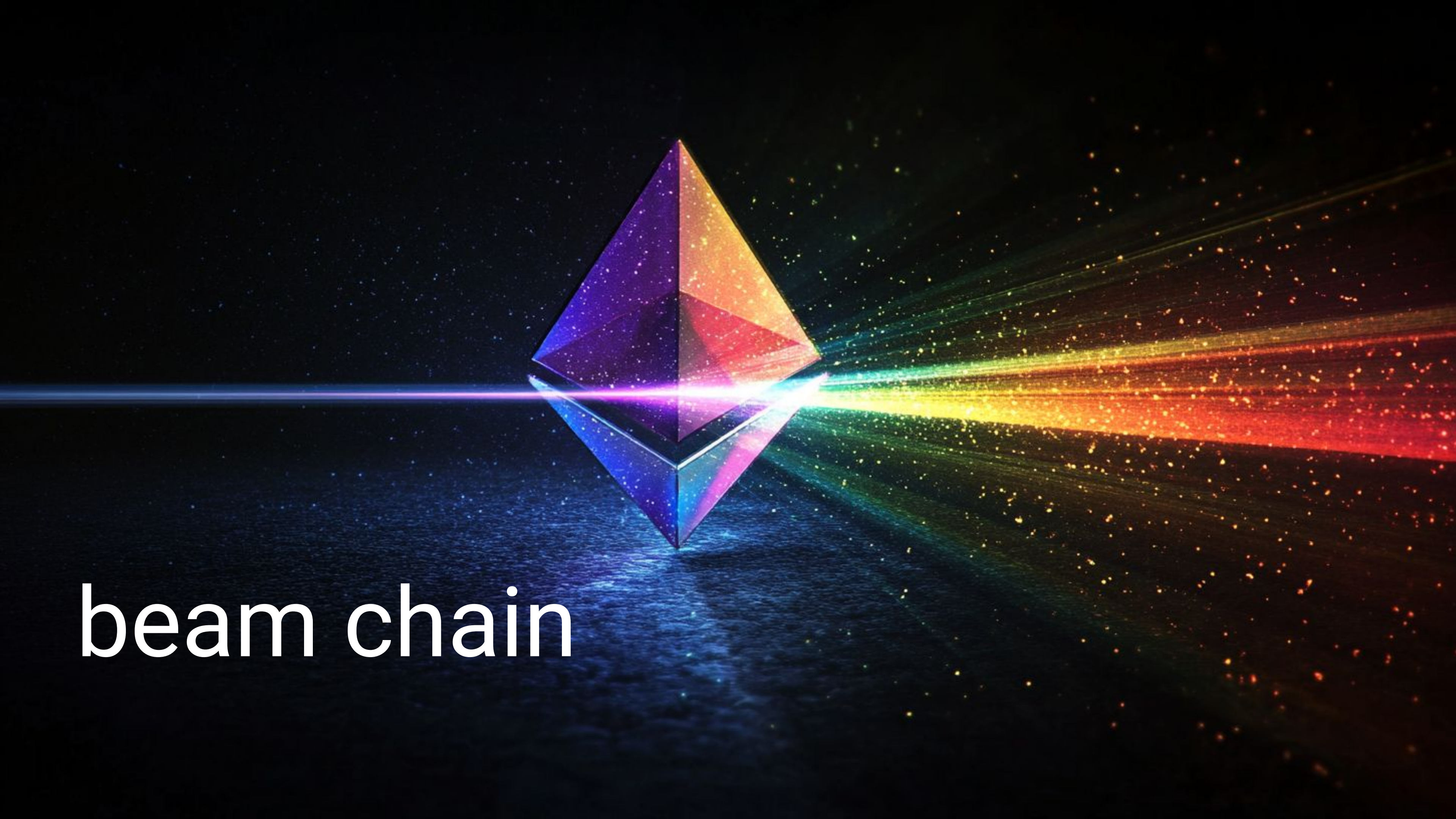
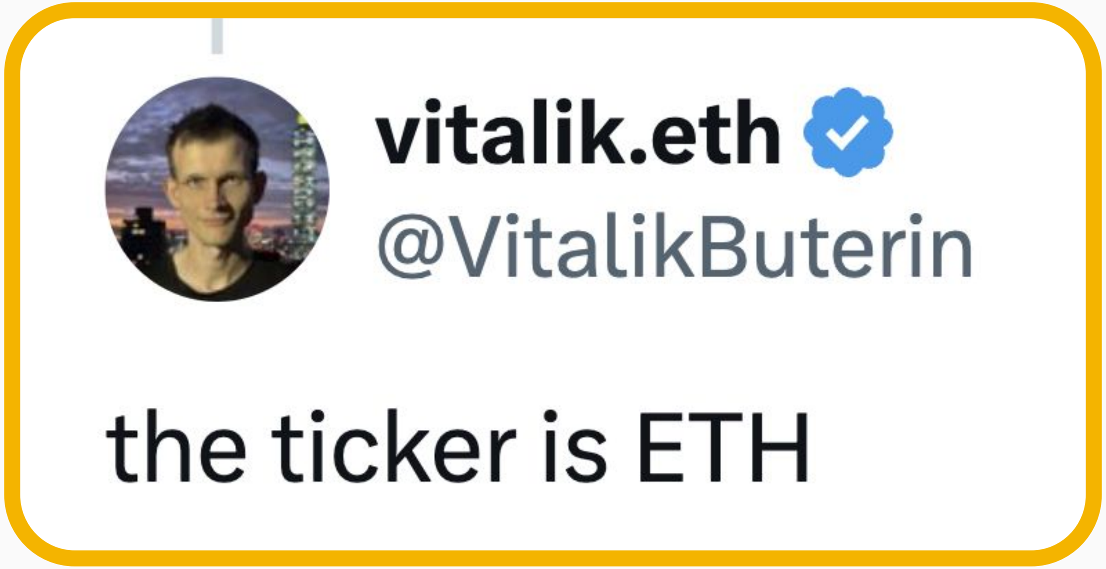
[redacted]



beam chain

disclaimer #1—a proposal

disclaimer #2—same ticker


vitalik.eth
@VitalikButerin

the ticker is ETH

part 1—vision

part 2—tech

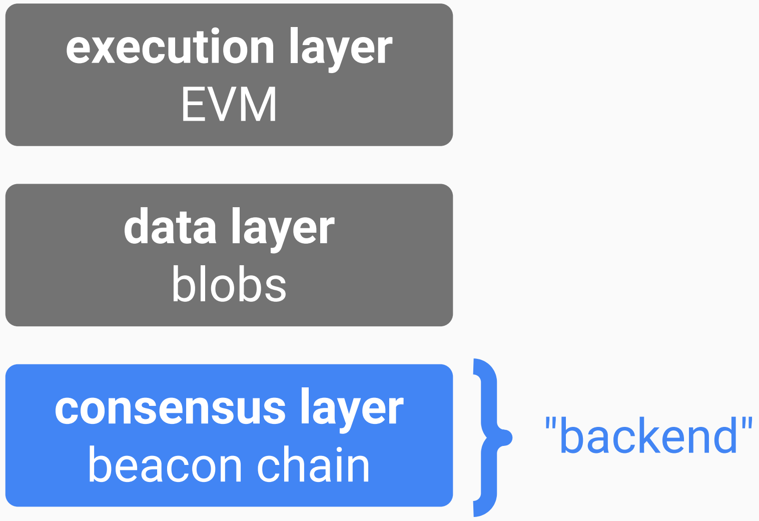
part 3—plan

part 1—vision

part 2—tech

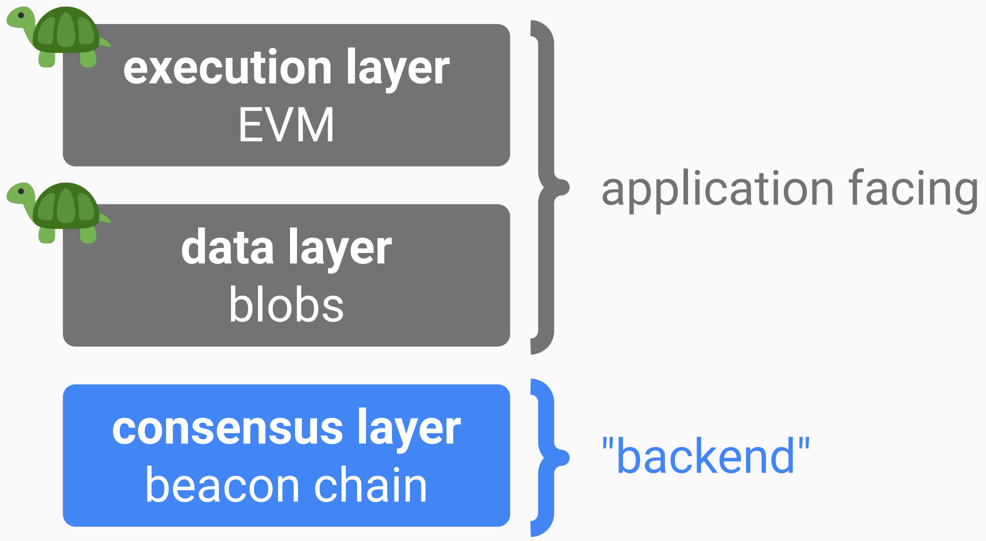
part 3—plan

scope

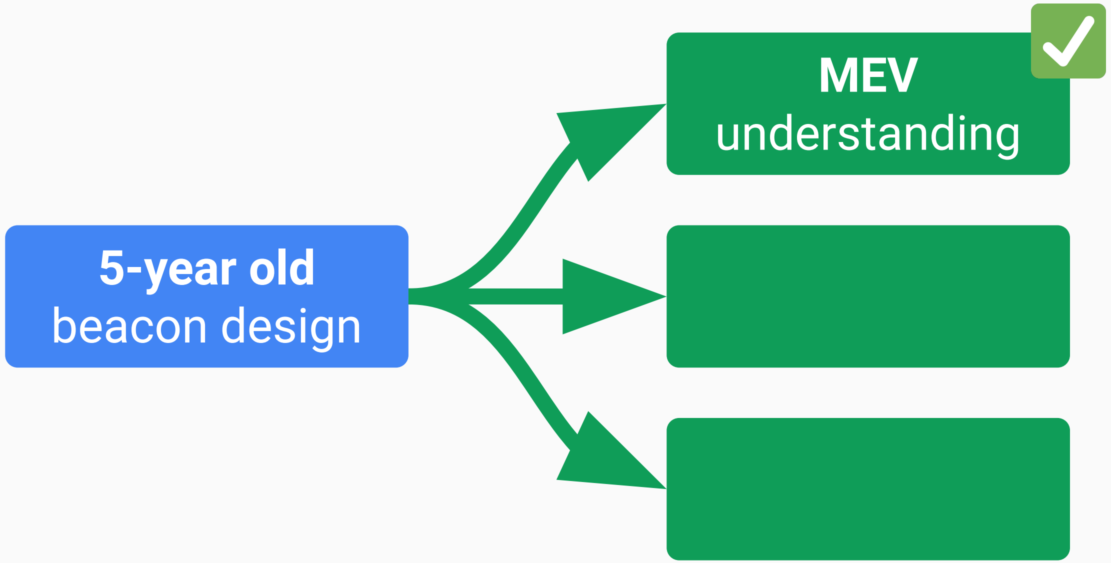


execution layer
EVM
data layer
blobs
consensus layer
beacon chain
"backend"

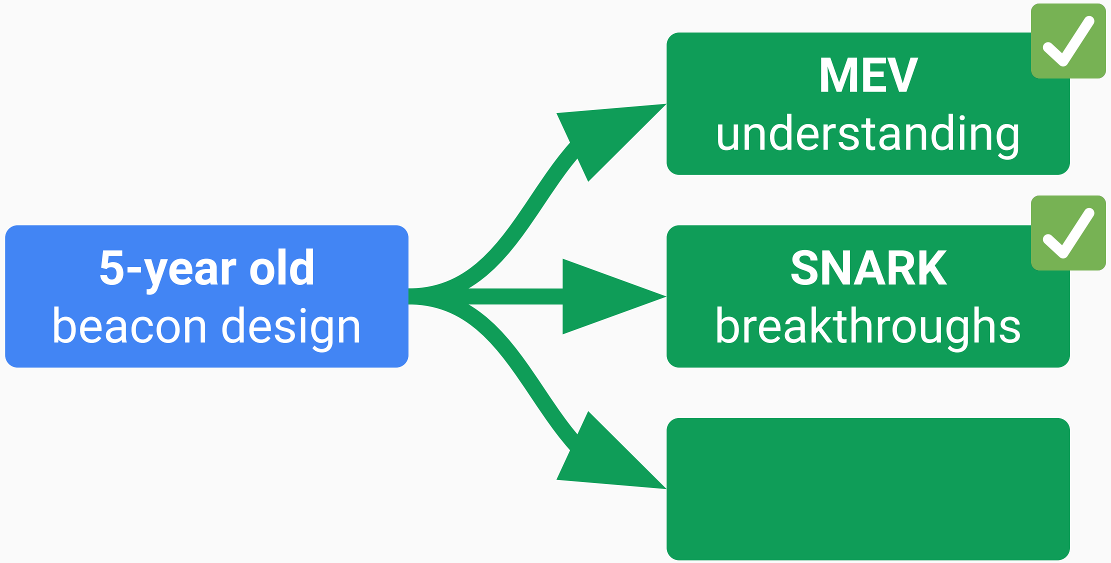
scope


execution layer
EVM
data layer
blobs
consensus layer
beacon chain
application facing
"backend"

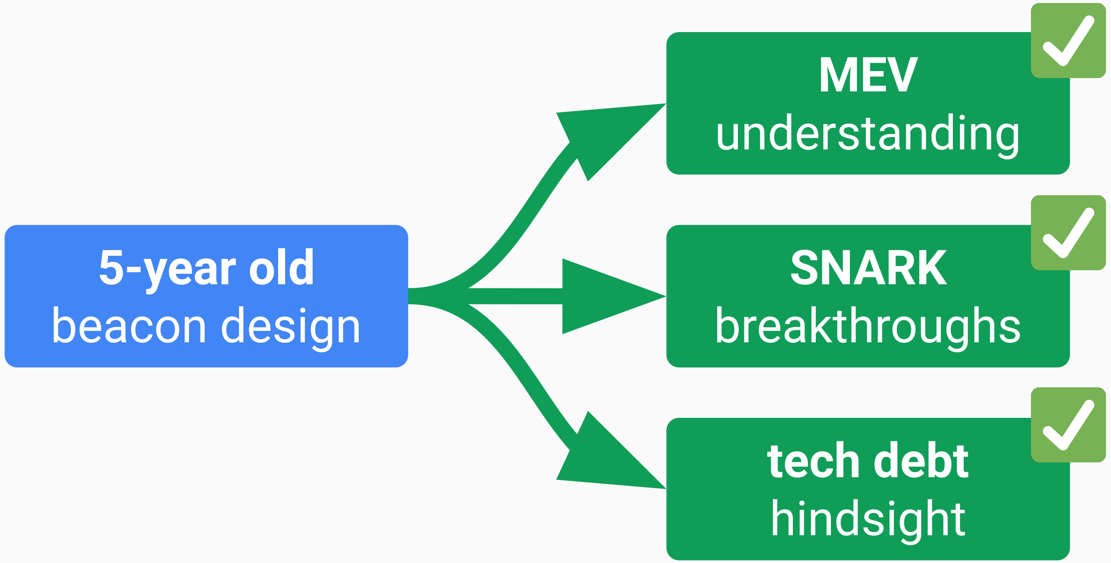
why now?



why now?


