
Ethereum Consensus:
The Road to Faster Finality
Yann Vonlanthen
Ethereum Foundation

Ethereum Consensus:
The Road to Faster Finality

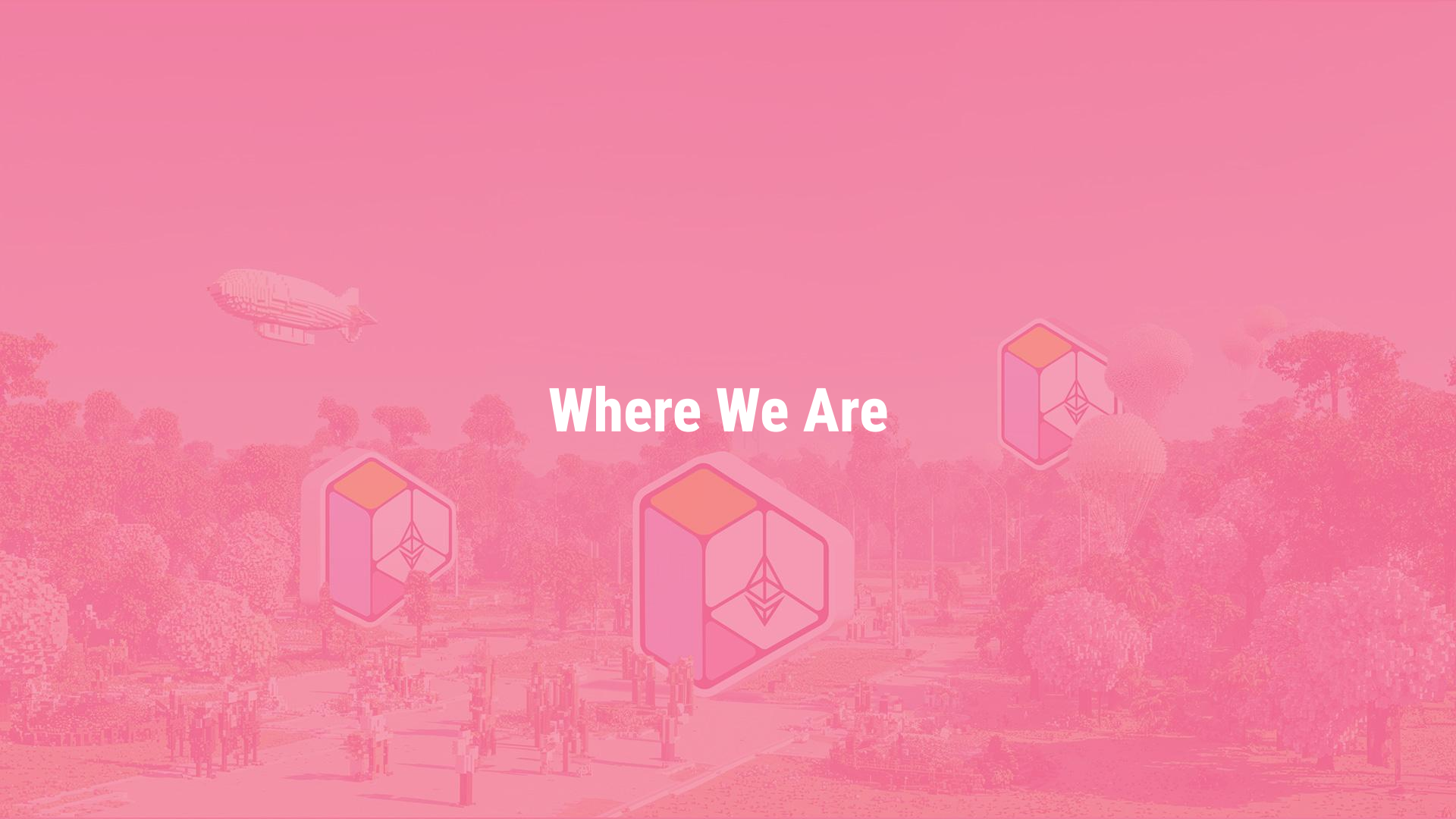
Yann Vonlanthen
Consensus Team @ Ethereum Foundation

Francesco D'Amato

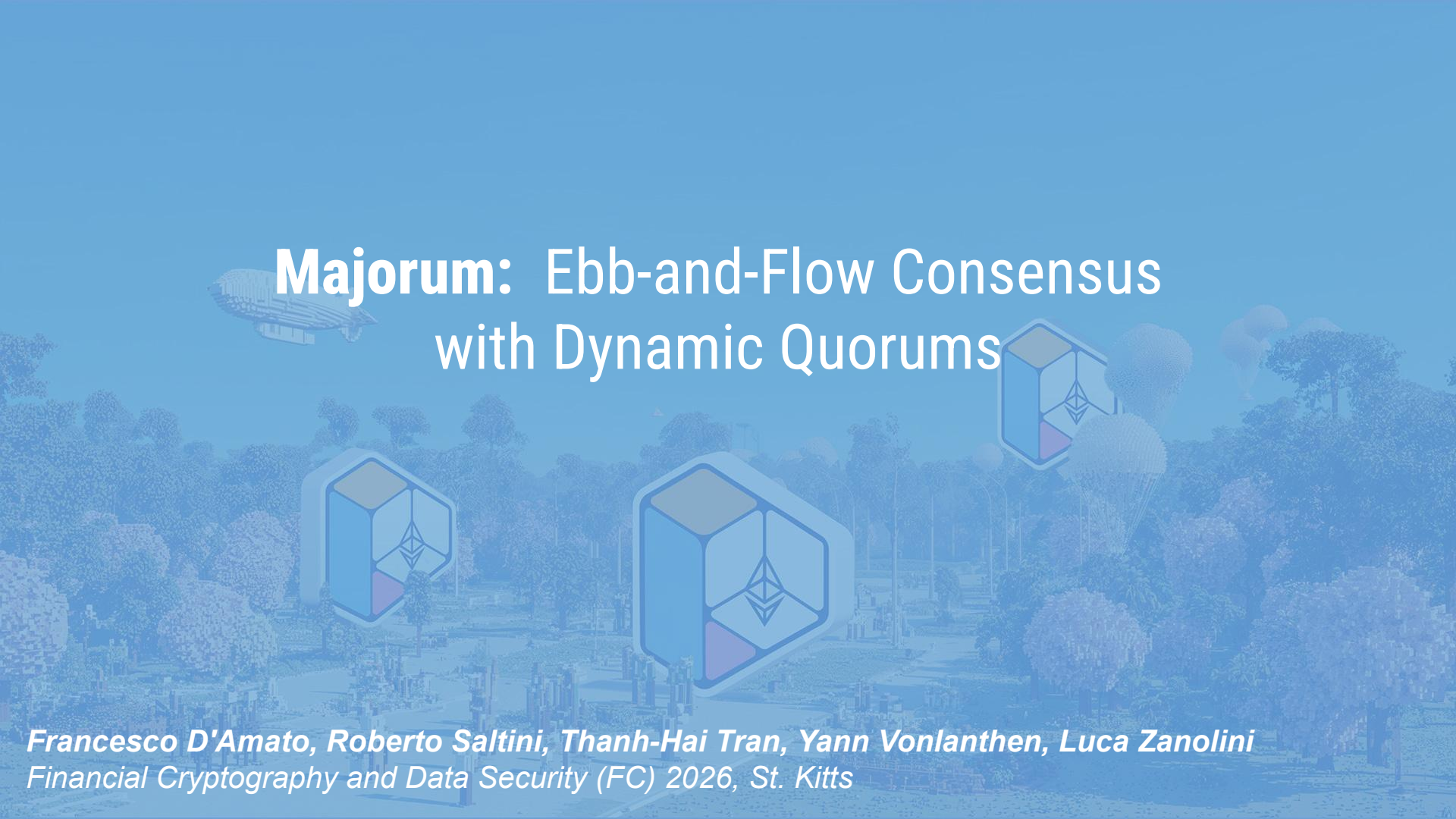
Roberto Saltini

Luca Zanolini

Where We Are

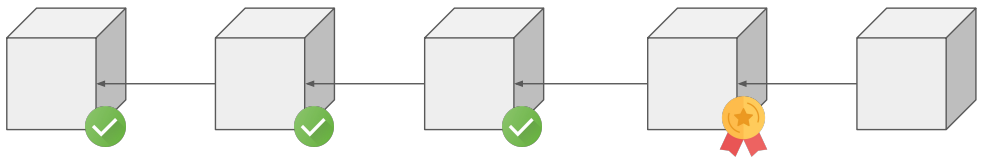


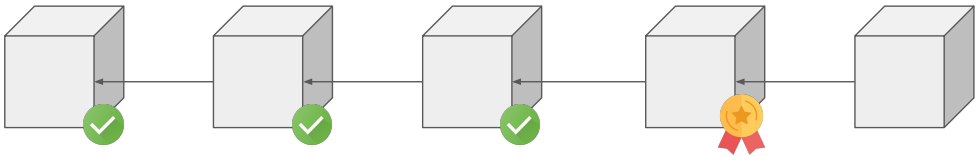
Majorum: Ebb-and-Flow Consensus with Dynamic Quorums



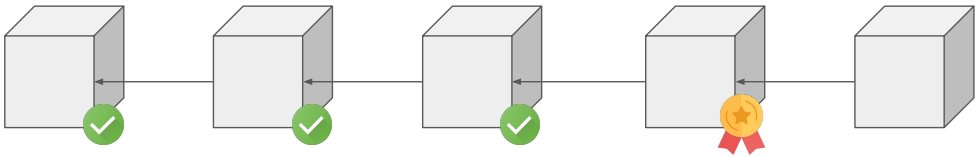
Francesco D'Amato, Roberto Saltini, Thanh-Hai Tran, Yann Vonlanthen, Luca Zanolini

Financial Cryptography and Data Security (FC) 2026, St. Kitts

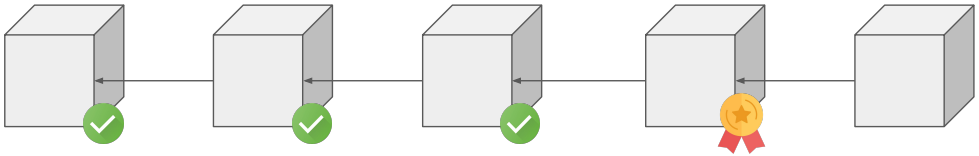




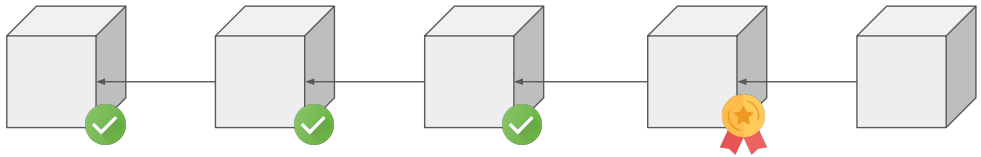

aws

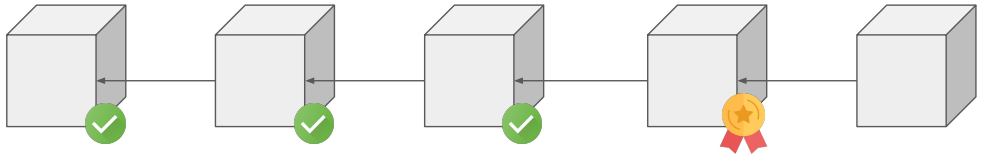


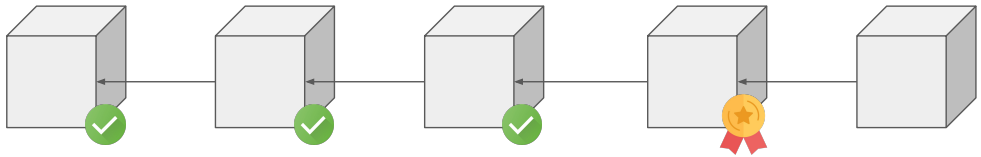

aws

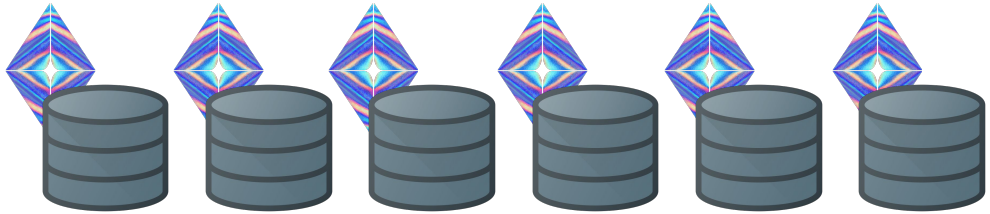


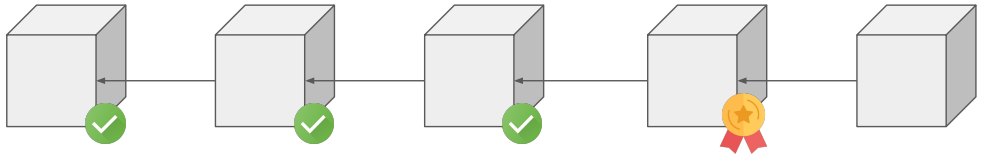

aws

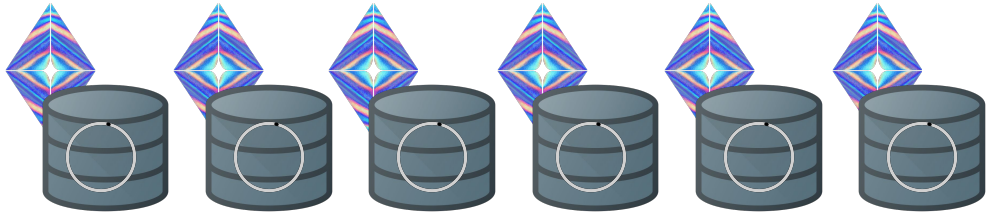


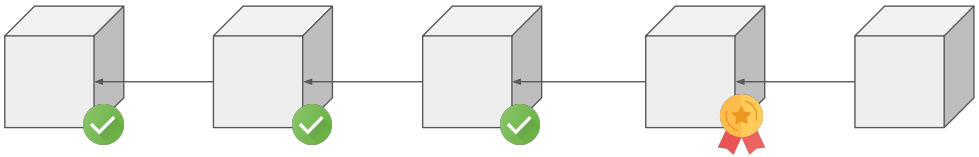


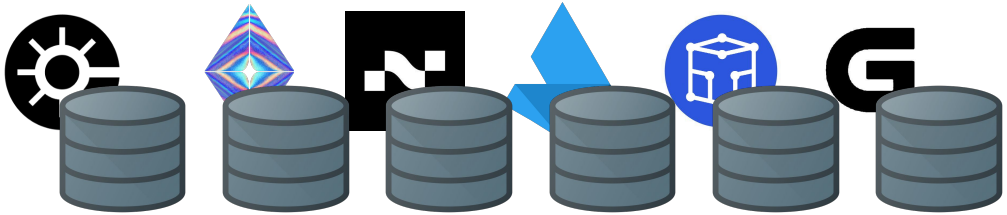


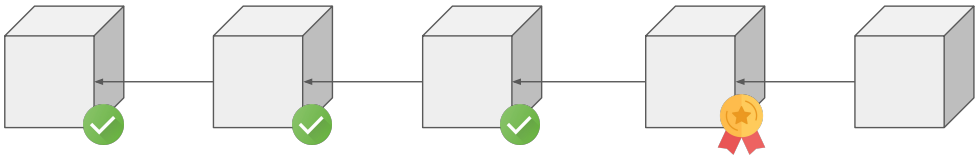


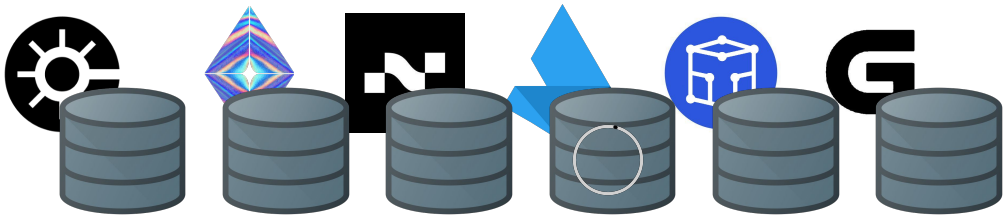


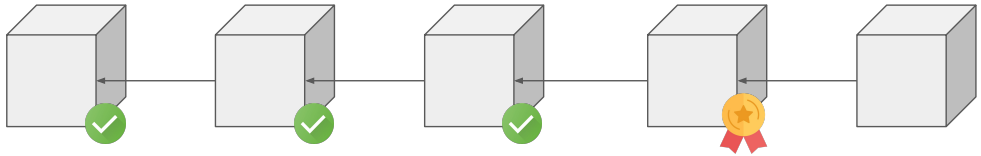


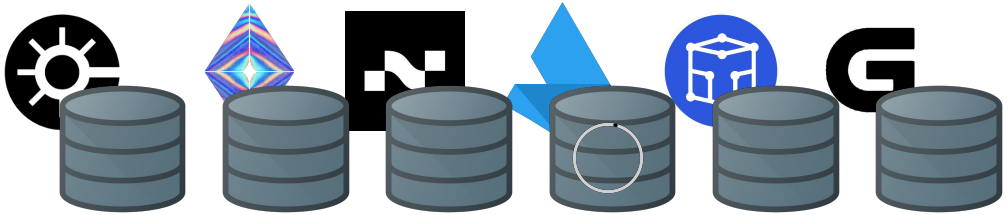










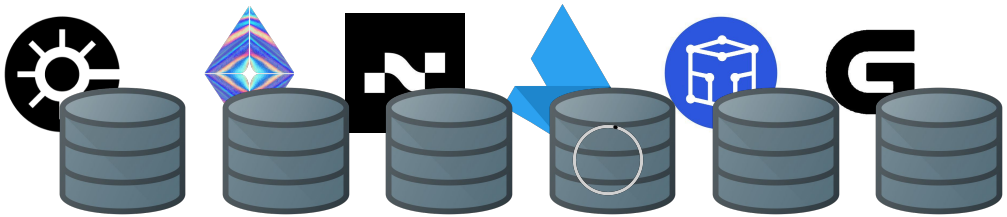


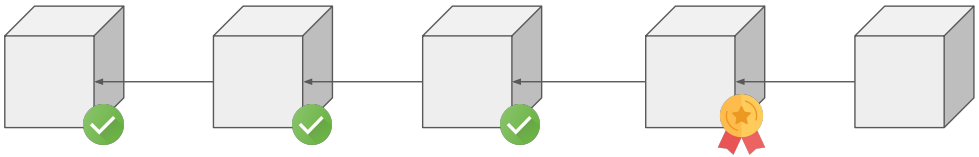
Validator Activity in Ethereum Drops 25% Following Fusaka Deployment

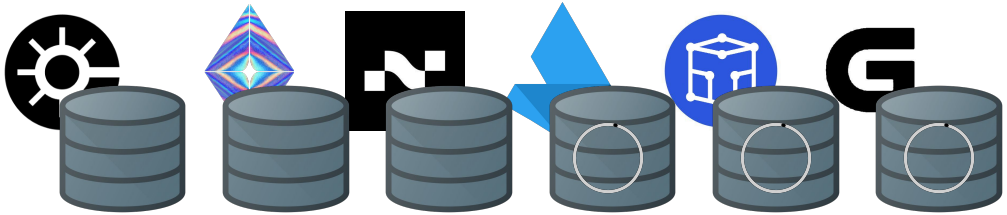
Validator activity in Ethereum drops 25% following Fusaka deployment.

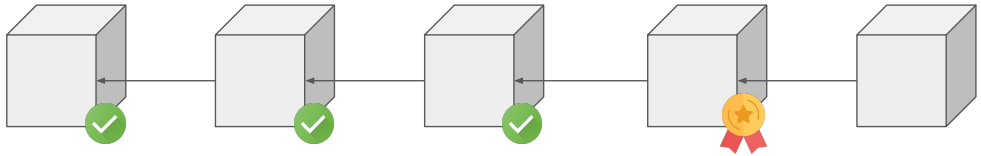
05.12.2025 • ForkLog

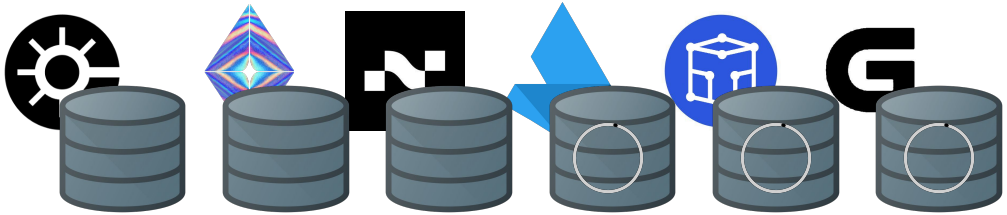
Shortly after the deployment of Fusaka, a malfunction occurred in the popular consensus client Prysm, disabling a portion of Ethereum validators.



