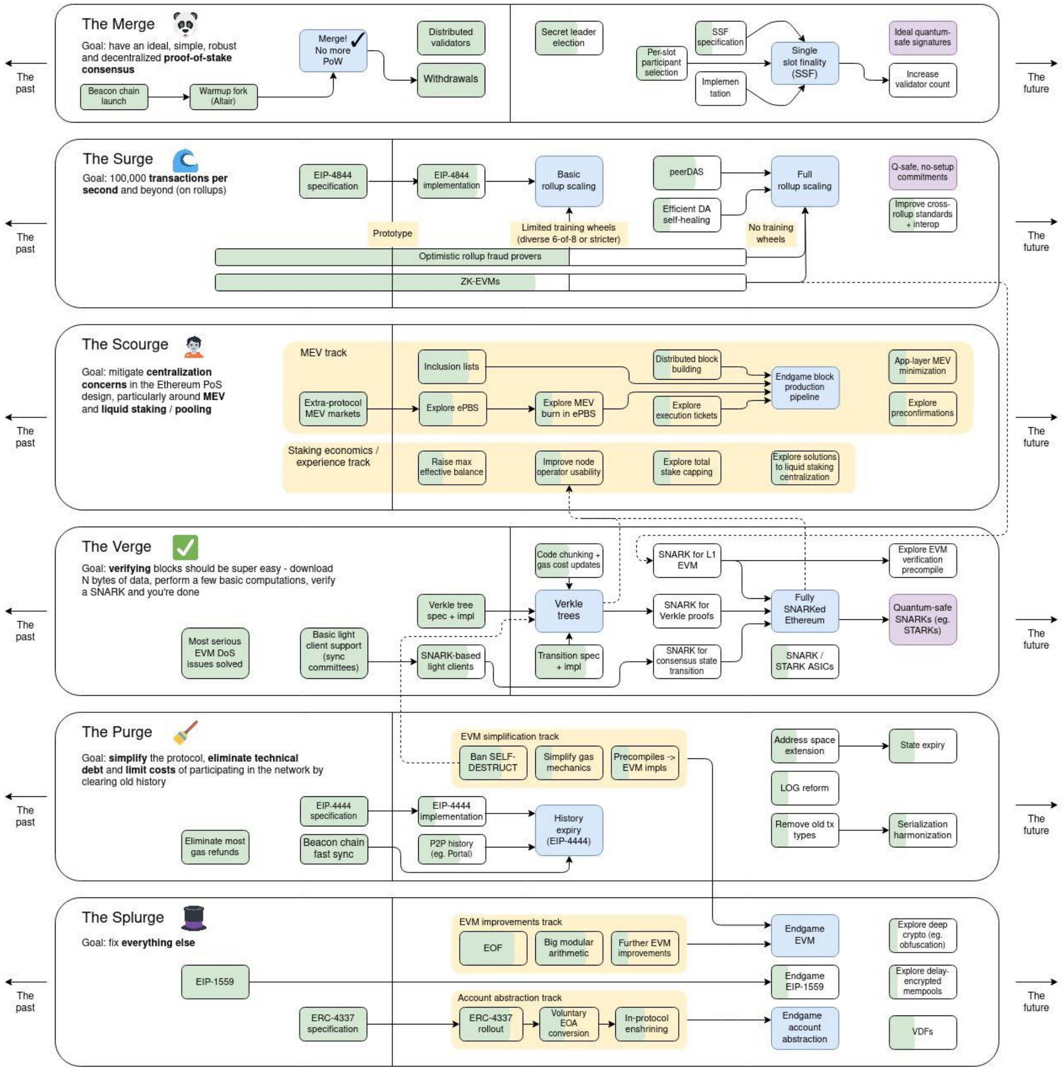
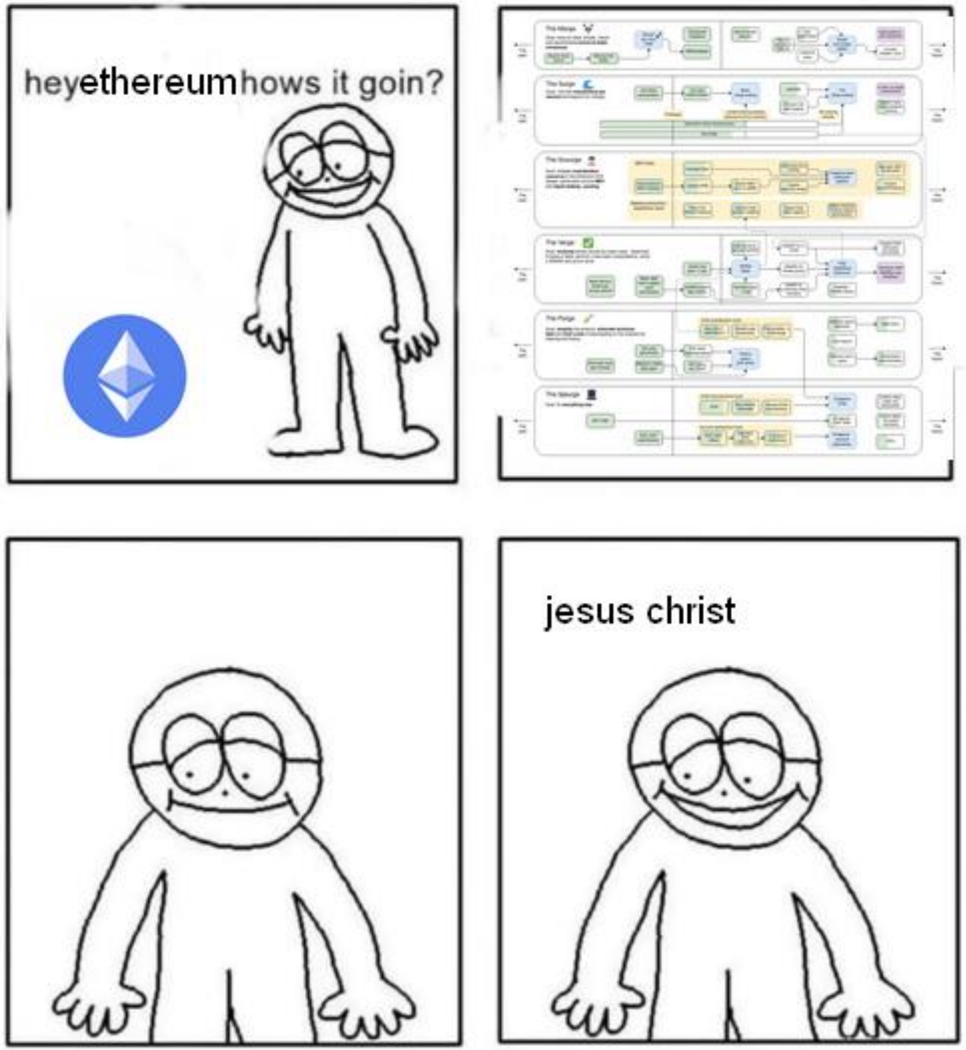
# Ethereum Roadmap

By domothy
Twitter & Telegram: @domothy
domothy@ethereum.org


The Merge
Goal: have an ideal, simple, robust and decentralized proof-of-stake consensus
Beacon chain launch
Warmup fork (Altair)
Merge! No more PoW ✓
Distributed validators
Withdrawals
Secret leader election
SSF specification
Per-slot participant selection
Implementation
Single slot finality (SSF)
Ideal quantum-safe signatures
Increase validator count
The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs
The Scourge
Goal: mitigate centralization concerns in the Ethereum PoS design, particularly around MEV and liquid staking / pooling
MEV track
Inclusion lists
Extra-protocol MEV markets
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
App-layer MEV minimization
Explore preconfirmations
Staking economics / experience track
Raise max effective balance
Improve node operator usability
Explore total stake capping
Explore solutions to liquid staking centralization
The Verge
Goal: verifying blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)
The Purge
Goal: simplify the protocol, eliminate technical debt and limit costs of participating in the network by clearing old history
EVM simplification track
Ban SELF-DESTRUCT
Simplify gas mechanics
Precompiles -> EVM impls
Eliminate most gas refunds
EIP-4444 specification
Beacon chain fast sync
EIP-4444 implementation
P2P history (eg. Portal)
History expiry (EIP-4444)
Address space extension
State expiry
LOG reform
Remove old tx types
Serialization harmonization
The Splurge
Goal: fix everything else
EVM improvements track
EOF
Big modular arithmetic
Further EVM improvements
EIP-1559
ERC-4337 specification
Account abstraction track
ERC-4337 rollout
Voluntary EOA conversion
In-protocol enshrining
Endgame EVM
Endgame EIP-1559
Endgame account abstraction
Explore deep crypto (eg. obfuscation)
Explore delay-encrypted mempools
VDFs
The past
The future

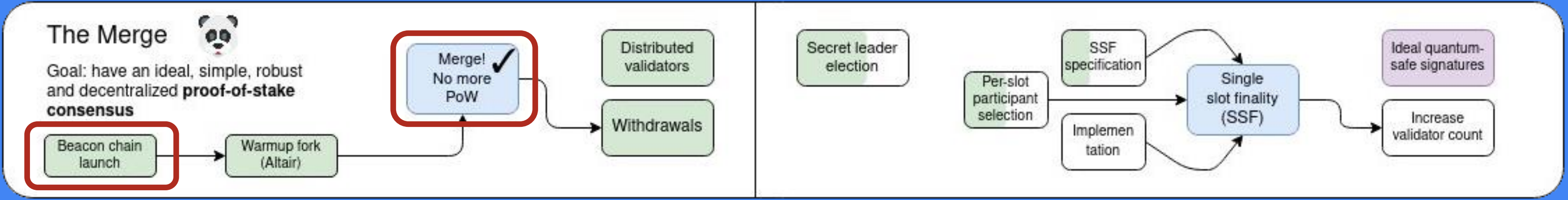

heyethereumhows it goin?
The Merge
The Surge
The Scourge
The Verge
The Purge
The Splurge
jesus christ

## Roadmap tl;dr

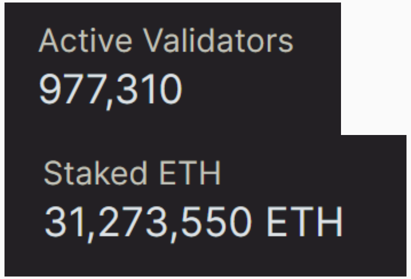
- Merge: Better Proof of Stake
- Surge: More data (availability) for rollups
- Scourge: Less MEV downsides
- Verge: Easier verification
- Purge: Simpler protocol
- Splurge: Miscellaneous goodies

## Roadmap tl;dr

- **Merge: Better Proof of Stake**
- Surge: More data (availability) for rollups
- Scourge: Less MEV downsides
- Verge: Easier verification
- Purge: Simpler protocol
- Splurge: Miscellaneous goodies