5-year old
beacon design
MEV
understanding
✓
SNARK
breakthroughs
✓

why now?


5-year old beacon design
MEV understanding ✓
SNARK breakthroughs ✓
tech debt hindsight ✓

consensus layer roadmap

block production

P0

censorship resistance
e.g. FOCIL

🌶️ **P1**

isolated validators
e.g. execution auctions

P2

faster slots
e.g. 4 seconds

staking

cryptography

consensus layer roadmap

block production

P0
censorship resistance
e.g. FOCIL

P1
isolated validators
e.g. execution auctions

P2
faster slots
e.g. 4 seconds

staking

P0
smarter issuance
e.g. stake cap

P1
smaller validators
e.g. 1 ETH Orbit staking

P2
faster finality
e.g. 3-slot FFG

cryptography

consensus layer roadmap

block production

- 🟩 **P0** **censorship resistance** e.g. FOCIL
- 🟩 🌶️ **P1** **isolated validators** e.g. execution auctions
- 🟥 **P2** **faster slots** e.g. 4 seconds

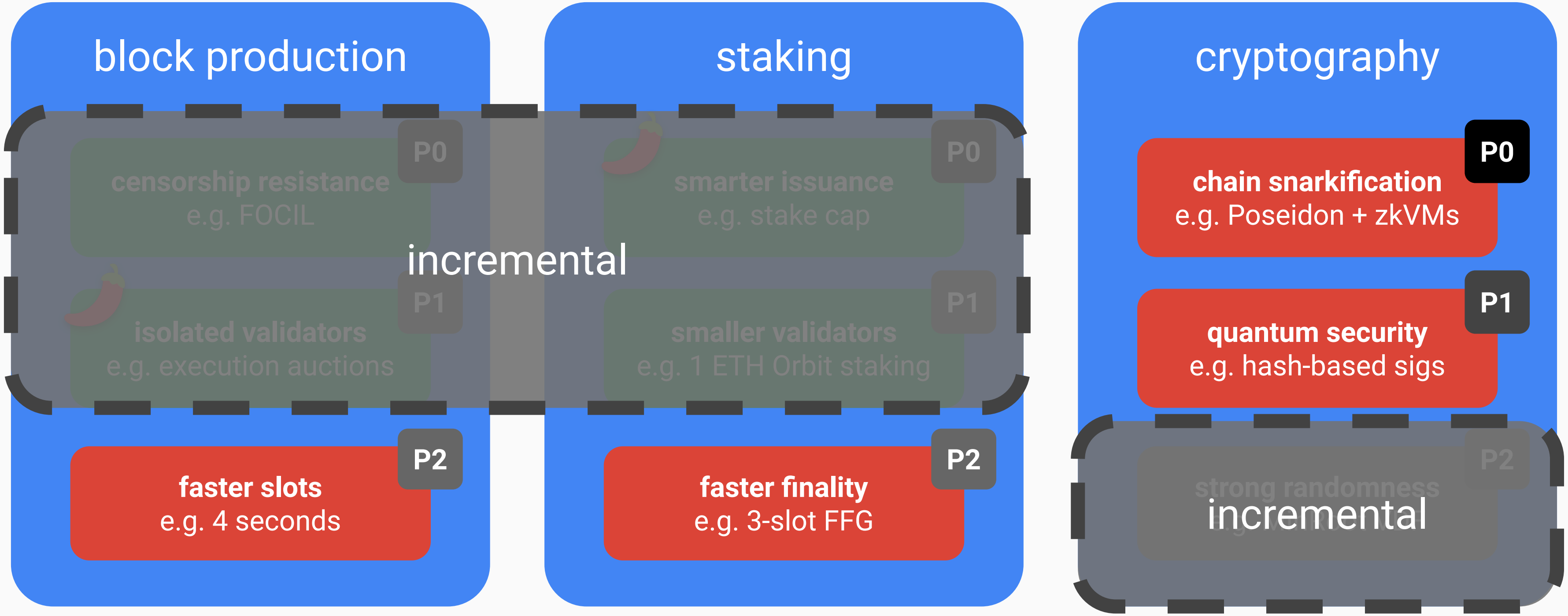
staking

- 🟩 🌶️ **P0** **smarter issuance** e.g. stake cap
- 🟩 **P1** **smaller validators** e.g. 1 ETH Orbit staking
- 🟥 **P2** **faster finality** e.g. 3-slot FFG

cryptography

- 🟥 **P0** **chain snarkification** e.g. Poseidon + zkVMs
- 🟥 **P1** **quantum security** e.g. hash-based sigs
- ⬜ **P2** **strong randomness** e.g. MinRoot VDF

consensus layer roadmap



Vitalik's roadmap (Dec 2023)

block production

P0 **censorship resistance**
e.g. FOCIL

P1 **isolated validators**
e.g. execution auctions

P2 **faster slots**
e.g. 4 seconds

staking

P0 **smarter issuance**
e.g. stake ca

P1 **smaller validators**
e.g. 1 ETH Orbit staking

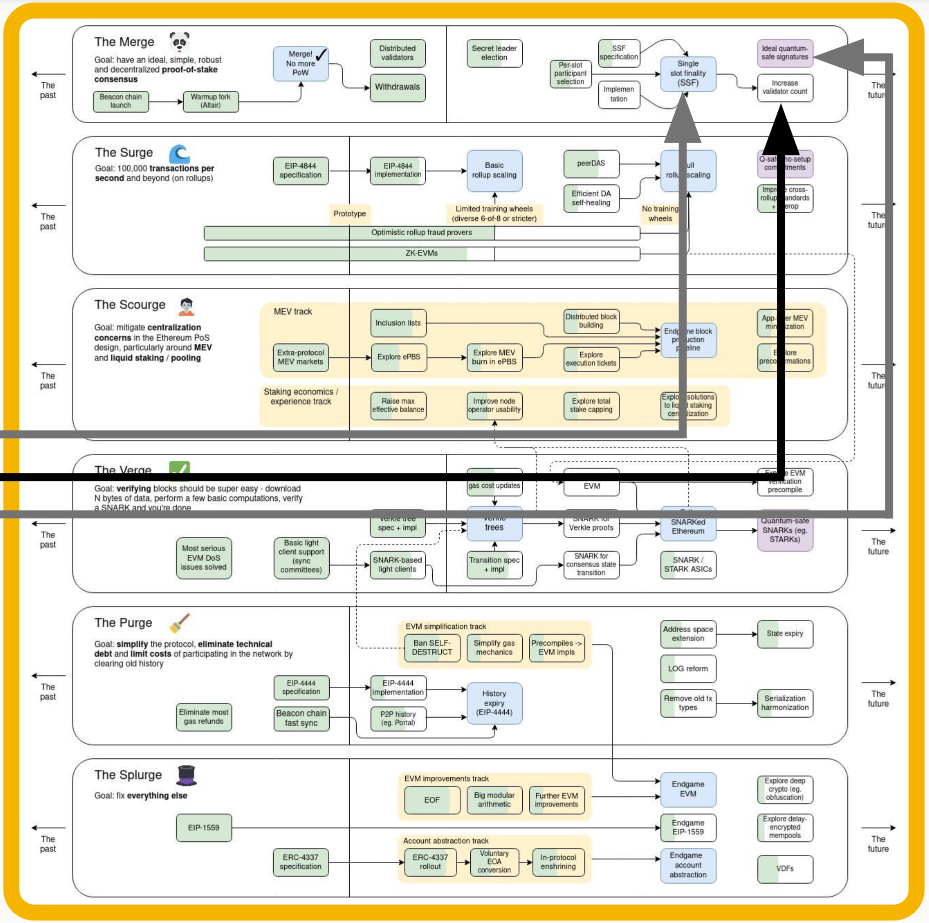
P2 **faster finality**
e.g. 3-slot FFG

cryptography

P0 **chain snarkification**
e.g. Poseidon + zkVMs

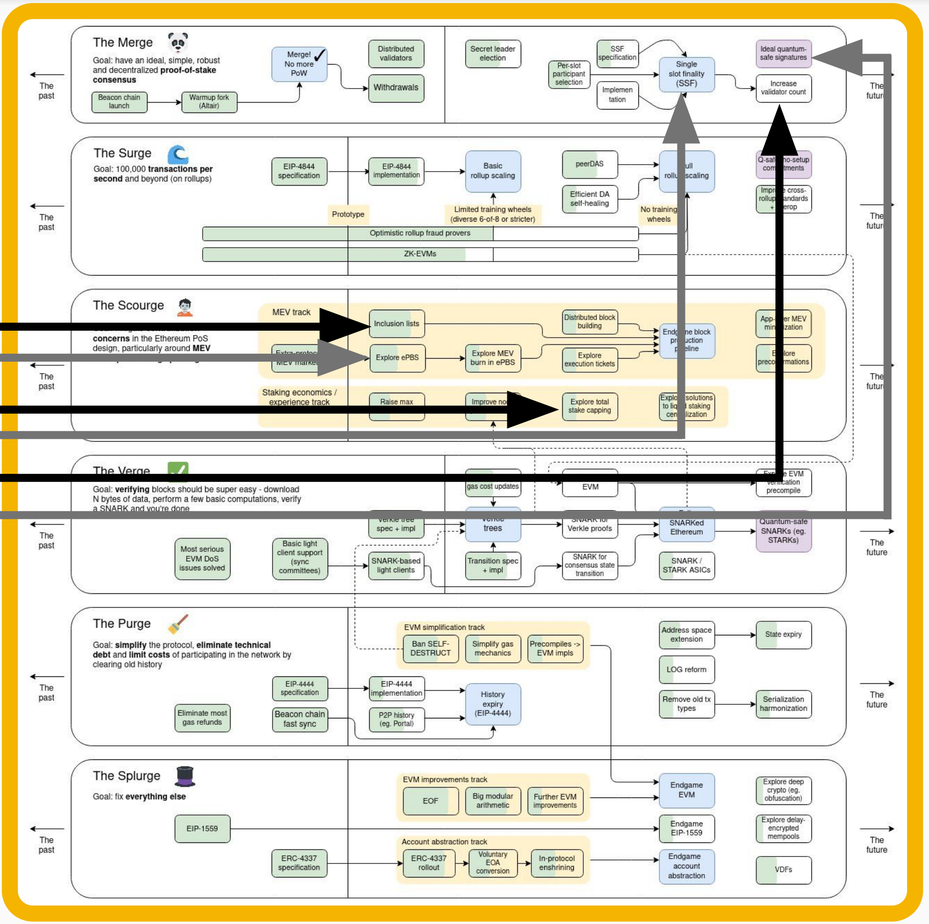
P1 **quantum security**
e.g. hash-based sigs

P2 **strong randomness**
e.g. MinRoot VDF



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
Per-slot participant selection
SSF specification
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
Distributed block building
Endgame block production pipeline
App-layer MEV minimization
Extra-protocol MEV markets
Explore ePBS
Explore MEV burn in ePBS
Explore execution tickets
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
gas cost updates
EVM
Explore EVM verification precompile
Verkle tree spec + impl
Verkle trees
SNARK for Verkle proofs
SNARKed Ethereum
Quantum-safe SNARKs (eg. STARKs)
Most serious EVM DoS issues solved
Basic light client support (sync committees)
SNARK-based light clients
Transition spec + impl
SNARK for consensus state transition
SNARK / STARK ASICs
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
Address space extension
State expiry
LOG reform
Remove old tx types
Serialization harmonization
EIP-4444 specification
EIP-4444 implementation
History expiry (EIP-4444)
Eliminate most gas refunds
Beacon chain fast sync
P2P history (eg. Portal)
The Splurge
Goal: fix everything else
EVM improvements track
EOF
Big modular arithmetic
Further EVM improvements
Endgame EVM
Explore deep crypto (eg. obfuscation)
EIP-1559
Endgame EIP-1559
Explore delay-encrypted mempools
Account abstraction track
ERC-4337 specification
ERC-4337 rollout
Voluntary EOA conversion
In-protocol enshrining
Endgame account abstraction
VDFs
The past
The future

Vitalik's roadmap (Dec 2023)