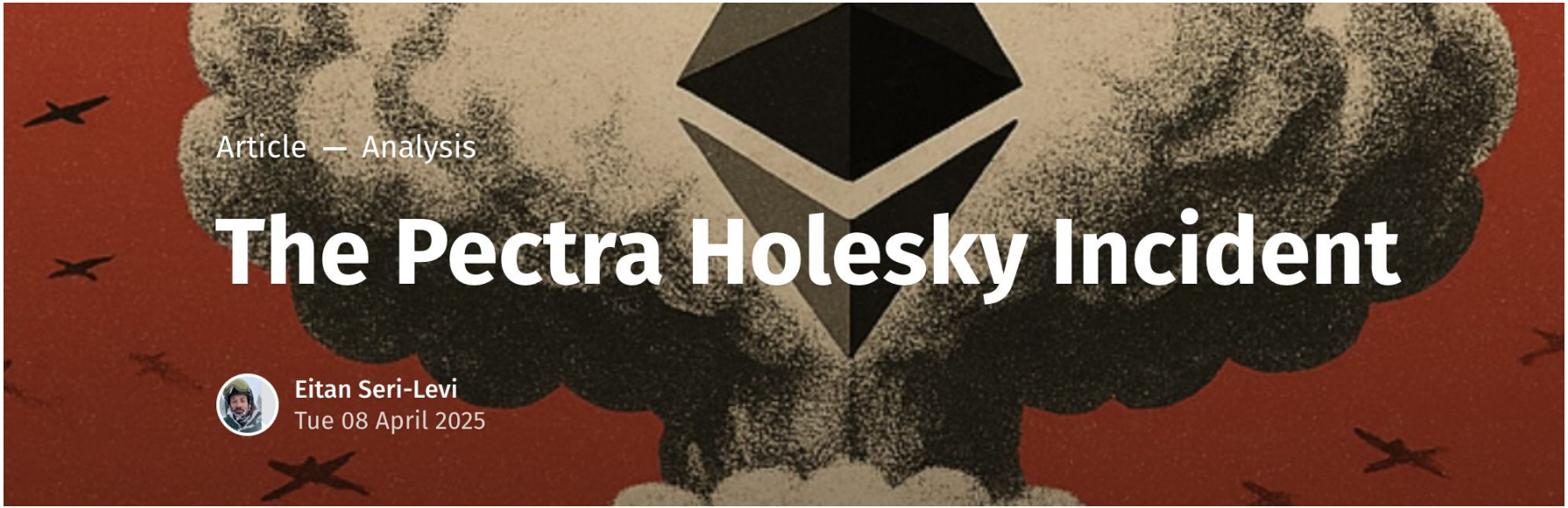


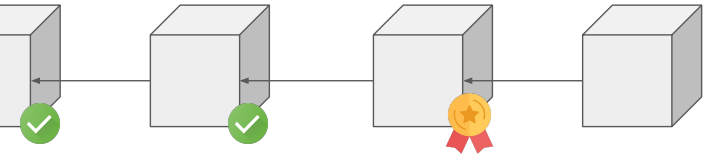


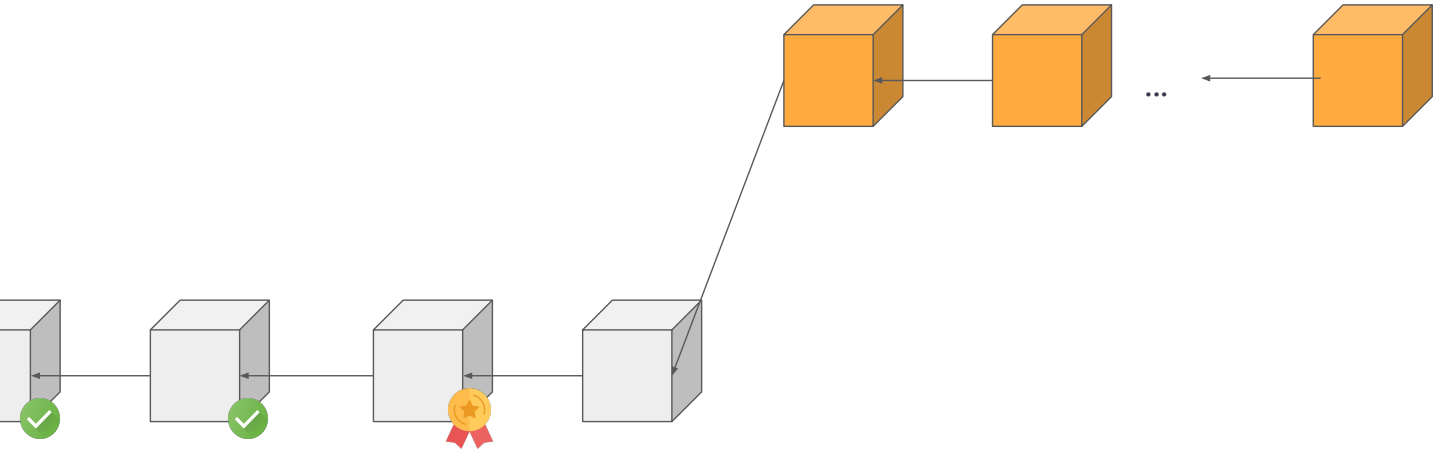


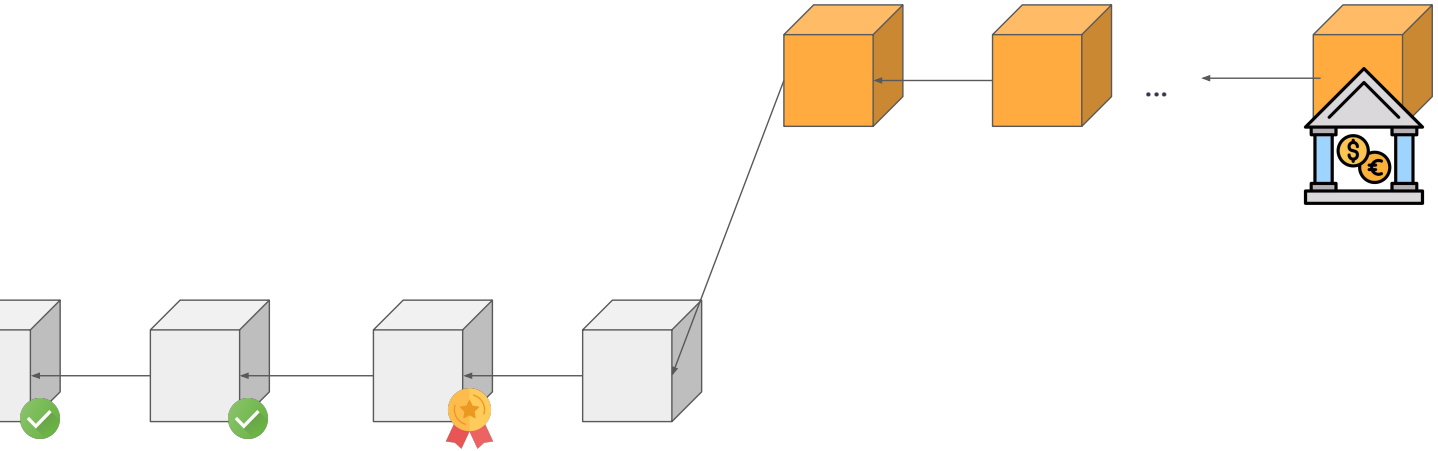


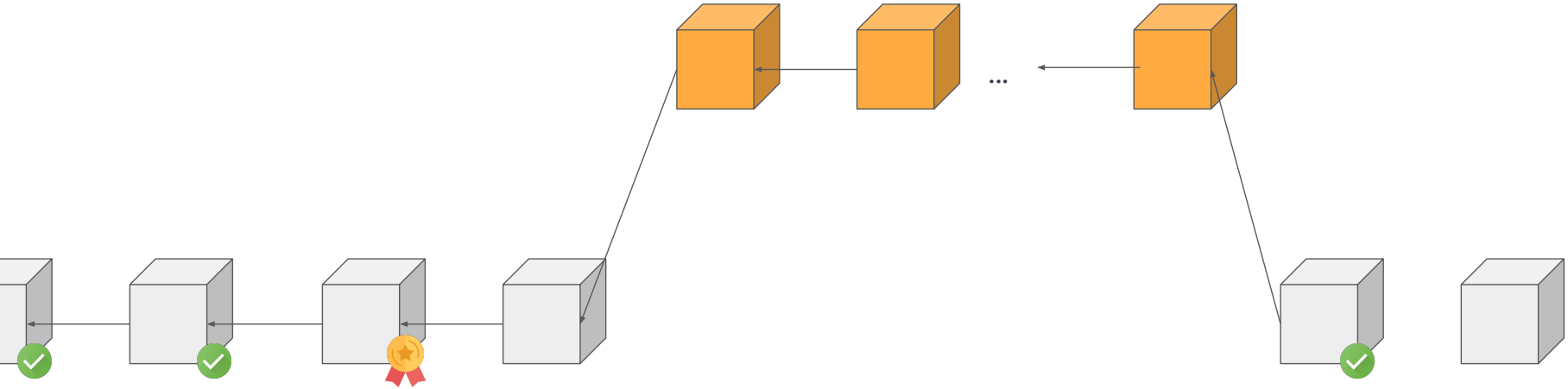
https://blog.sigmaprime.io/pectra-holesky-incident.html

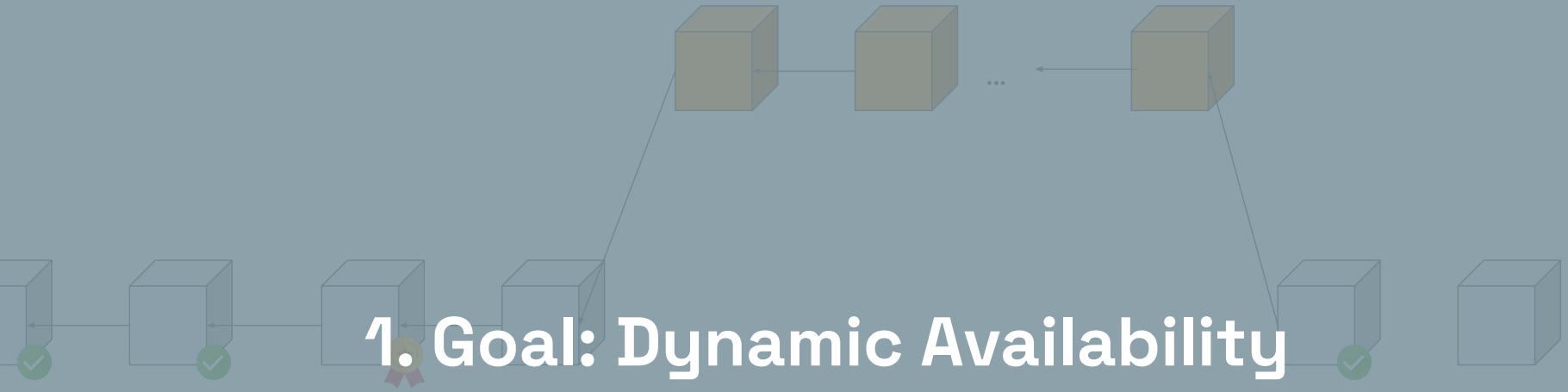












1. Goal: Dynamic Availability

Why Ethereum Needs a Dynamically Available Protocol

Consensus

luca_zanolini 2 12d

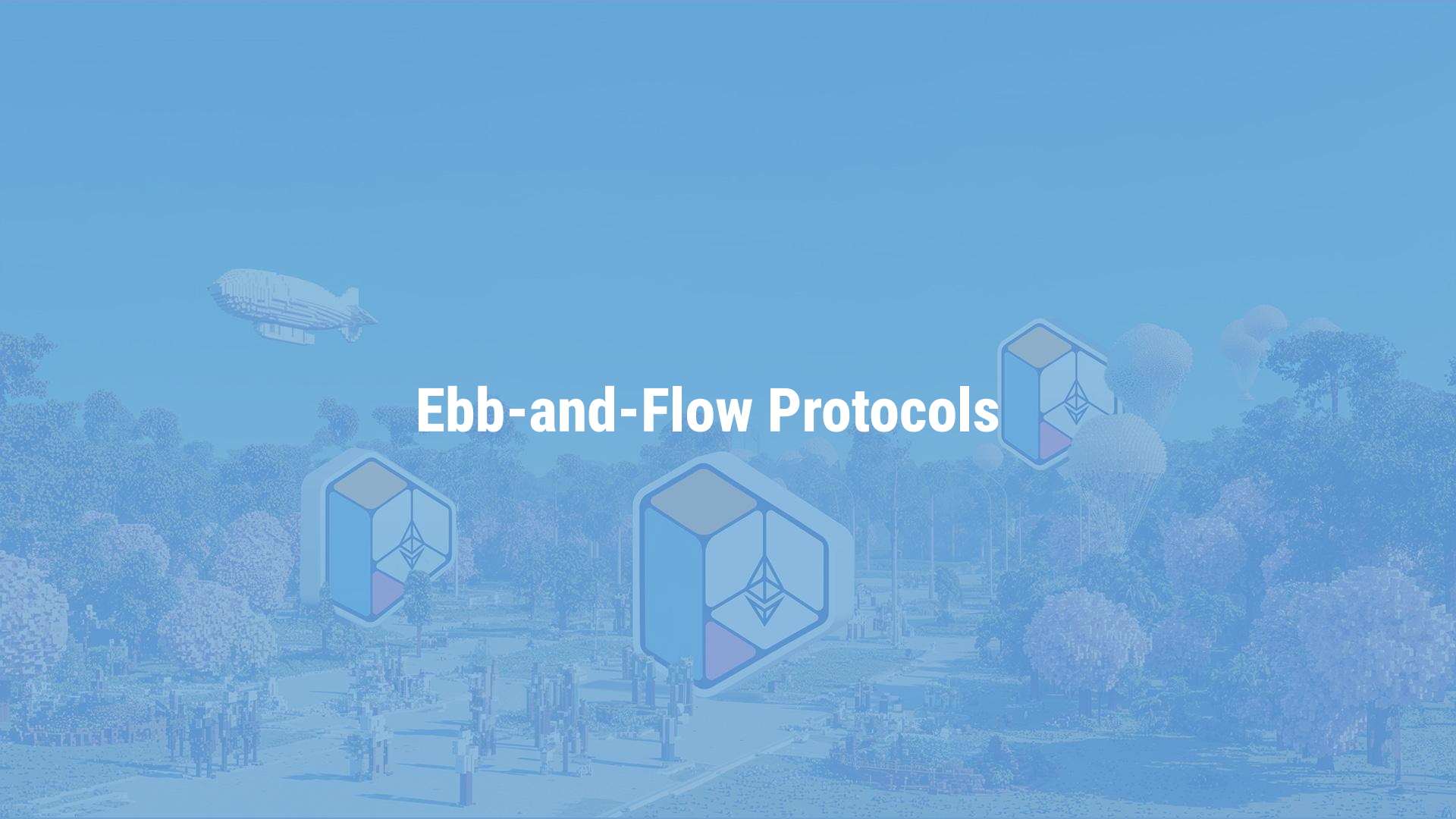
Author: Luca Zanolini 9

Huge thanks to Ben 9 *, Francesco* 3 *, Joachim* 4 *, Justin* 6 *, Mikhail* 3 *, Roberto* 1 *, Thomas* 3 *, Vitalik* 1 *, and Yann* 4 *for their feedback.*

We are working on a proposal for the next consensus protocol for Ethereum. A central piece of the new design is a two-layer architecture: a fast available chain — the *heartbeat* — produced by a small randomly-sampled committee, and a separate finality mechanism that trails behind, finalizing blocks the heartbeat has already produced — crucially, with the two layers fully **decoupled**, unlike the current Gasper 2 design where LMD-GHOST and Casper FFG interact in ways that have proven difficult to reason about 6 . Vitalik outlined this direction in a recent post 11 .

This post focuses on the first layer — the heartbeat — and on a property we believe should be a strict requirement for it: **dynamic availability**.

Ebb-and-Flow Protocols



Ebb-and-Flow Protocols: A Resolution of the Availability-Finality Dilemma

Joachim Neu
jneu@stanford.edu

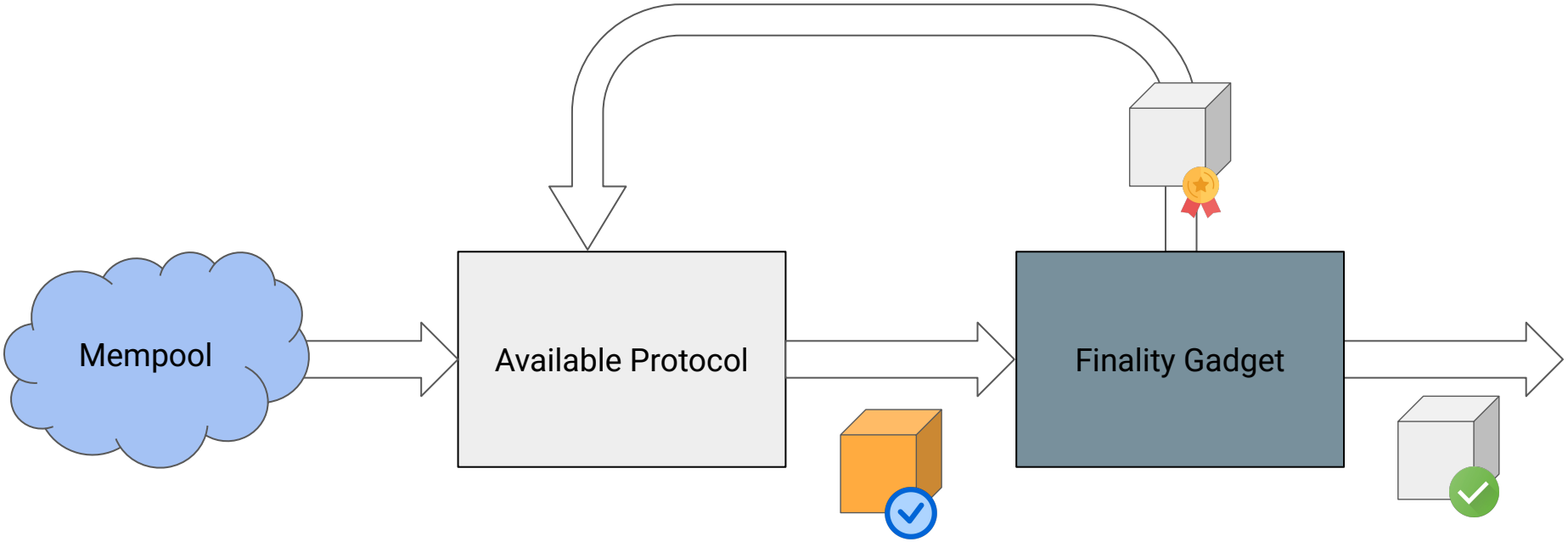
Ertem Nusret Tas
nusret@stanford.edu

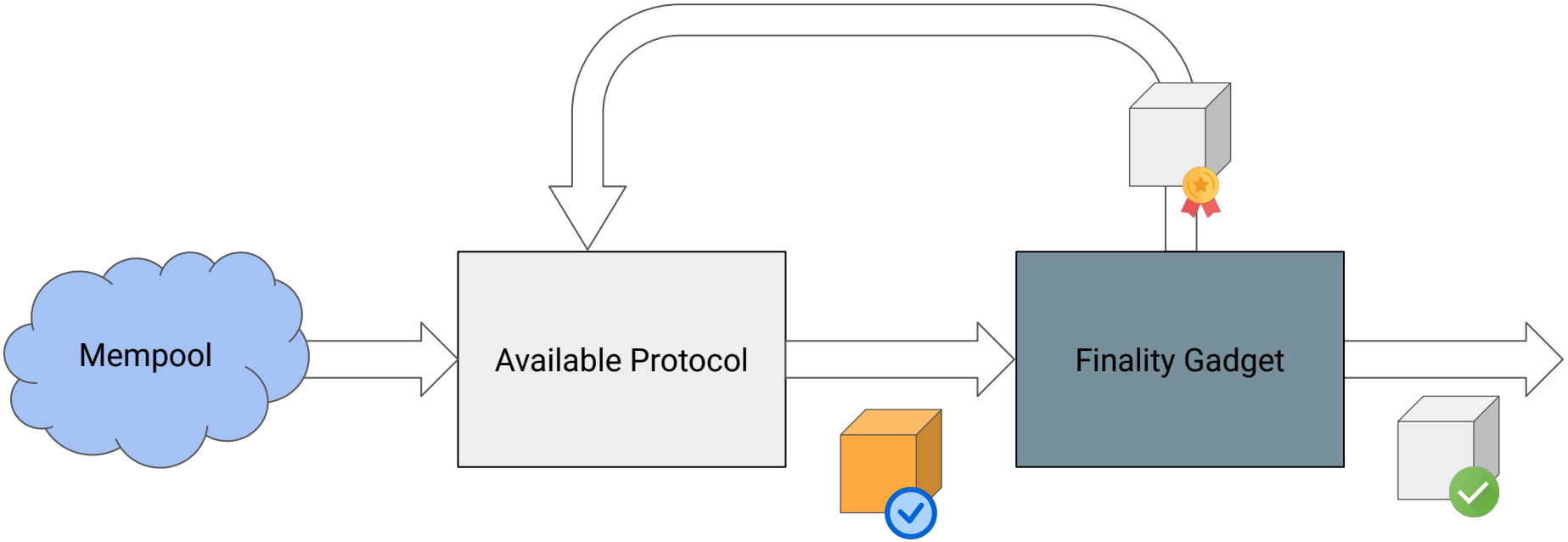
David Tse
dntse@stanford.edu

***Abstract*—The CAP theorem says that no blockchain can be live under dynamic participation and safe under temporary network partitions. To resolve this availability-finality dilemma, we formulate a new class of flexible consensus protocols, *ebb-and-flow protocols*, which support a full dynamically available ledger in conjunction with a finalized prefix ledger. The finalized ledger falls behind the full ledger when the network partitions but catches up when the network heals. Gasper, the current candidate protocol for Ethereum 2.0's beacon chain, combines the finality gadget Casper FFG with the LMD GHOST fork choice rule and aims to achieve this property. However, we discovered an attack in the standard synchronous network model, highlighting a general difficulty with existing finality-gadget-based designs. We present a construction of provably secure ebb-and-flow protocols with optimal resilience. Nodes run an off-the-shelf dynamically available protocol, take snapshots of the growing available ledger, and input them into a separate off-the-shelf BFT protocol to finalize a prefix. We explore connections with flexible BFT and improve upon the state-of-the-art for that problem.**

need to assume all adversary nodes are awake at the beginning [3], [6] or a trusted setup for nodes to join the network [4], [5], but recently it has been shown that these restrictions can be removed using verifiable delay functions [7].

One limitation of dynamically available protocols is that they are not tolerant to network partition: when the network partitions, honest nodes in a dynamically available protocol will think that many nodes are asleep, continue to confirm transactions, and thus is not safe.[1] This is in contrast to permissioned BFT protocols designed for partially synchronous networks, such as PBFT [8], Tendermint [9], [10], Hotstuff [11] and Streamlet [12]. This type of protocols is the basis for permissioned blockchains such as Libra [13], [14] and PoS blockchains such as Algorand [15], [16]. In these protocols, a quorum of two-thirds of the signatures of all the nodes is required to finalize transactions, and hence is safe under


Mempool
Available Protocol
Finality Gadget



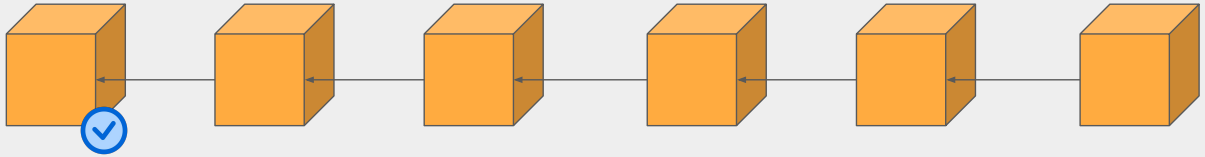
Prefix Property: The available chain is a prefix of the final chain.

Available Chain

$f < n / 2$

Δ (Synchronous)

1 honest awake

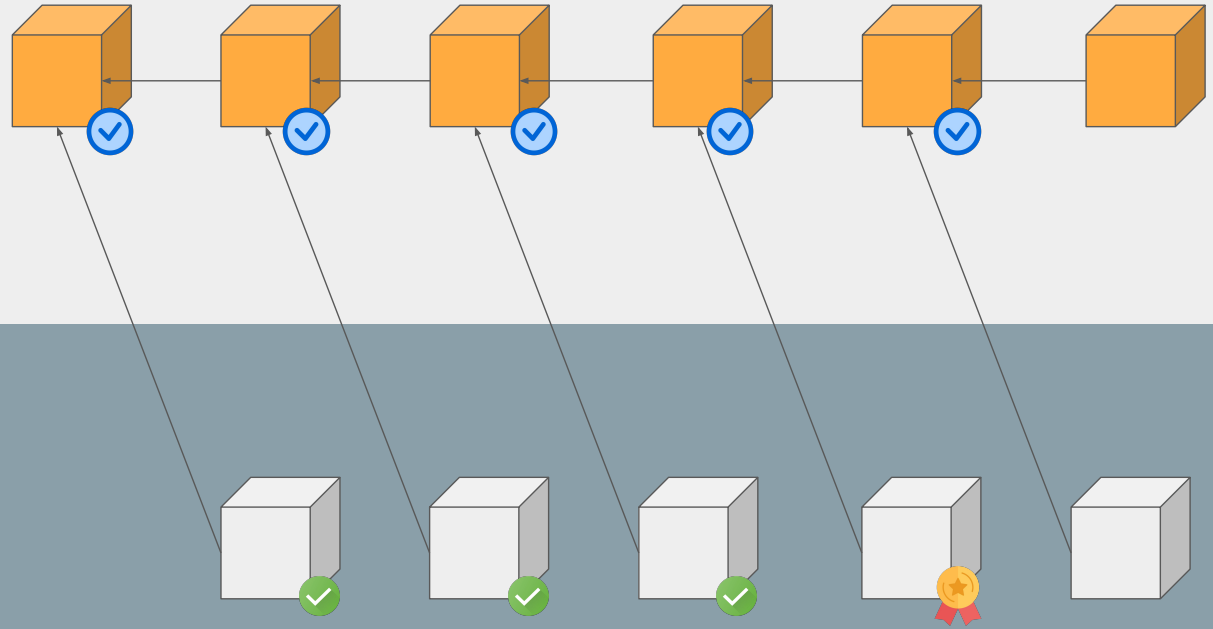


Available Chain

f < n / 2

Δ (Synchronous)

1 honest awake



Finality Chain

f < n / 3

Δ after **GST** (part. sync)

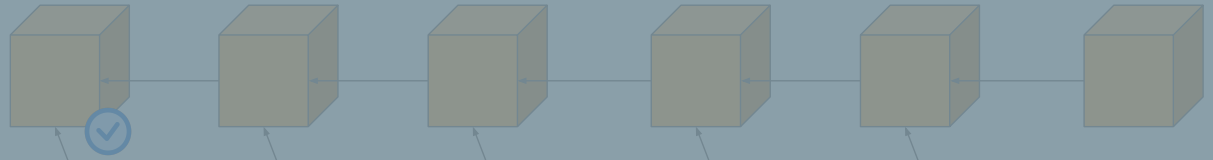
All online after **GAT**

Available Chain

f < n / 2

Δ (Synchronous)

1 honest awake



2. GOAL: Low-Latency

Finality Chain

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**

Slot 13812415

Epoch 431637

Fulu

6.3 sec

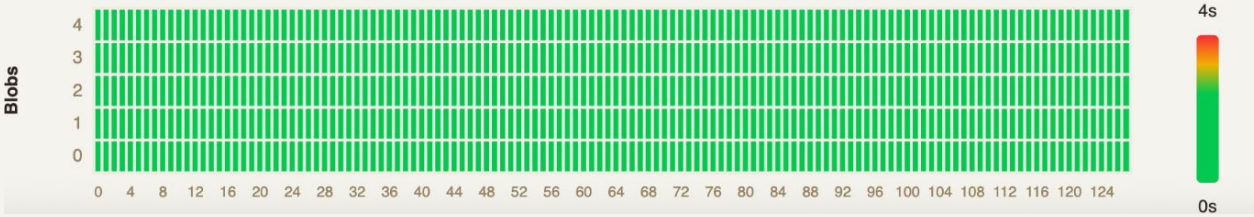
Block | Attestations | Aggregations

0s 4s 8s 12s

6.0s	Attest	3 validators
6.1s	Attest	8 validators
6.2s	Attest	11 validators
6.2s	Attest	1 validator
6.3s	Attest	1 validator
6.3s	Attest	1 validator
6.3s	Attest	2 validators
6.4s	Attest	9 validators
7.5s	Attest	1 validator
9.5s	Attest	1 validator