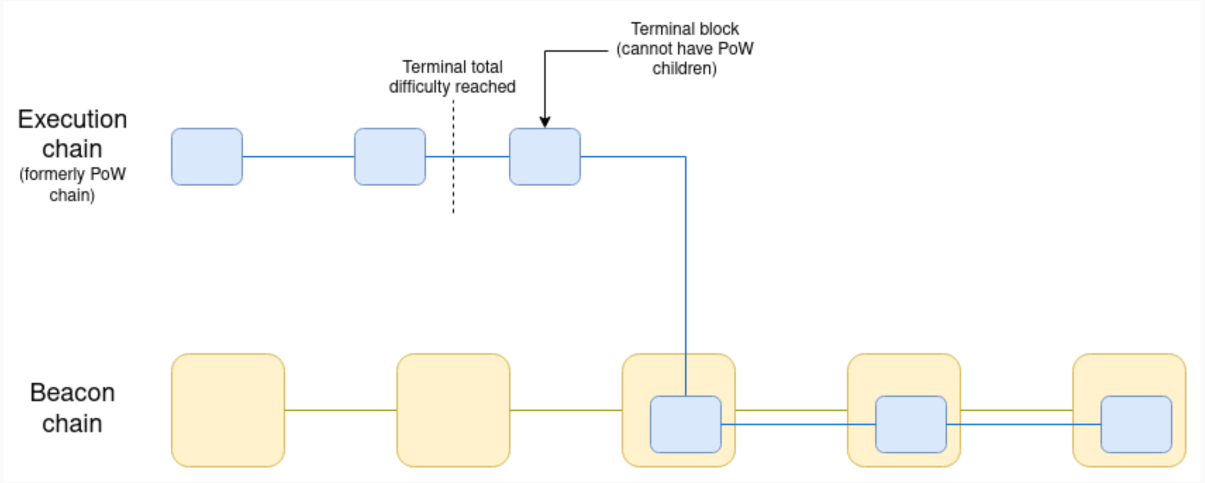


## Beacon Chain

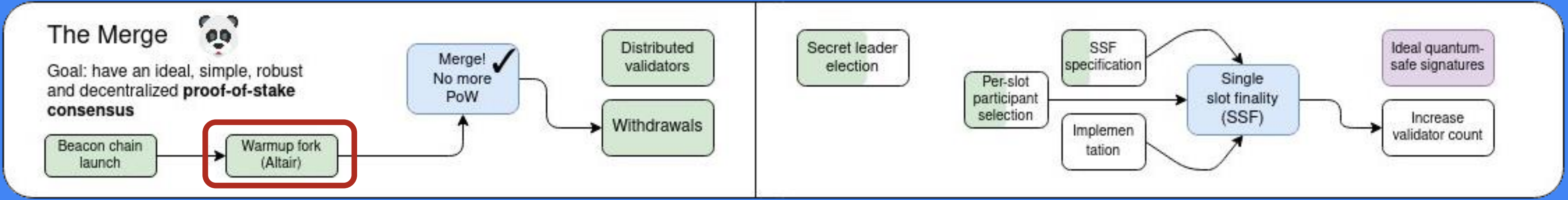

Active Validators
977,310

Staked ETH
31,273,550 ETH

(~$110B)


Terminal block (cannot have PoW children)
Terminal total difficulty reached
Execution chain (formerly PoW chain)
Beacon chain

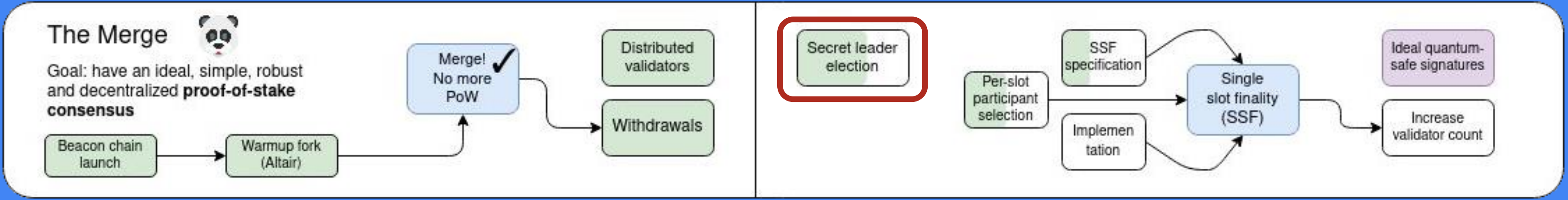
*https://github.com/ethereum/annotated-spec/blob/master/merge/beacon-chain.md*



## Sync committee / Light client protocol

- 512 validators, rotated every 256 epochs (~27 hours)
- Light-weight; 512 signatures to check vs. ~1M
- Trust-*minimized* rather than *trustless*

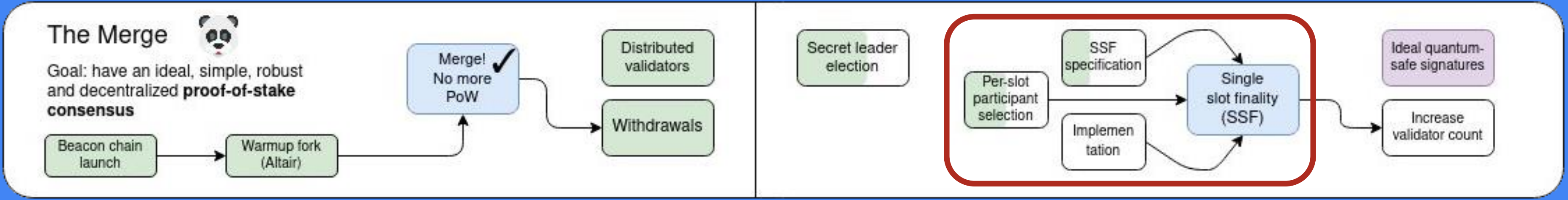
See a16z's Helios client
*https://a16zcrypto.com/posts/article/building-helios-ethereum-light-client/*



## Secret Leader Election

- Currently, leader/proposer is revealed a bit ahead of time
- Protection against Denial of Service attacks
- Low priority (unless…)

**EIP-7441** – Whisk shuffling

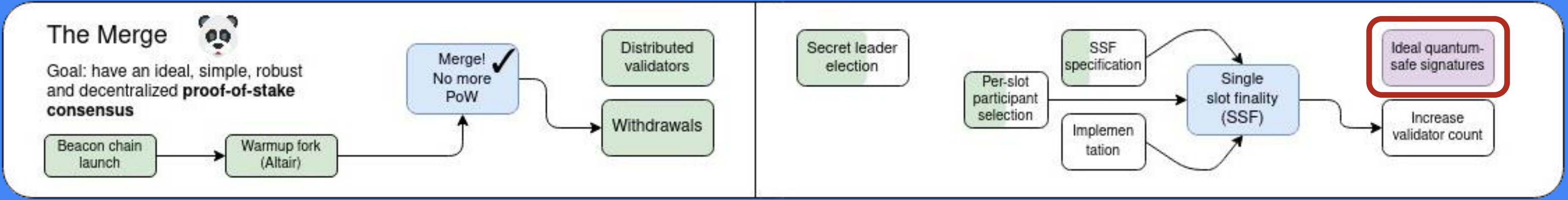


## Single Slot Finality

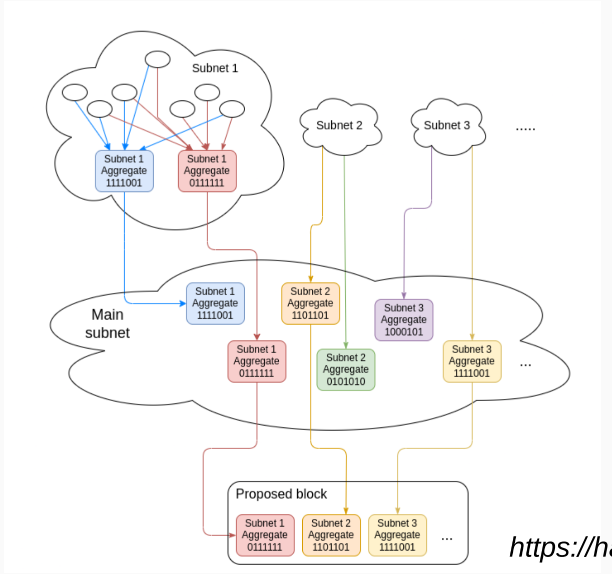
- From 12.6 minutes to 12 seconds
- Main problem: Too many signatures to check and aggregate

Solution paths:

- Fewer validators (MaxEB)
- Fewer *active* validators
- Way fewer validators (8192) + Distributed Validators Tech
- Better signature aggregation schemes



## Quantum-proof Beacon Chain

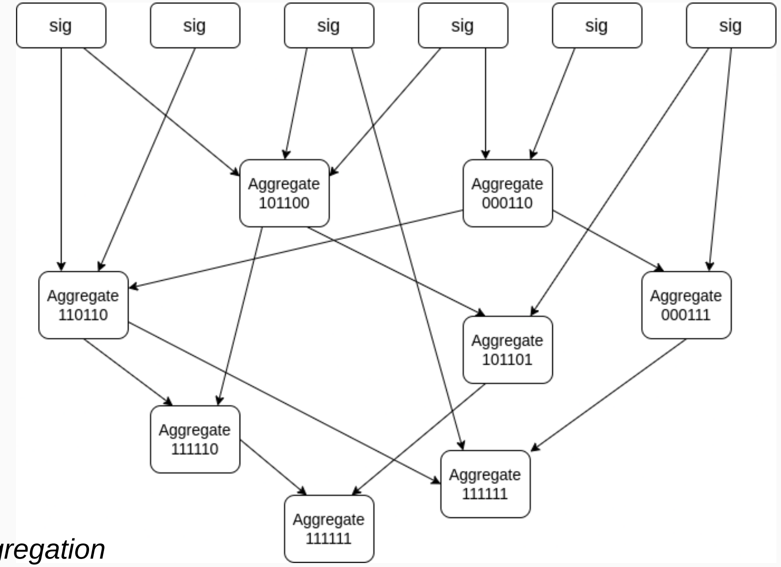

Subnet 1
Subnet 1 Aggregate 1111001
Subnet 1 Aggregate 0111111
Subnet 2
Subnet 3
.....
Main subnet
Subnet 1 Aggregate 1111001
Subnet 2 Aggregate 1101101
Subnet 3 Aggregate 1000101
Subnet 1 Aggregate 0111111
Subnet 2 Aggregate 0101010
Subnet 3 Aggregate 1111001
...
Proposed block
Subnet 1 Aggregate 0111111
Subnet 2 Aggregate 1101101
Subnet 3 Aggregate 1111001
...