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
Per-slot participant selection
SSF specification
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
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs
The Scourge
concerns in the Ethereum PoS design, particularly around MEV
MEV track
Inclusion lists
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
Staking economics / experience track
Raise max
Explore total stake capping
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
gas cost updates
EVM
Verkle trees
SNARK for Verkle proofs
SNARKed Ethereum
Quantum-safe SNARKs (eg. STARKs)
Most serious EVM DoS issues solved
Basic light client support (sync committees)
SNARK-based light clients
Transition spec + impl
SNARK for consensus state transition
SNARK / STARK ASICs
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 SELFDESTRUCT
Simplify gas mechanics
Precompiles -> EVM impls
Address space extension
State expiry
LOG reform
Remove old tx types
Serialization harmonization
EIP-4444 specification
EIP-4444 implementation
History expiry (EIP-4444)
Eliminate most gas refunds
Beacon chain fast sync
P2P history (eg. Portal)
The Splurge
Goal: fix everything else
EVM improvements track
EOF
Big modular arithmetic
Further EVM improvements
Endgame EVM
EIP-1559
Endgame EIP-1559
Account abstraction track
ERC-4337 specification
ERC-4337 rollout
Voluntary EOA conversion
In-protocol enshrining
Endgame account abstraction
Explore deep crypto (eg. obfuscation)
Explore delay-encrypted mempools
VDFs
The past
The future

block production

P0 **censorship resistance**
e.g. FOCIL

P1 **isolated validators**
e.g. execution auctions

P2 **faster slots**
e.g. 4 seconds

staking

P0 **smarter issuance**
e.g. stake cap

P1 **smaller validators**
e.g. 1 ETH Orbit staking

P2 **faster finality**
e.g. 3-slot FFG

cryptography

P0 **chain snarkification**
e.g. Poseidon + zkVMs

P1 **quantum security**
e.g. hash-based sigs

P2 **strong randomness**
e.g. MinRoot VDF

Vitalik's roadmap (Dec 2023)

block production

P0 **censorship resistance** e.g. FOCIL

P1 **isolated validators** e.g. execution auctions

P2 **faster slots** e.g. 4 seconds

staking

P0 **smarter issuance** e.g. stake cap

P1 **smaller validators** e.g. 1 ETH Orbit staking

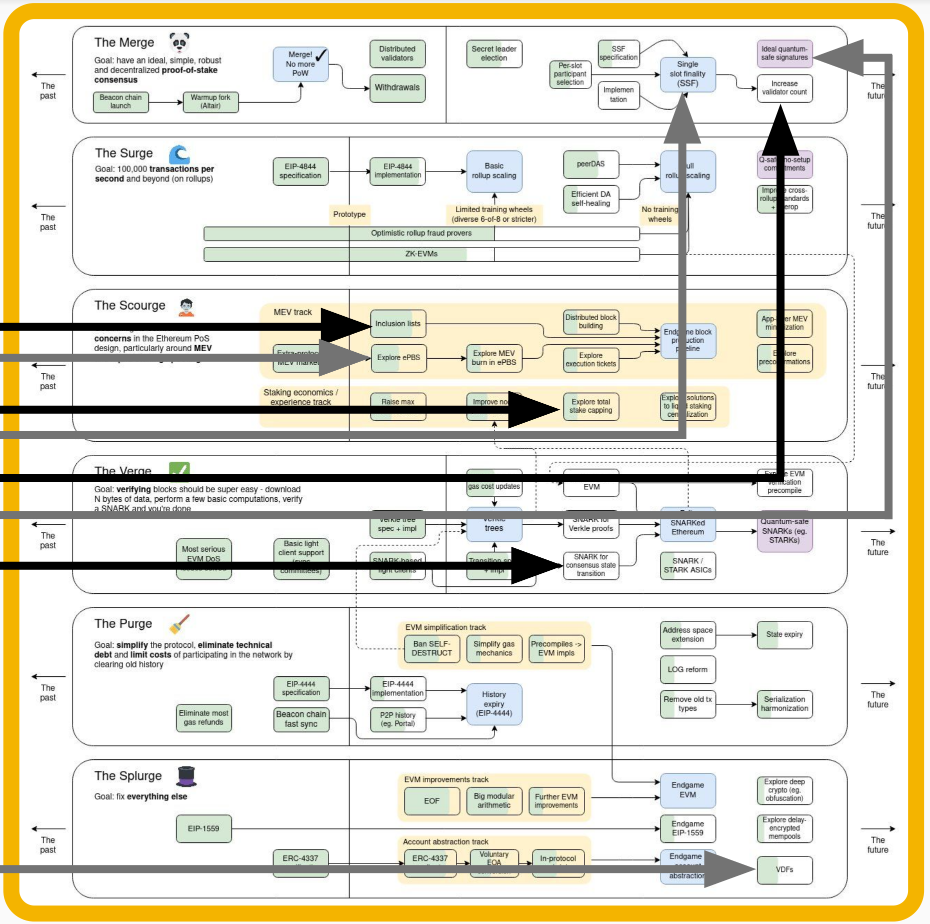
P2 **faster finality** e.g. 3-slot FFG

cryptography

P0 **chain snarkification** e.g. Poseidon + zkVMs

P1 **quantum security** e.g. hash-based sigs

P2 **strong randomness** e.g. MinRoot VDF



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
Per-slot participant selection
SSF specification
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
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs
Q-safe no-setup commitments
Improve cross-rollup standards + interop
The Scourge
concerns in the Ethereum PoS design, particularly around MEV
MEV track
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
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
Basic light client support
Verkle tree spec + impl
SNARK-based light clients
gas cost updates
Verkle trees
Transition spec + impl
EVM
SNARK for Verkle proofs
SNARK for consensus state transition
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
EIP-4444 implementation
Beacon chain fast sync
P2P history (eg. Portal)
History expiry (EIP-4444)
Address space extension
LOG reform
Remove old tx types
State expiry
Serialization harmonization
The Splurge
Goal: fix everything else
EVM improvements track
EOF
Big modular arithmetic
Further EVM improvements
EIP-1559
ERC-4337
Account abstraction track
Voluntary EOA
In-protocol
Endgame EVM
Endgame EIP-1559
Endgame account abstraction
Explore deep crypto (eg. obfuscation)
Explore delay-encrypted mempools
VDFs
The past
The future

Vitalik's roadmap (Dec 2023)

NEW

block production

P0 **censorship resistance** e.g. FOCIL

P1 **isolated validators** e.g. execution auctions

P2 **faster slots** e.g. 4 seconds

staking

P0 **smarter issuance** e.g. stake cap

P1 **smaller validators** e.g. 1 ETH Orbit staking

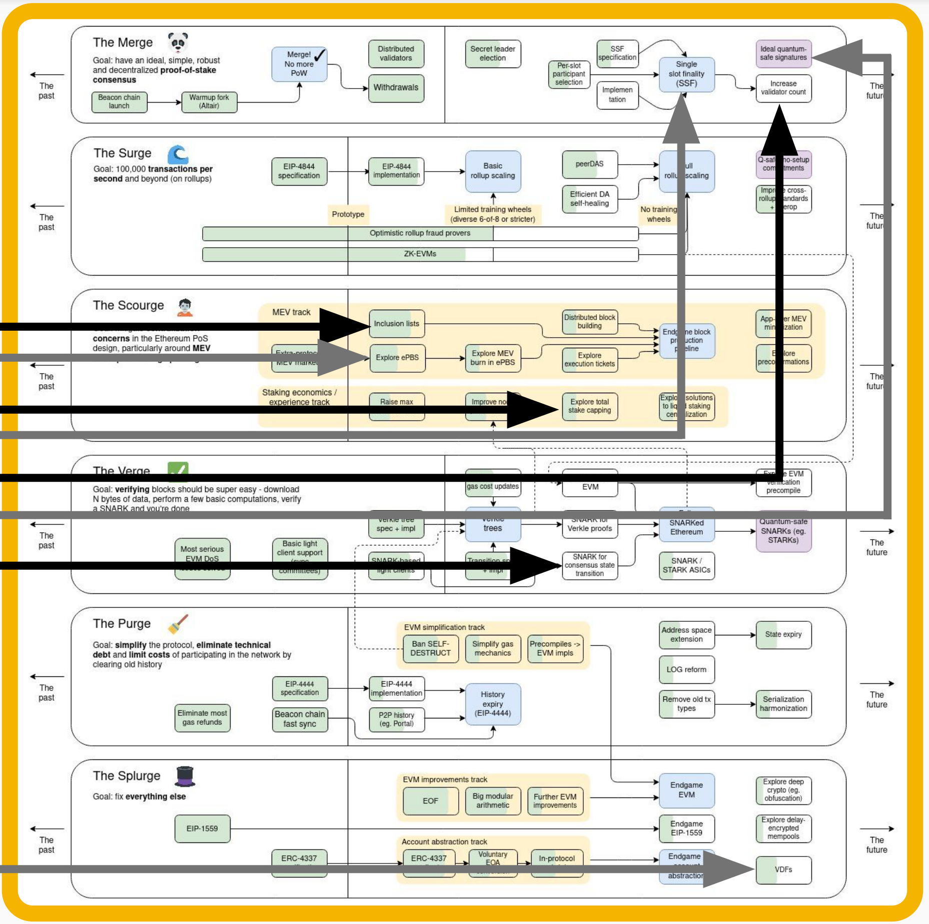
P2 **faster finality** e.g. 3-slot FFG

cryptography

P0 **chain snarkification** e.g. Poseidon + zkVMs

P1 **quantum security** e.g. hash-based sigs

P2 **strong randomness** e.g. MinRoot VDF



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
Prototype
Limited training wheels (diverse 6-of-8 or stricter)
No training wheels
Optimistic rollup fraud provers
ZK-EVMs
The Scourge
concerns in the Ethereum PoS design, particularly around MEV
MEV track
Inclusion lists
Explore ePBS
Explore MEV burn in ePBS
Distributed block building
Explore execution tickets
Endgame block production pipeline
Staking economics / experience track
Raise max
Explore total stake capping
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
gas cost updates
EVM
Verkle trees
SNARK for Verkle proofs
SNARKed Ethereum
Quantum-safe SNARKs (eg. STARKs)
Most serious EVM DoS issues solved
Basic light client support (sync committees)
SNARK-based light clients
SNARK for consensus state transition
SNARK / STARK ASICs
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
Address space extension
State expiry
LOG reform
Remove old tx types
Serialization harmonization
EIP-4444 specification
EIP-4444 implementation
History expiry (EIP-4444)
Eliminate most gas refunds
Beacon chain fast sync
P2P history (eg. Portal)
The Splurge
Goal: fix everything else
EVM improvements track
EOF
Big modular arithmetic
Further EVM improvements
Endgame EVM
EIP-1559
Endgame EIP-1559
Account abstraction track
ERC-4337
Voluntary EOA conversion
In-protocol
Endgame account abstraction
Explore deep crypto (eg. obfuscation)
Explore delay-encrypted mempools
VDFs
The past
The future