DATA COLUMN AVAILABILITY

✓ Data available 128 on time



ATTESTATION ARRIVALS

lab.ethpandaops.io

Slot 13812415

Epoch 431637

Fulu

6.3 sec

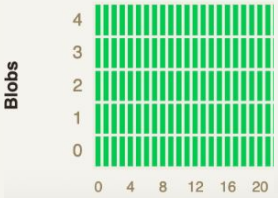
Block | Attestations | Aggregations

0s 4s 8s 12s

6.0s Attest 3 validators

Data from nodes contributing

DATA COLUMN AVAILABILITY



ATTESTATION ARRIVALS

0s

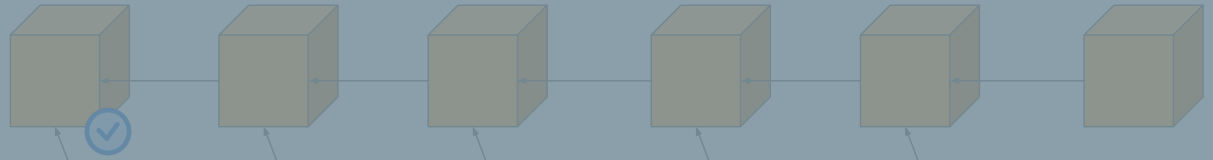
lab.ethpandaops.io

Available Chain

f < n / 2

Δ (Synchronous)

1 honest awake



2. GOAL: Low-Latency

Finality Chain

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**

Available Chain

f < n / 2

Δ (Synchronous)

1 honest awake

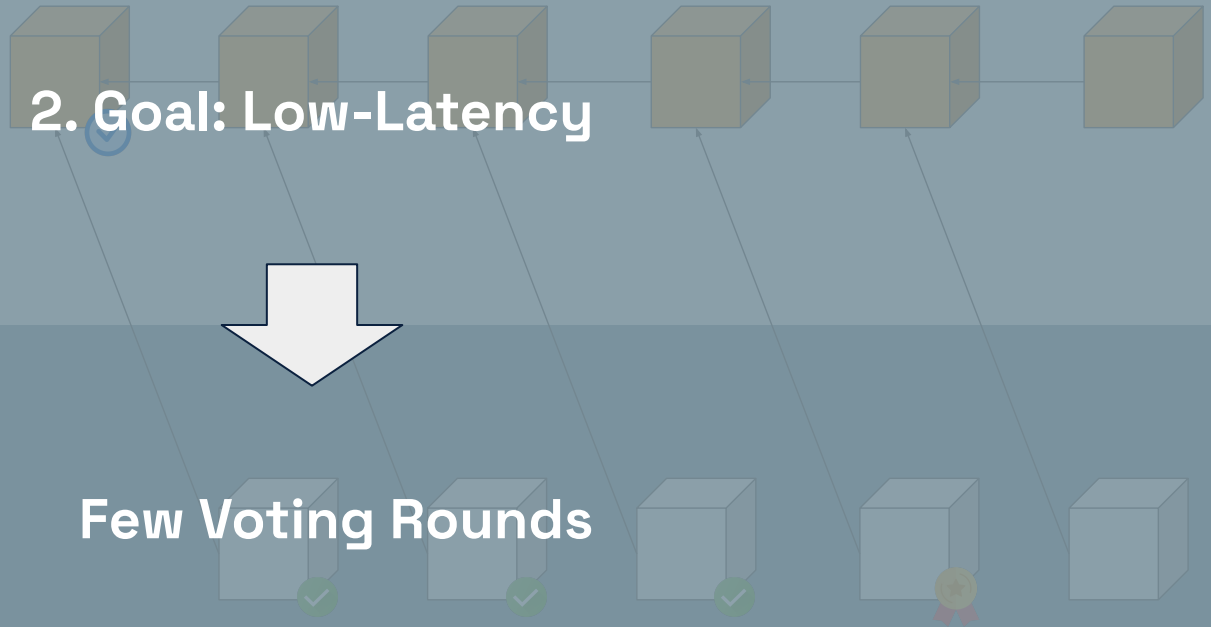
Finality Chain

f < n / 3

Δ after **GST** (part. sync)

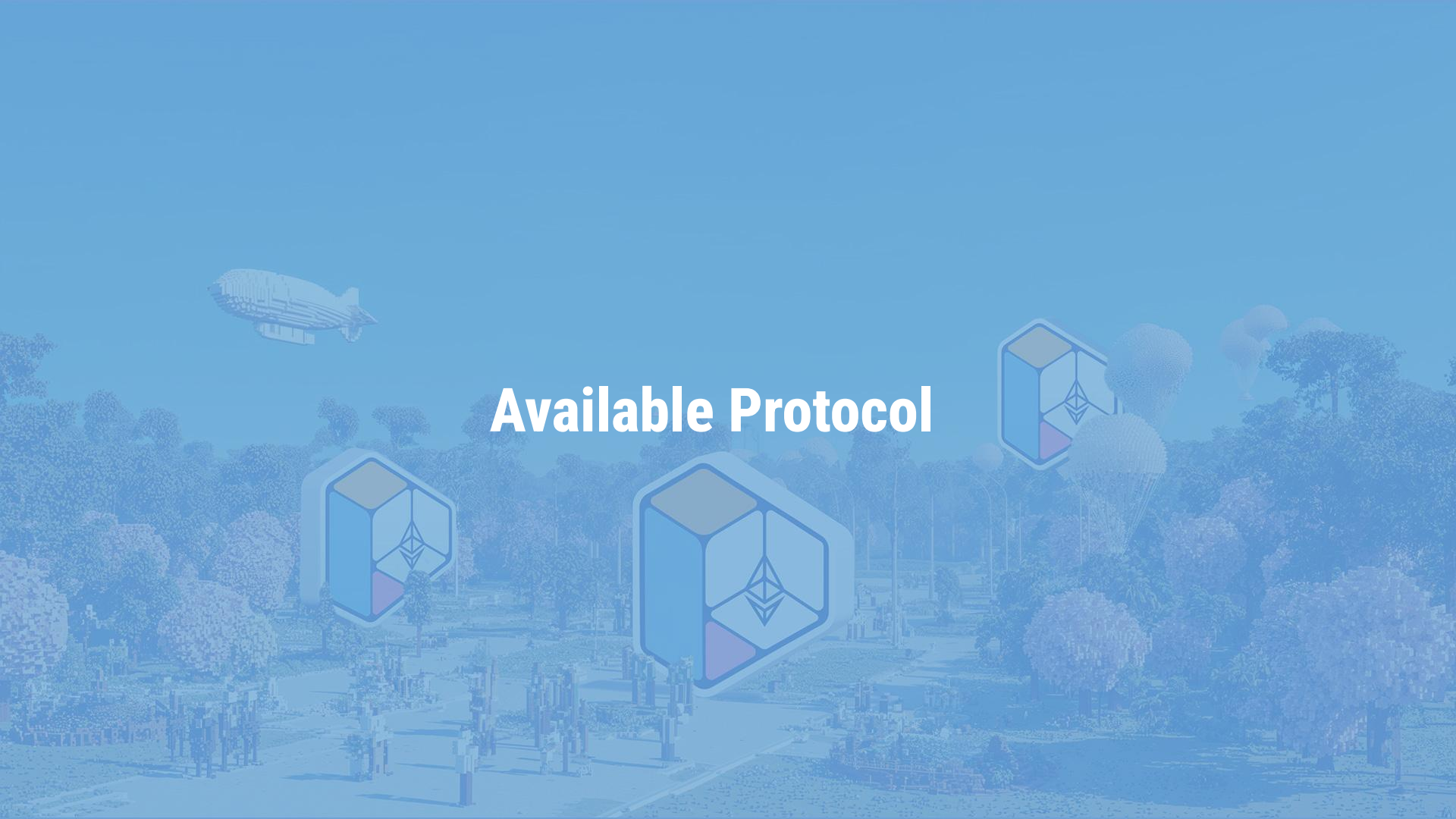
All online after **GAT**

2. Goal: Low-Latency



Few Voting Rounds

Available Protocol

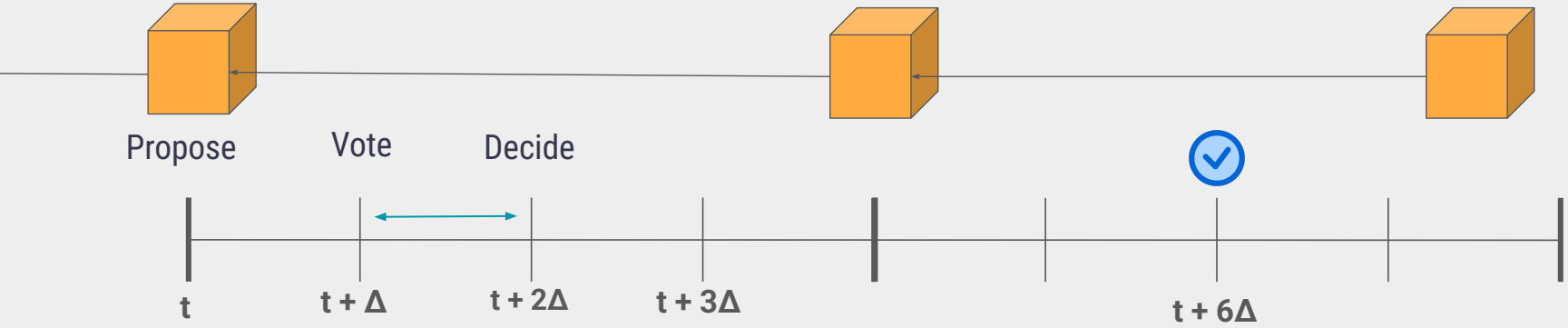


TOB-SVD

$f < n / 2$

Δ (Synchronous)

1 honest awake



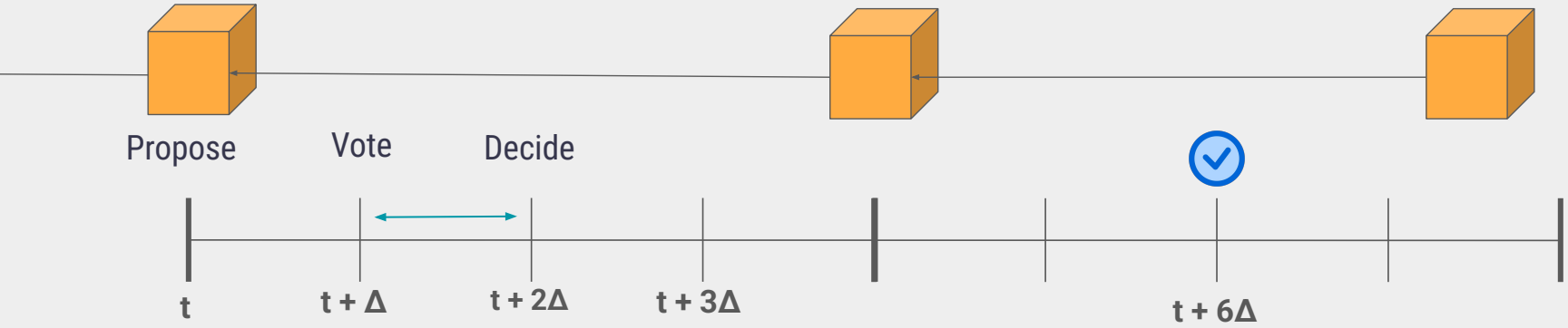
TOB-SVD

Single vote decision!

$f < n / 2$

Δ (Synchronous)

1 honest awake

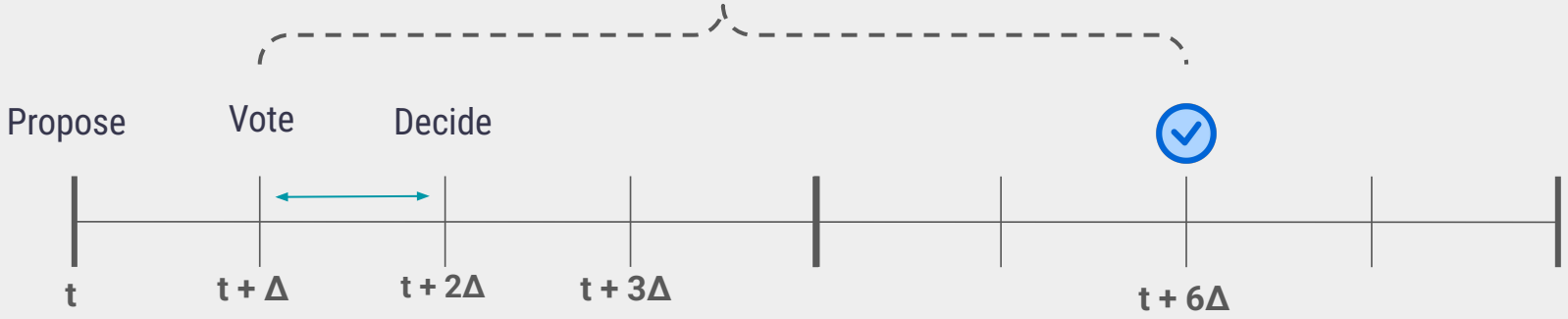


TOB-SVD

f < n / 2

Δ (Synchronous)

1 honest awake

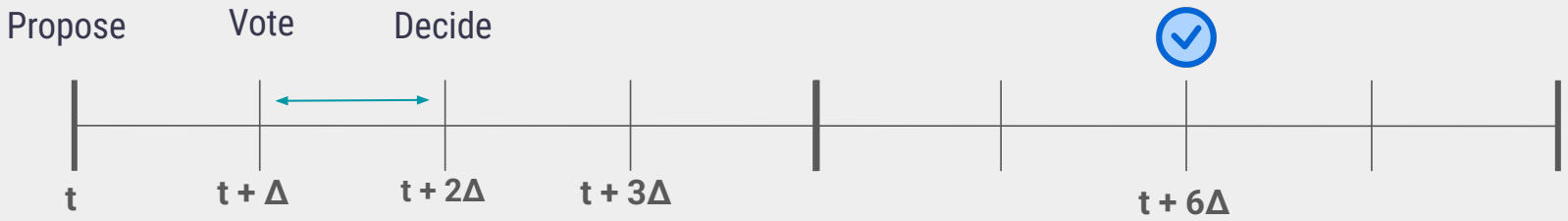


TOB-SVD: Dynamic Quorums

f < n / 2

Δ (Synchronous)

1 honest awake

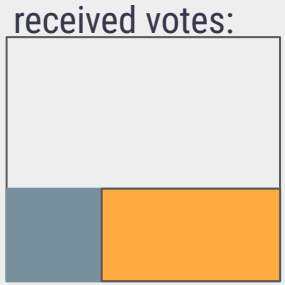


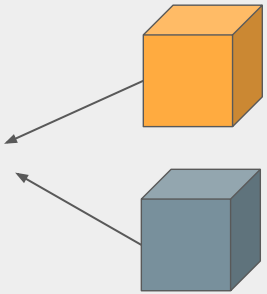
TOB-SVD

$f < n / 2$

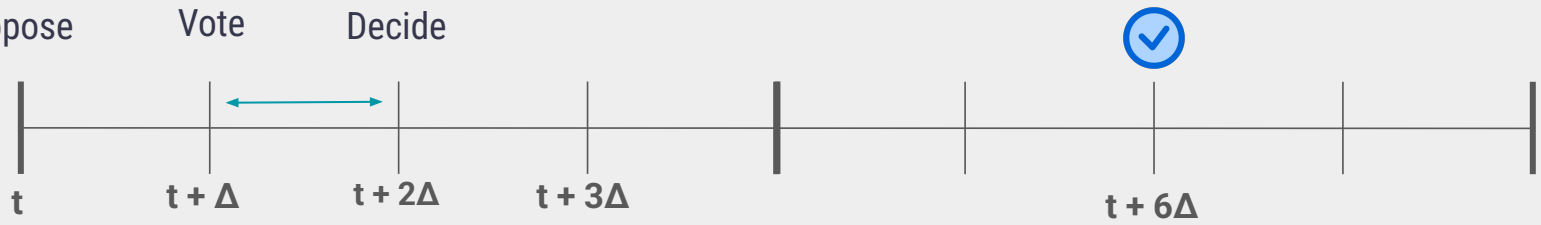
Δ (Synchronous)

1 honest awake





Solution: Time shifted quorums!

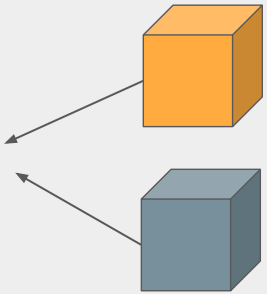


TOB-SVD: Dynamic Quorums

$f < n / 2$

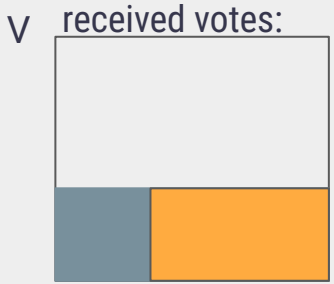
Δ (Synchronous)

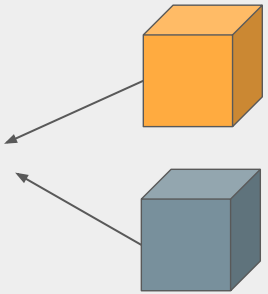
1 honest awake



Propose Vote Decide

TOB-SVD

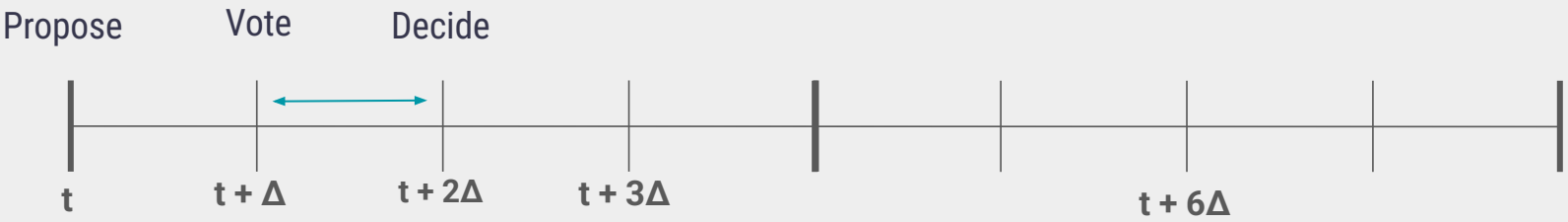




f < n / 2

Δ (Synchronous)

1 honest awake



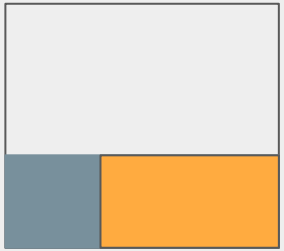
TOB-SVD

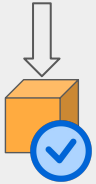
$f < n / 2$

Δ (Synchronous)

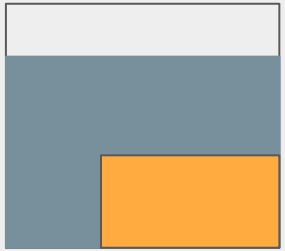
1 honest awake

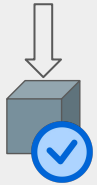
V received votes:

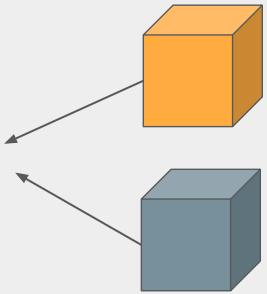




U received votes:







Propose Vote Decide

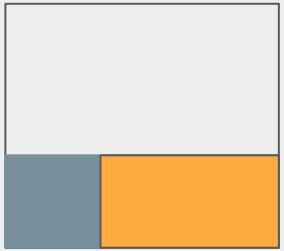
TOB-SVD

f < n / 2

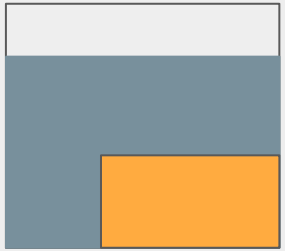
Δ (Synchronous)

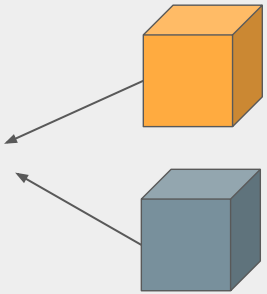
1 honest awake

V received votes:



U received votes:





Solution: Time shifted quorums!

Propose Vote Decide

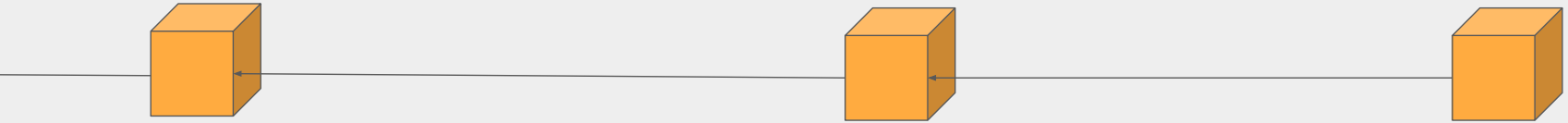
t t + Δ t + 2Δ t + 3Δ t + 6Δ

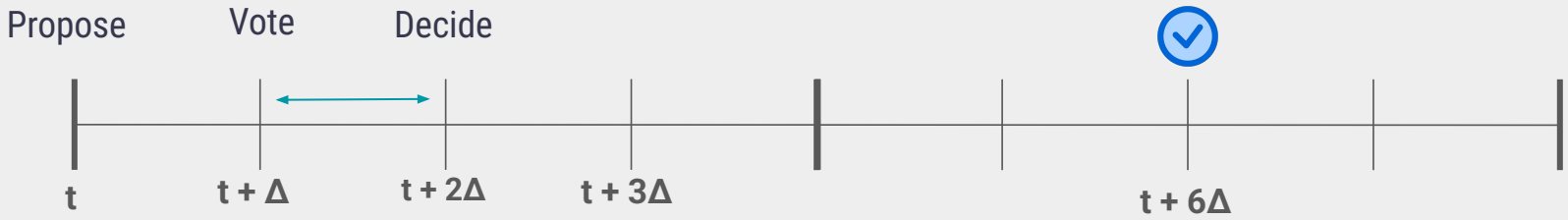
TOB-SVD

$f < n / 2$

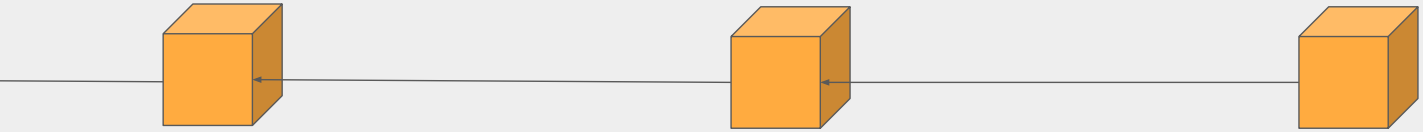
Δ (Synchronous)

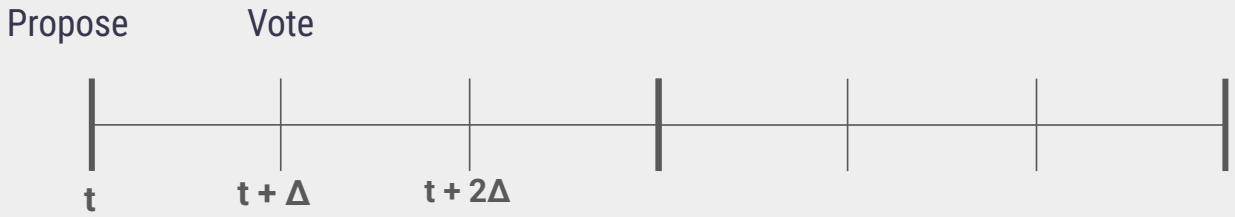
1 honest awake





Step 1: Probabilistic TOB-SVD





$f < n / 2$

Δ (Synchronous)

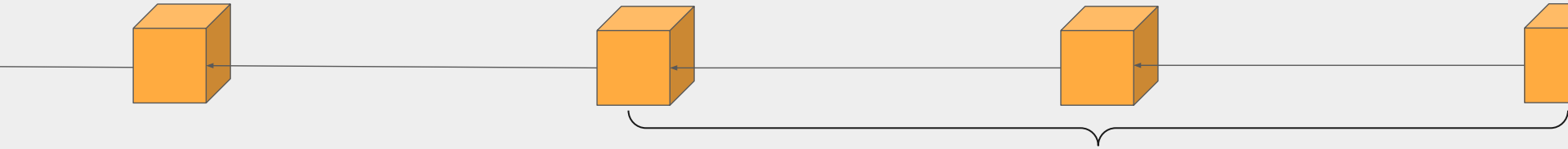
1 honest awake

Step 1: Probabilistic TOB-SVD

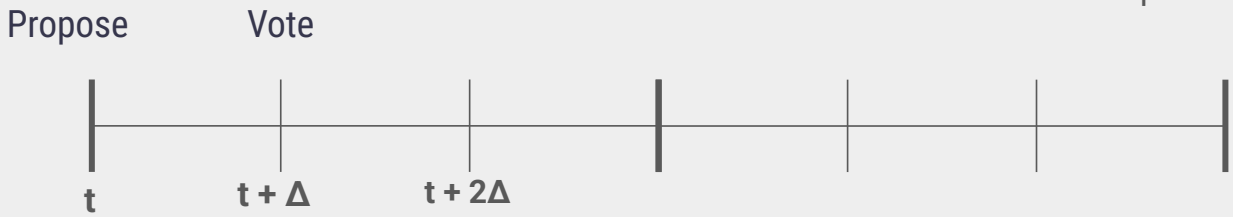
f < n / 2

Δ (Synchronous)