**Solution**

STARKs

(and STARKs of STARKs)

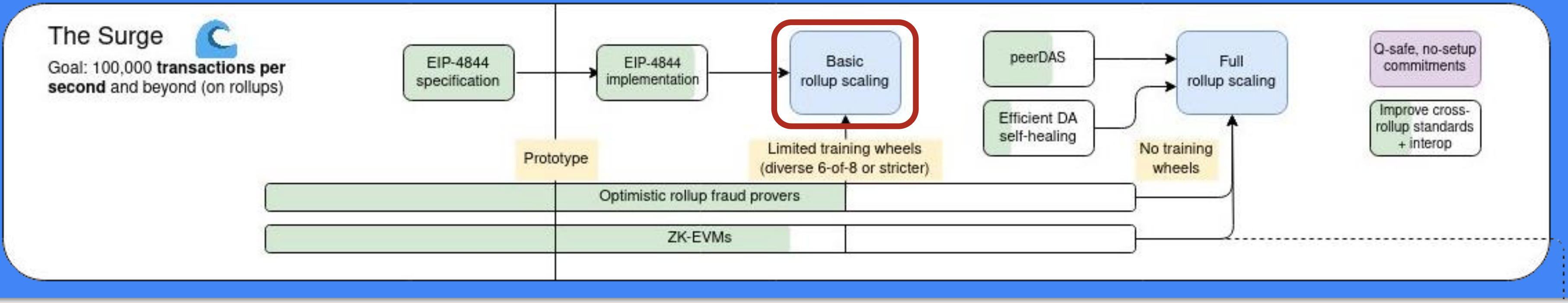
((and STARKs of STARKs of STARKs))



*https://hackmd.io/@vbuterin/stark_aggregation*

## Roadmap tl;dr

- Merge: Better Proof of Stake
- **Surge: More data (availability) for rollups**
- Scourge: Less MEV downsides
- Verge: Easier verification
- Purge: Simpler protocol
- Splurge: Miscellaneous goodies


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Full rollup scaling
Q-safe, no-setup commitments
Efficient DA self-healing
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

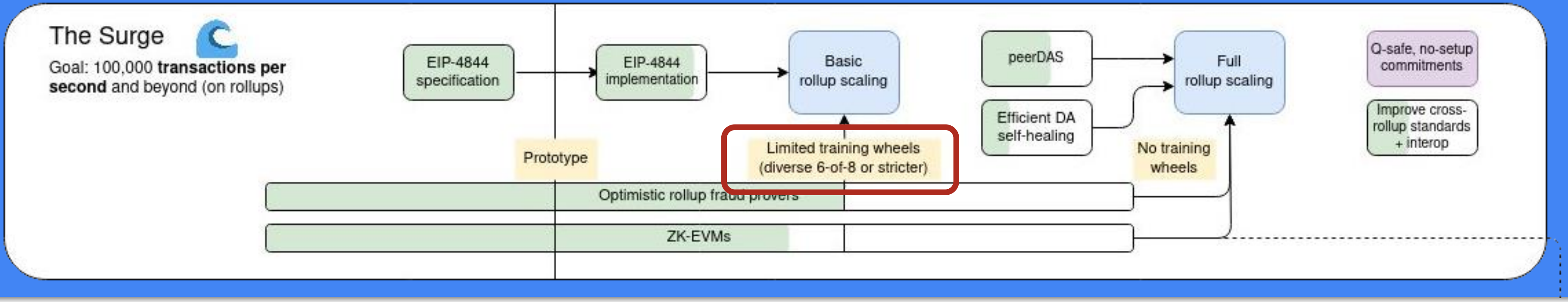
## Rollups from 10,000 feet

- Safely scaling L1 execution is hard
- But safely scaling L1 data is easy(-er)
- Rollups convert L1 data into L2 execution – *with 1-of-N trust assumption!*
- **Rollup-centric roadmap**

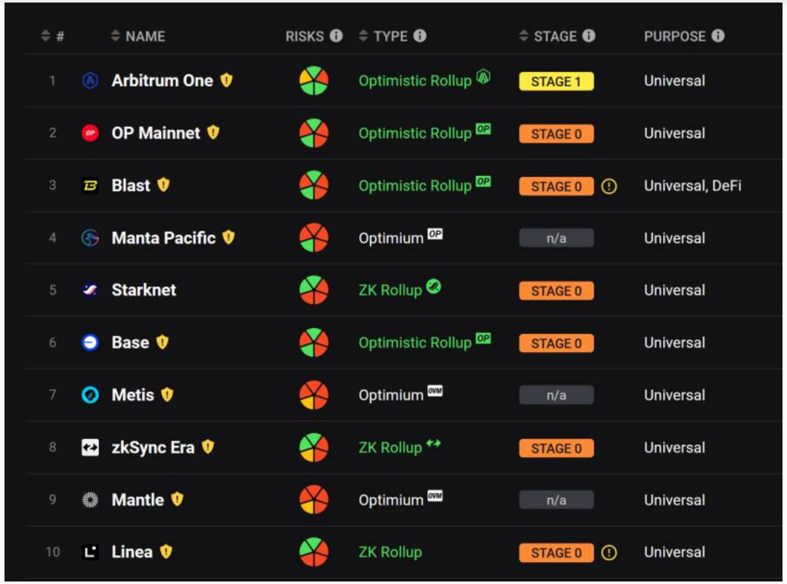
*Optimistic Rollups*
- Assume all transactions are valid
- Slash sequencer if not (fraud proofs)

*Zero-Knowledge Rollups*
- Sequencer proves transactions are valid
- Short proofs verified by L1

All rollup data must be **available** on Layer 1 (for those who *need/want* it)
- Force L2 transaction inclusion (i.e. to exit back to L1)

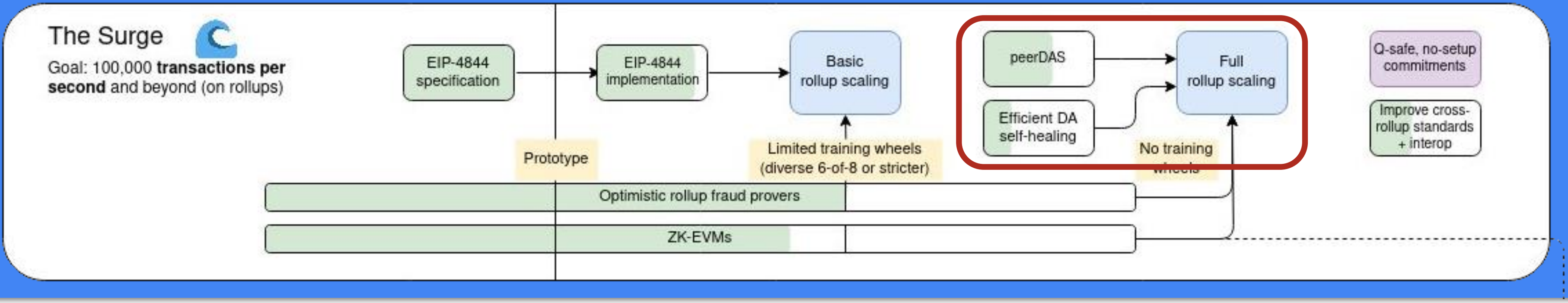

The Surge
Goal: 100,000 **transactions per second** and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Full rollup scaling
Q-safe, no-setup commitments
Efficient DA self-healing
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

- Upgradability / mutability
- Multisig / governance
- Permissioned elements

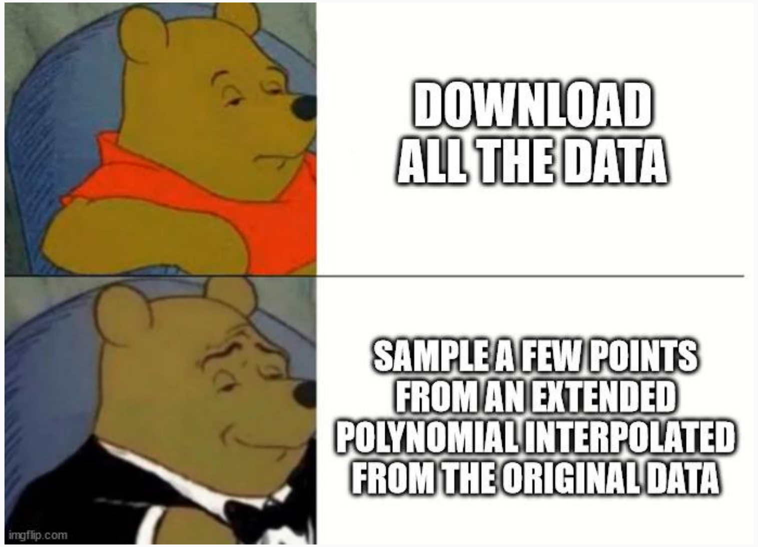


| # | NAME | RISKS | TYPE | STAGE | PURPOSE |
|---|---|---|---|---|---|
| 1 | Arbitrum One | ● | Optimistic Rollup | STAGE 1 | Universal |
| 2 | OP Mainnet | ● | Optimistic Rollup OP | STAGE 0 | Universal |
| 3 | Blast | ● | Optimistic Rollup OP | STAGE 0 | Universal, DeFi |
| 4 | Manta Pacific | ● | Optimium OP | n/a | Universal |
| 5 | Starknet | ● | ZK Rollup | STAGE 0 | Universal |
| 6 | Base | ● | Optimistic Rollup OP | STAGE 0 | Universal |
| 7 | Metis | ● | Optimium OVM | n/a | Universal |
| 8 | zkSync Era | ● | ZK Rollup | STAGE 0 | Universal |
| 9 | Mantle | ● | Optimium OVM | n/a | Universal |
| 10 | Linea | ● | ZK Rollup | STAGE 0 | Universal |

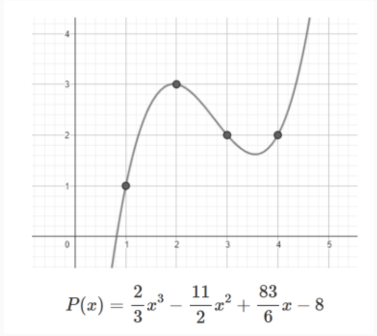
*L2beat.com*


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

## Data Availability Sampling / *"is the data available?"*


DOWNLOAD ALL THE DATA
SAMPLE A FEW POINTS FROM AN EXTENDED POLYNOMIAL INTERPOLATED FROM THE ORIGINAL DATA
imgflip.com

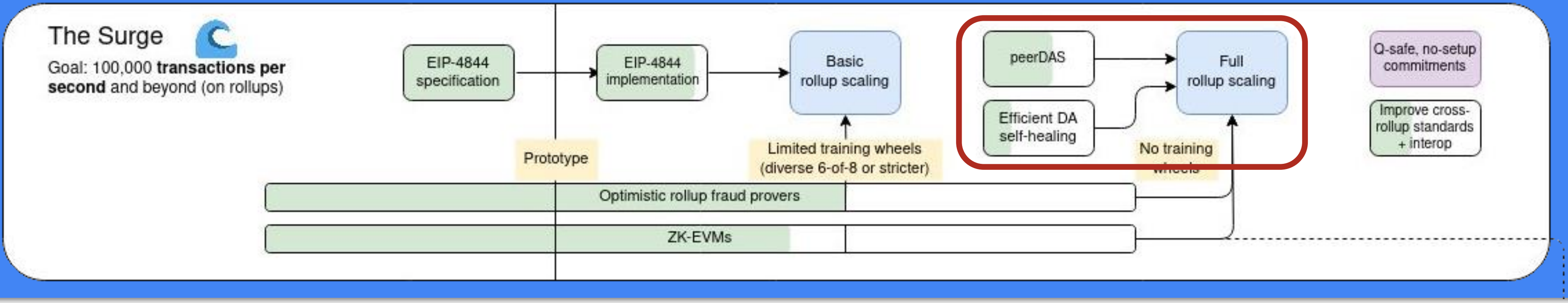
- Two points make a line
- Three points make a parabola
- 4096 points make a… 4095-degree polynomial