consensus purge

epochs
only slots

deposit contract
only precompile

sync committees
only SNARK proofs

max balance
only min balance

deposit tree
only EVM root

withdrawal credentials
only EVM addresses

effective balances
only balances

committee indices
only attestation slots

swap-or-not
only RANDAO

consensus purge

epochs
only slots

deposit contrac
only precom

sync committees
only SNARK proofs

max balance
only min balance

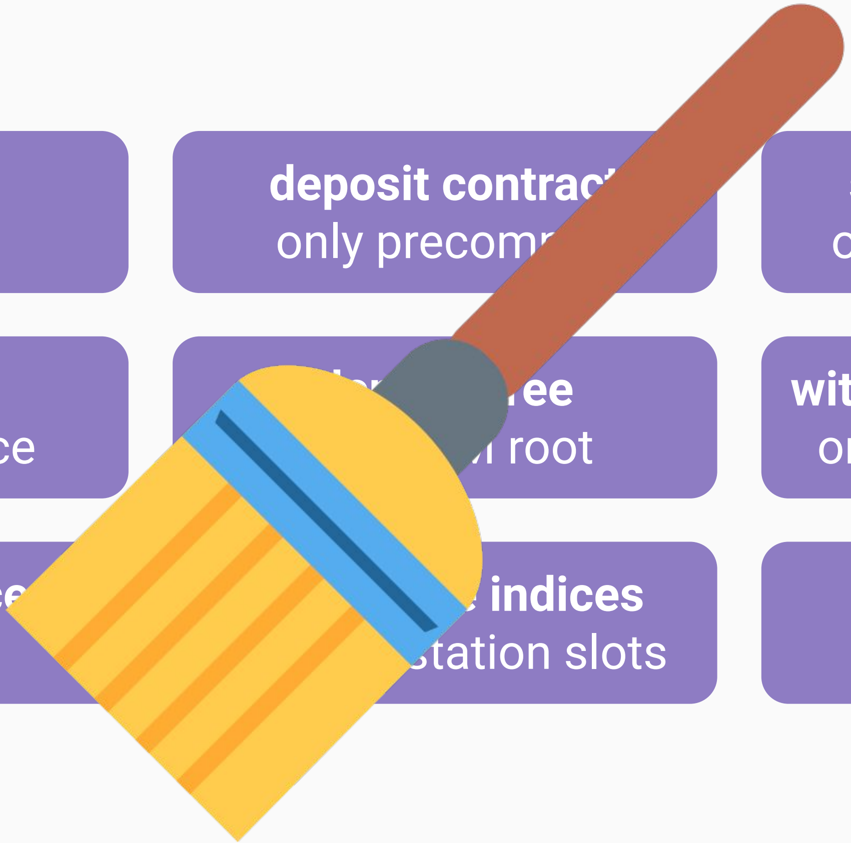
ee
root

withdrawal credentials
only EVM addresses

effective balance
only balances

indices
tation slots

swap-or-not
only RANDAO

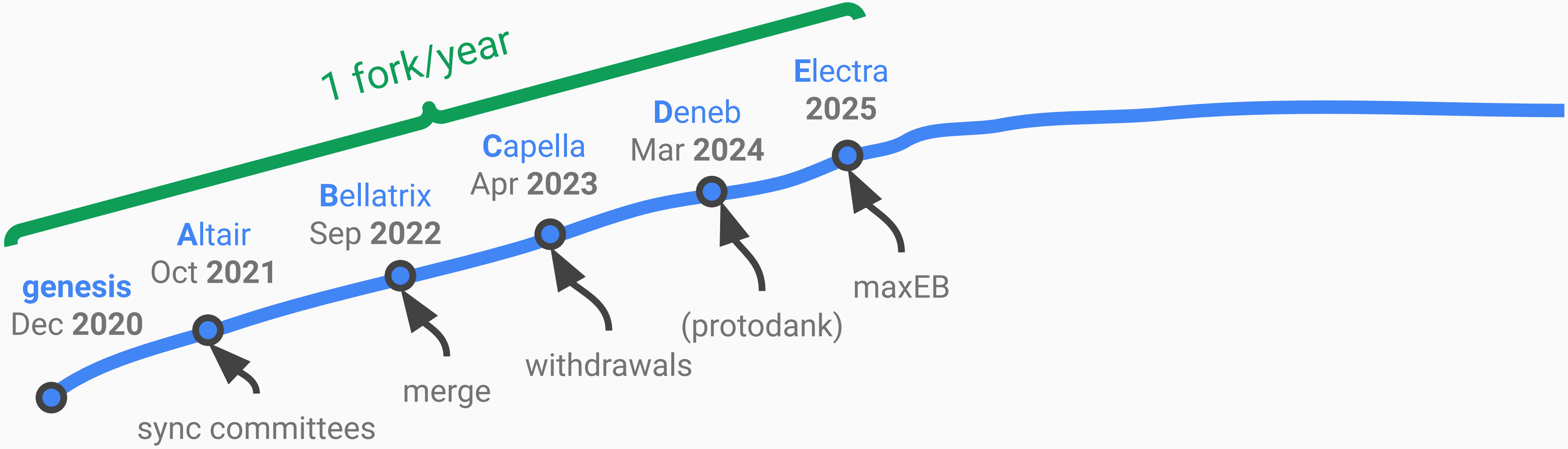


technical debt hindsight

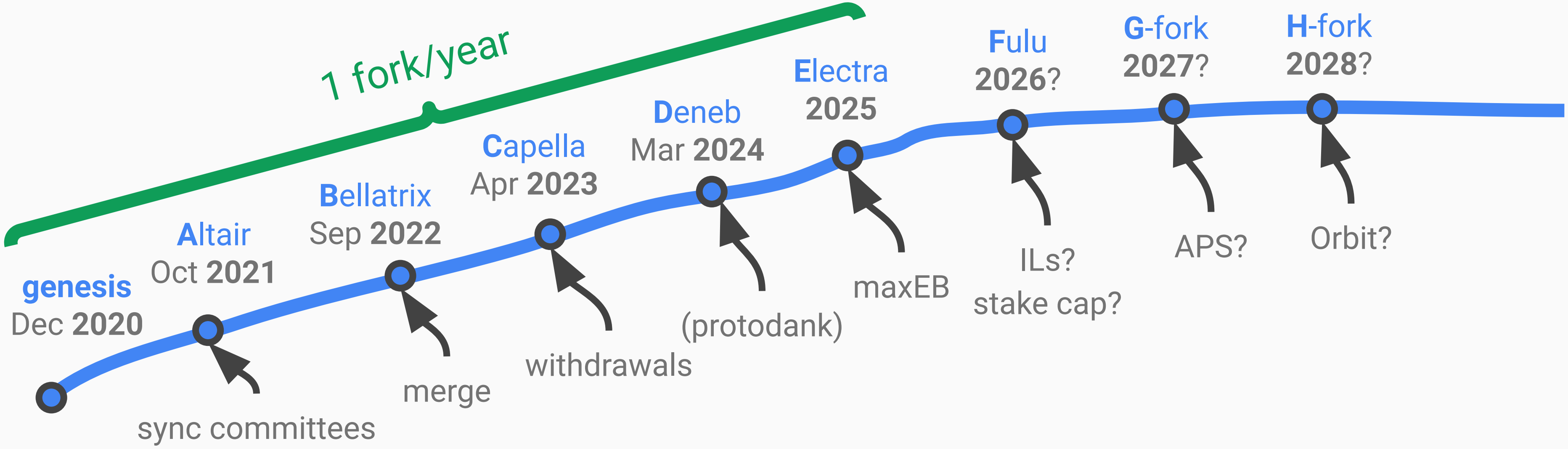
Devconnect staking gathering

Nov 2023, Istanbul (talk link)

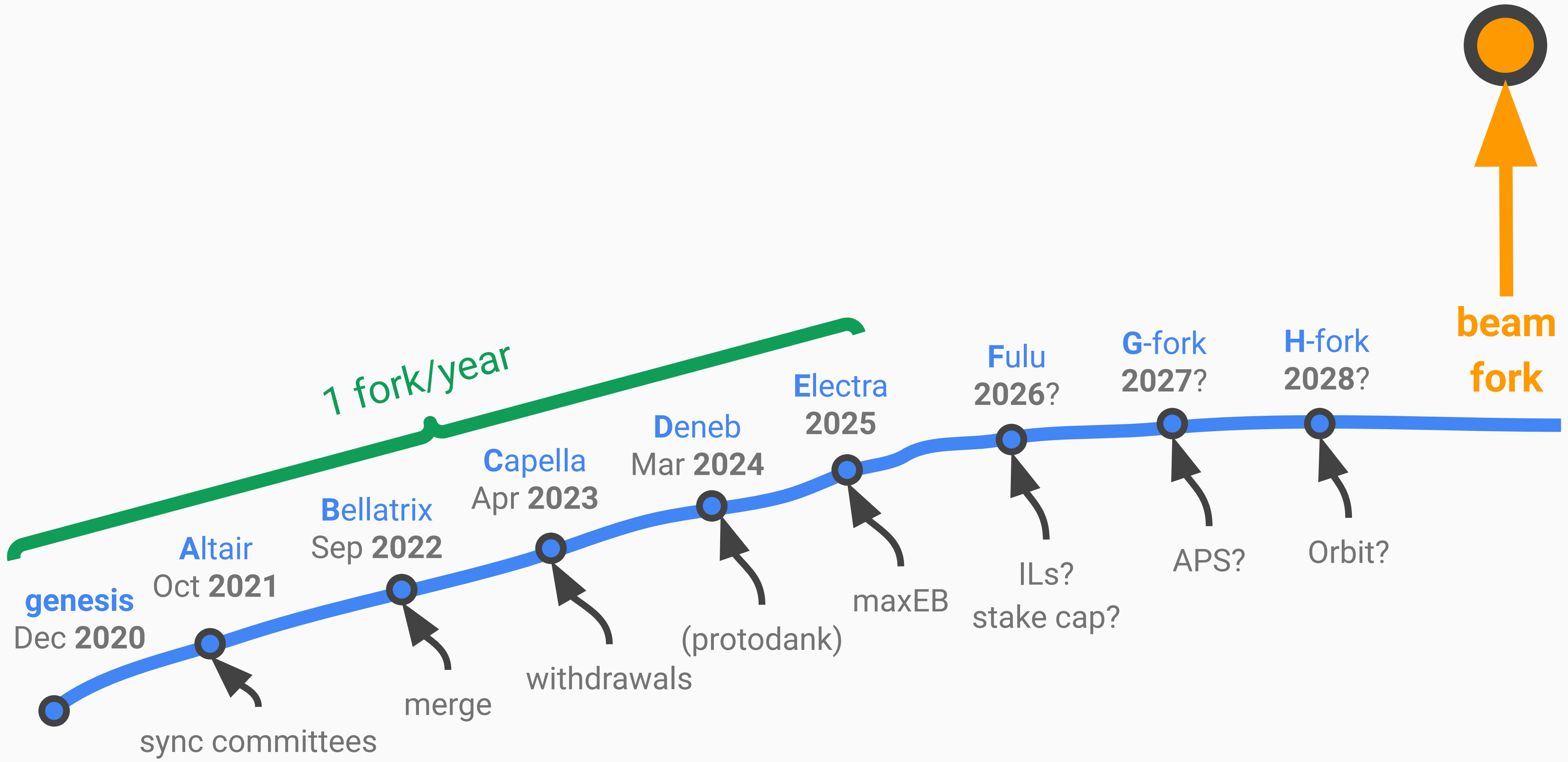
ossification accelerationism (oss/acc)



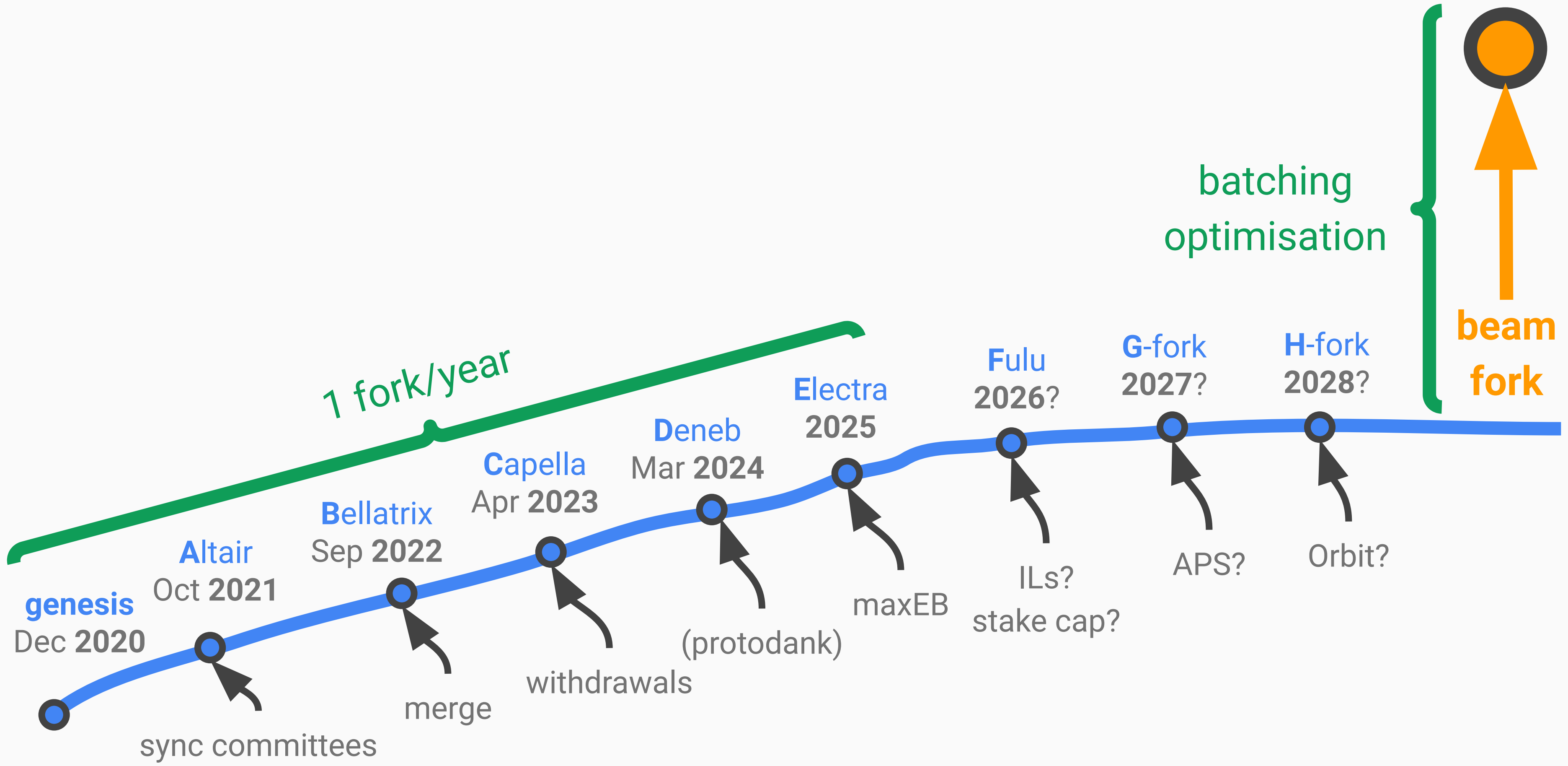
ossification accelerationism (oss/acc)



ossification accelerationism (oss/acc)



ossification accelerationism (oss/acc)