1 honest awake



k-deep ⇒ *confirmed*

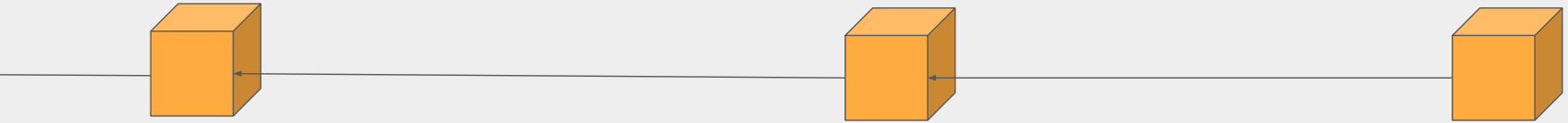


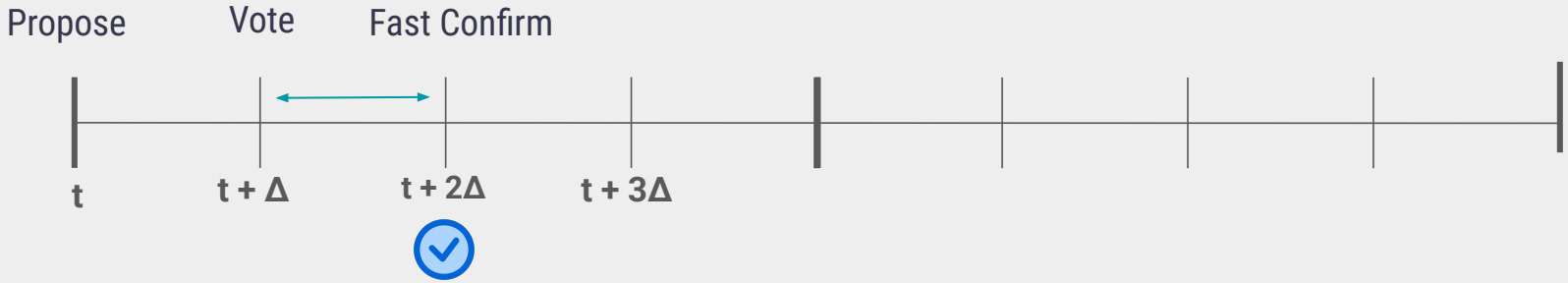
Step 2: Fast Confirmation

f < n / 2

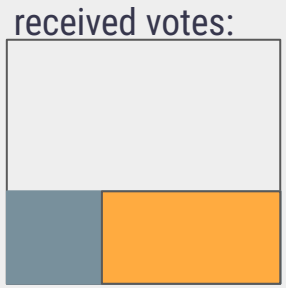
Δ (Synchronous)

1 honest awake





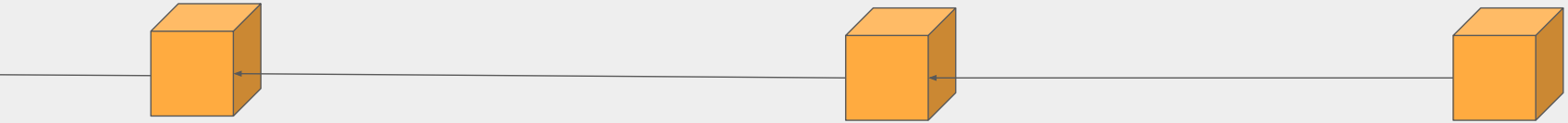
Step 2: Fast Confirmation



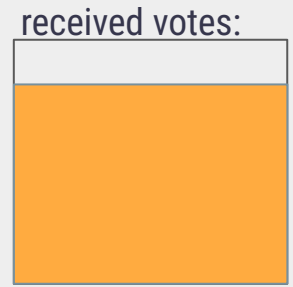
f < n / 2

Δ (Synchronous)

1 honest awake



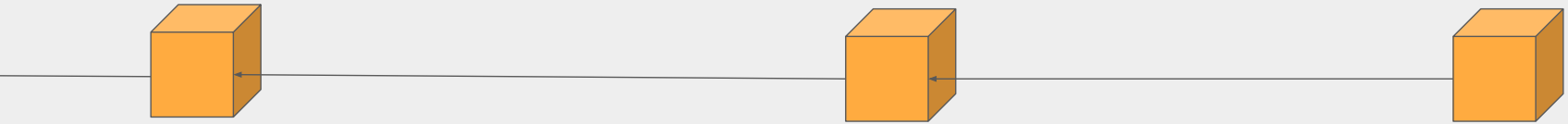
Step 2: Fast Confirmation

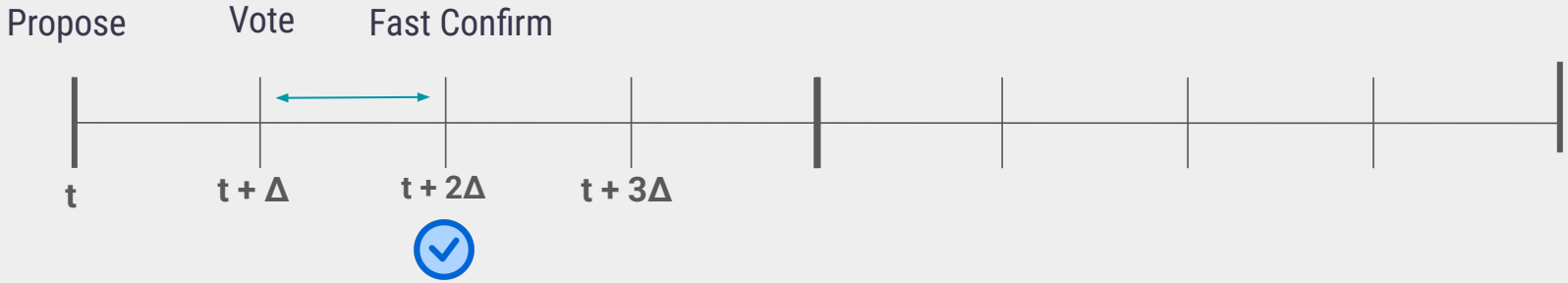


f < n / 2

Δ (Synchronous)

1 honest awake



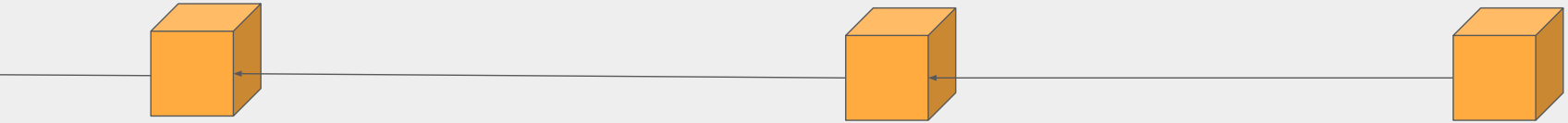


Probabilistic TOB-SVD with Fast Confirmation

$f < n / 2$

Δ (Synchronous)

1 honest awake

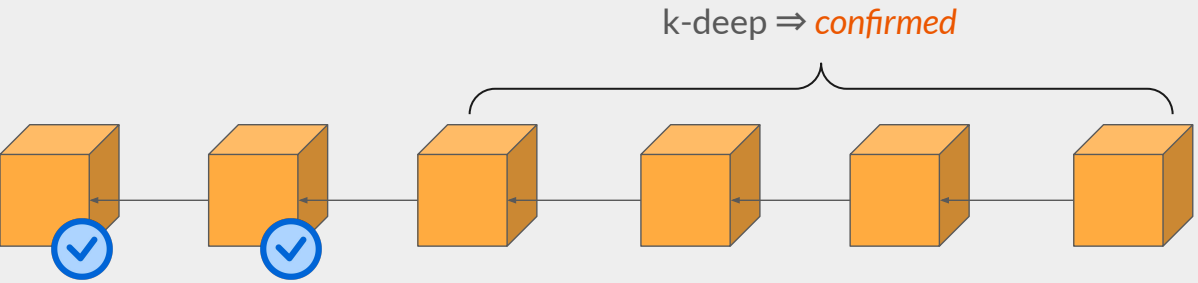


TOB-SVD: Dynamic Quorums

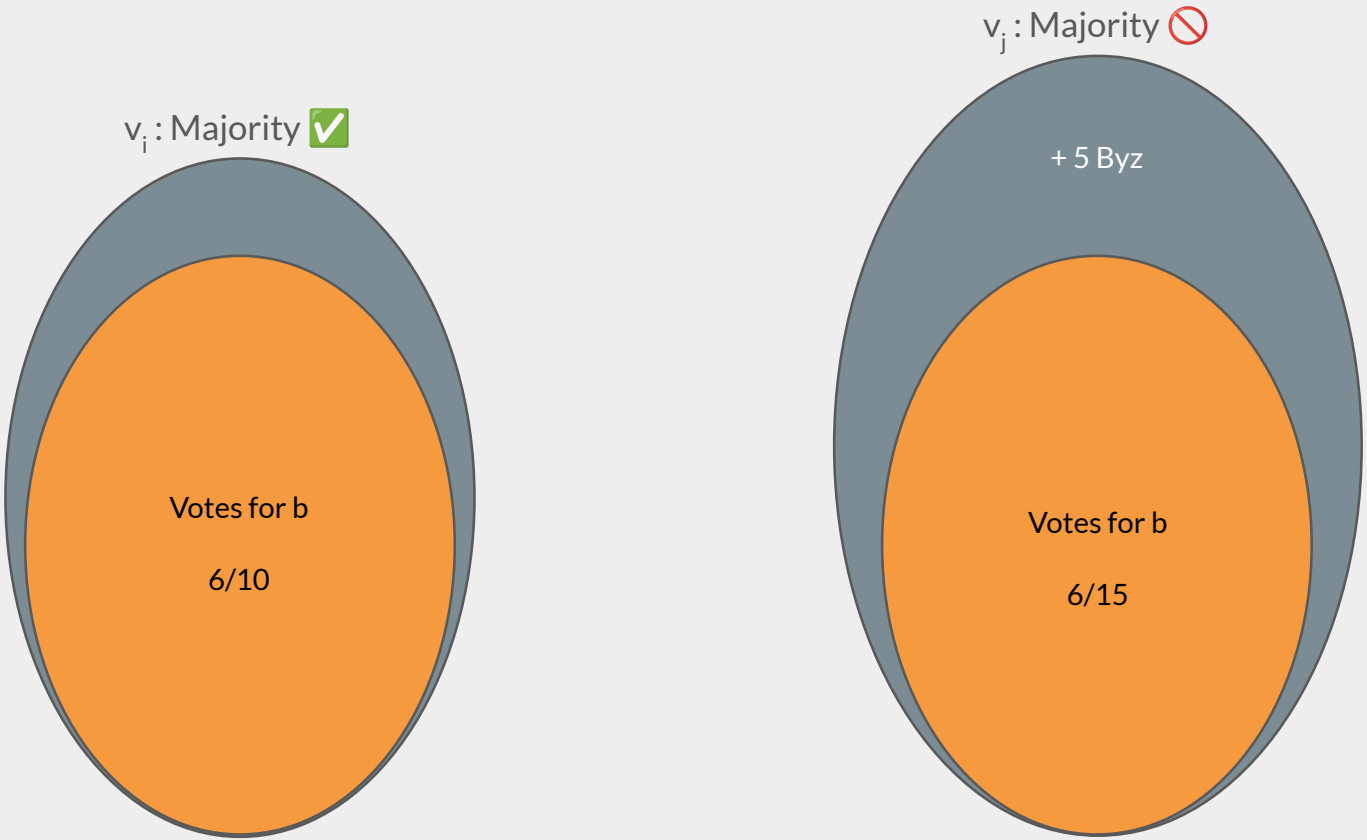
$f < n / 2$

Δ (Synchronous)

1 honest awake

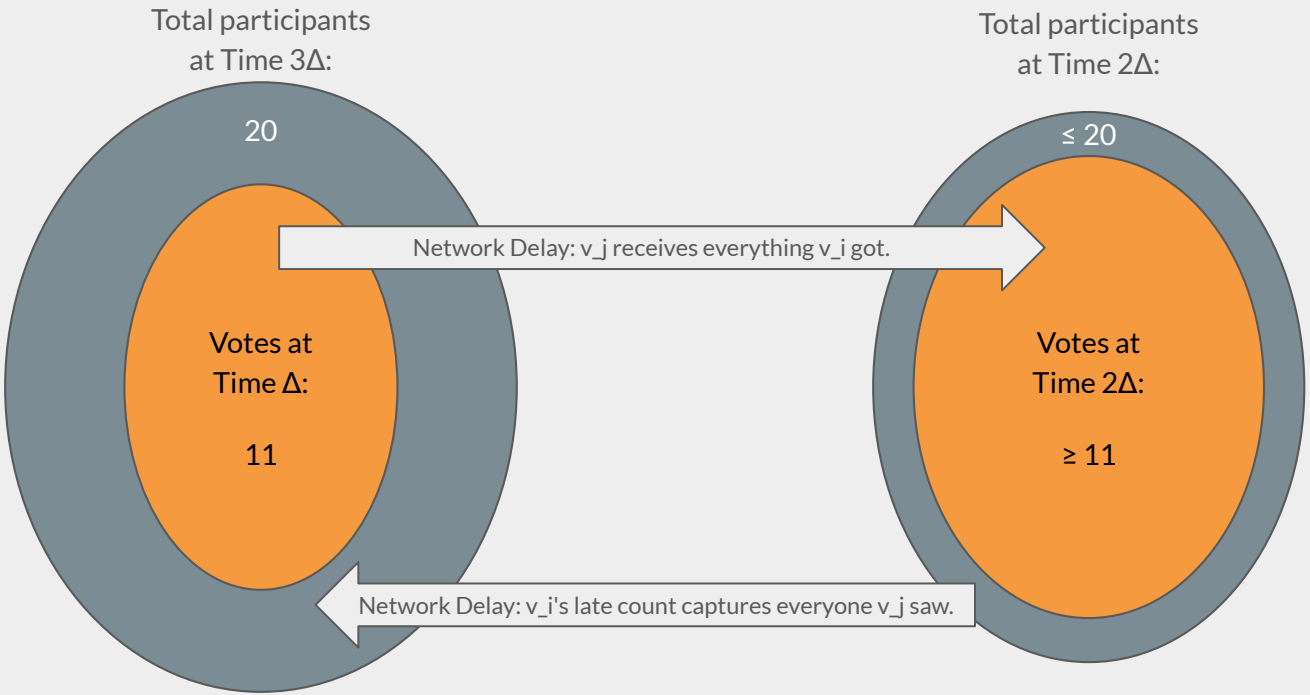


The Problem: Dynamic Quorum Non-Transferability



Byzantine validators announce themselves selectively so the same certificate is valid for v_i but not for v_j.

The Solution: Time-Shifted Quorums



v_i : Majority ✅

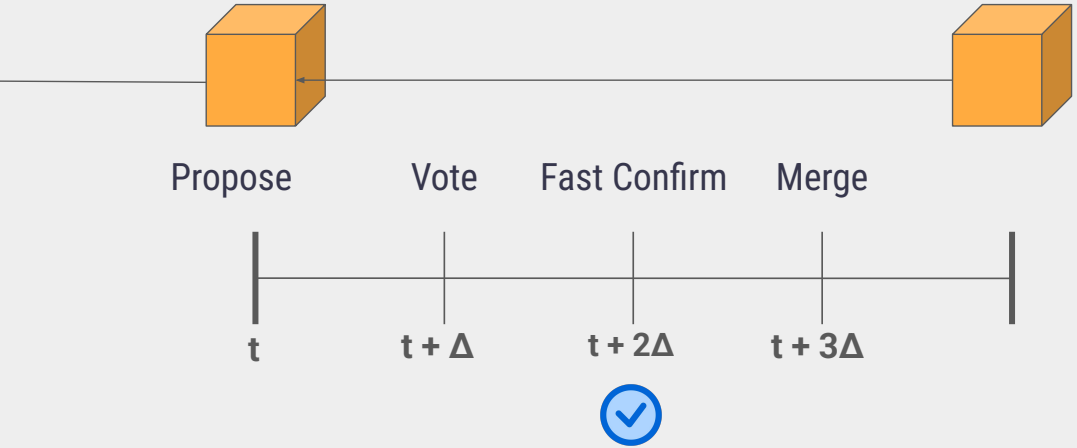
11 > 20/2

v_j : Majority ✅

(≥ 11) > (≤ 20)/2

Validator v_j is guaranteed to see *at least* all of v_i's early votes (Top Arrow), and v_j's late participant count will capture *at least* everyone v_j saw (Bottom Arrow).

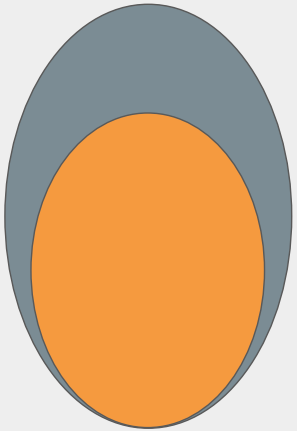
Probabilistic TOB-SVD

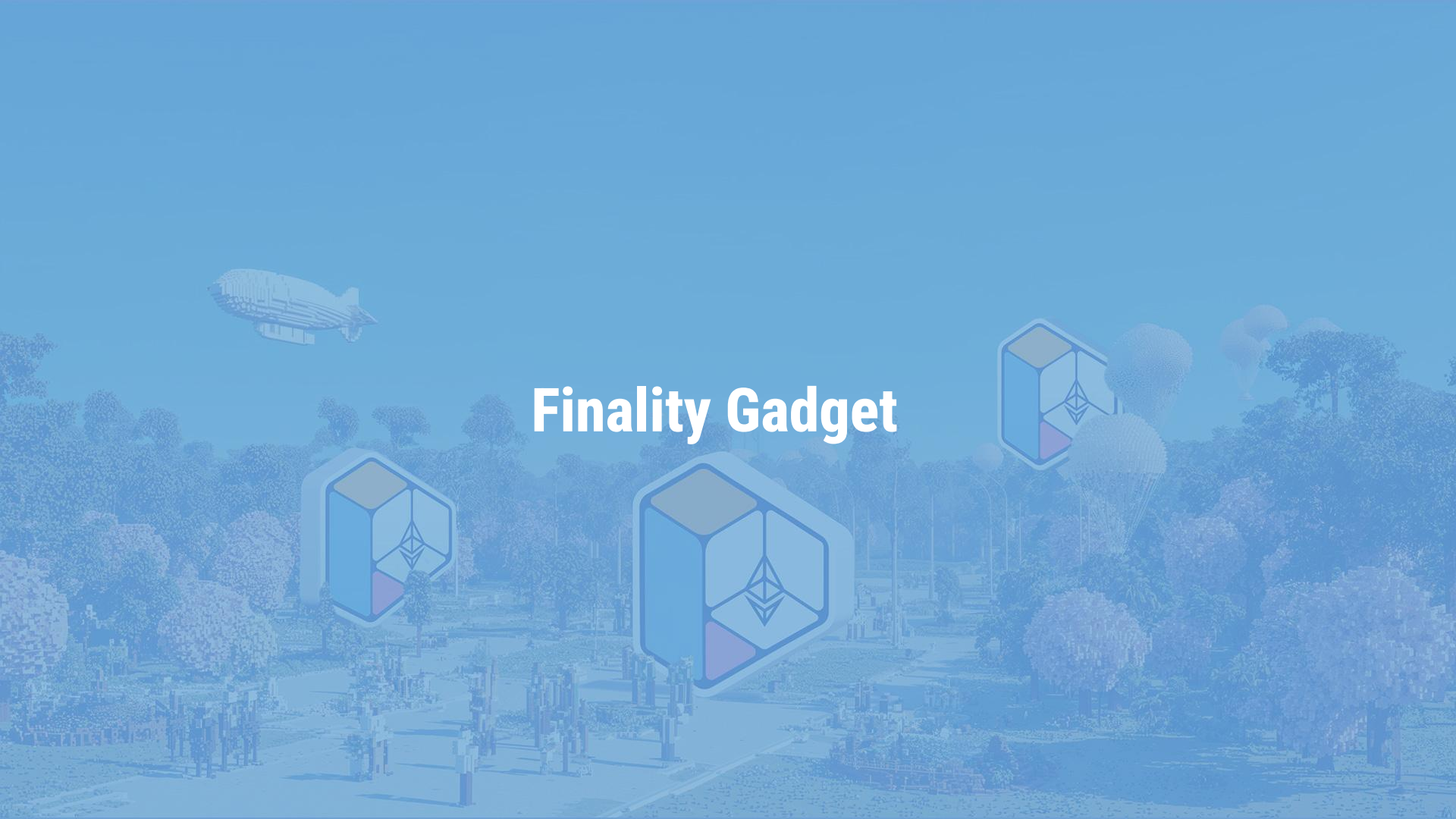


f < n / 2

Δ (Synchronous)

1 honest awake



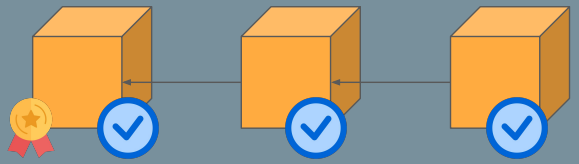

Finality Gadget

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



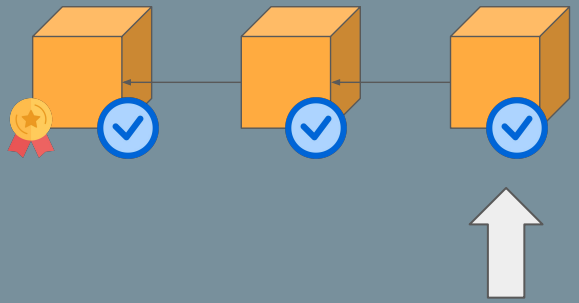
Pick voting target:
Confirmed head of available chain (respecting justification)

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



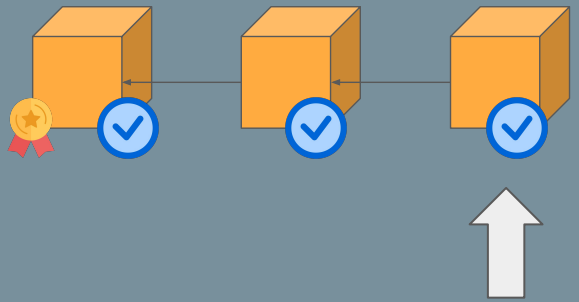
Pick voting target:
Confirmed head of available chain (respecting justification)

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



Pick voting target:
Confirmed head of available chain (respecting justification)

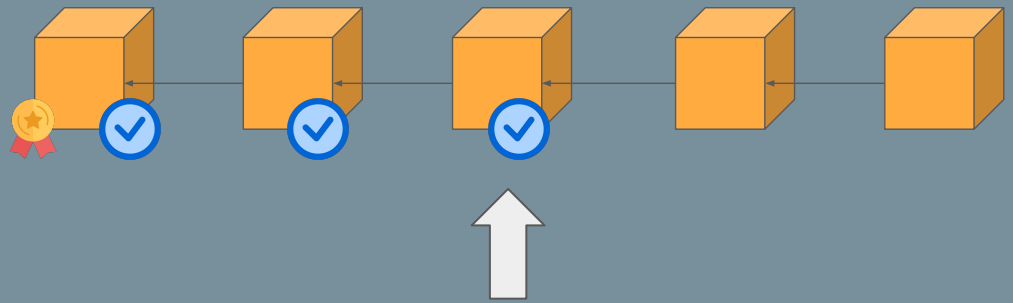
⇒ fast confirmed

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**



Pick voting target:
Confirmed head of available chain (respecting justification)