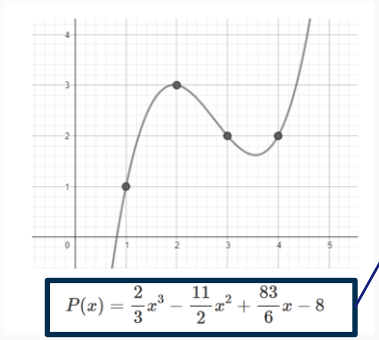
$$P(x) = \frac{2}{3}x^3 - \frac{11}{2}x^2 + \frac{83}{6}x - 8$$

Data: (1, 3, 2, 2)
Extension: (7, 21, 48, 92)

**50% of data + extension can recover 100% of data**


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

## Polynomial Commitment Schemes

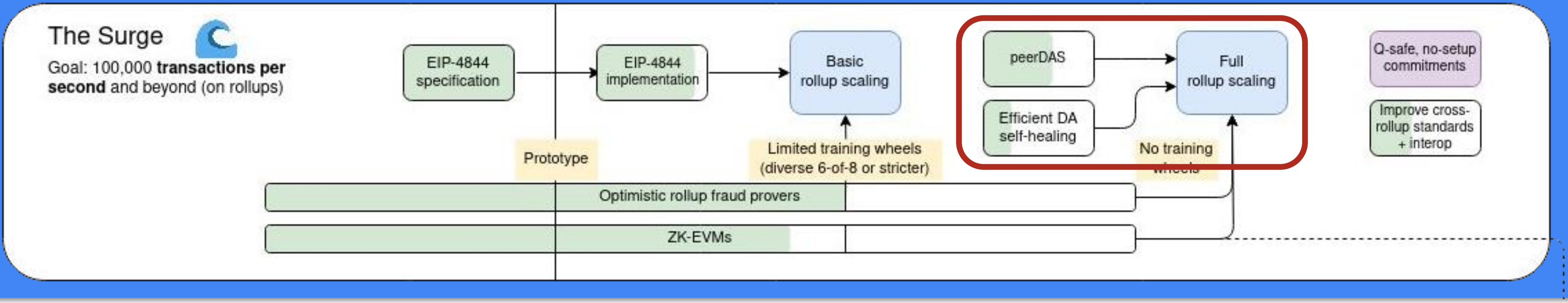


$$P(x) = \frac{2}{3}x^3 - \frac{11}{2}x^2 + \frac{83}{6}x - 8$$

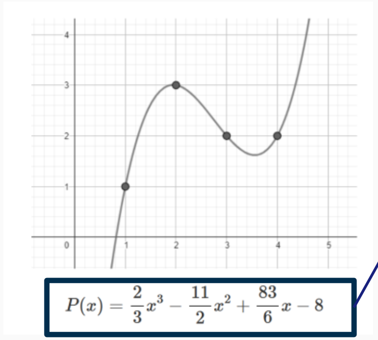
Data: (1, 3, 2, 2)
Extension: (7, 21, 48, 92)

*In practice, P(x) has thousands of coefficients*
C = commit(P) = a few bytes (*like a hash*) known to all nodes

- Ask for random data point (e.g. the 3rd one)
- Receive the value 2 along with proof π
- Verify proof π against C, is satisfied that *P(3) = 2*
- At most 50% odds of “being fooled”
- Ask for another random data point, odds become 25%
- Another sample: 12.5%
- 30 samples = 1 / $2^{30}$ = ~1 in a billion chance of being fooled


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

## Polynomial Commitment Schemes

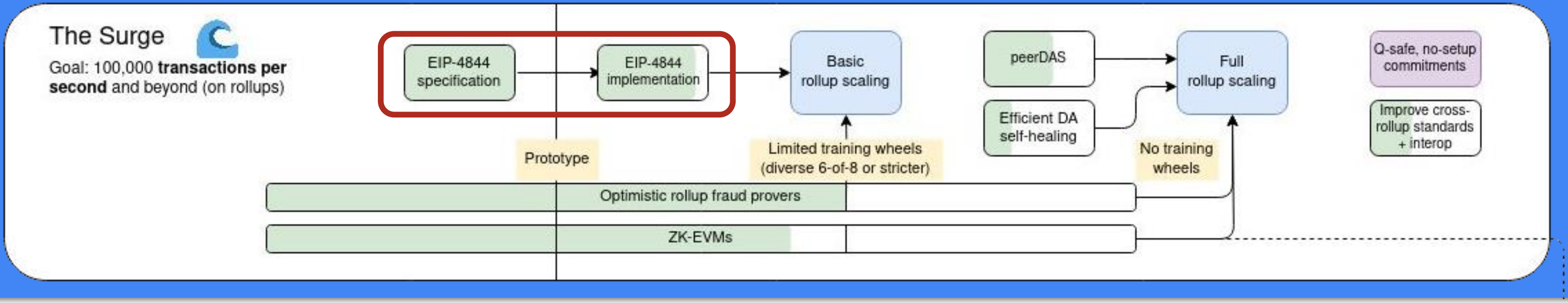


Data: (1, 3, 2, 2)
Extension: (7, 21, 48, 92)

*In practice, P(x) has thousands of coefficients*
C = commit(P) = a few bytes (*like a hash*) known o all nodes

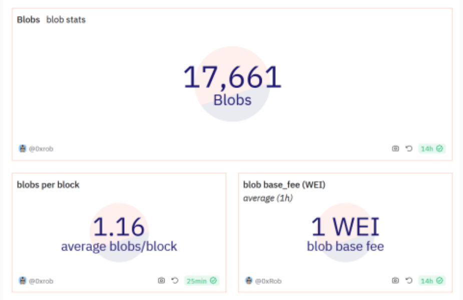
- Ask for random data point (e.g. the 3rd one)
- Receive the value 2 along with proof π
- Verify proof π against C, is satisfied that *P(3) = 2*
- At most 50% odds of "being fooled"
- Ask for another random data point, odds become 25%
- Another sample: 12.5%
- 30 samples = 1 / $2^{30}$ = ~1 in a billion chance of being fooled

**The node is quickly convinced that all the data is available, without having seen (downloaded) more than 30 samples**


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

## EIP4844 introduces Blobspace

- No fancy sampling yet, every node download all blobs (128 kilobytes*)
- Conservative initial values: Target of 3 blobs per block, max 6 (priced à la EIP1559, separately!)
- Sets the stage for Data Availability Sampling (using KZG commitment scheme)



**Blobs** blob stats

17,661
Blobs

@0xrob 14h

blobs per block

1.16
average blobs/block

@0xrob 25min

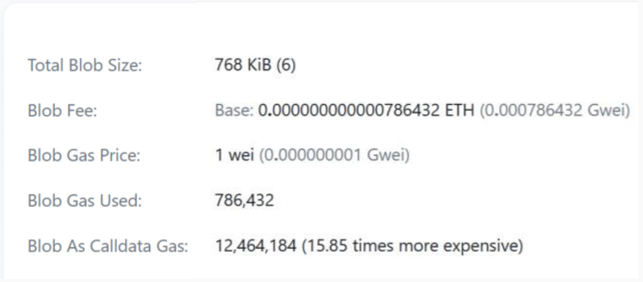
blob base_fee (WEI)
*average (1h)*

1 WEI
blob base fee

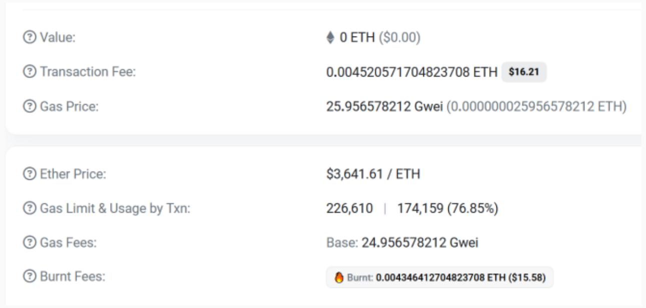
@0xRob 14h

*https://dune.com/0xRob/blobs*

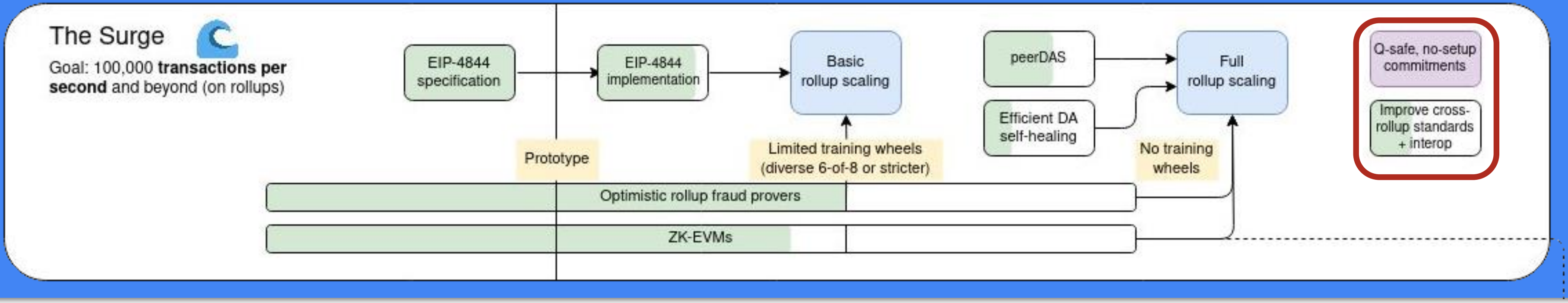
1 wei per blob gas × $2^{17}$ blob gas = 0.000131 *gwei* per blob



| | |
|---|---|
| Total Blob Size: | 768 KiB (6) |
| Blob Fee: | Base: 0.000000000000786432 ETH (0.000786432 Gwei) |
| Blob Gas Price: | 1 wei (0.000000001 Gwei) |
| Blob Gas Used: | 786,432 |
| Blob As Calldata Gas: | 12,464,184 (15.85 times more expensive) |



| | |
|---|---|
| Value: | 0 ETH ($0.00) |
| Transaction Fee: | 0.004520571704823708 ETH $16.21 |
| Gas Price: | 25.956578212 Gwei (0.000000025956578212 ETH) |
| Ether Price: | $3,641.61 / ETH |
| Gas Limit & Usage by Txn: | 226,610 \| 174,159 (76.85%) |
| Gas Fees: | Base: 24.956578212 Gwei |
| Burnt Fees: | Burnt: 0.004346412704823708 ETH ($15.58) |


The Surge
Goal: 100,000 transactions per second and beyond (on rollups)
EIP-4844 specification
EIP-4844 implementation
Basic rollup scaling
peerDAS
Efficient DA self-healing
Full rollup scaling
Q-safe, no-setup commitments
Improve cross-rollup standards + interop
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs

**Quantum-proof blobspace**

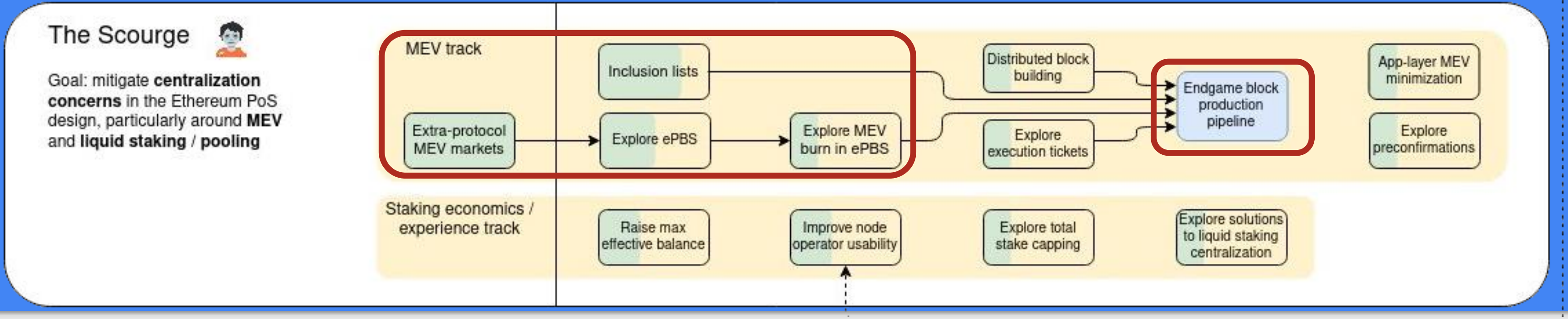
- KZG drawbacks: Not quantum-proof and required a trusted setup (>140k contributors)
- Eventually hot-swap KZG for something based on STARKs or Lattices

**Cross-rollup interopability**

- Establish standards between rollups
- Based rollups, preconfirmations, shared sequencing

## Roadmap tl;dr

- Merge: Better Proof of Stake
- Surge: More data (availability) for rollups
- **Scourge: Less MEV downsides**
- Verge: Easier verification
- Purge: Simpler protocol
- Splurge: Miscellaneous goodies

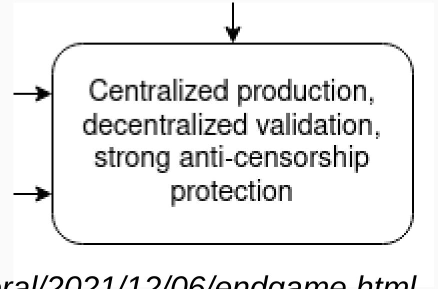

The Scourge
Goal: mitigate **centralization concerns** in the Ethereum PoS design, particularly around **MEV** and **liquid staking / pooling**
MEV track
Inclusion lists
Extra-protocol MEV markets
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
App-layer MEV minimization
Explore preconfirmations
Staking economics / experience track
Raise max effective balance
Improve node operator usability
Explore total stake capping
Explore solutions to liquid staking centralization

## Proposer/Builder Separation

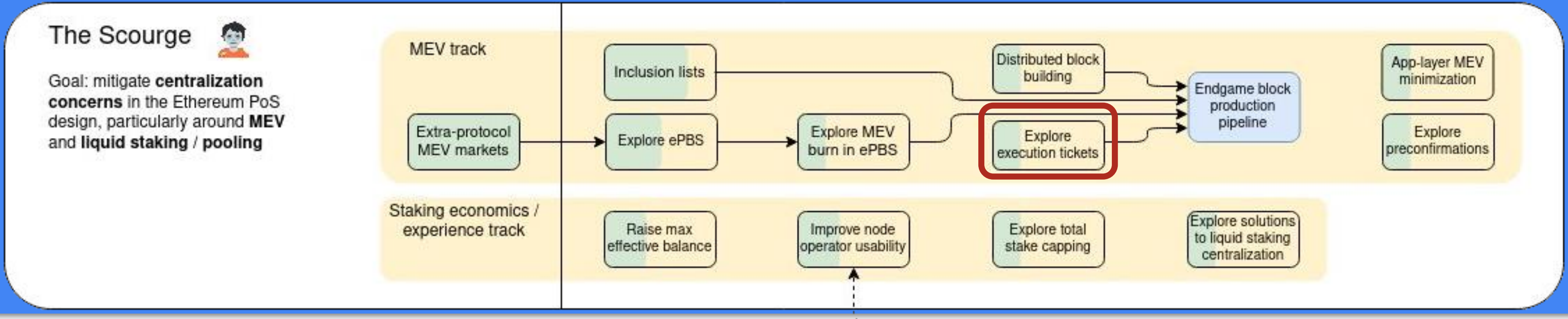
- *MEV is inevitable, untamed MEV markets hurt solo stakers*
- ***Goal:*** Minimize the choices validators have to make (reduce incentive to specialize)

- Out-of-protocol (today) with MEV Boost: Relays act as trusted brokers
- Enshrined PBS (ePBS): Remove relays, allow MEV burning to smooth the staking yield
- Inclusion lists: Put constraints on builders (reduce censorship by forcing inclusion)

**Endgame:**


Centralized production, decentralized validation, strong anti-censorship protection

*https://vitalik.eth.limo/general/2021/12/06/endgame.html*

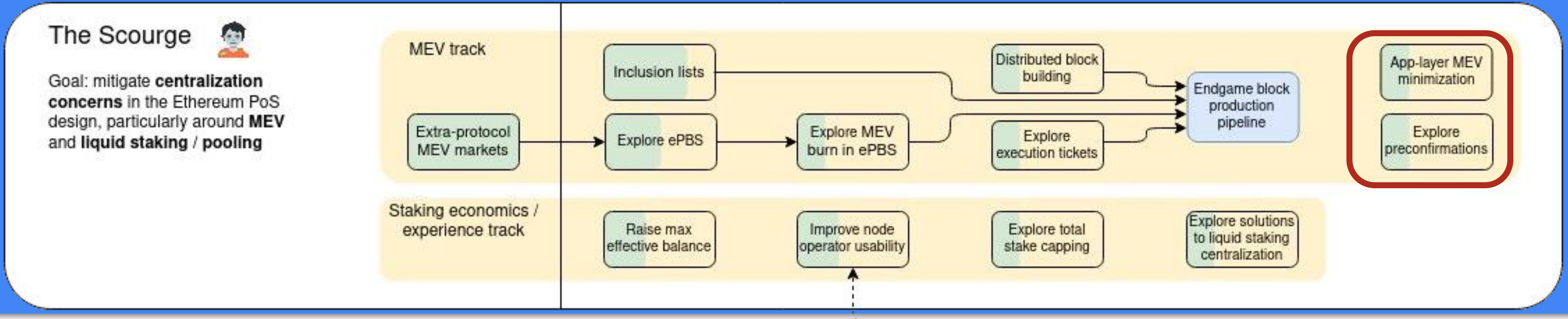

The Scourge
Goal: mitigate centralization concerns in the Ethereum PoS design, particularly around MEV and liquid staking / pooling
MEV track
Extra-protocol MEV markets
Inclusion lists
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
App-layer MEV minimization
Explore preconfirmations
Staking economics / experience track
Raise max effective balance
Improve node operator usability
Explore total stake capping
Explore solutions to liquid staking centralization

## Execution Tickets

- Even more role separation (between *attesting* and *proposing*)
- tl;dr Sell the right to propose a block ahead of time (like lottery tickets)

- Attesters remain simple, proposers can specialize (constrained by ILs)
- Permissionless degen MEV lottery (cost of ticket ≈ expected value of MEV per block)

ETH Research: https://ethresear.ch/t/execution-tickets/17944

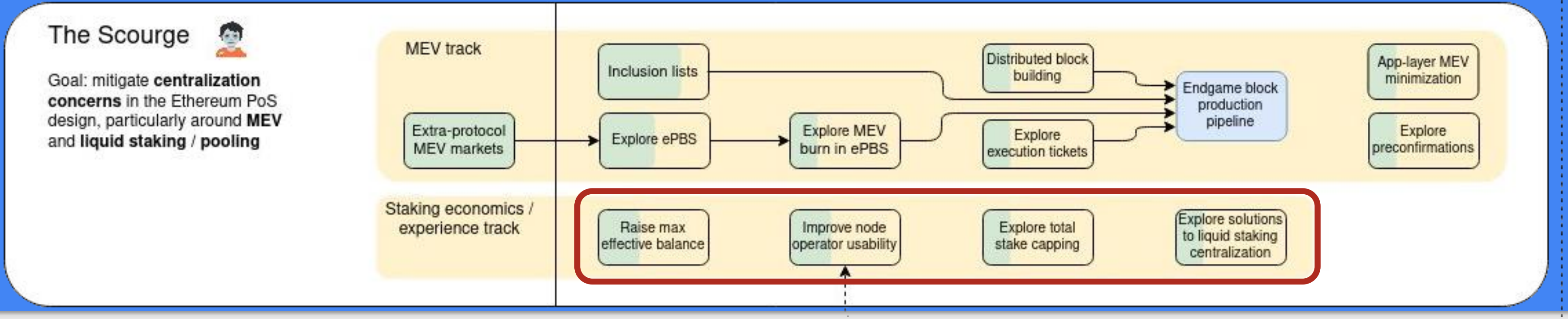

The Scourge
Goal: mitigate centralization concerns in the Ethereum PoS design, particularly around MEV and liquid staking / pooling
MEV track
Extra-protocol MEV markets
Inclusion lists
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
App-layer MEV minimization
Explore preconfirmations
Staking economics / experience track
Raise max effective balance
Improve node operator usability
Explore total stake capping
Explore solutions to liquid staking centralization

## App-layer MEV minimization

- Basically, develop better dapps with MEV in mind

## Preconfirmations

- Receive next-block inclusion guarantee from builder
- Pairs well with execution tickets and restaking schemes


The Scourge
Goal: mitigate **centralization concerns** in the Ethereum PoS design, particularly around **MEV** and **liquid staking** / **pooling**
MEV track
Extra-protocol MEV markets
Inclusion lists
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
App-layer MEV minimization
Explore preconfirmations
Staking economics / experience track
Raise max effective balance
Improve node operator usability
Explore total stake capping
Explore solutions to liquid staking centralization

**Raise max effective balance (*MaxEB*)**

- Today: Minimum 32 ETH, maximum 32 ETH
  After MaxEB: Minimum 32 ETH, maximum 2048
  + automatic compounding
  + fewer validators for the same amount of stake (lower overhead)