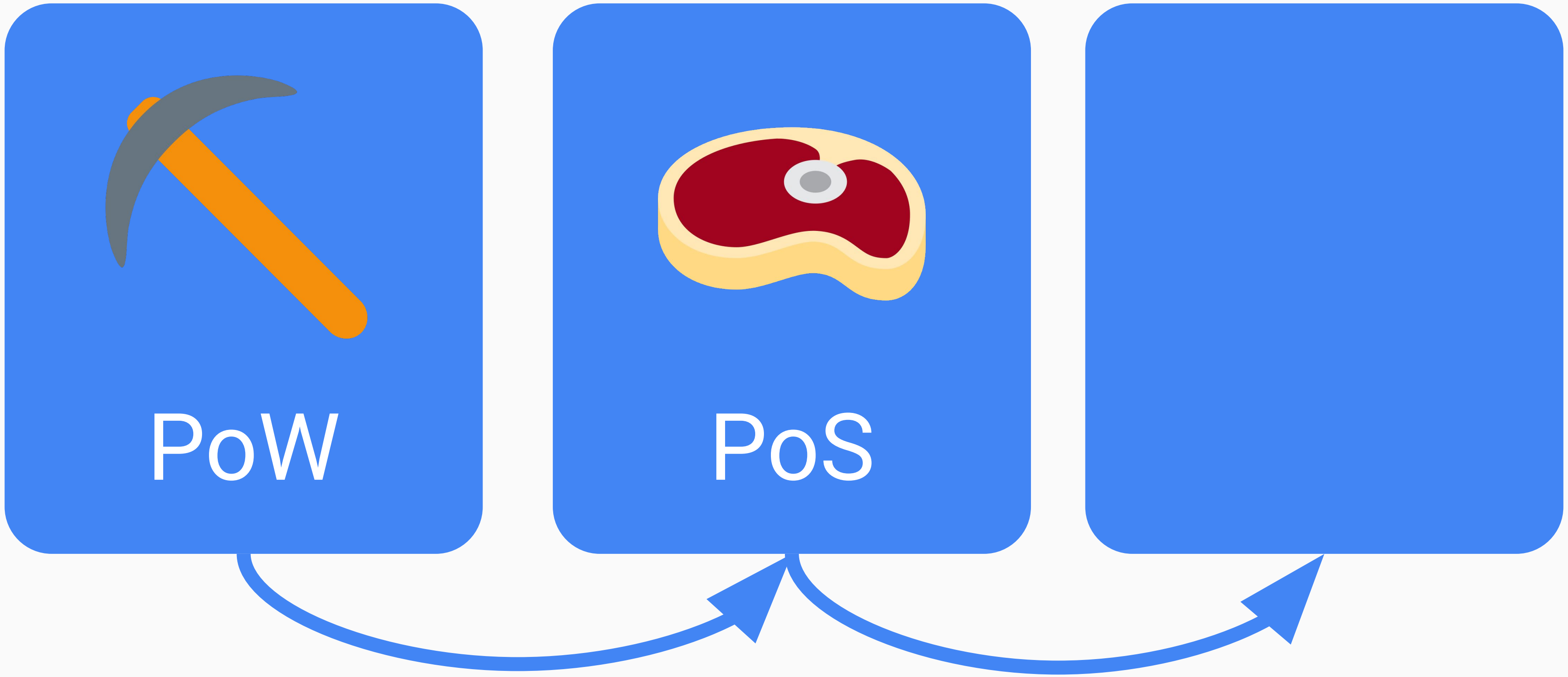


part 1—vision

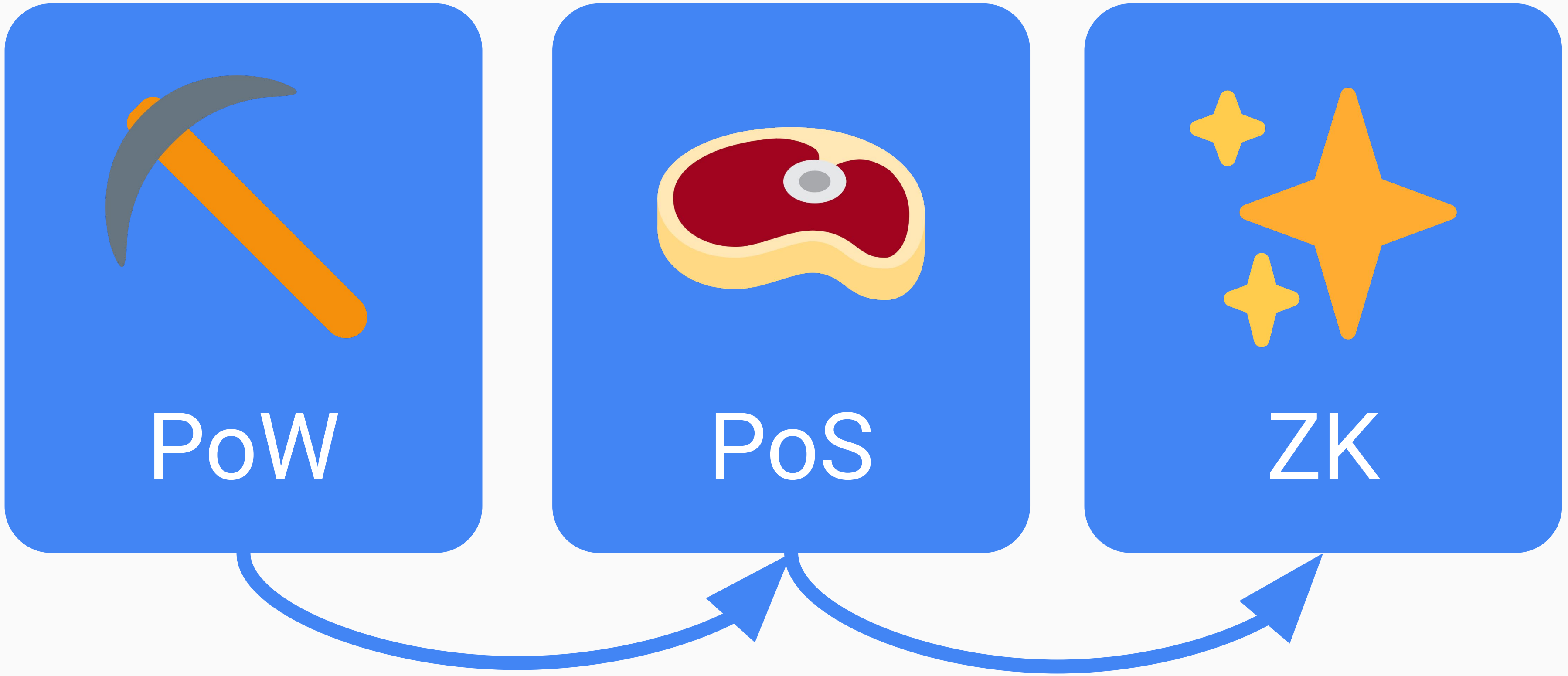
part 2—tech

part 3—plan

Ethereum eras

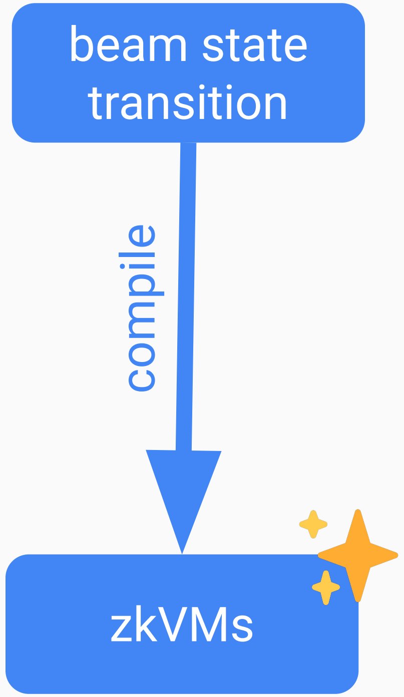


Ethereum eras

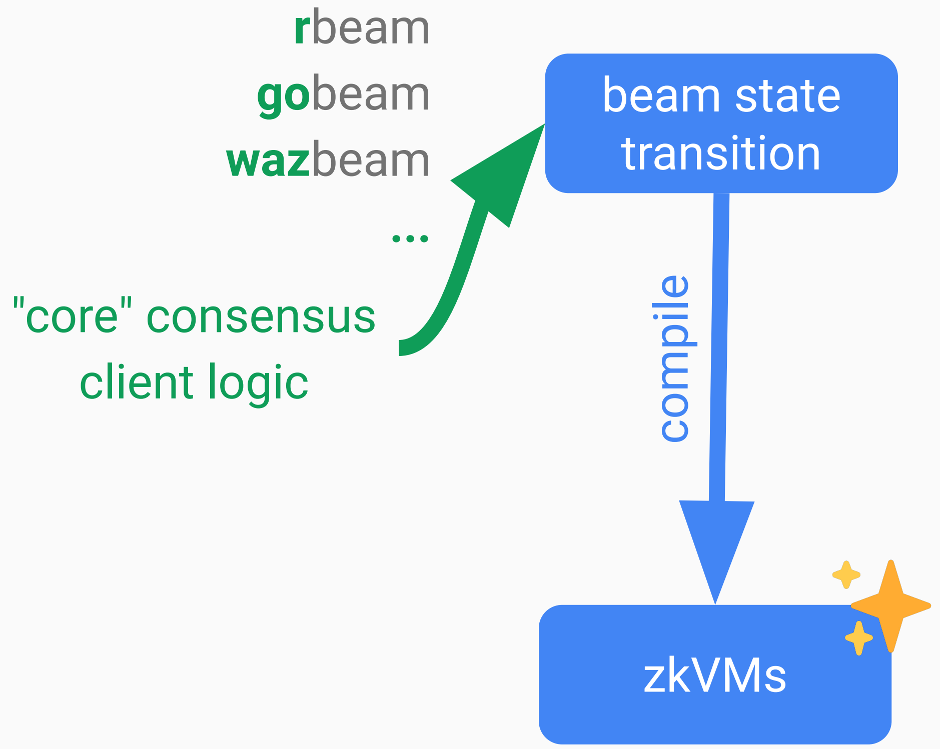


chain snarkification

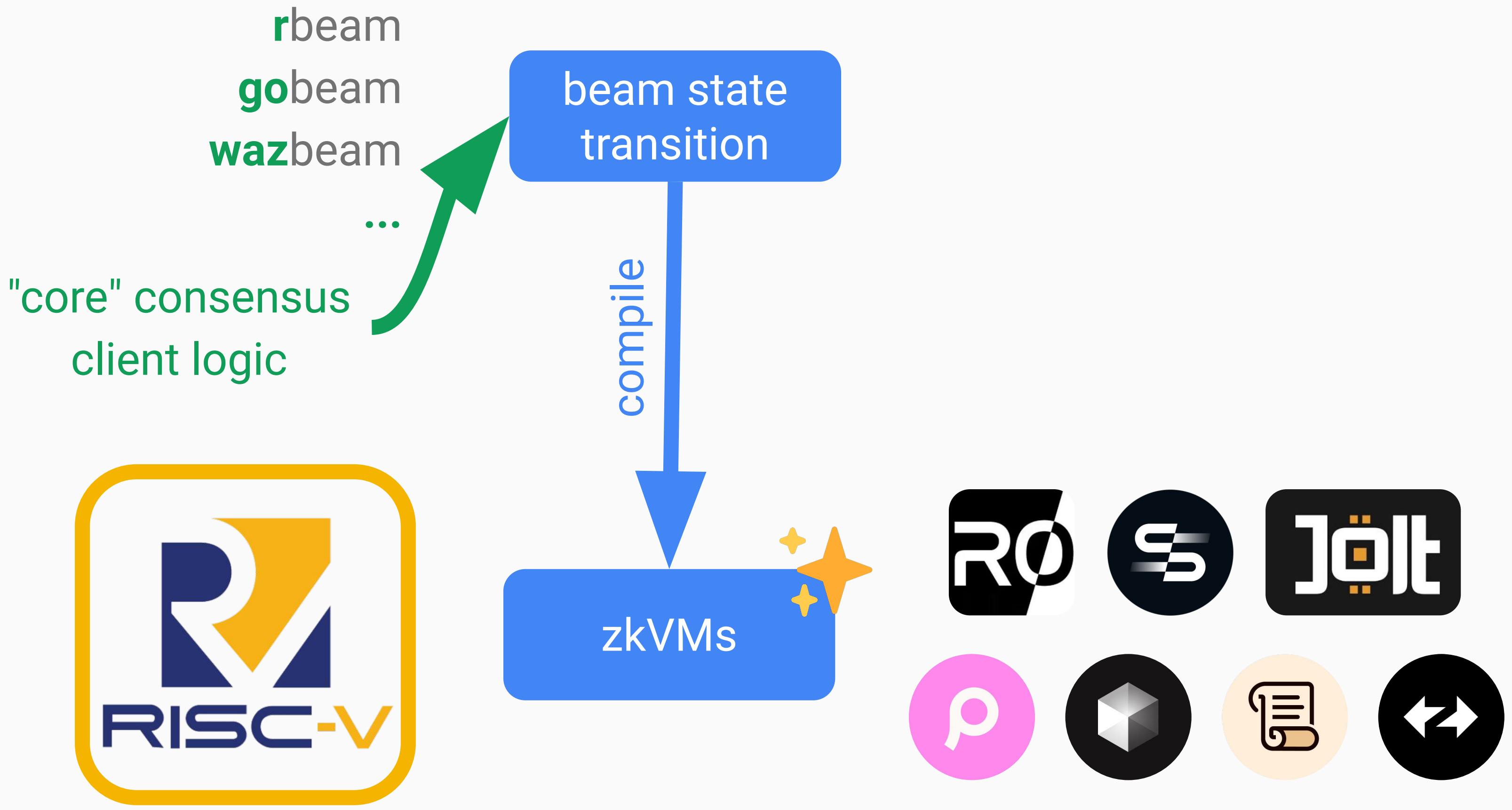
rbeam
gobeam
wazbeam
...