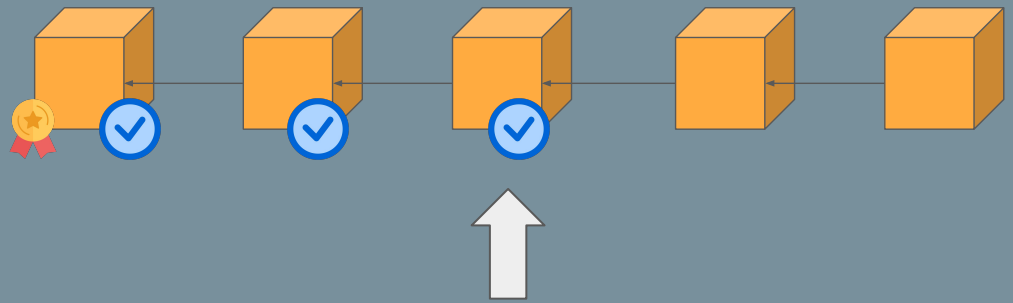
⇒ fast confirmed
⇒ k - deep

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



Pick voting target:
Confirmed head of available chain (respecting justification)

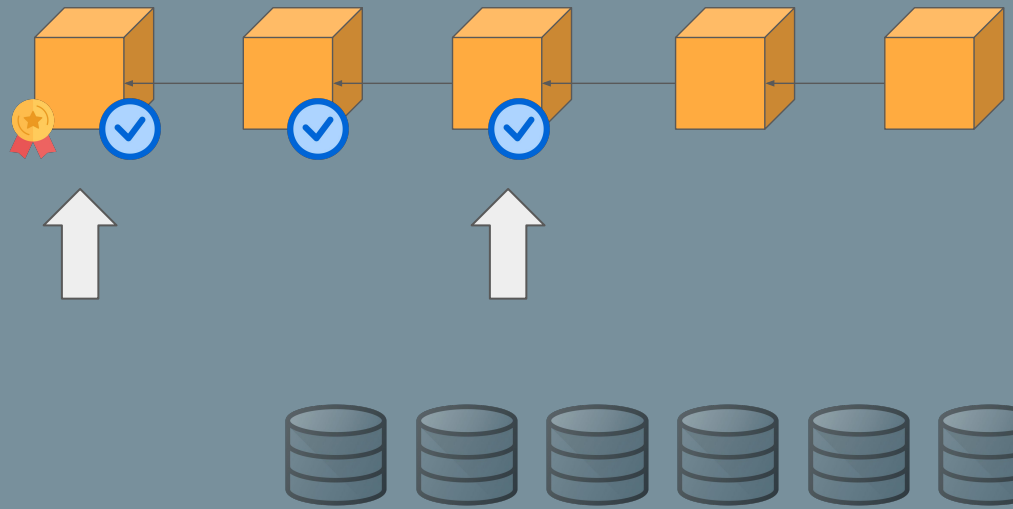
⇒ fast confirmed
⇒ k - deep
(under sleepiness, or asynchrony)

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**



Pick voting target:
Confirmed head of available chain (respecting justification)

⇒ fast confirmed
⇒ k - deep
(under sleepiness, or asynchrony)

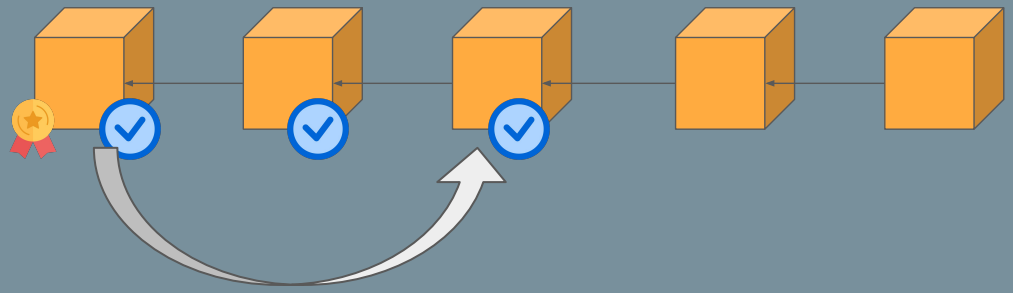
Pick voting source:
Latest justified block

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



Pick voting target:
Confirmed head of available chain (respecting justification)

⇒ fast confirmed
⇒ k - deep
(under sleepiness, or asynchrony)

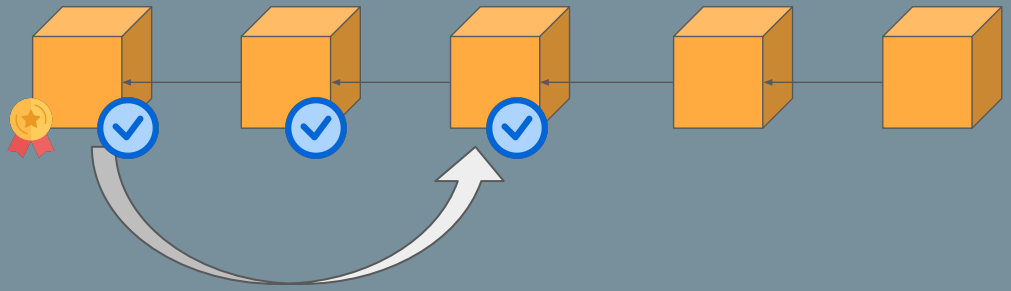
Pick voting source:
Latest justified block

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



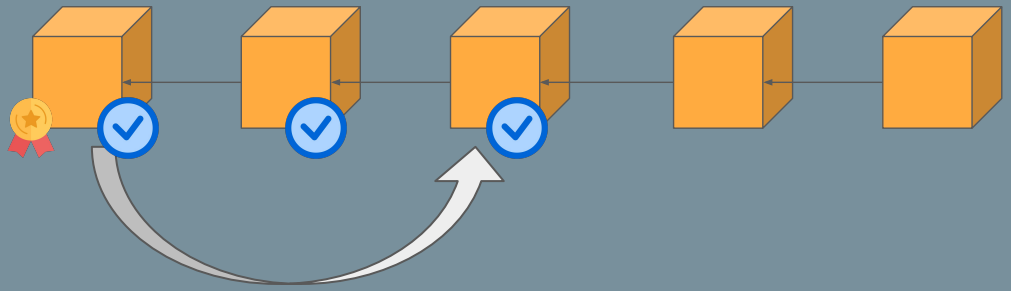
Justify block B once:
≥ 67% votes with target B

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



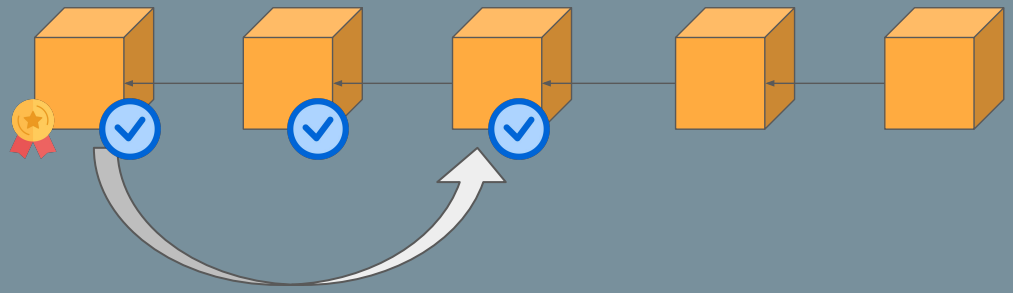
Justify block B once:
≥ 67% votes with target B

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**



Justify block B once:
≥ 67% votes with target B

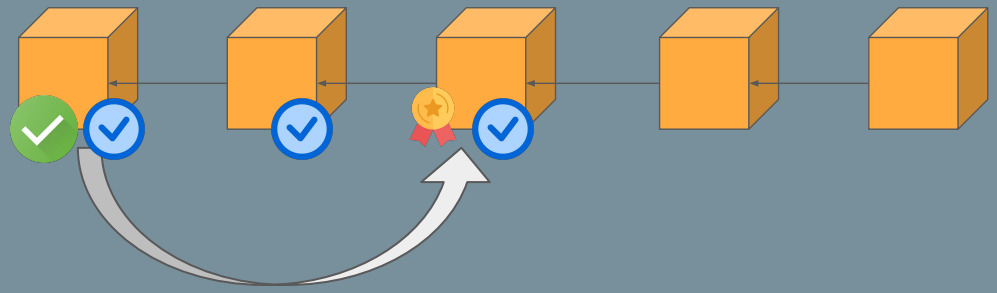
Finalize block B once:
≥ 67% votes with source B and target B' > B

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

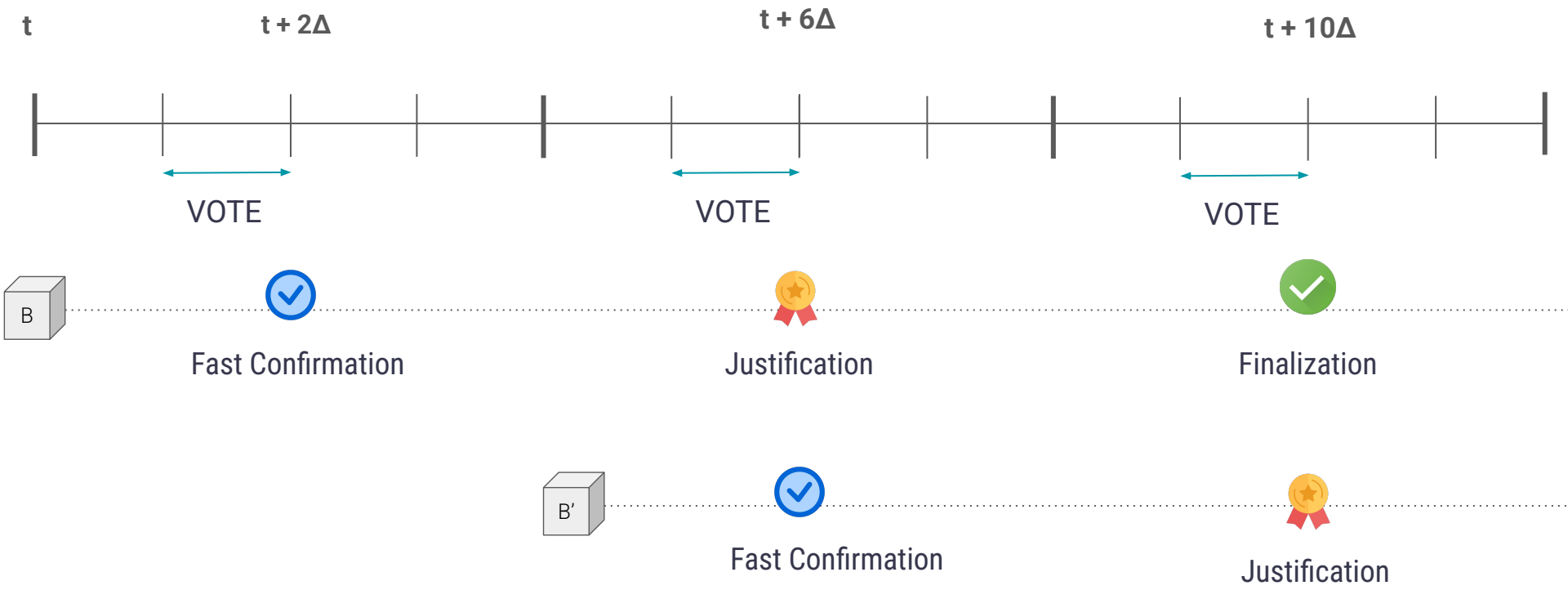
All online after **GAT**

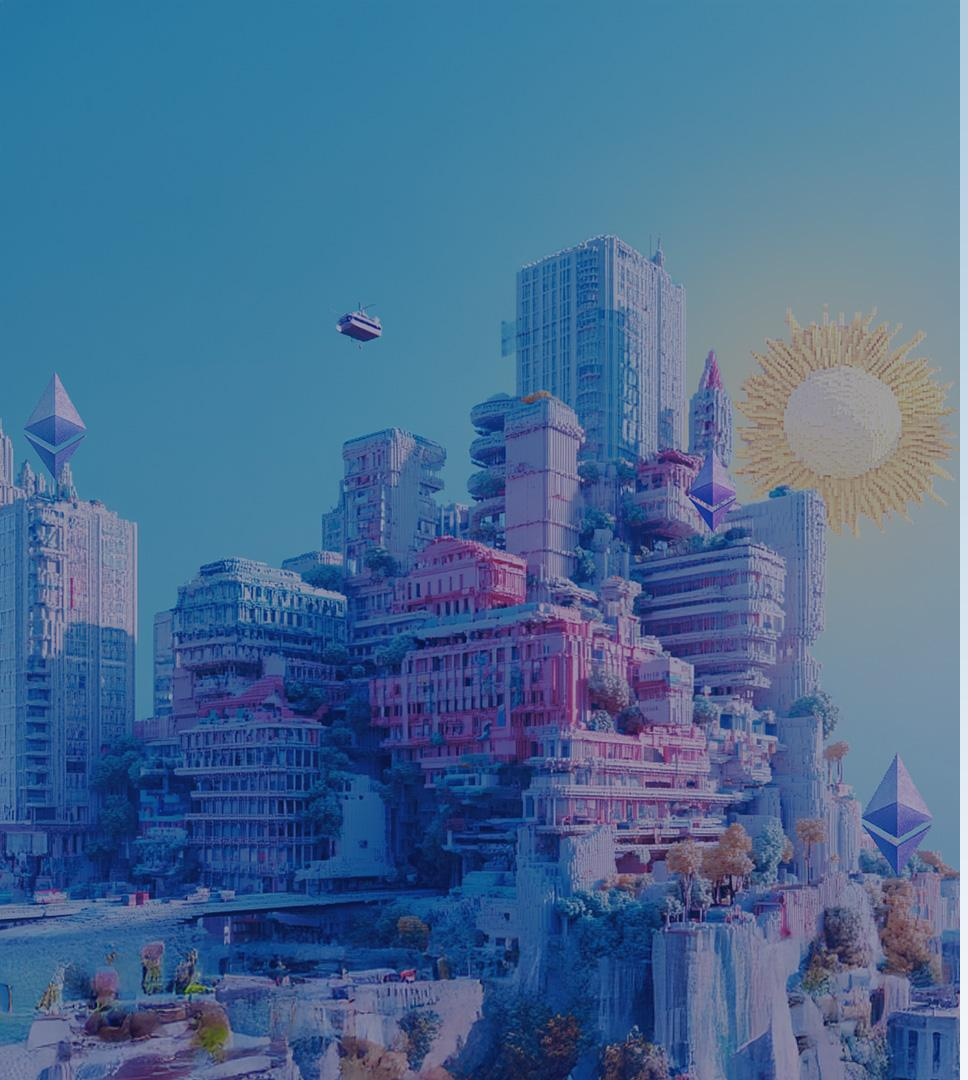


Justify block B once:
≥ 67% votes with target B

Finalize block B once:
≥ 67% votes with source B and target B' > B

Majorum

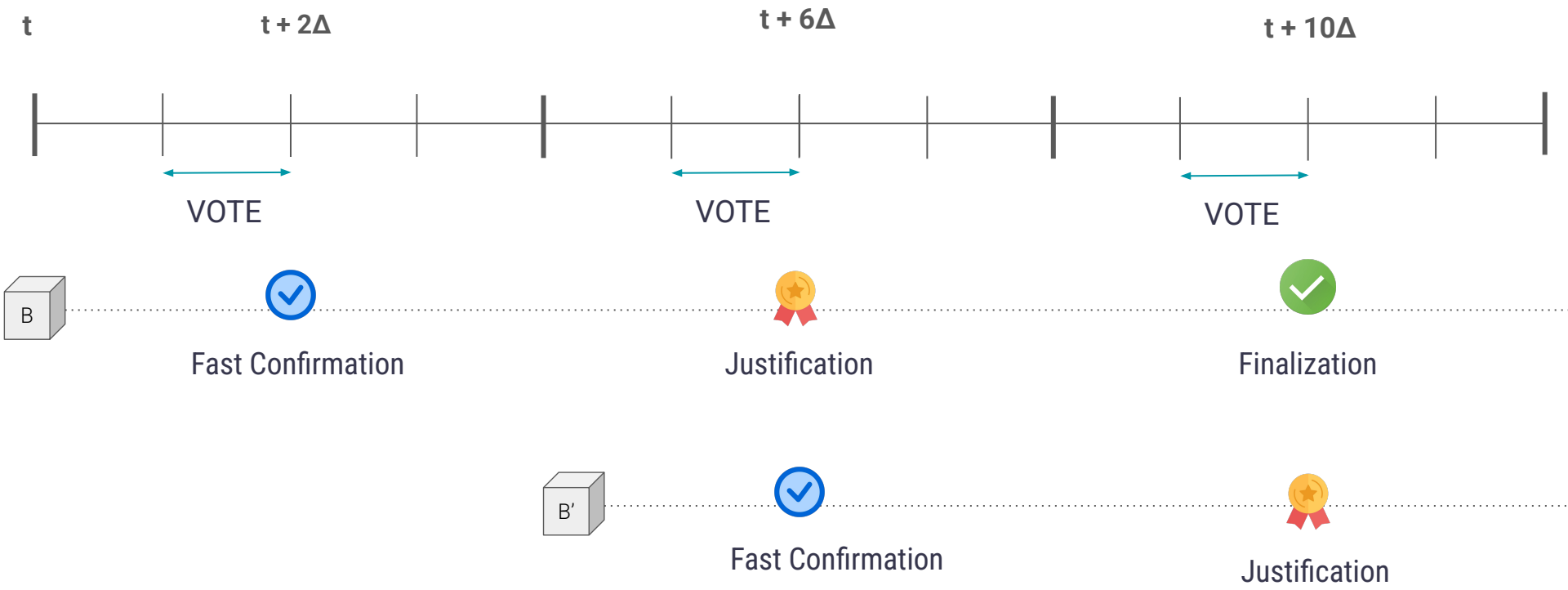

t
t + 2Δ
t + 6Δ
t + 10Δ
VOTE
VOTE
VOTE
B
Fast Confirmation
Justification
Finalization
B'
Fast Confirmation
Justification



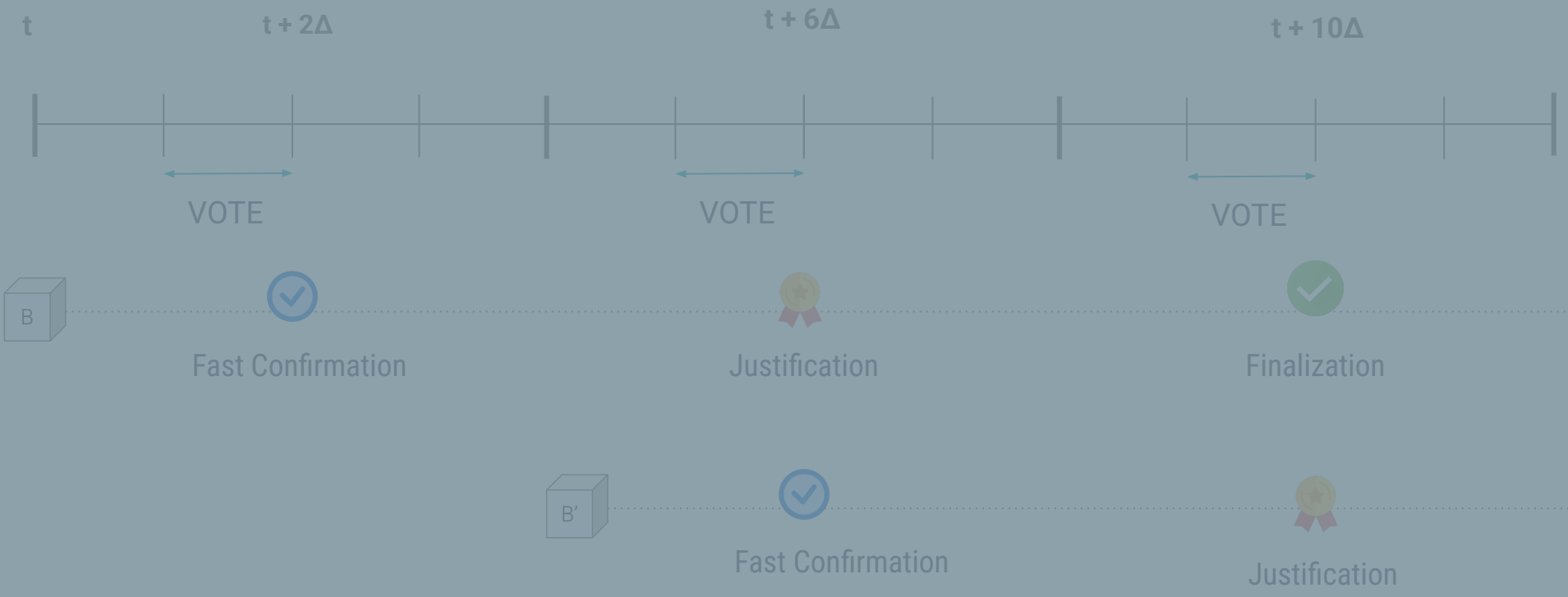
Contributions

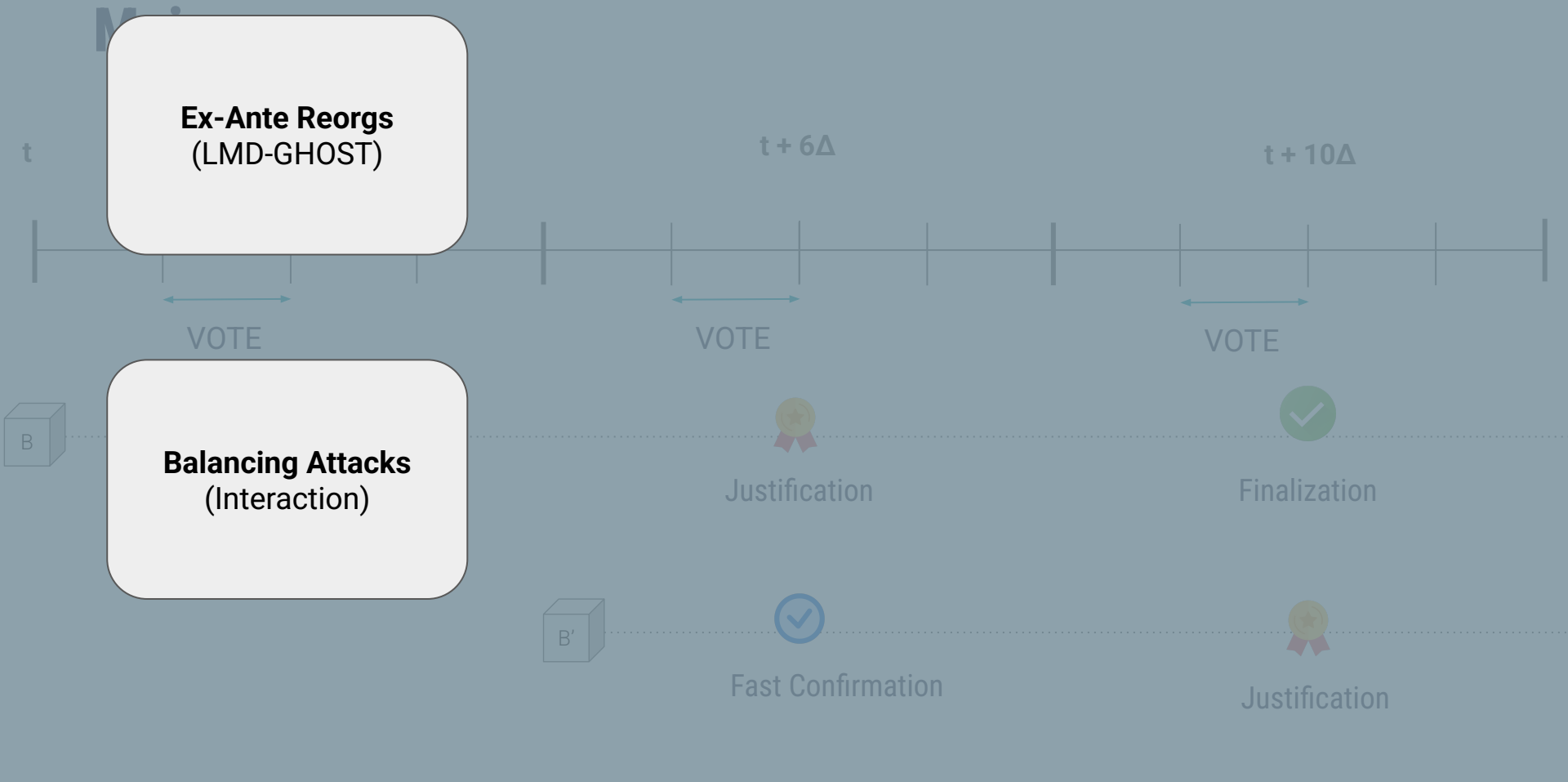
- **Ebb-and-Flow** construction with **dynamic quorums**
- Combination of **k-deep** and **fast confirmation** with a single voting phase per slot
  - Finalization in 3 slots
  - Expected transaction latency: 12**Δ**
- Opens the way for future improvements

Majorum


t
t + 2Δ
t + 6Δ
t + 10Δ
VOTE
VOTE
VOTE
B
Fast Confirmation
Justification
Finalization
B'
Fast Confirmation
Justification