**Explore total stake capping**

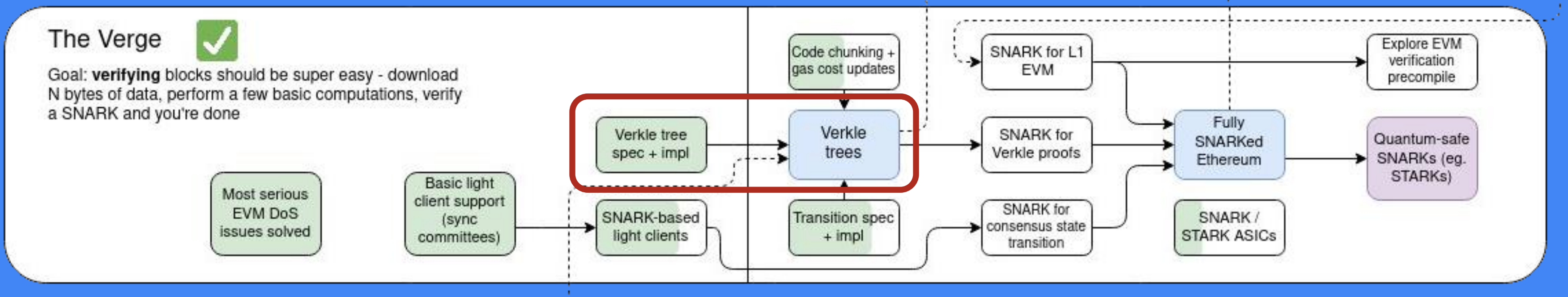
- Also related to overhead / SSF
- Research in progress: changing issuance curve (possibly negative), stake targetting

**Liquid staking centralization**

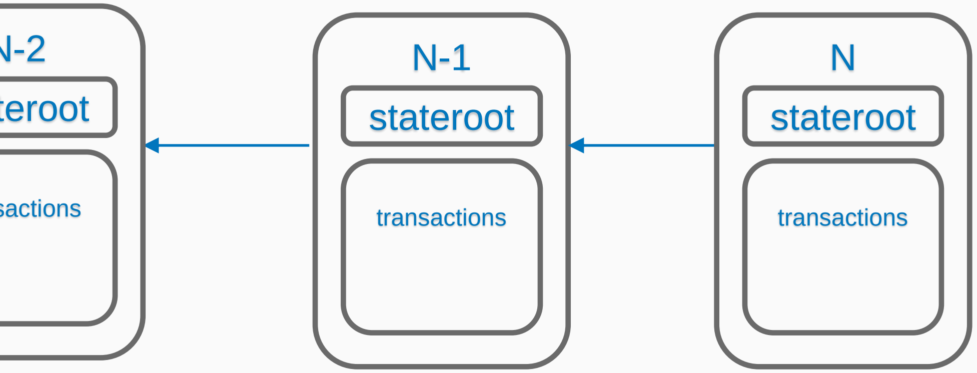
- Research in progress: Enshrine? Cap slashing penalties?

## Roadmap tl;dr

- Merge: Better Proof of Stake
- Surge: More data (availability) for rollups
- Scourge: Less MEV downsides
- **Verge: Easier verification**
- Purge: Simpler protocol
- Splurge: Miscellaneous goodies


The Verge ✓
Goal: **verifying** blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

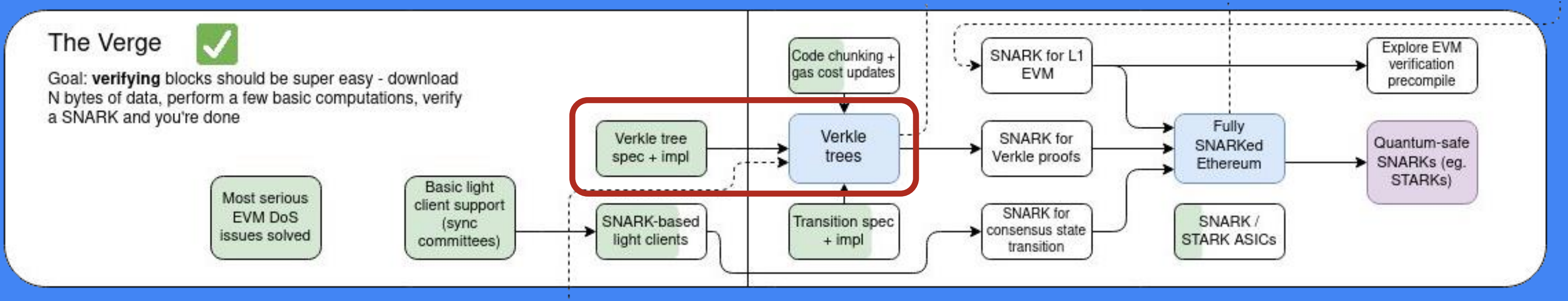
## Merkle -> Verkle



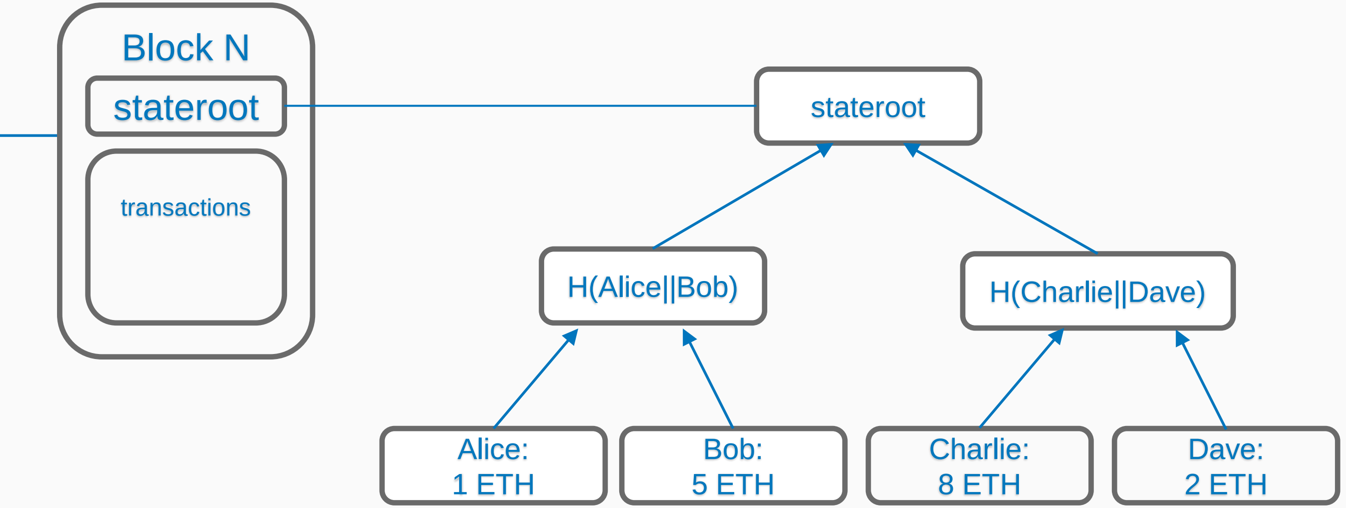
- State = all current balances (and more)
- History = all past transfers/transactions

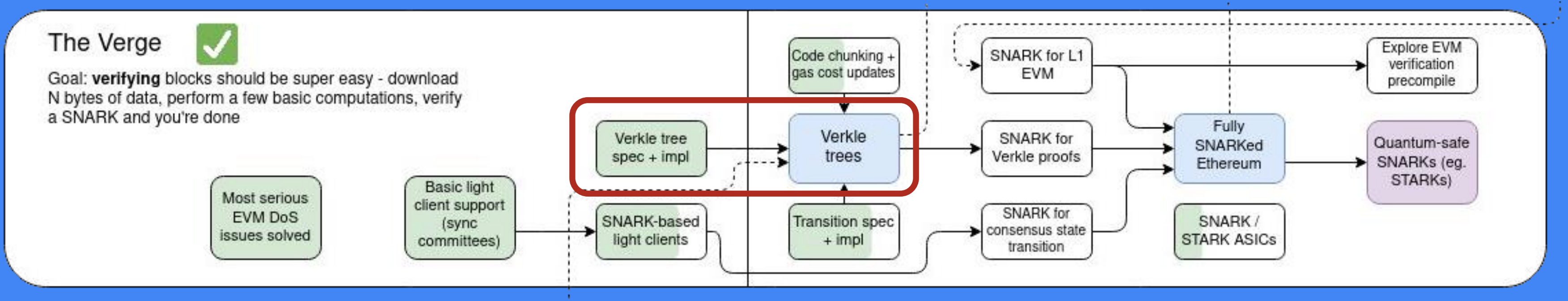
- With the history, you can compute the state to check balances and validate new transactions

- Is there a better way?

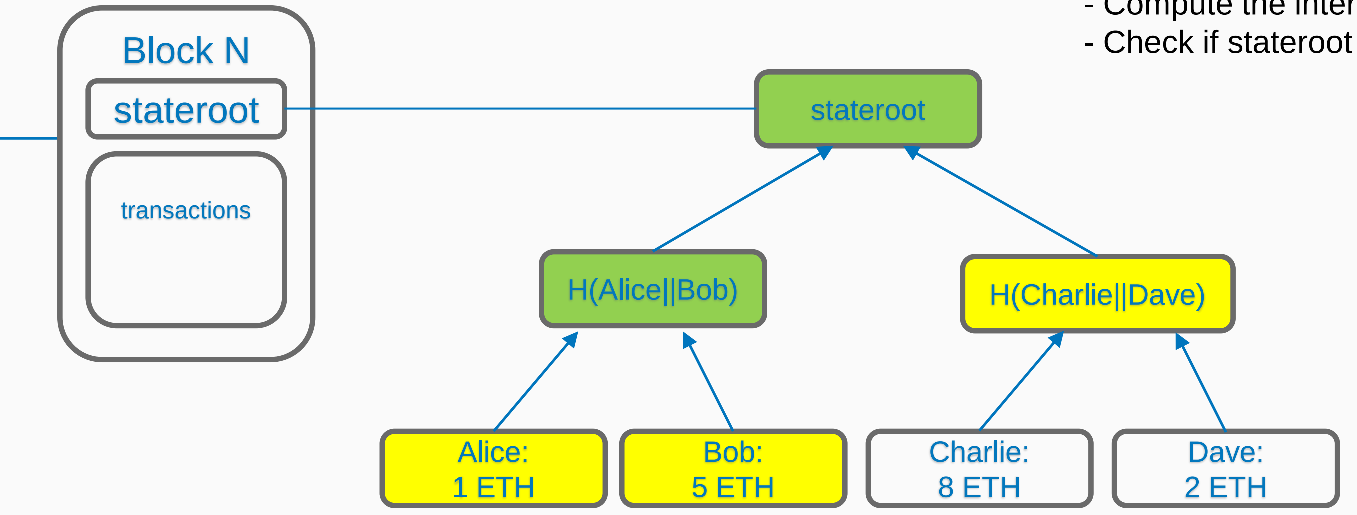

The Verge
Goal: **verifying** blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