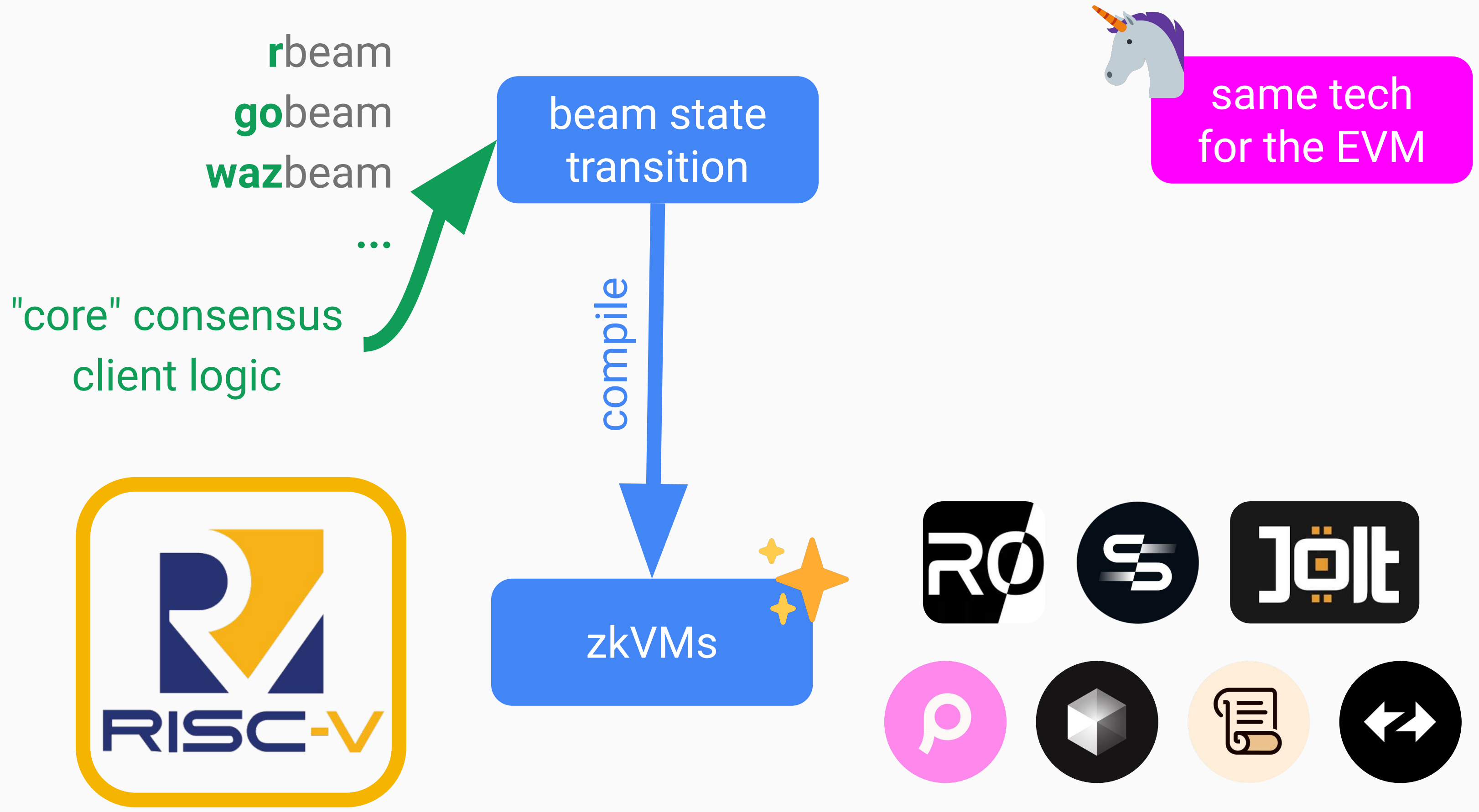
chain snarkification



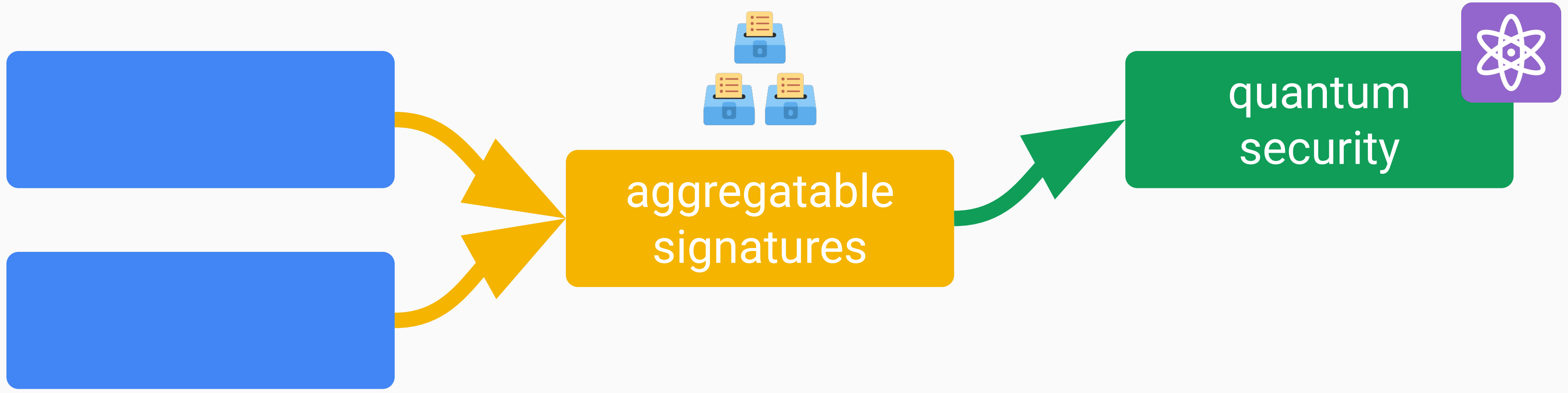
chain snarkification



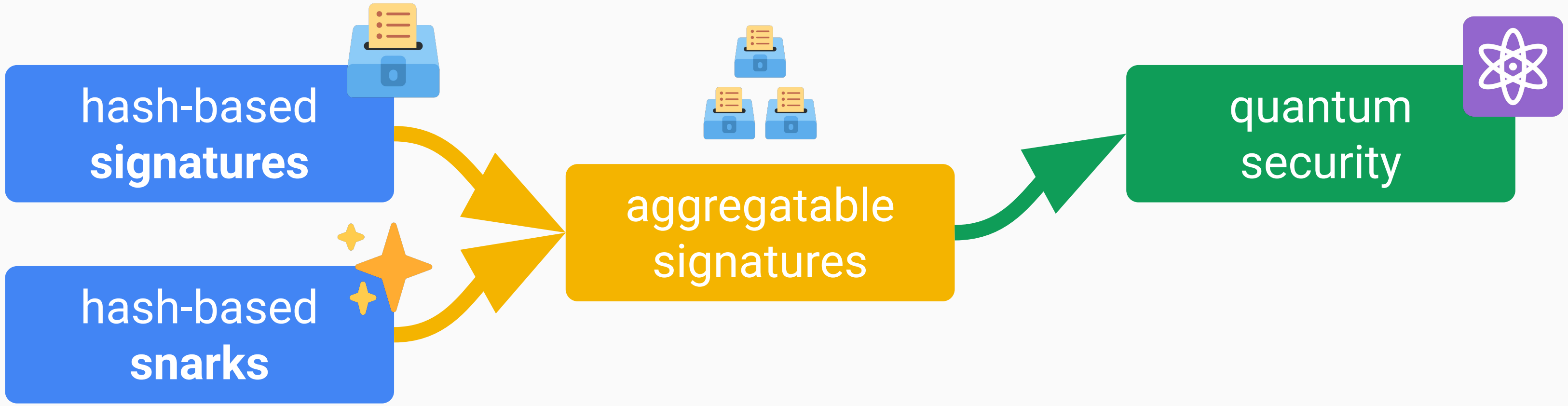
chain snarkification



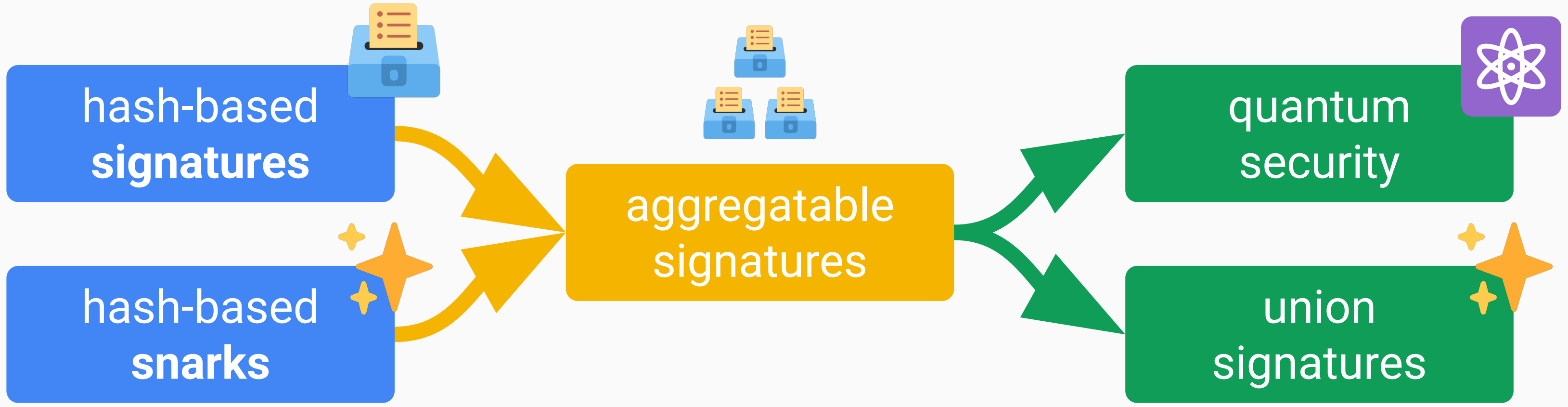
attestation snarkification



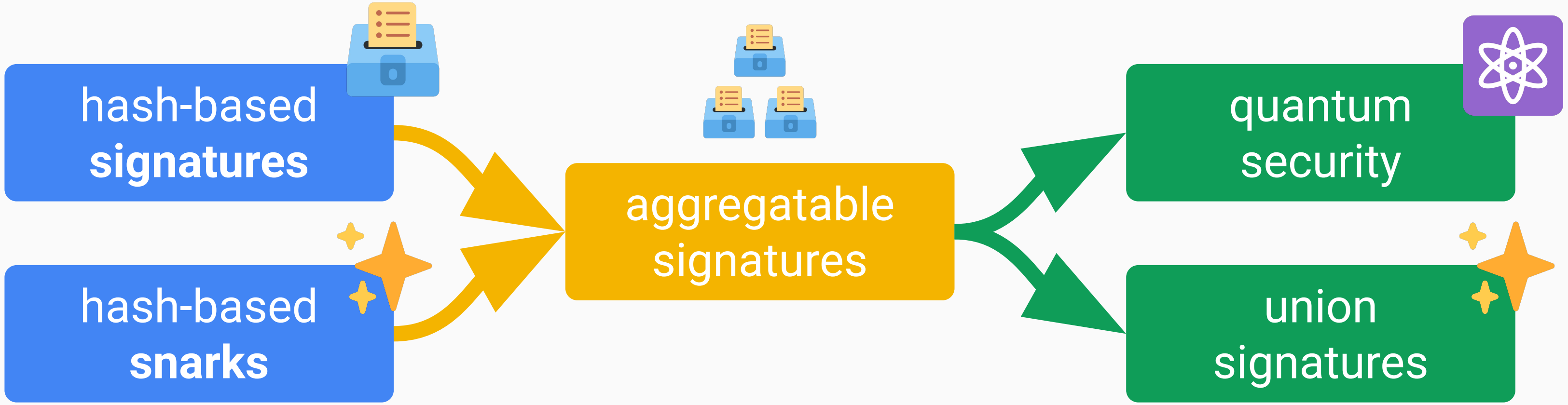
attestation snarkification

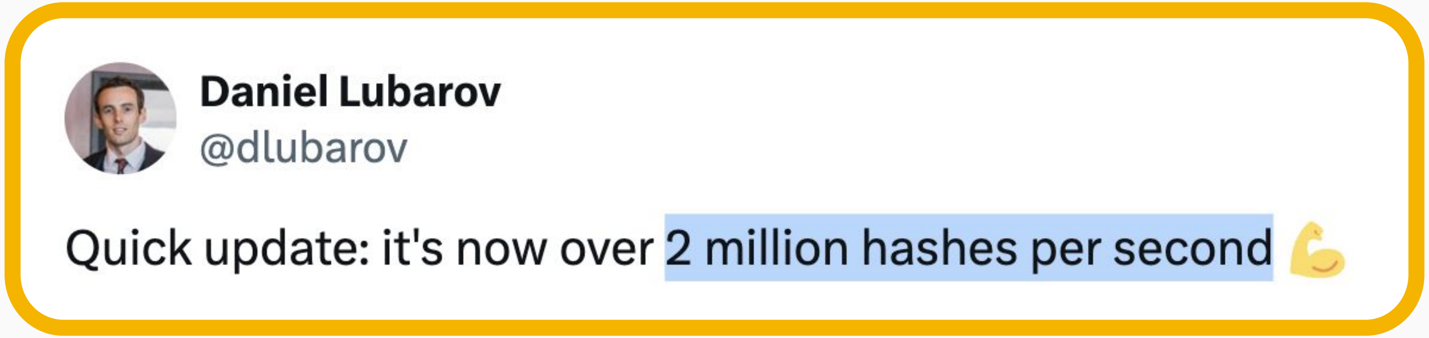


attestation snarkification



attestation snarkification




Daniel Lubarov
@dlubarov

Quick update: it's now over 2 million hashes per second 💪

reuse infrastructure

libp2p

SSZ

Simple**S**eriali**Z**e

reuse infrastructure

libp2p

PySpec

Python executable **Spec**

SSZ

Simple**S**eriali**Z**e

ProtocolGuild

leverage existing teams

CL devs

clientdiversity.org

EthPanda**Ops**

Protocol
Security
Research

leverage existing teams

CL devs

clientdiversity.org

EthPanda**Ops**

Protocol
Security
Research

Robust
Incentives
Group

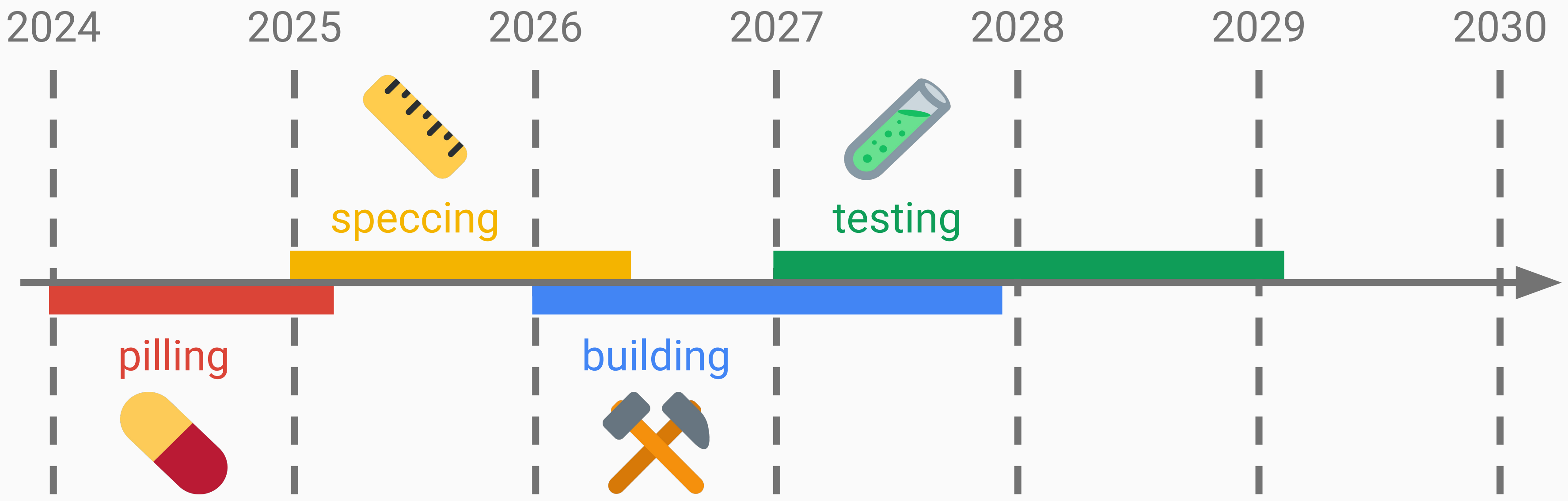
Applied
Research
Group

part 1—vision

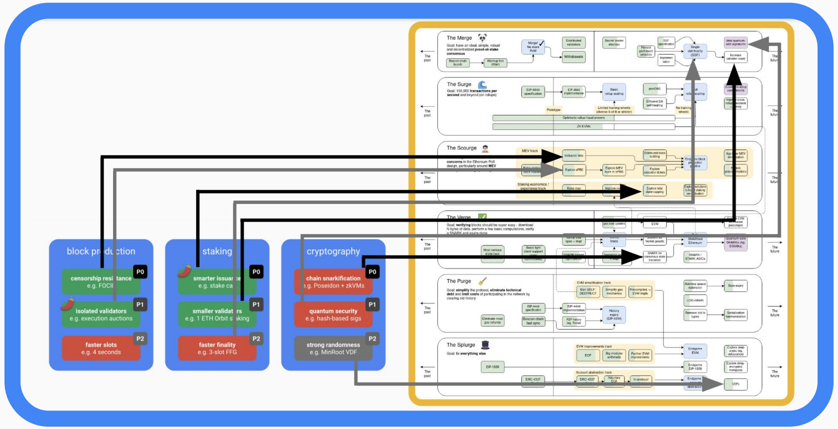
part 2—tech

part 3—plan

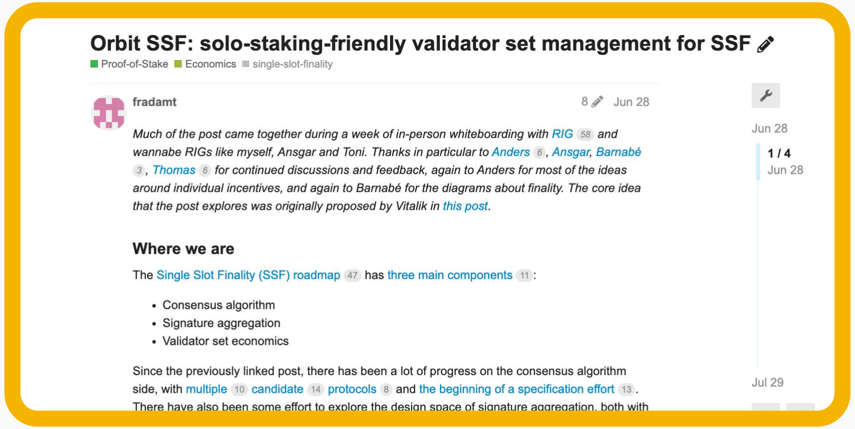
strawman timeline



2025 mission—"executable roadmap"


block production
censorship resistance e.g. FOCIL
P0
isolated validators e.g. execution auctions
P1
faster slots e.g. 4 seconds
P2
staking
smarter issuance e.g. stake cap
P0
smaller validators e.g. 1 ETH Orbit staking
P1
faster finality e.g. 3-slot FFG
P2
cryptography
chain snarkification e.g. Poseidon + zkVMs
P0
quantum security e.g. hash-based sigs
P1
strong randomness e.g. MinRoot VDF
P2
The Merge
The Surge
The Scourge
The Verge
The Purge
The Splurge

pixels



Orbit SSF: solo-staking-friendly validator set management for SSF

Proof-of-Stake Economics single-slot-finality

fradamt 8 Jun 28

Much of the post came together during a week of in-person whiteboarding with RIG 58 and wannabe RIGs like myself, Ansgar and Toni. Thanks in particular to Anders 6, Ansgar, Barnabé 3, Thomas 6 for continued discussions and feedback, again to Anders for most of the ideas around individual incentives, and again to Barnabé for the diagrams about finality. The core idea that the post explores was originally proposed by Vitalik in this post.

Where we are

The Single Slot Finality (SSF) roadmap 47 has three main components 11:

- Consensus algorithm
- Signature aggregation
- Validator set economics

Since the previously linked post, there has been a lot of progress on the consensus algorithm side, with multiple 10 candidate 14 protocols 8 and the beginning of a specification effort 13.

There have also been some effort to explore the design space of signature aggregation, both with

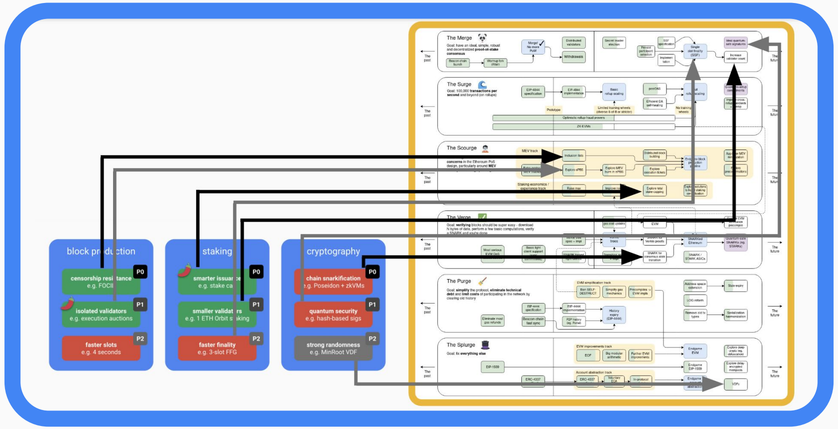
Jun 28

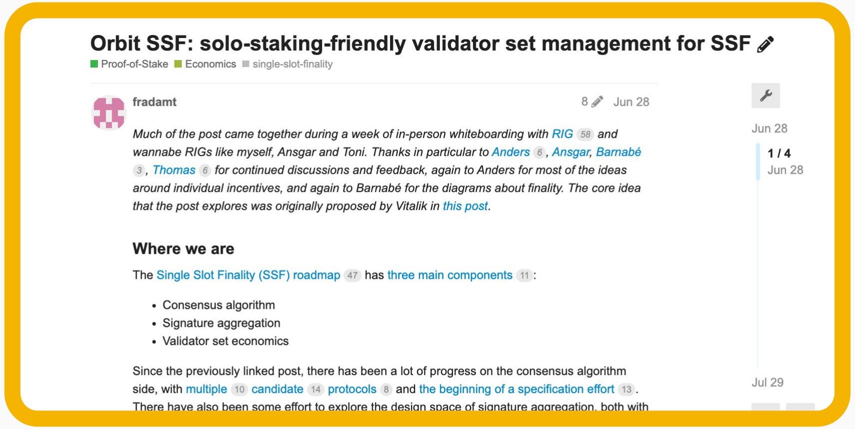
1 / 4
Jun 28

Jul 29

words