Majorum


t
t + 2Δ
t + 6Δ
t + 10Δ
VOTE
VOTE
VOTE
B
Fast Confirmation
Justification
Finalization
B'
Fast Confirmation
Justification


Ex-Ante Reorgs
(LMD-GHOST)
t
t + 6Δ
t + 10Δ
VOTE
VOTE
VOTE
Balancing Attacks
(Interaction)
B
Justification
Finalization
B'
Fast Confirmation
Justification

Ex-Ante Reorgs
(LMD-GHOST)

Balancing Attacks
(Interaction)

t

t + 6Δ

t + 10Δ

VOTE

VOTE

VOTE

B

Justification

Finalization

B'

Fast Confirmation

Justification

Ex-Ante Reorgs
(LMD-GHOST)

Balancing Attacks
(Interaction)

Block production
vs.
Time to finality

t

t + 6Δ

t + 10Δ

VOTE

VOTE

B

Justification

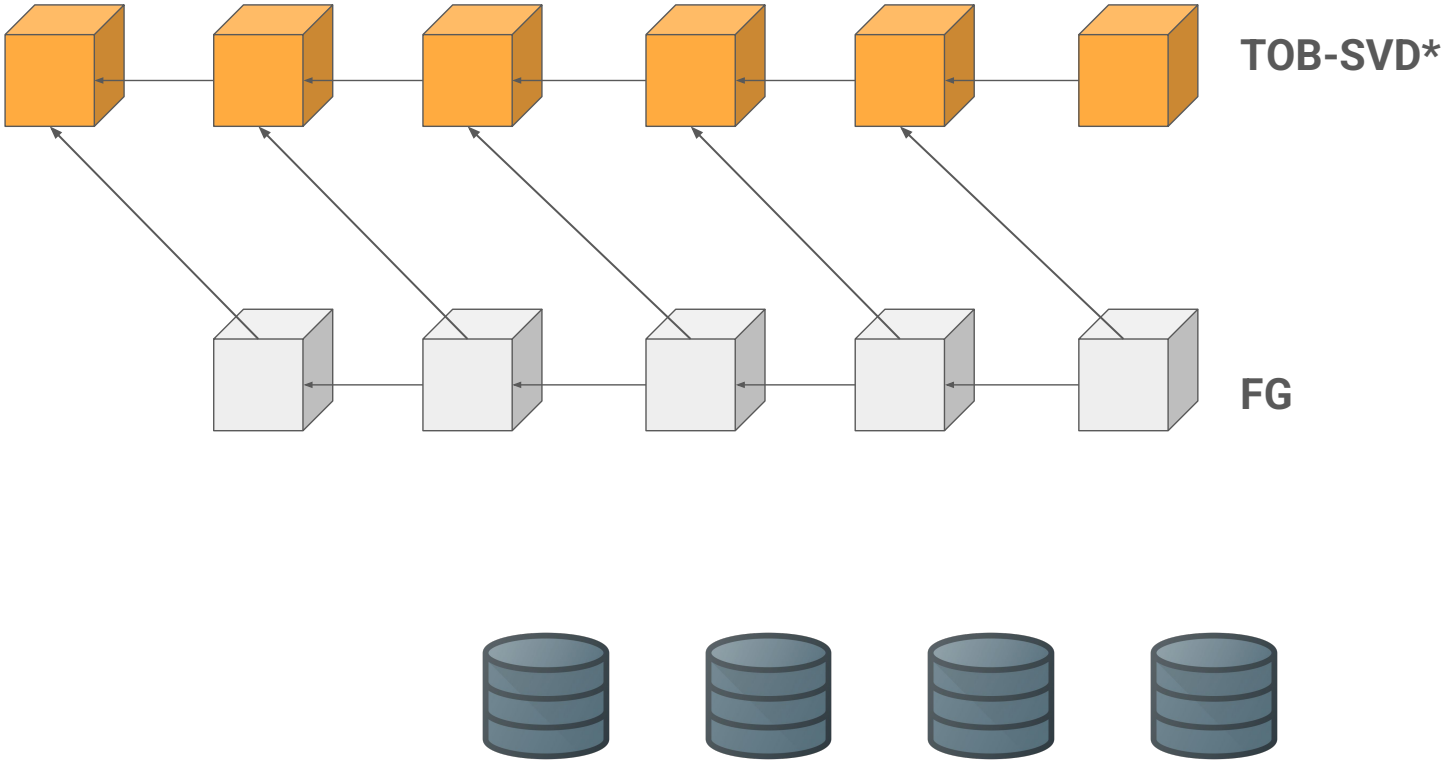
Finalization

B'

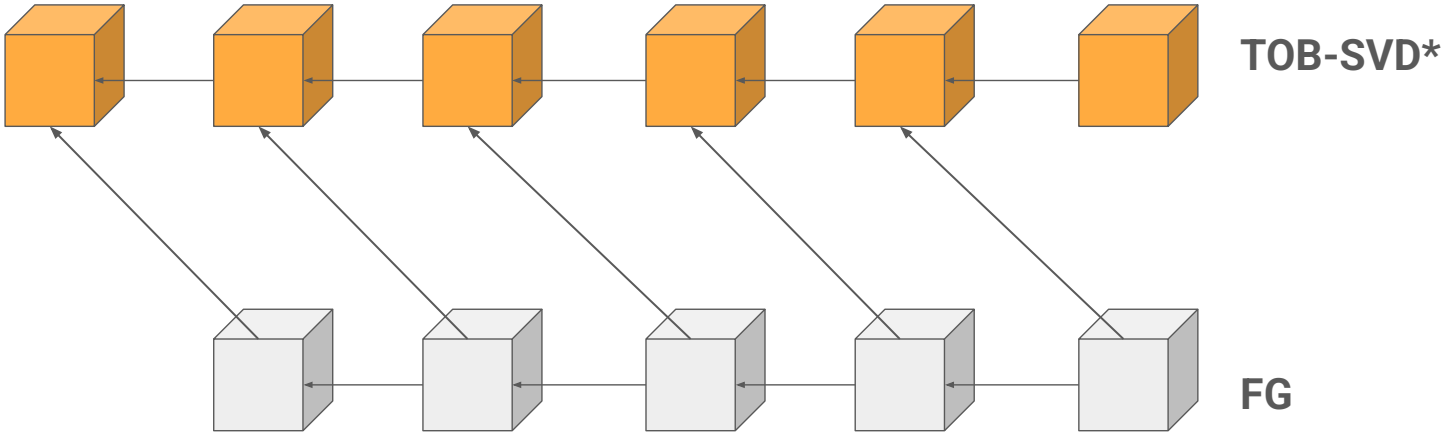
Fast Confirmation

Justification

Majorum



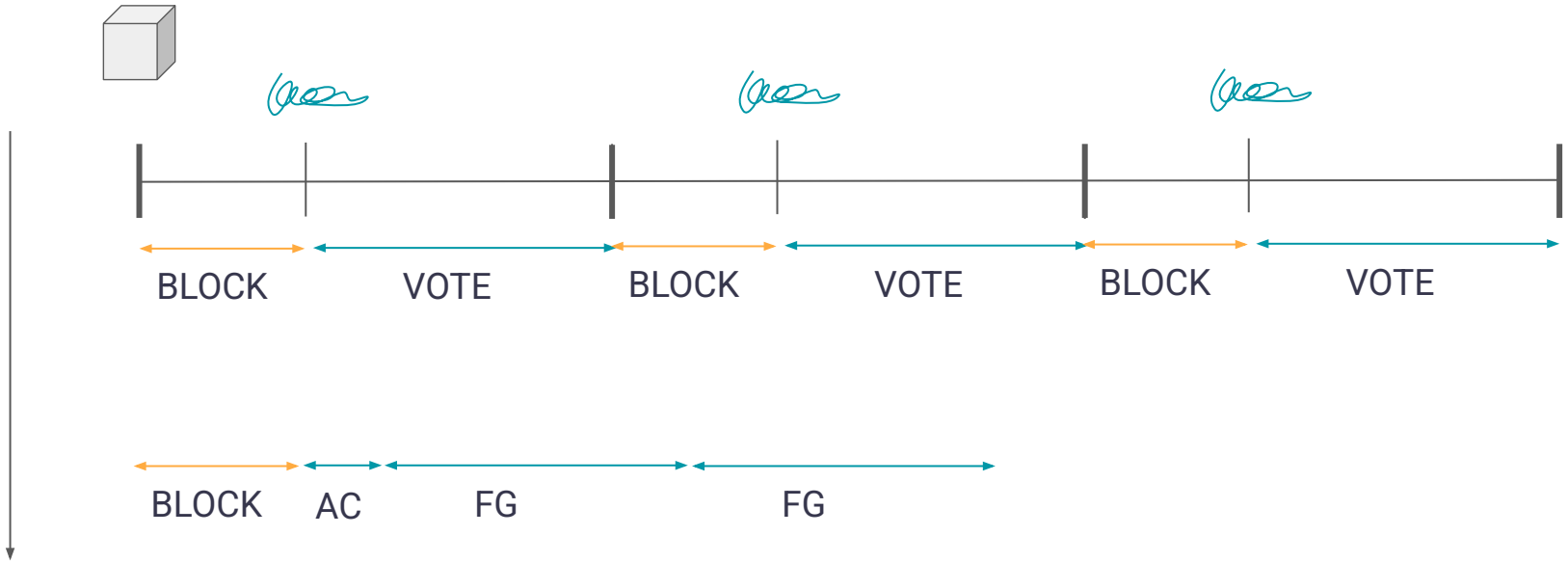
Majorum



Coupled timelines in Majorum

- **Finality slows down block production**
  - ➢ full voting round in the critical path
  - ➢ need small validator set to have fast slots
- **Block production slows down finality**
  - ➢ Slot isn't just voting (block propagation, execution, proposer synchronization mechanisms)

Coupled timelines in Majorum



Ex-Ante Reorgs
(LMD-GHOST)

Balancing Attacks
(Interaction)

Block production
vs.
finality trade-off

t

t + 6Δ

t + 10Δ

VOTE

VOTE

B

Justification

Finalization

B'

Fast Confirmation

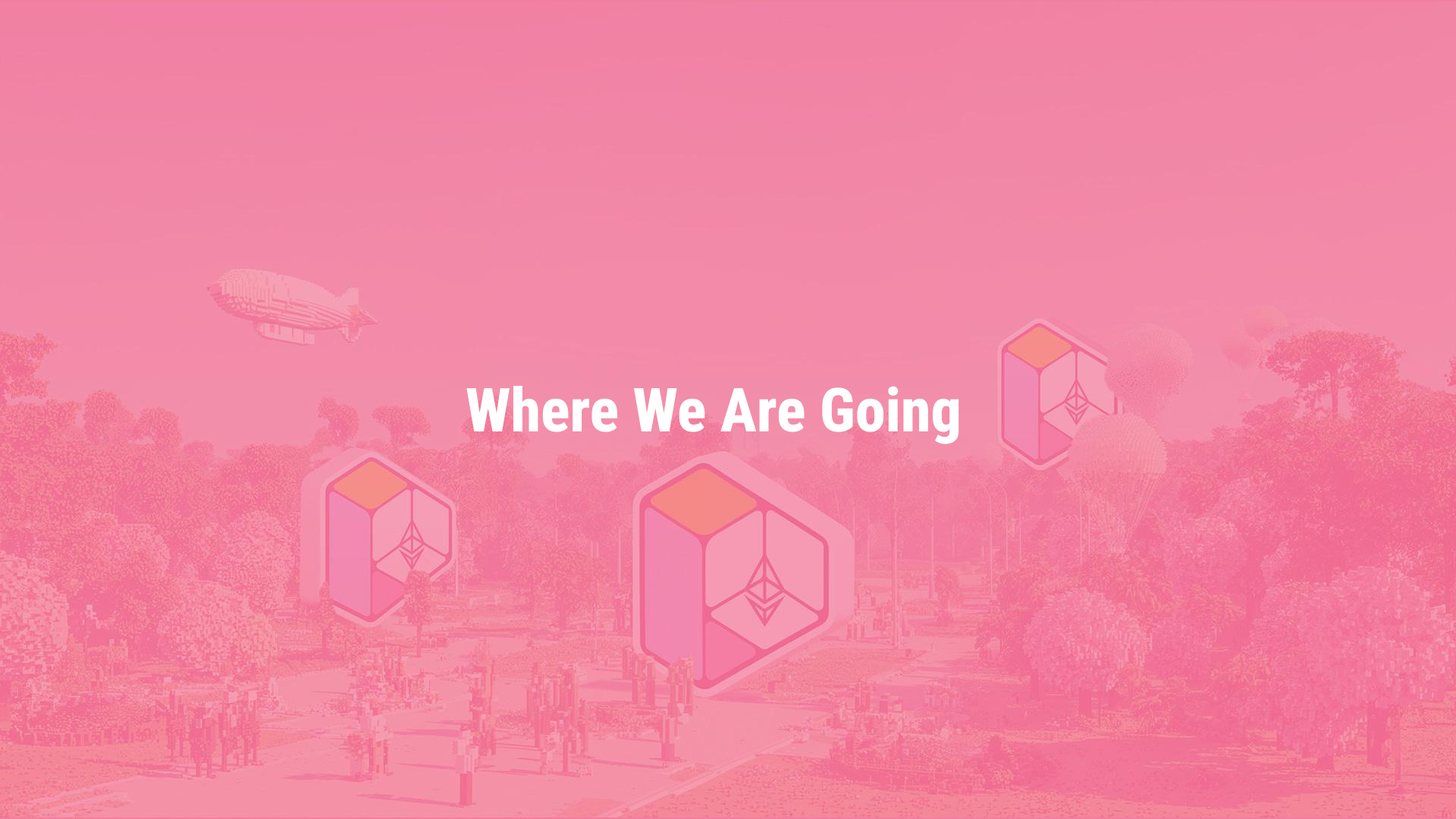
Justification

Ex-Ante Reorgs
(LMD-GHOST)

Balancing Attacks
(Interaction)

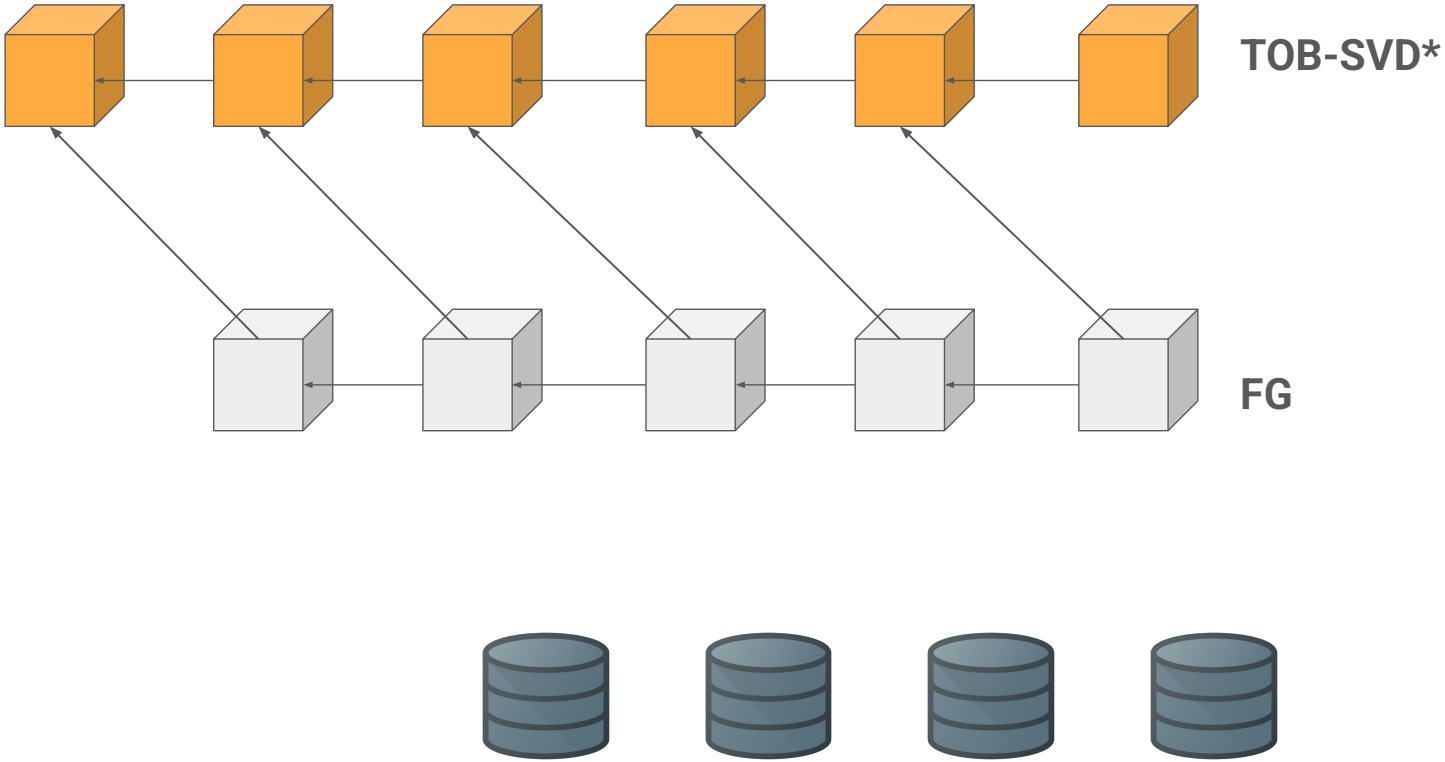
Block production
vs.
finality trade-off

Where We Are Going

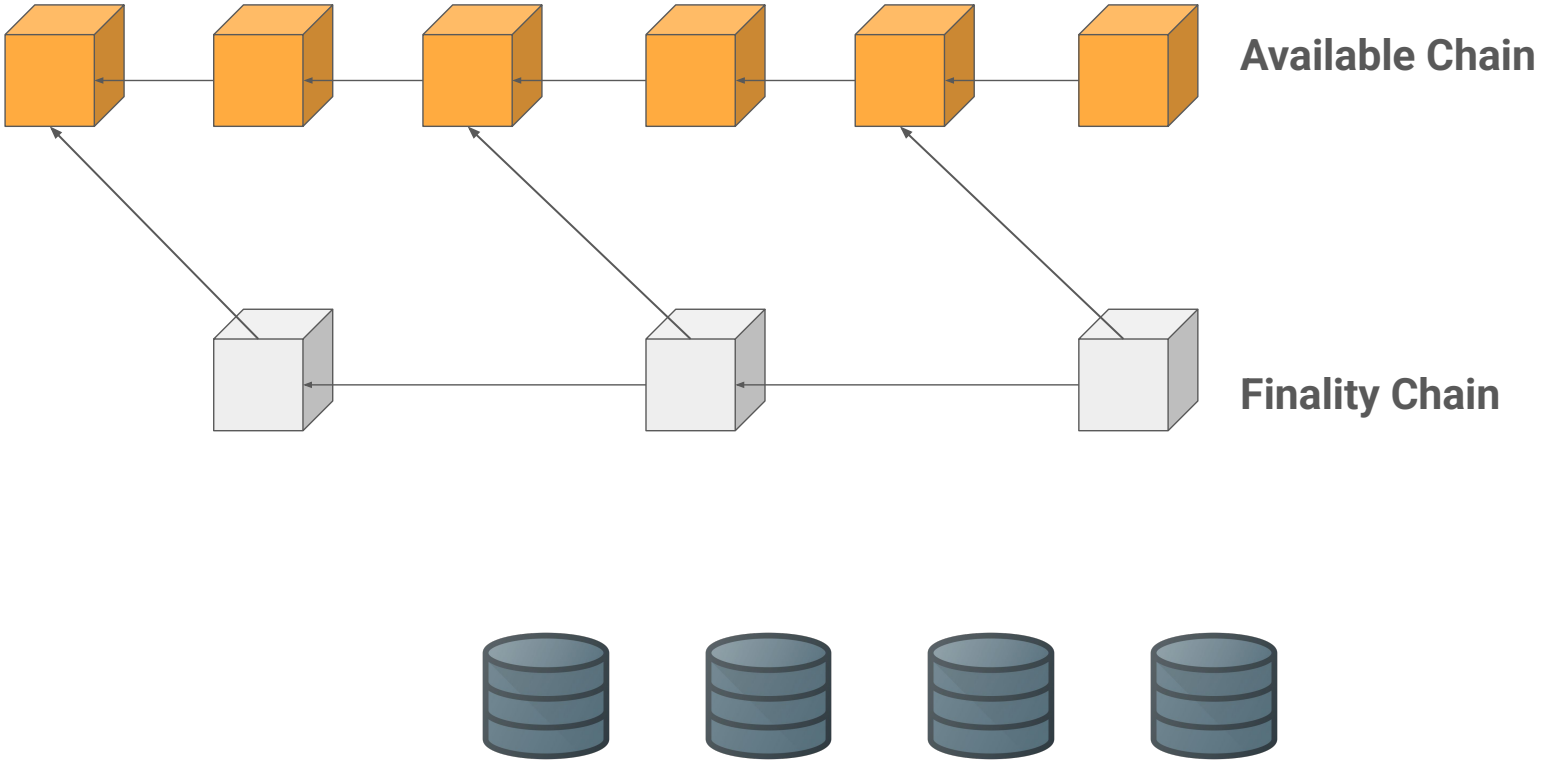


Decoupled Consensus

Majorum

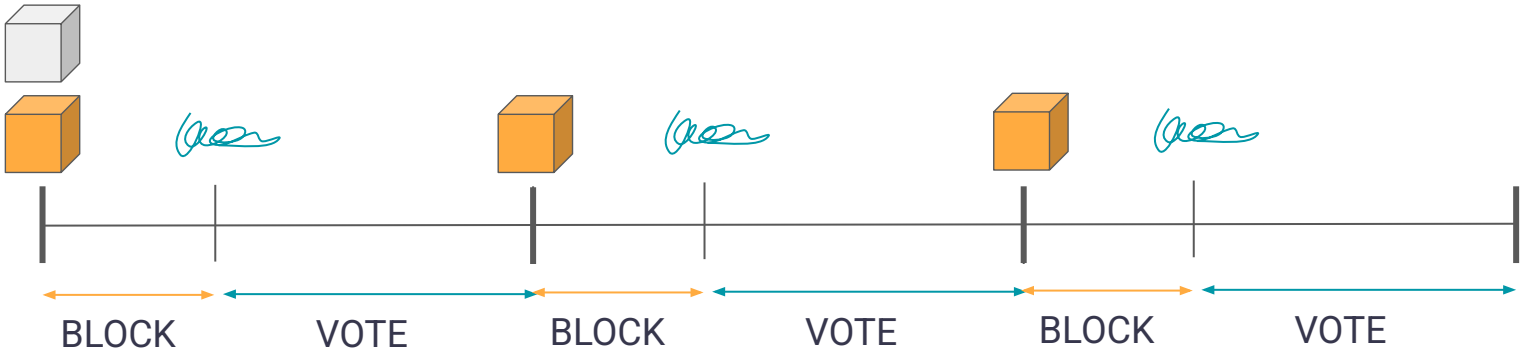


Decoupled Consensus



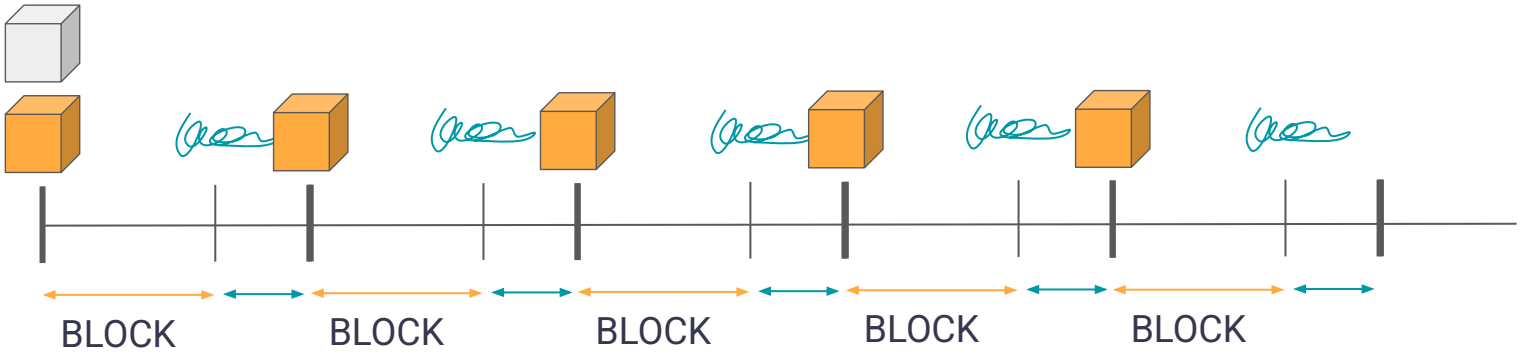
Coupled timelines in Gasper

1M Attestations
of ~100B over 32 slots!