## Merkle -> Verkle



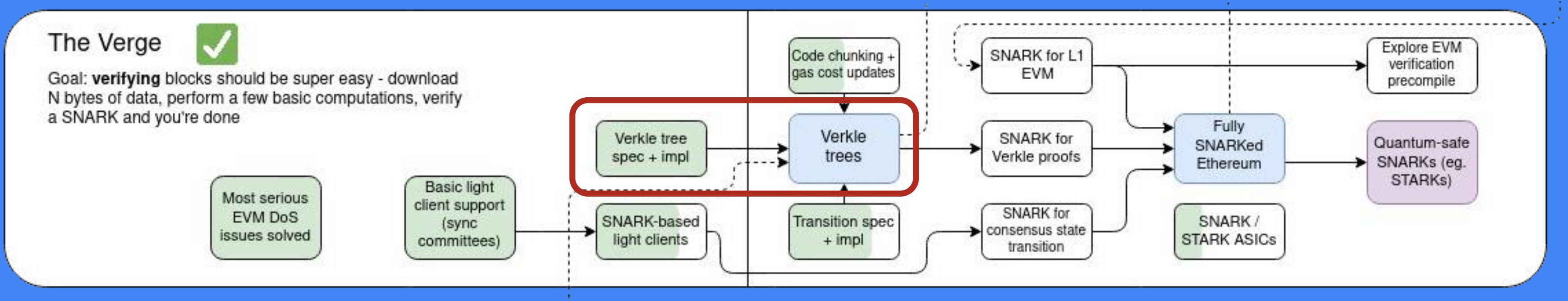

The Verge ✓
Goal: **verifying** blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

## Merkle -> Verkle



## Merkle-prove that Bob owns 5 ETH

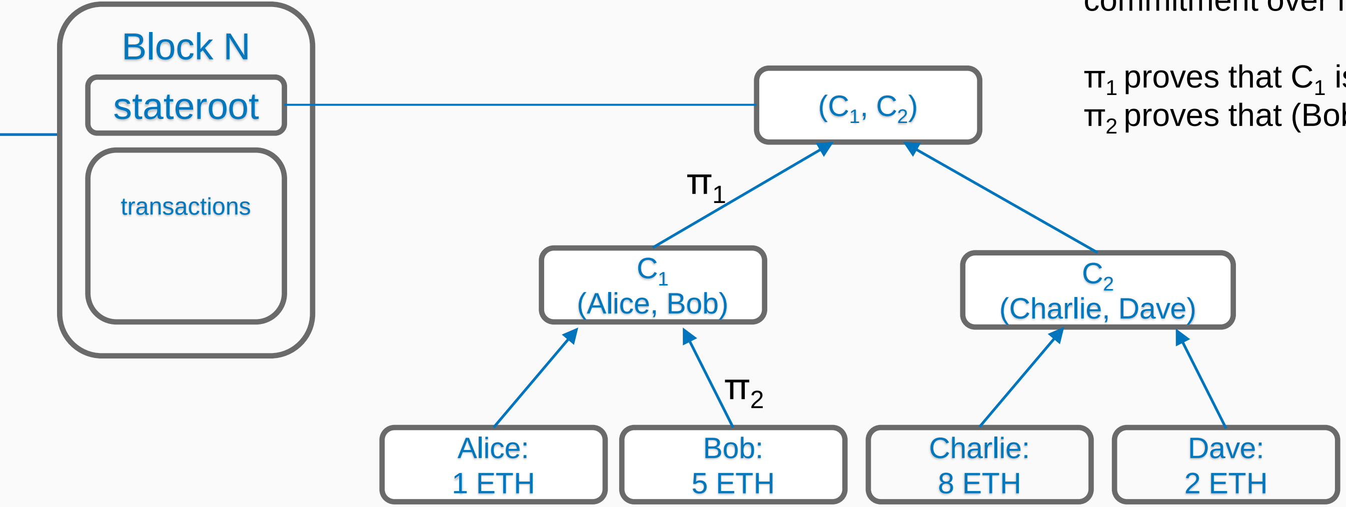
- Receive needed nodes (yellow)
- Compute the intermediary nodes (green)
- Check if stateroot matches with block header


The Verge ✓

Goal: **verifying** blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done

Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

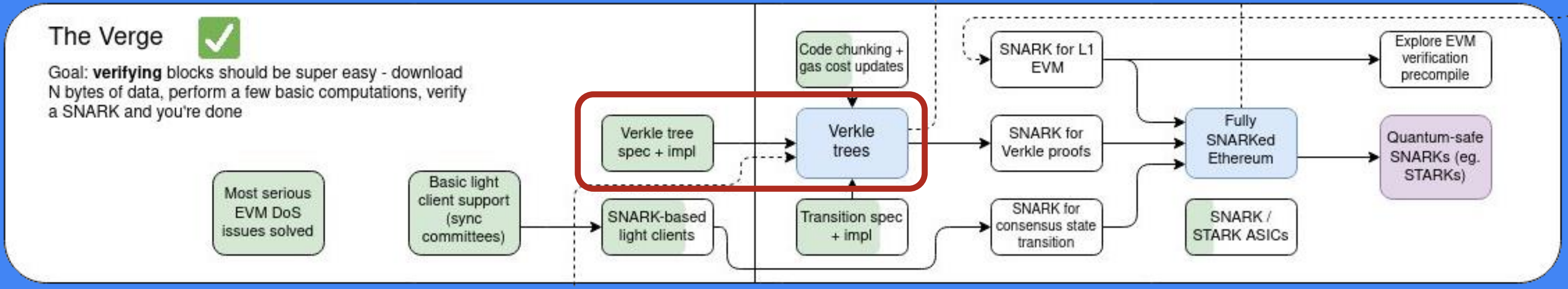
## Merkle -> Verkle



## Verkle-prove that Bob owns 5 ETH

Every node is now a polynomial commitment over its children

$\pi_1$ proves that $C_1$ is the stateroot's first child
$\pi_2$ proves that (Bob, 5 ETH) is $C_1$'s second child

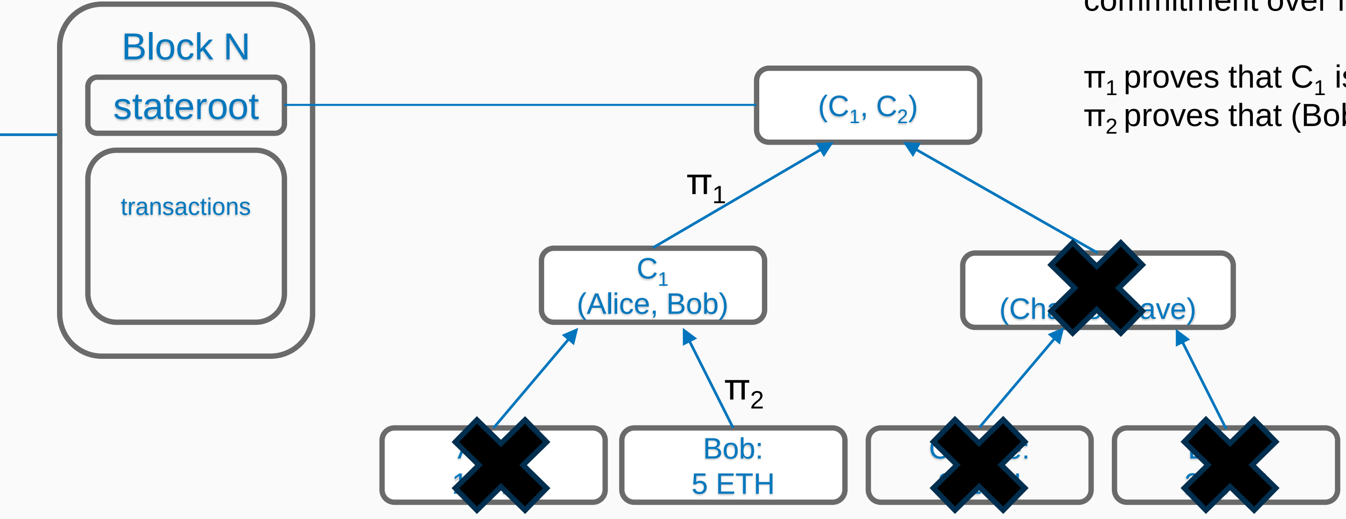


The Verge ✓

Goal: **verifying** blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done

Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

## Merkle -> Verkle



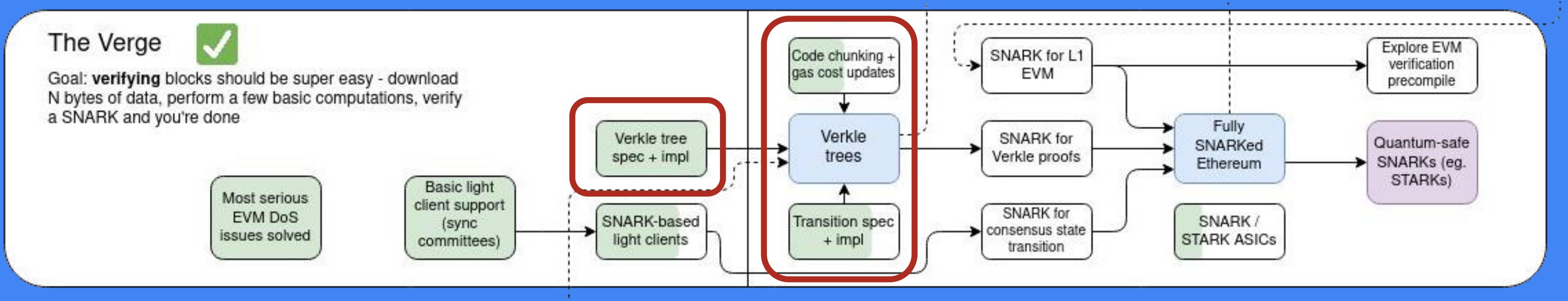
## Verkle-prove that Bob owns 5 ETH

Every node is now a polynomial commitment over its children

$\pi_1$ proves that $C_1$ is the stateroot's first child
$\pi_2$ proves that (Bob, 5 ETH) is $C_1$'s second child

Siblings no longer necessary
Only path, intermediary nodes and opening proofs are needed

Bonus: $\pi_1$ and $\pi_2$ can be merged!