2025 mission—"executable roadmap"


block production
censorship resistance e.g. FOCIL
P0
isolated validators e.g. execution auctions
P1
faster slots e.g. 4 seconds
P2
staking
smarter issuance e.g. stake cap
P0
smaller validators e.g. 1 ETH Orbit staking
P1
faster finality e.g. 3-slot FFG
P2
cryptography
chain snarkification e.g. Poseidon + zkVMs
P0
quantum security e.g. hash-based sigs
P1
strong randomness e.g. MinRoot VDF
P2
The Merge
The Surge
The Scourge
The Verge
The Purge
The Splurge



Orbit SSF: solo-staking-friendly validator set management for SSF

Proof-of-Stake · Economics · single-slot-finality

fradamt 8 Jun 28

Much of the post came together during a week of in-person whiteboarding with RIG 58 and wannabe RIGs like myself, Ansgar and Toni. Thanks in particular to Anders 6, Ansgar, Barnabé 3, Thomas 6 for continued discussions and feedback, again to Anders for most of the ideas around individual incentives, and again to Barnabé for the diagrams about finality. The core idea that the post explores was originally proposed by Vitalik in this post.

Where we are

The Single Slot Finality (SSF) roadmap 47 has three main components 11:

- Consensus algorithm
- Signature aggregation
- Validator set economics

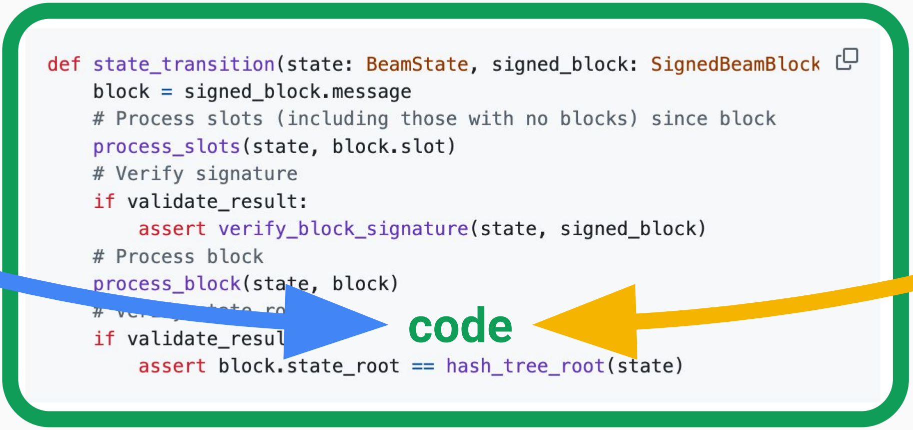
Since the previously linked post, there has been a lot of progress on the consensus algorithm side, with multiple 10 candidate 14 protocols 8 and the beginning of a specification effort 13.

There have also been some effort to explore the design space of signature aggregation, both with

Jun 28
1 / 4
Jun 28
Jul 29

pixels

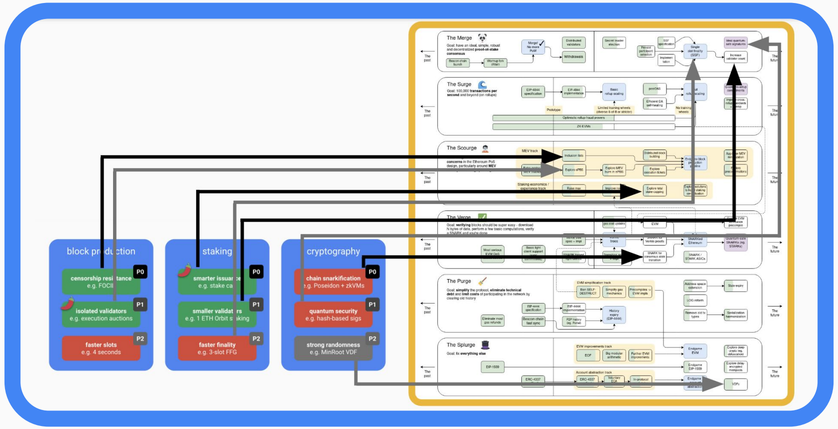
words

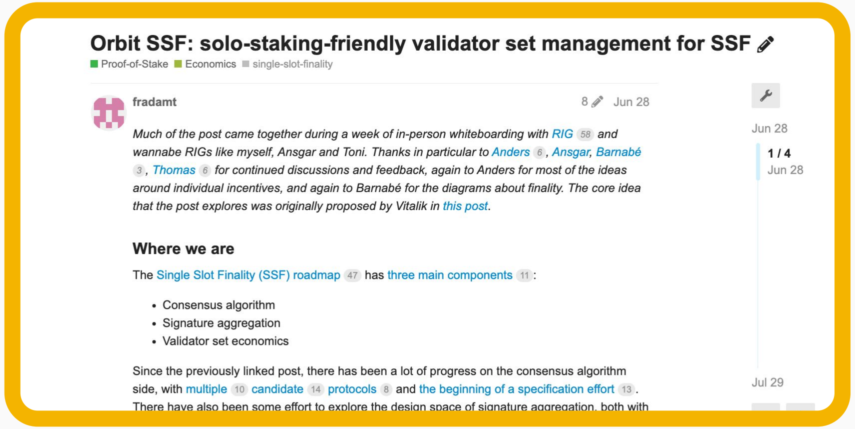


```
def state_transition(state: BeamState, signed_block: SignedBeamBlock
    block = signed_block.message
    # Process slots (including those with no blocks) since block
    process_slots(state, block.slot)
    # Verify signature
    if validate_result:
        assert verify_block_signature(state, signed_block)
    # Process block
    process_block(state, block)
    # Ver[illegible]tate ro[illegible]
    if validate_resul[illegible]
        assert block.state_root == hash_tree_root(state)
```

code

2025 mission—"executable roadmap"





Orbit SSF: solo-staking-friendly validator set management for SSF
Proof-of-Stake Economics single-slot-finality

fradamt 8 Jun 28

Much of the post came together during a week of in-person whiteboarding with RIG 58 and wannabe RIGs like myself, Ansgar and Toni. Thanks in particular to Anders 6, Ansgar, Barnabé 3, Thomas 6 for continued discussions and feedback, again to Anders for most of the ideas around individual incentives, and again to Barnabé for the diagrams about finality. The core idea that the post explores was originally proposed by Vitalik in this post.