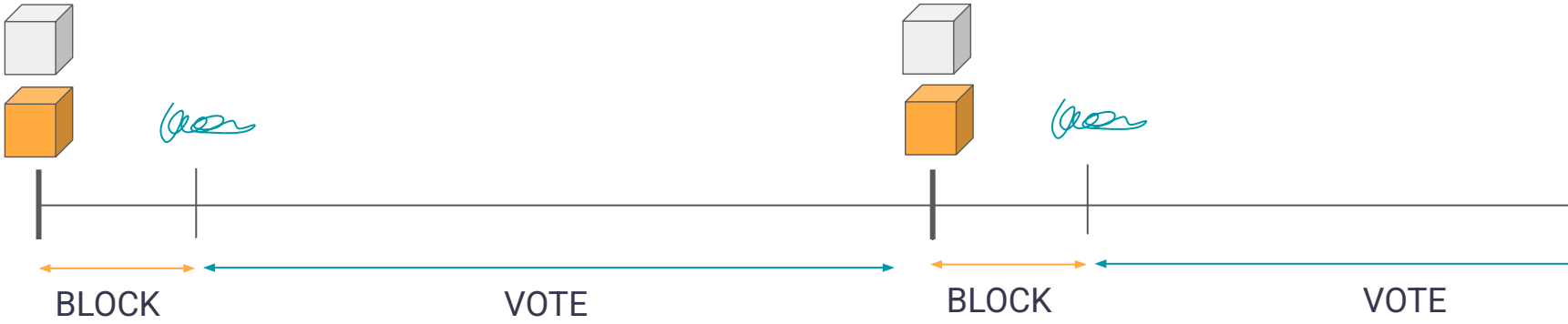
Coupled timelines in Gasper

Small committees
→ long epochs
→ slow finality



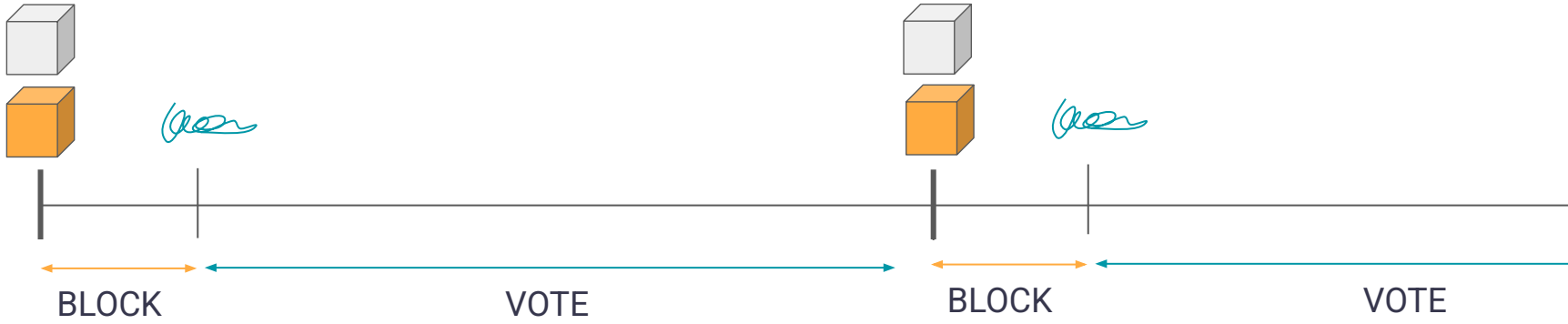
Coupled timelines in Gasper

Large committees
→ long slots
→ slow block production



Coupled timelines in Gasper

Large committees
→ long slots
→ slow block production



Block production and finality timeline are coupled!

LMD GHOST with ~256 validators and a fast-following finality gadget

vbuterin 3 Aug 1

Epistemic status: early exploration

Recently, there has been discussion about more aggressive ways to reduce Ethereum's slot time. This can be done in two ways:

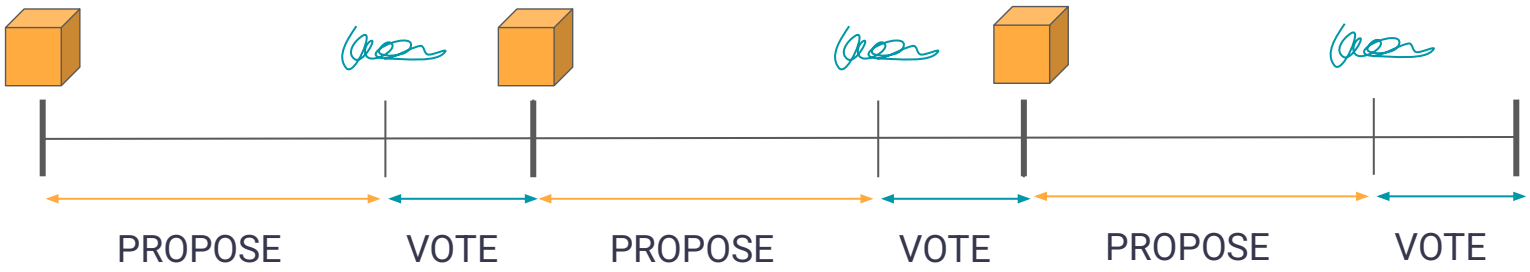
1. Reducing the δ parameter (our assumption on maximum expected network latency). This can only be done safely if we get improvements at the p2p layer that reduce latency
2. Re-architecting the slot structure to reduce the number of network latency rounds in one slot.

There is significant p2p hardening and optimization work going on to enable (1); the top is erasure coding. Research work is focusing on (2).

Quote Copy Quote Share

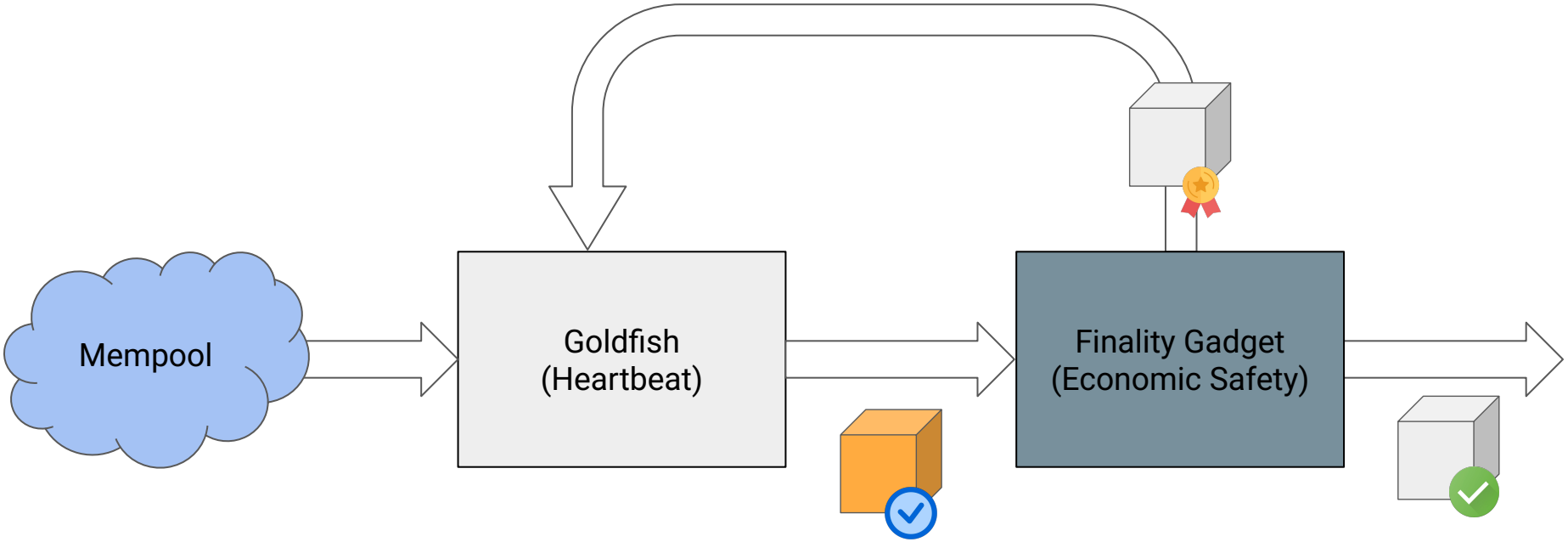
This post will argue that the optimal approach to (2) may be to move somewhat away from the tight coupling between slots and finality introduced in 3SF 14, and instead have a more separate LMD GHOST fork choice rule and finality gadget, with different participant counts.

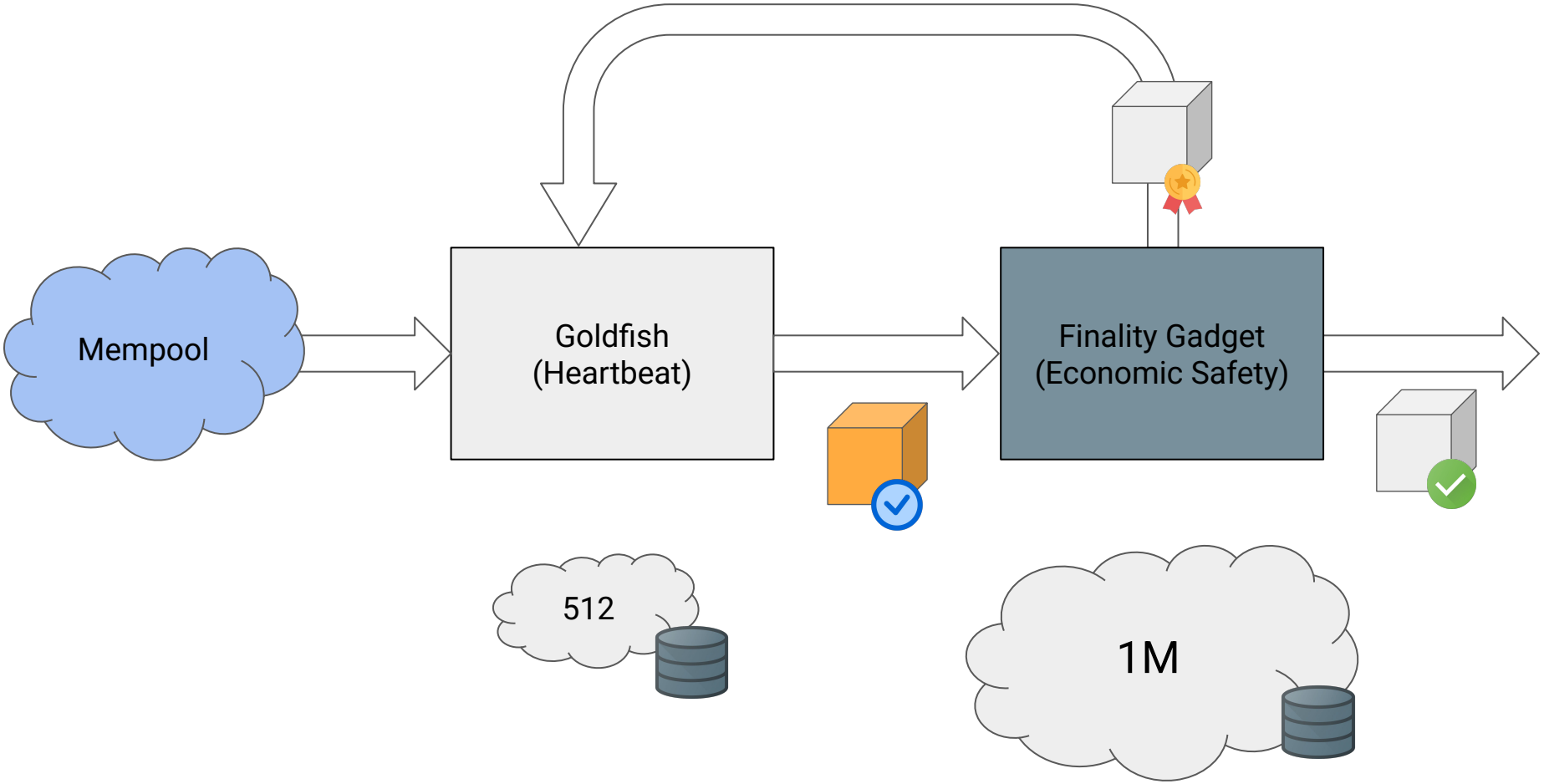
First, let's look at the current slot structure (source 13):



AC

FG


Mempool
Goldfish
(Heartbeat)
Finality Gadget
(Economic Safety)

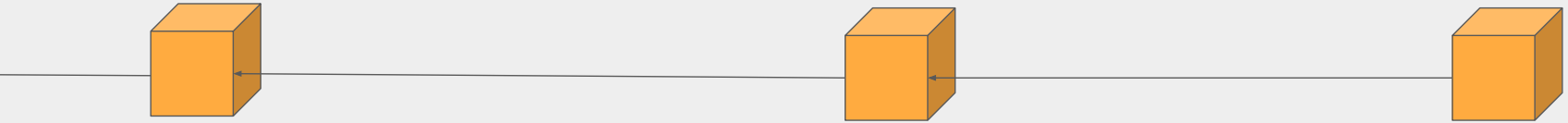

Mempool
Goldfish
(Heartbeat)
Finality Gadget
(Economic Safety)
512
1M

Goldfish

f < n / 2

Δ (Synchronous)

1 honest awake



Goldfish

f < n / 2

Δ (Synchronous)

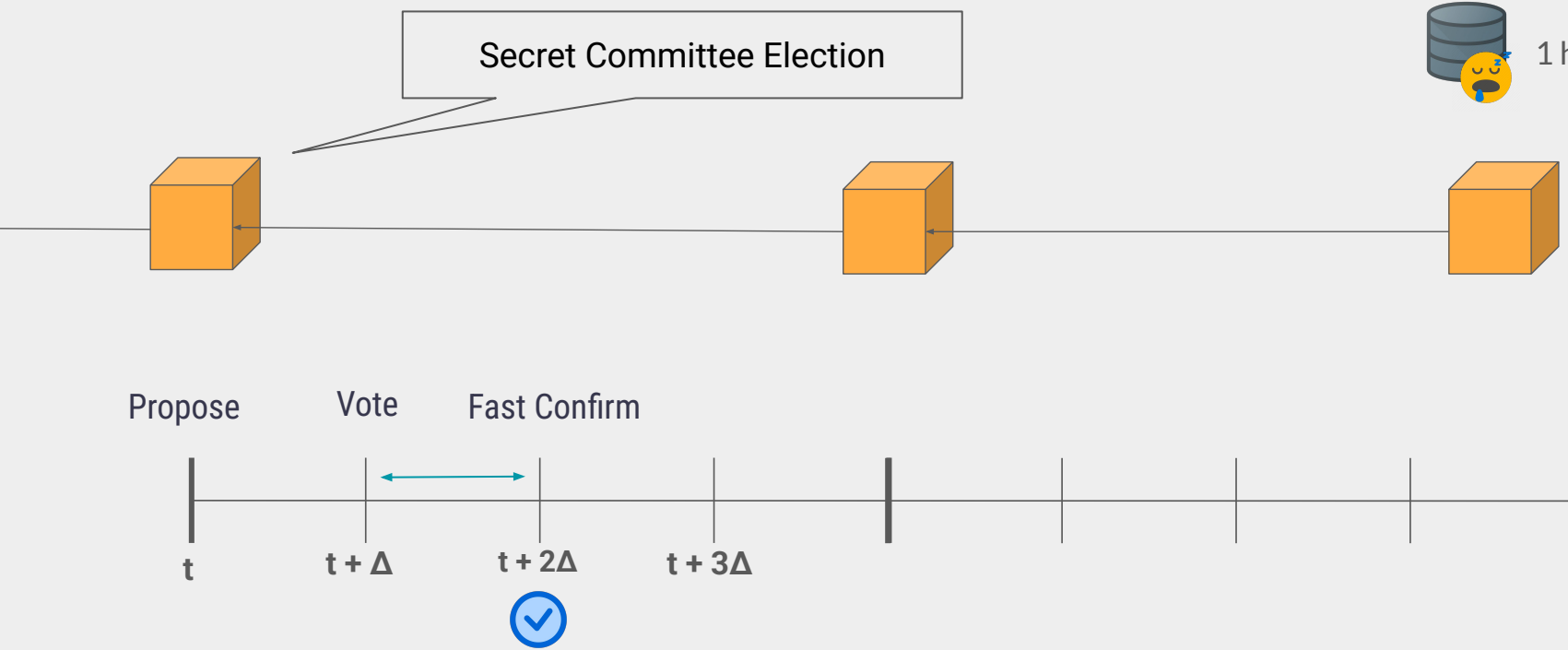
1 honest awake

Goldfish

f < n / 2

Δ (Synchronous)

1 honest awake

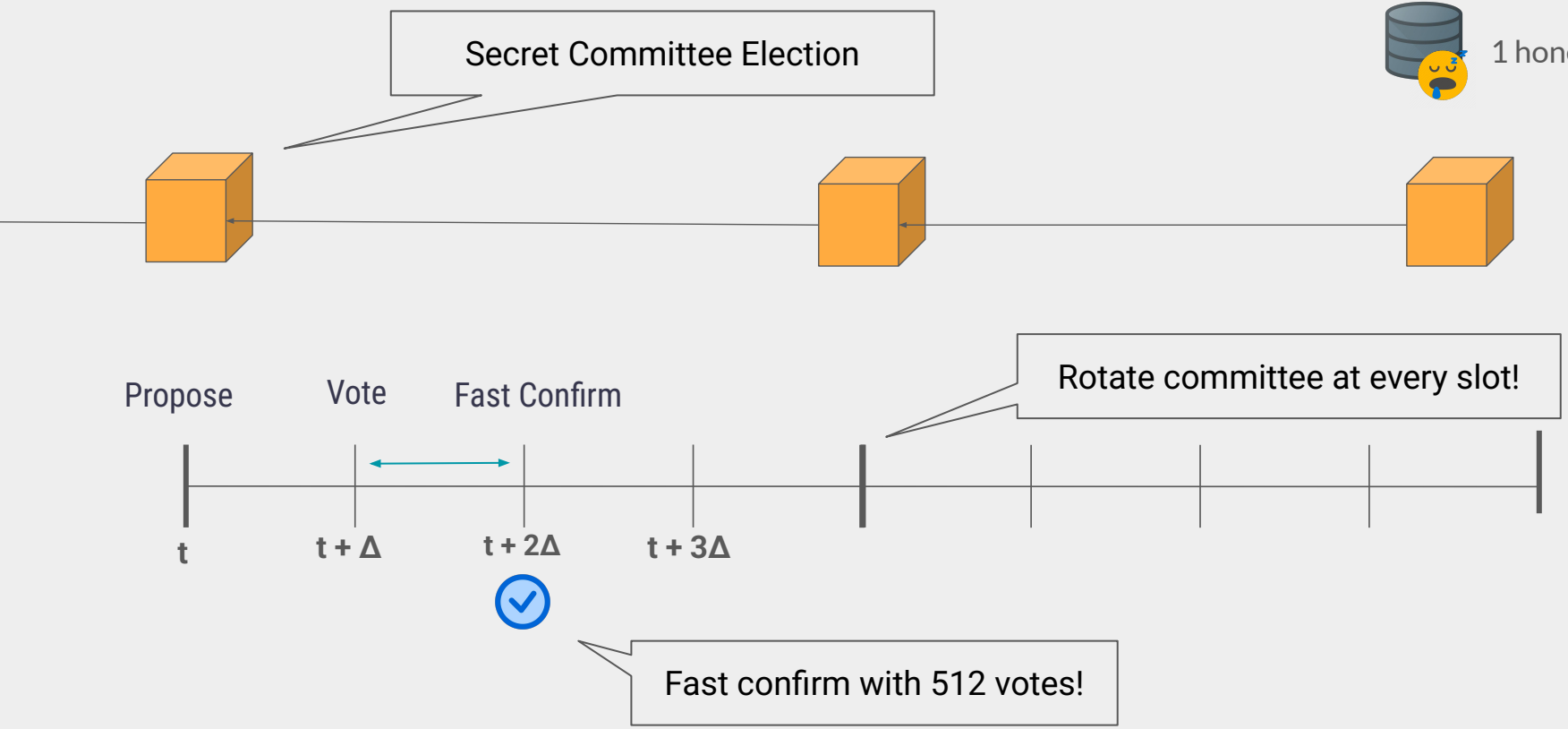


Goldfish

f < n / 2

Δ (Synchronous)