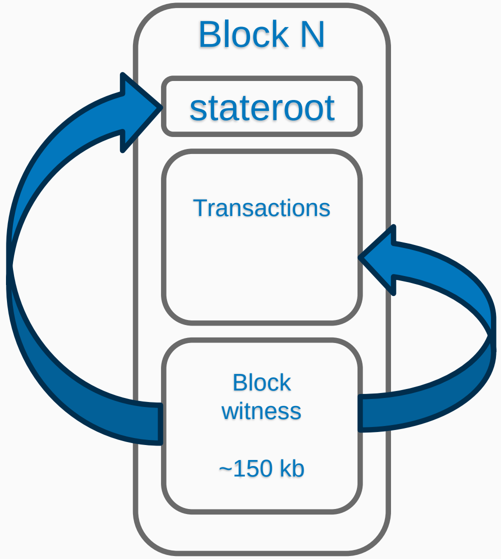

The Verge
Goal: verifying blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
Verkle tree spec + impl
SNARK-based light clients
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

## Verkle trees

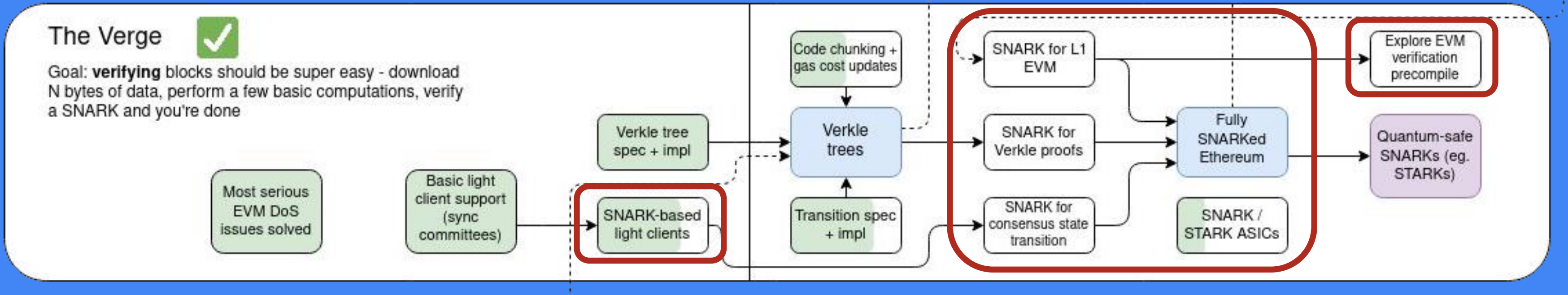
- Much shorter state proofs
- Wider tree (256 vs 16 siblings)
- ZK-friendly proofs

- Allow **stateless** validators
  (no history needed – instant sync)

- Light clients become even lighter
- Lower dev reliance on centralized indexers



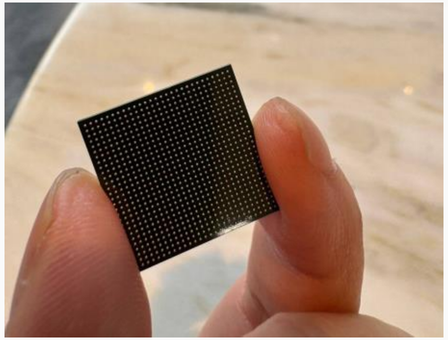
All the proofs needed to start computing transactions


The Verge
Goal: verifying blocks should be super easy - download N bytes of data, perform a few basic computations, verify a SNARK and you're done
Most serious EVM DoS issues solved
Basic light client support (sync committees)
SNARK-based light clients
Verkle tree spec + impl
Code chunking + gas cost updates
Verkle trees
Transition spec + impl
SNARK for L1 EVM
SNARK for Verkle proofs
SNARK for consensus state transition
Fully SNARKed Ethereum
SNARK / STARK ASICs
Explore EVM verification precompile
Quantum-safe SNARKs (eg. STARKs)

**SNARKify all the things**

- Light client protocol (sync committee transitions)
- All beacon chain transitions (signatures, balance changes, etc.)

- Verkle state access proofs / block witnesses
- Eventually all EVM execution (thank you zkRollups!)

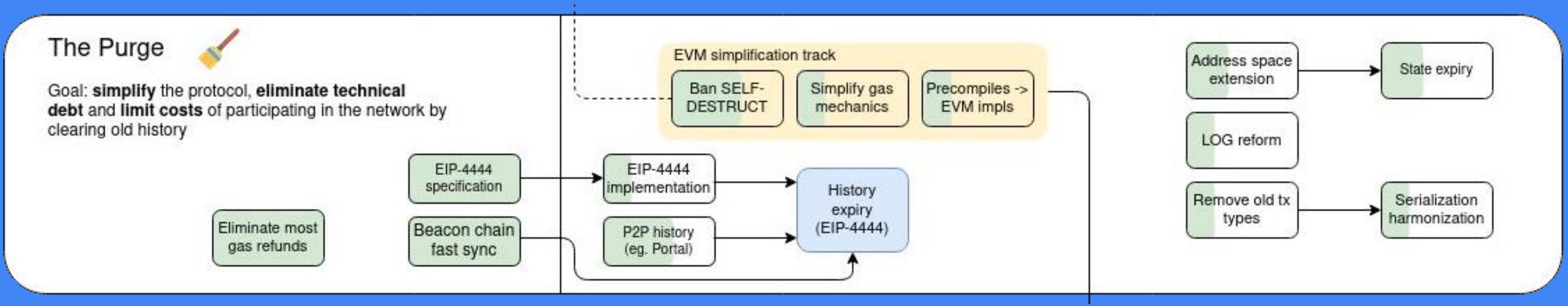

*First ever SNARK proving ASIC*
*https://x.com/drakefjustin/status/1755929540700807211*

**zkEVM opcode/precompile**

- Verify EVM execution proof inside the EVM (or inside an EVM execution proof…)

## Roadmap tl;dr

- Merge: Better Proof of Stake
- Surge: More data (availability) for rollups
- Scourge: Less MEV downsides
- Verge: Easier verification
- **Purge: Simpler protocol**
- Splurge: Miscellaneous goodies



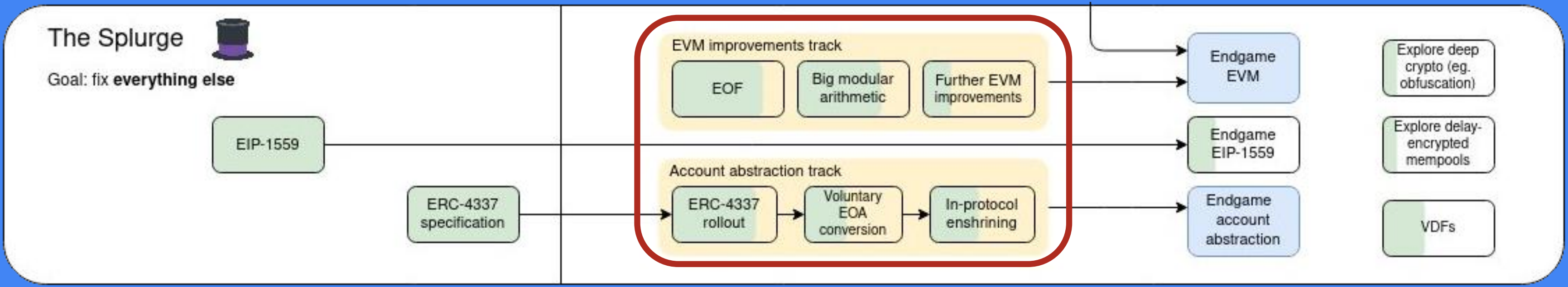
**History expiry (EIP-4444) – autoprune history older than 1 year**
- Simplifies client codebases (no need to support earlier forks)
- Alleviate node storage requirements
- History must reliably be accessible by other means (Portal network, torrents, block explorers, etc.)

**State expiry**
- Lower priority now with PBS and Statelessness
- Requires many breaking changes (e.g. address length)

**Various harmonizations**
- Serialization: RLP (execution layer) vs SSZ (consensus layer)
- Slowly phase out old transaction types (e.g. pre-EIP1559 legacy type)

## Roadmap tl;dr

- Merge: Better Proof of Stake
- Surge: More data (availability) for rollups
- Scourge: Less MEV downsides
- Verge: Easier verification
- Purge: Simpler protocol
- **Splurge: Miscellaneous goodies**



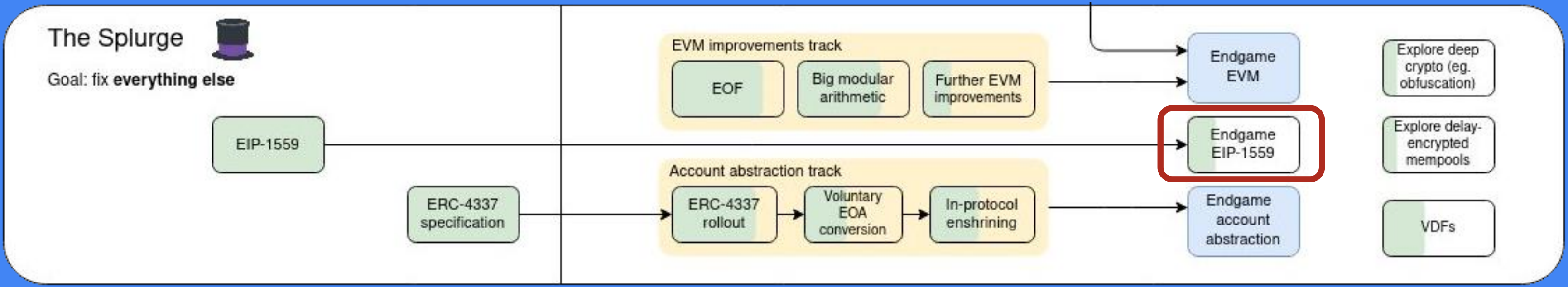
**EVM improvements / EVM Object Format**

- Series of EIPs to restructure aspects of EVM, makes future upgrades easier
  https://notes.ethereum.org/@ipsilon/evm-object-format-overview

**Account Abstraction**

- UX around Externally Owned Accounts (EOAs) is *bad* (like, *terrible*)
  Gas sponsorship, tx batching, key security,
  spending conditions, social recovery

- EIP-3074 to delegate control of EOAs to smart contract
- ERC-4337 for smart wallet standards across EVM chains/rollups (potential eventual enshrinement)



## Endgame EIP-1559

*More like an AMM curve*

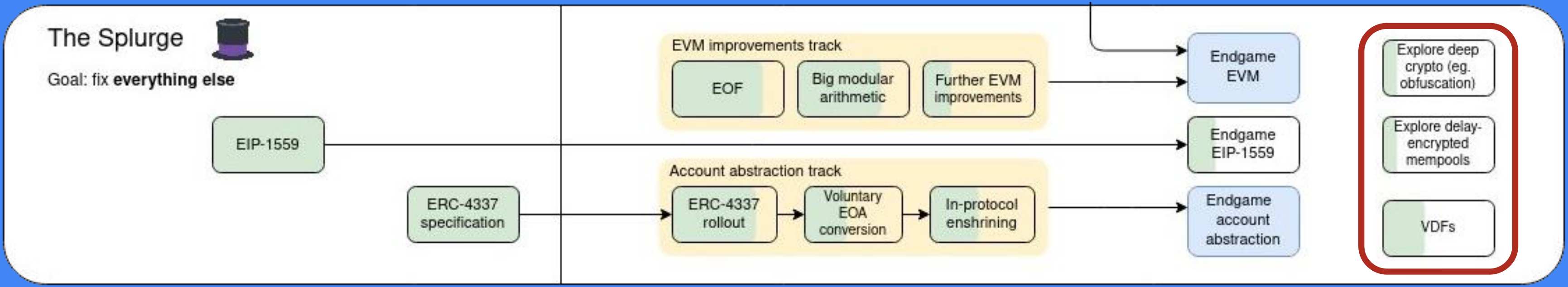
- Track excess gas instead of previous block's gas usage
- Higher censorship cost (entire fee vs. just priority fee)

*Multidimensional EIP-1559*

- Like gas vs blobs today, but for more resources: call data, state reads/writes, block size, witnesses etc.
- More efficient pricing – demand for one resource won't affect price for other resources

*Time-Aware base fee calculation (EIP-4396)*

- Avoid treating missed slots as sudden spike in demand



**Deep crypto**
- Fully Homomorphic Encryption
- One-Shot Signatures
- …

**Encrypted mempools**
- Toxic MEV disappears completely

**Verifiable Delay Functions**
- “*Non-parallelizable proof of work*”
  slow computation in one direction, fast verification after the fact
- Would enhance beacon chain randomness

Thabk you

Q&A