Where we are

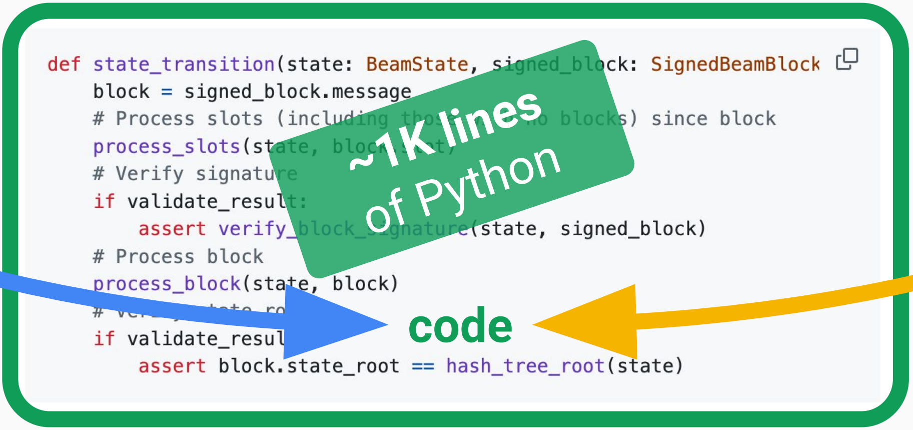
The Single Slot Finality (SSF) roadmap 47 has three main components 11:

- Consensus algorithm
- Signature aggregation
- Validator set economics

Since the previously linked post, there has been a lot of progress on the consensus algorithm side, with multiple 10 candidate 14 protocols 8 and the beginning of a specification effort 13.
There have also been some effort to explore the design space of signature aggregation, both with

Jun 28
1 / 4
Jun 28
Jul 29

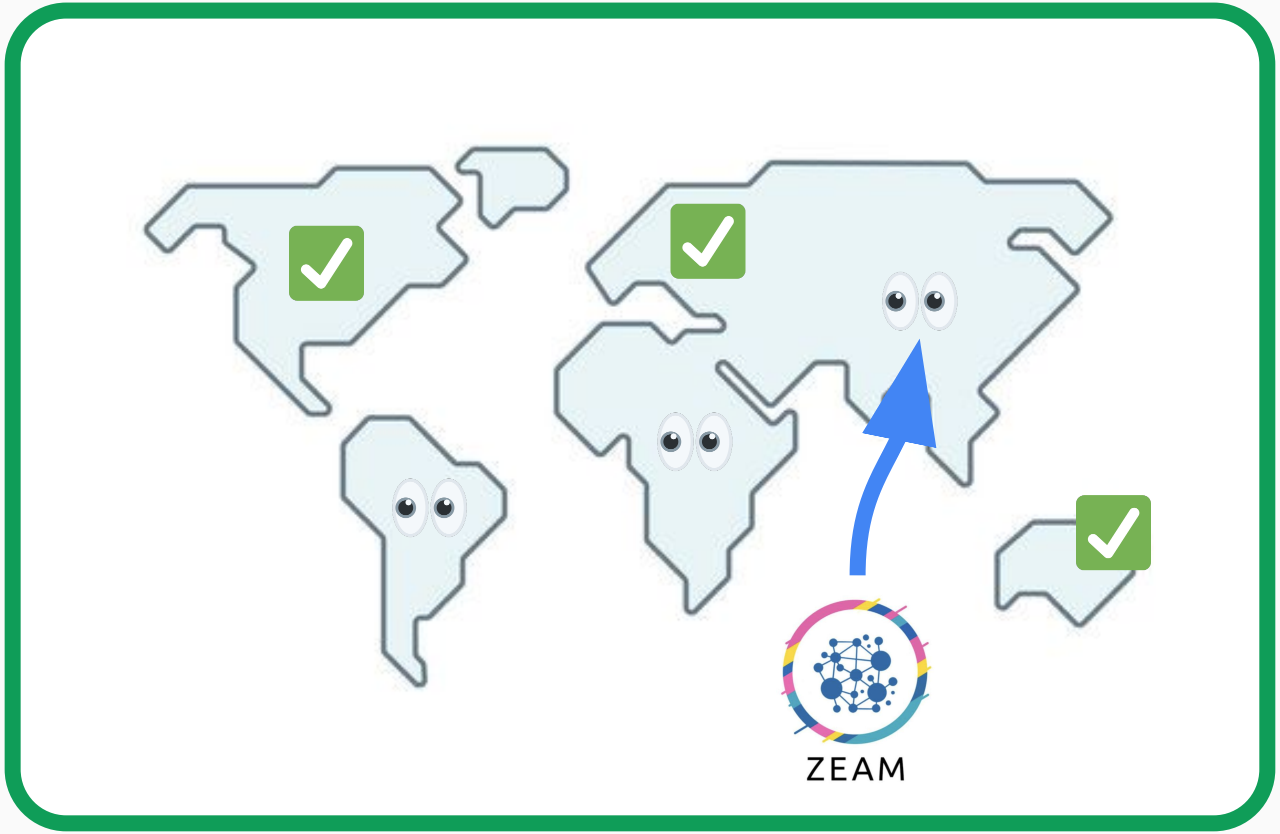
pixels **words**



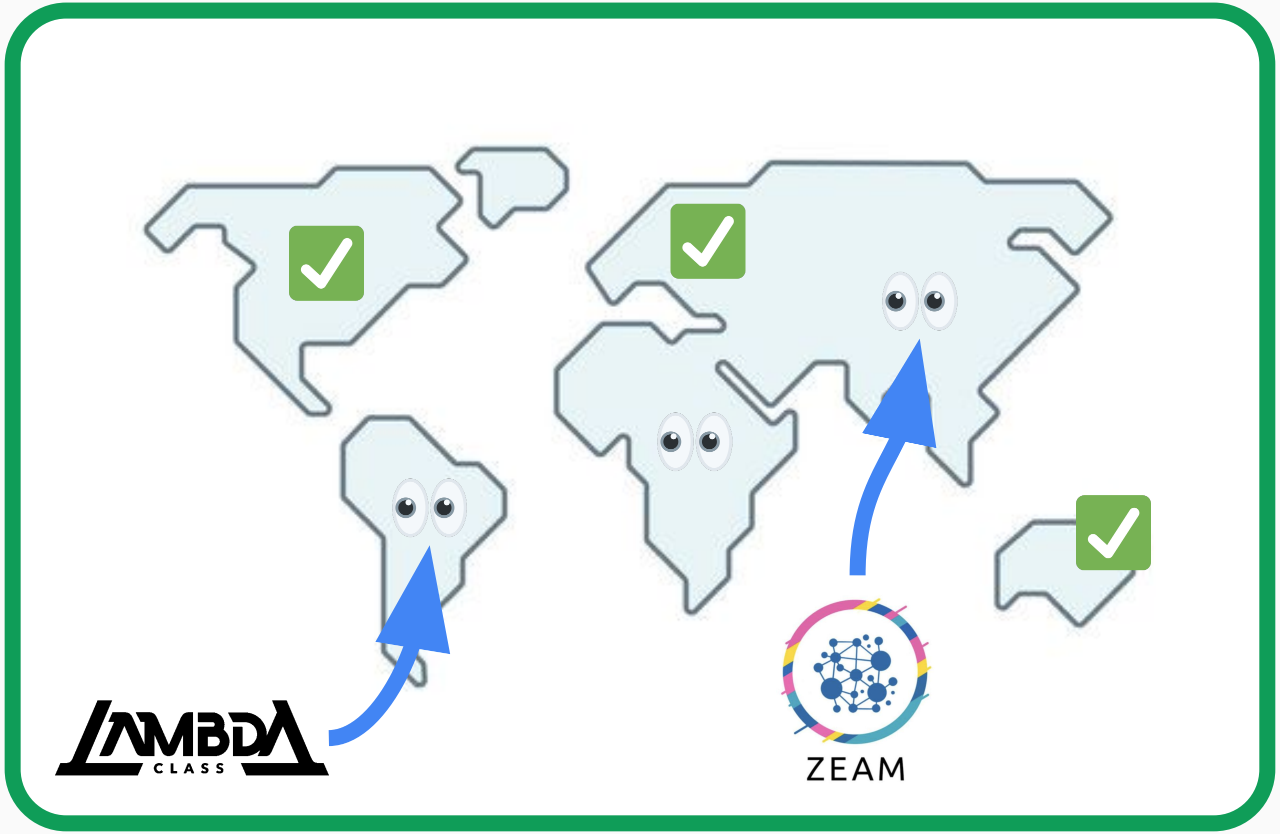
```
def state_transition(state: BeamState, signed_block: SignedBeamBlock
    block = signed_block.message
    # Process slots (including those with no blocks) since block
    process_slots(state, block.slot)
    # Verify signature
    if validate_result:
        assert verify_block_signature(state, signed_block)
    # Process block
    process_block(state, block)
    # Verify state root
    if validate_result:
        assert block.state_root == hash_tree_root(state)
```

onboarding opportunity

onboarding opportunity



onboarding opportunity



beam up

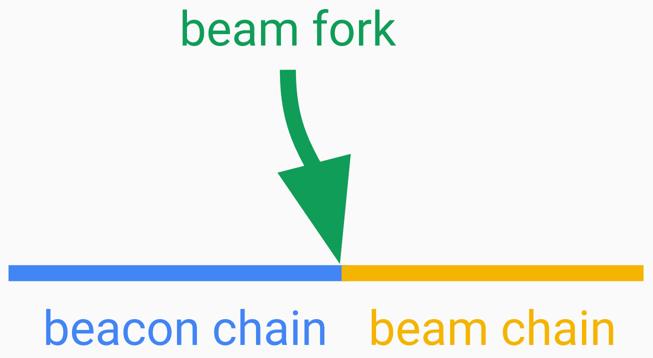
- speccing
- networking
- coordination
- snarks
- client devs

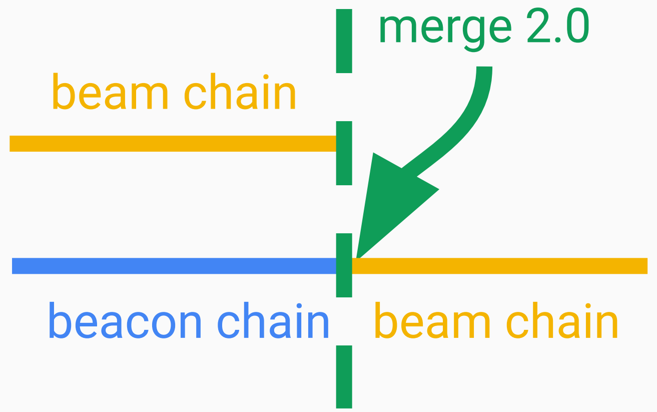
beam.chain@ethereum.org

thank you :)

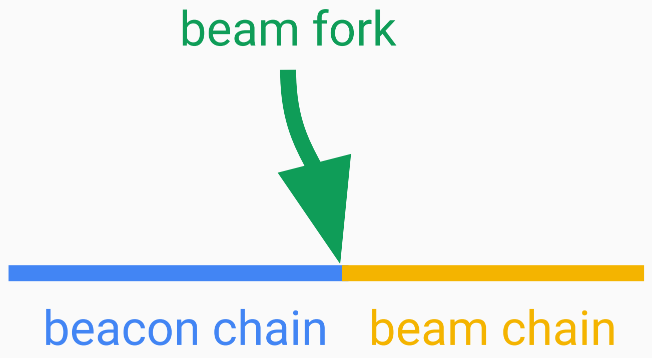
beam.chain@ethereum.org

to merge or not to merge?

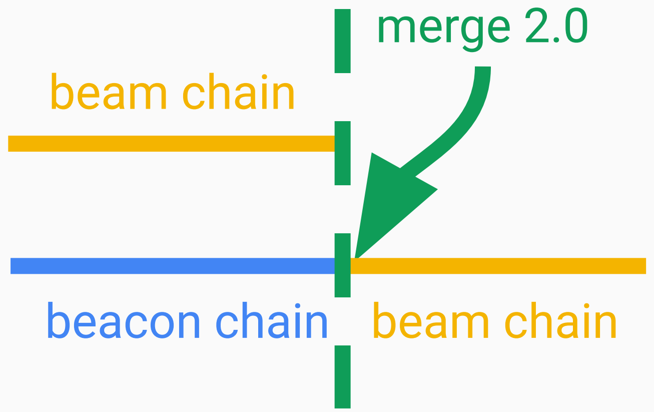




to merge or not to merge?



- same validator set
- same issuance curve
- day-1 withdrawals
- simpler, speedier
- compute light
- reuse client shell
- tax friendly?



- "incentivised testnet"
- available if required

beam staking

zen staking

execution auctions

- no relays
- no timing games
- no MEV spikes
- no preconfs

fish staking

Orbit staking

- 1 ETH validators
- millions of vals
- less pooling

feather staking

chain snarkification

- synchless
- stateless
- computeless

RISC-V support

✓

- official
  - RV64
- tinygo
  - RV32?

- OpenJDK
  - RV64

G extension

- integer (I)
- floating (F, D)
- mul/div (M)
- atomic (A)

embrace incrementalism—political derisking

block production

EIP-7805

censorship resistance
e.g. FOCIL

isolated validators
e.g. execution auctions

- no MEV for stakers
- no execution self-building
- agentic vs deterministic

staking

EIP soon™

smarter issuance
e.g. stake cap

cryptography

consensus devs wanted