1 honest awake

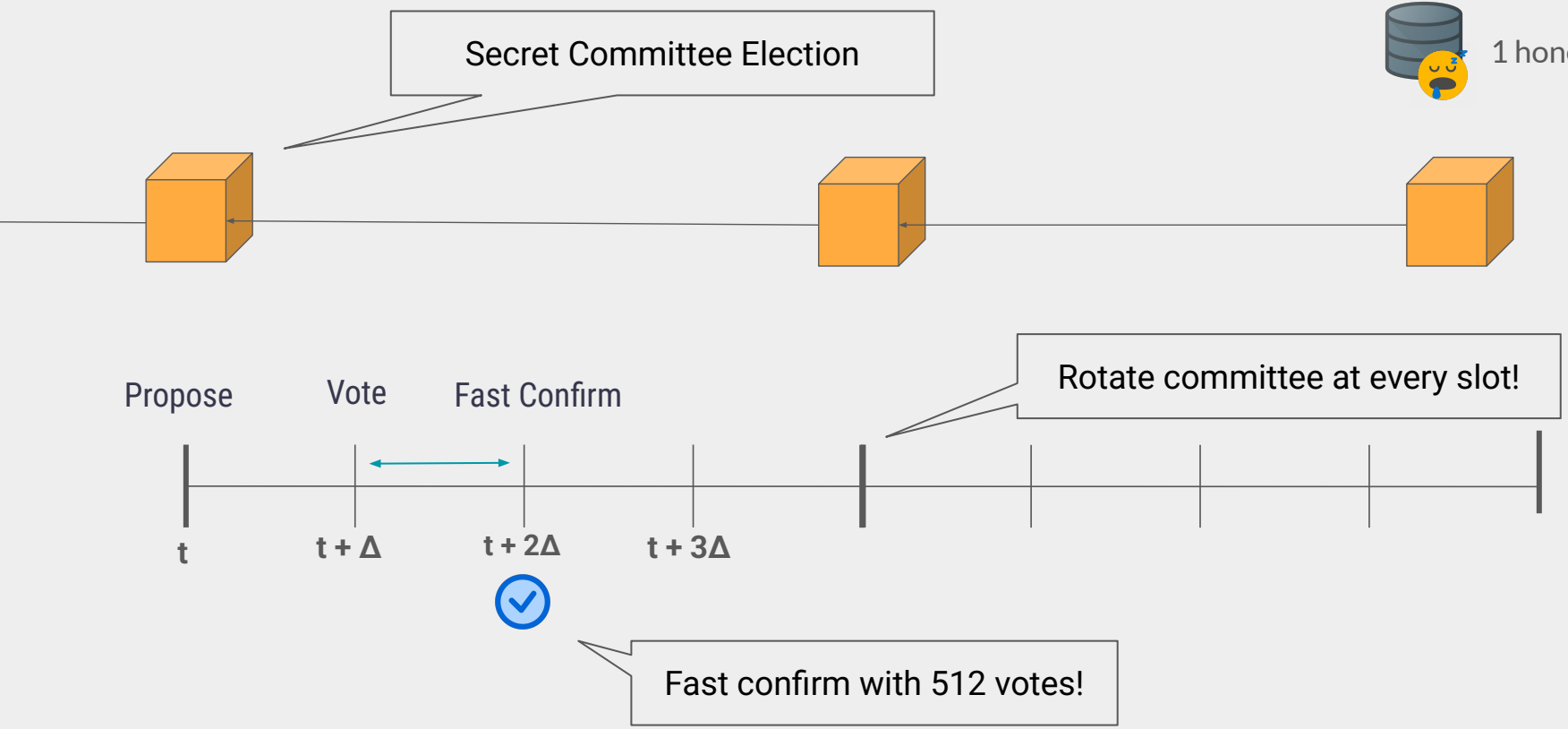


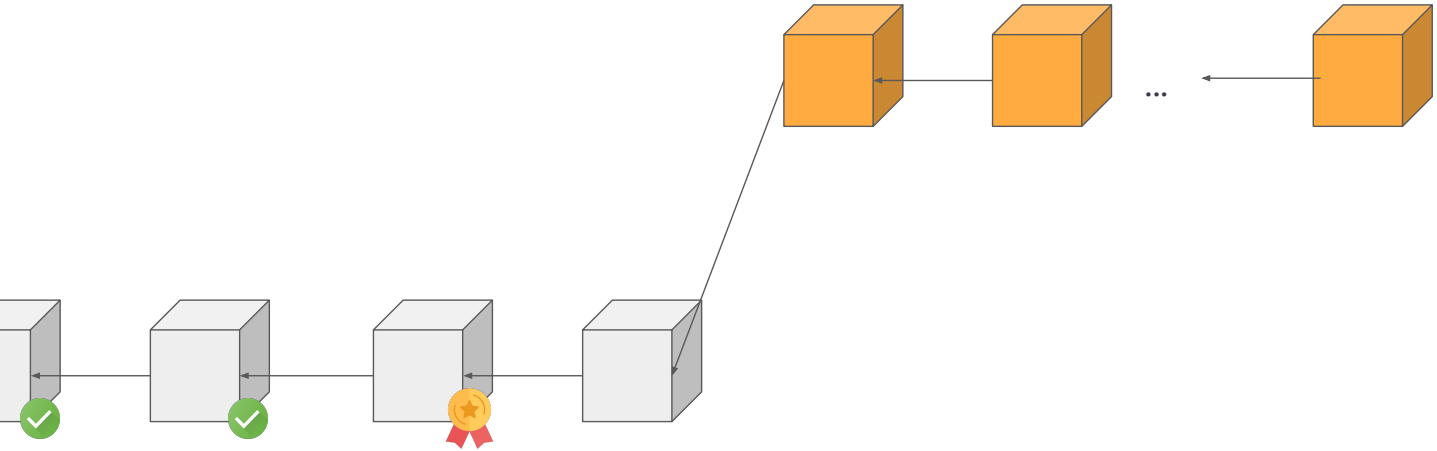
Goldfish

f < n / 2

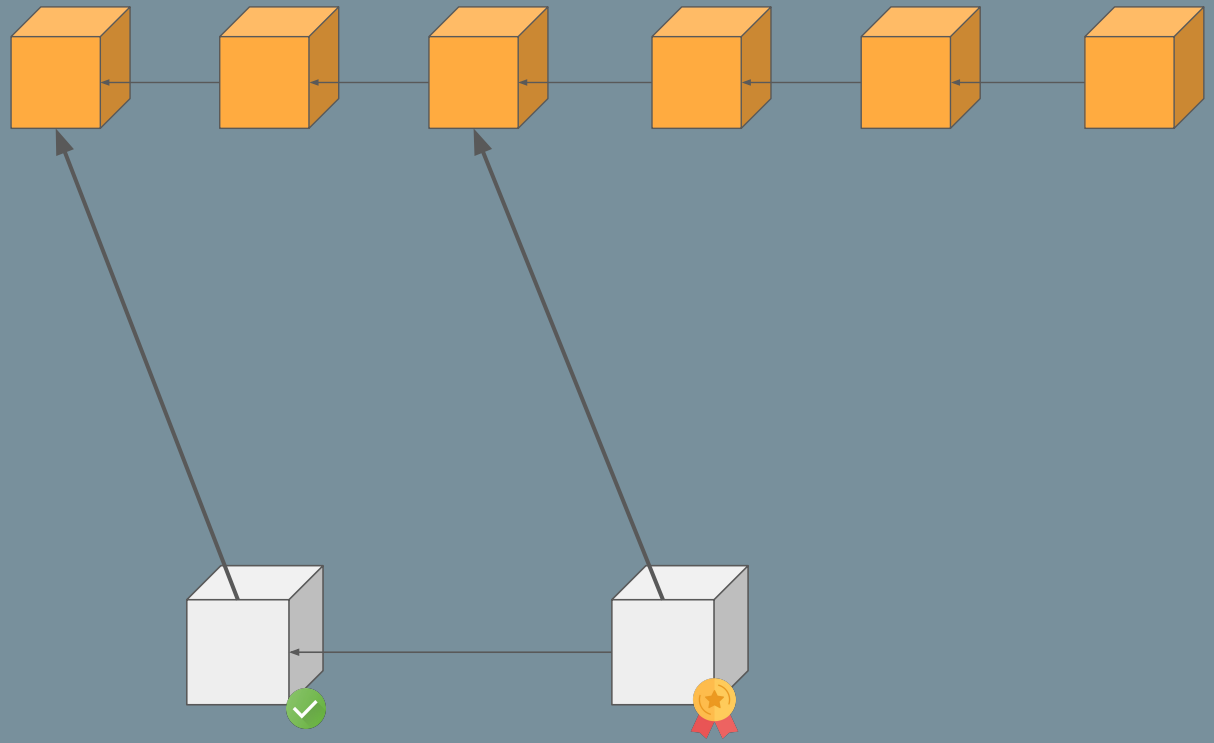
Δ (Synchronous)

1 honest awake





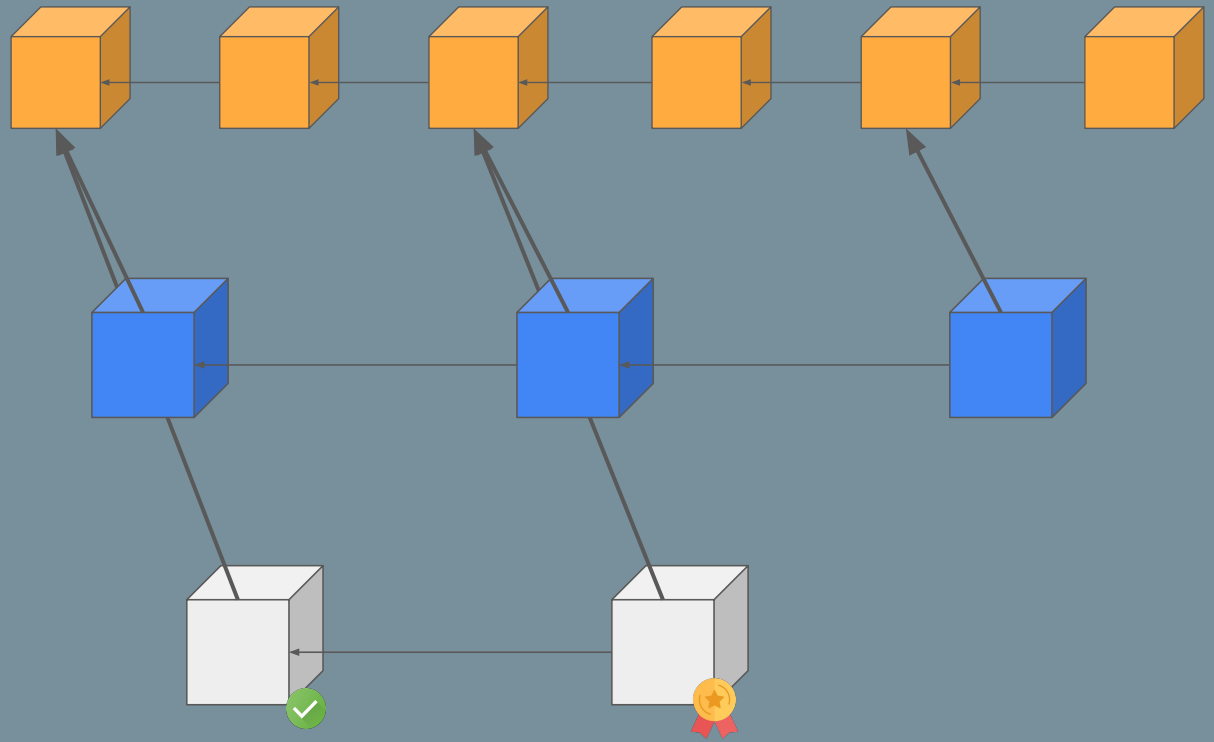
Stabilization Gadget



Goldfish (AC)

BFT (FG)

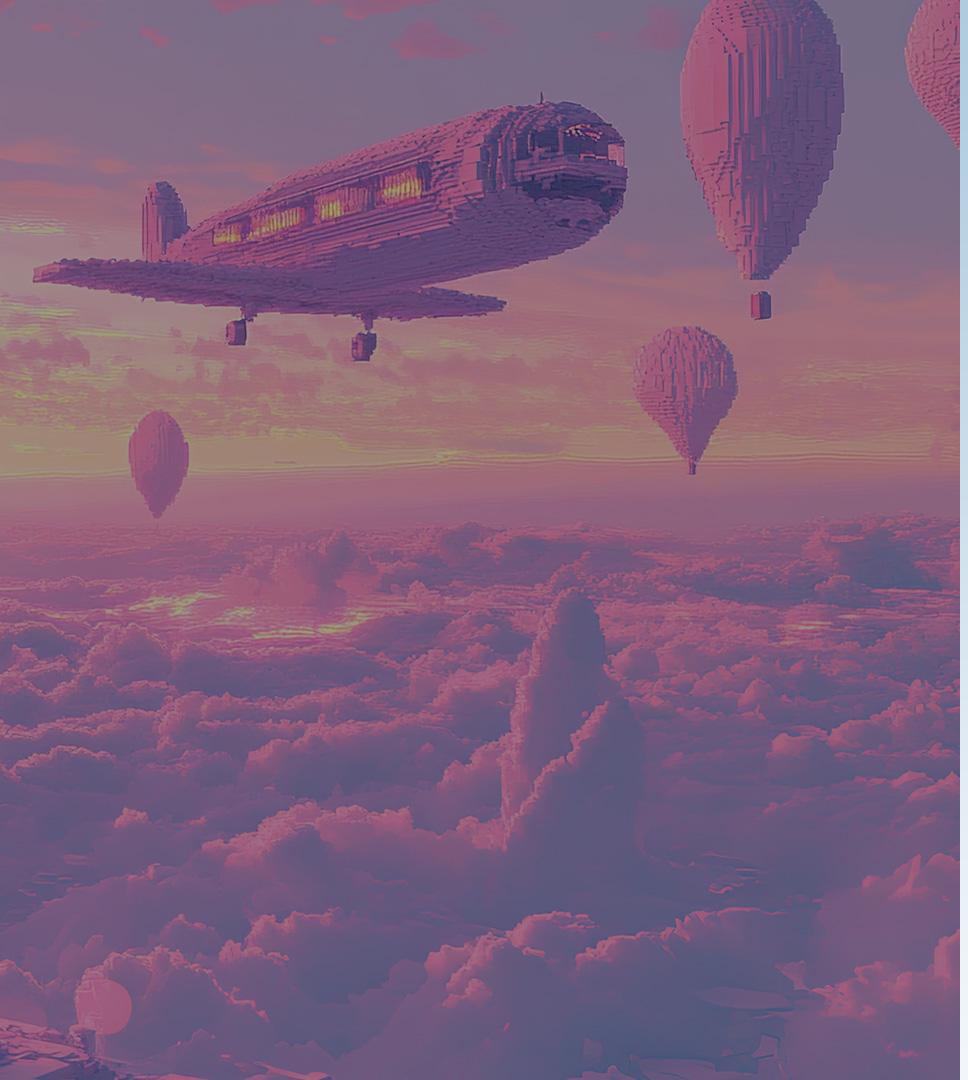
Stabilization Gadget



Goldfish (AC)

RLMD-GHOST (AR)

BFT (FG)



Advantages

- **PQ Heartbeat**
  - AC doesn't need aggregation → ship PQ chain fast
- **Benefit from incremental improvements**
  - Consolidation → incentivizes it too
  - p2p
- **Flexible**
  - Hard work up front
  - Possibility to switch out components in the future
- **Exposes trade-offs**
  - Privacy
  - Simplifies many other things

Unblocking faster finality with decoupled consensus

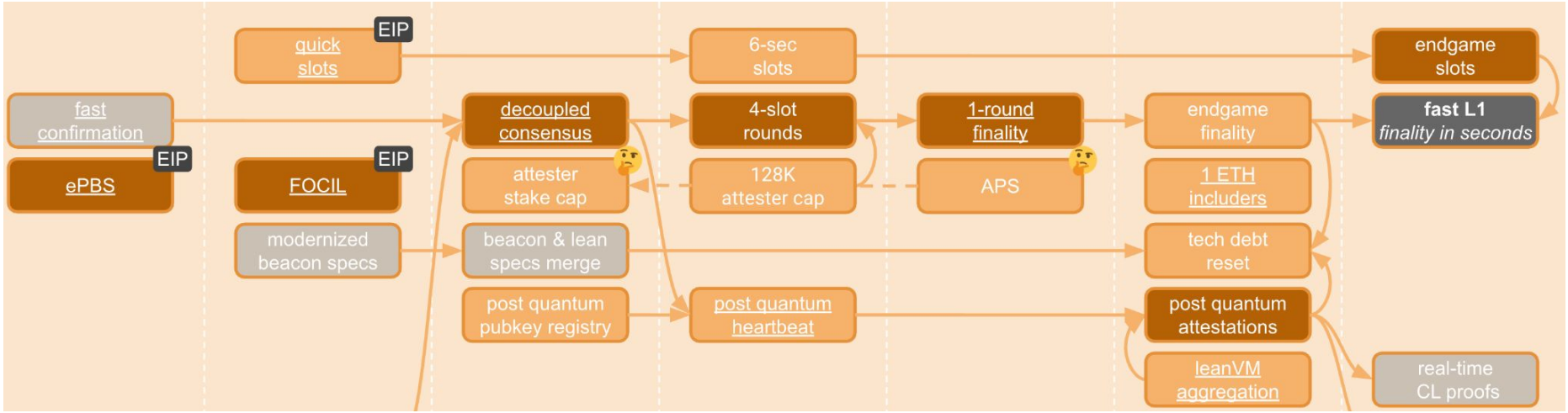
Consensus

fradamt 9 13h

Thanks to the EF Consensus team for discussion and comments

In the current consensus protocol, **the timelines of block production and finality are coupled**. A large committee (consisting of 1/32 of the whole validator set) attests in every slot, in the critical path. Shrinking the size of this committee, for example to 1/64 of the validator set, would enable faster slots, but it would slow down finality, because completing a full voting round (one epoch) would now accrue the overhead of 64 slots instead of 32. At one extreme of this tradeoff space, only a single validator votes in every slot, and a full voting round is only completed once every 1M slots or so, which would take more than a day even with 100ms slots. Even for less extreme committee sizes (and resulting epoch lengths), the overhead from doing “finality via slot-by-slot accumulation” cannot be discounted.

On the other hand, increasing the size of the committee would make finality faster, at the cost of slower block production (longer slots). At this other extreme of the tradeoff space, the whole validator set votes in every slot, and slot time largely depends on how fast we can aggregate it.



strawmap.org

Consensus Goals*

Fast Finality that allows

- Post-Quantum signatures?
- Validator anonymity?
- 100’000-1M validators?

Fairness & Decentralization

⇒ geographic decentralization
⇒ no optimistic responsiveness

No Downtime: tolerate, e.g., crash faults by multiple clients

⇒ (pseudo) dynamic availability?

Fast Blocks

⇒ fast (available) chain

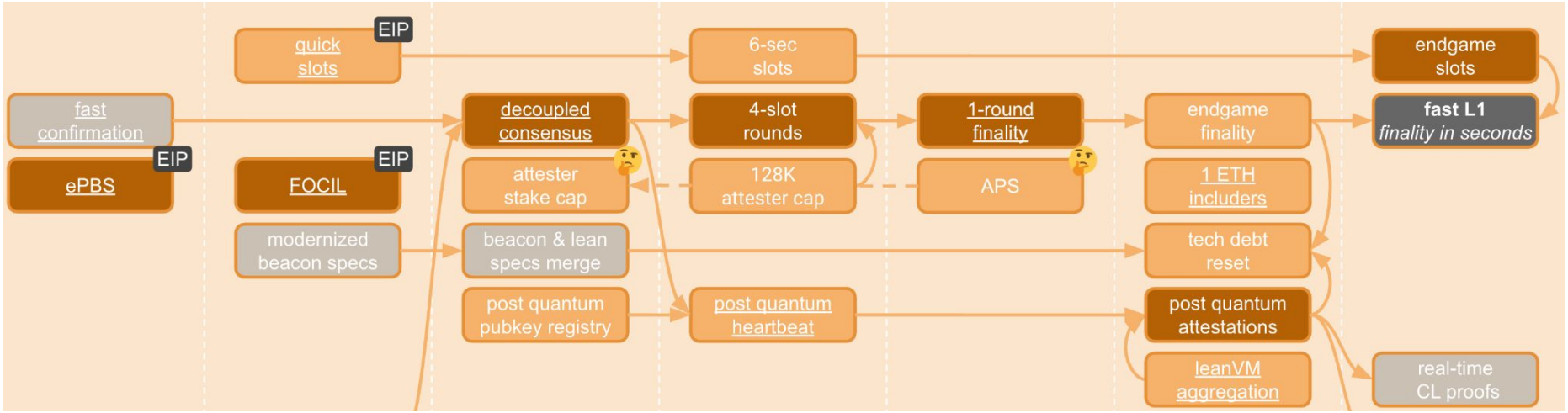
Accountable Safety
⇒ economic security

Accountable Liveness
⇒ minority forks have good UX

Veto Power by Minority
No bug can finalize a wrong chain

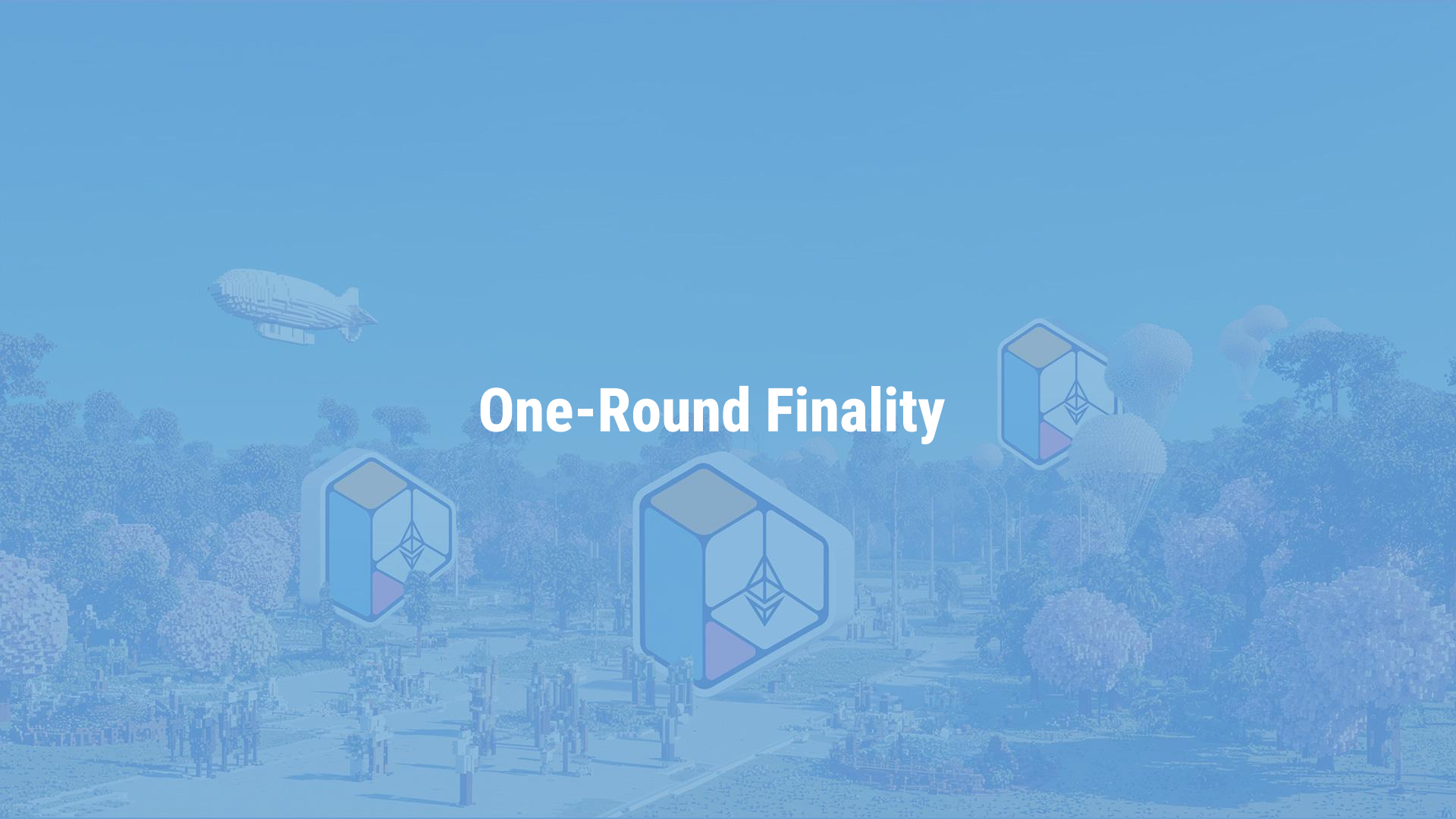
⇒ liveness threshold of 83% is fine

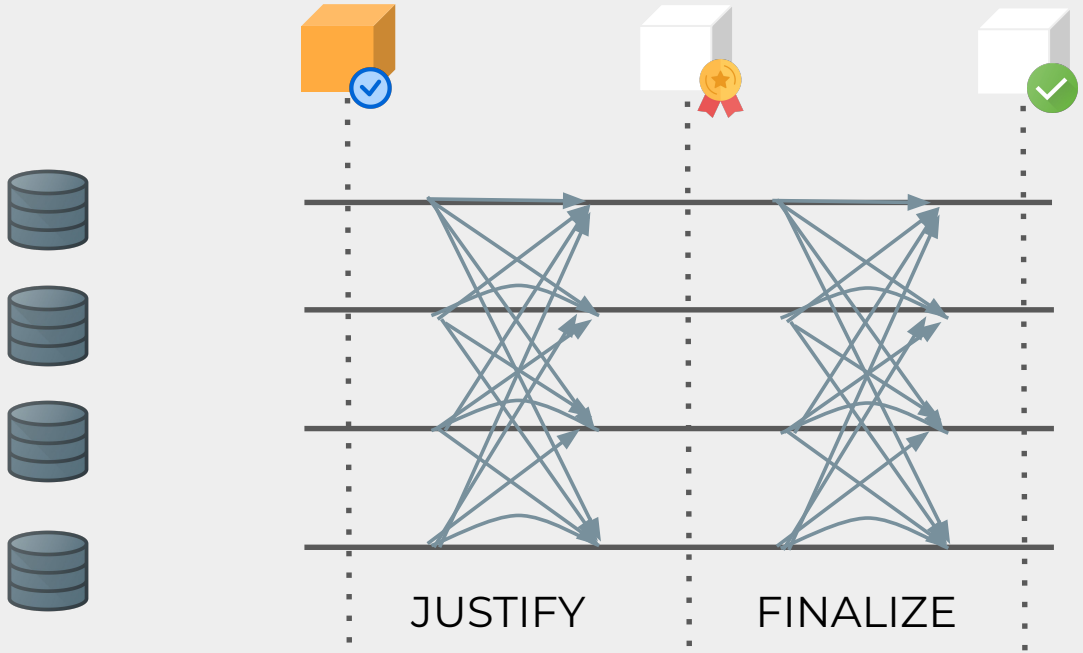
Endgame Components

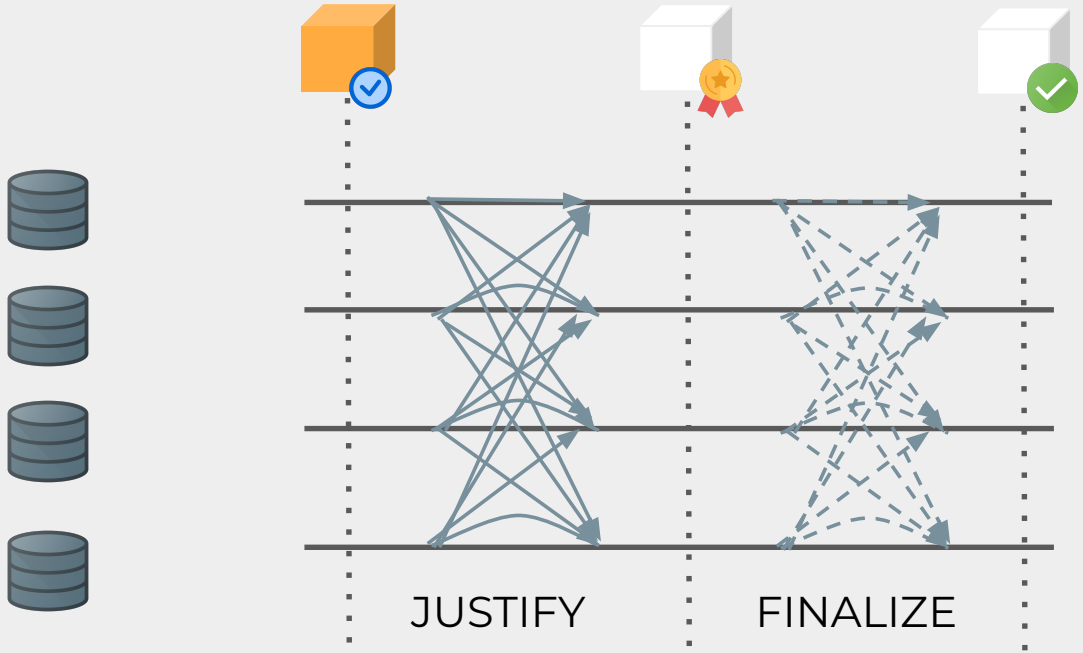


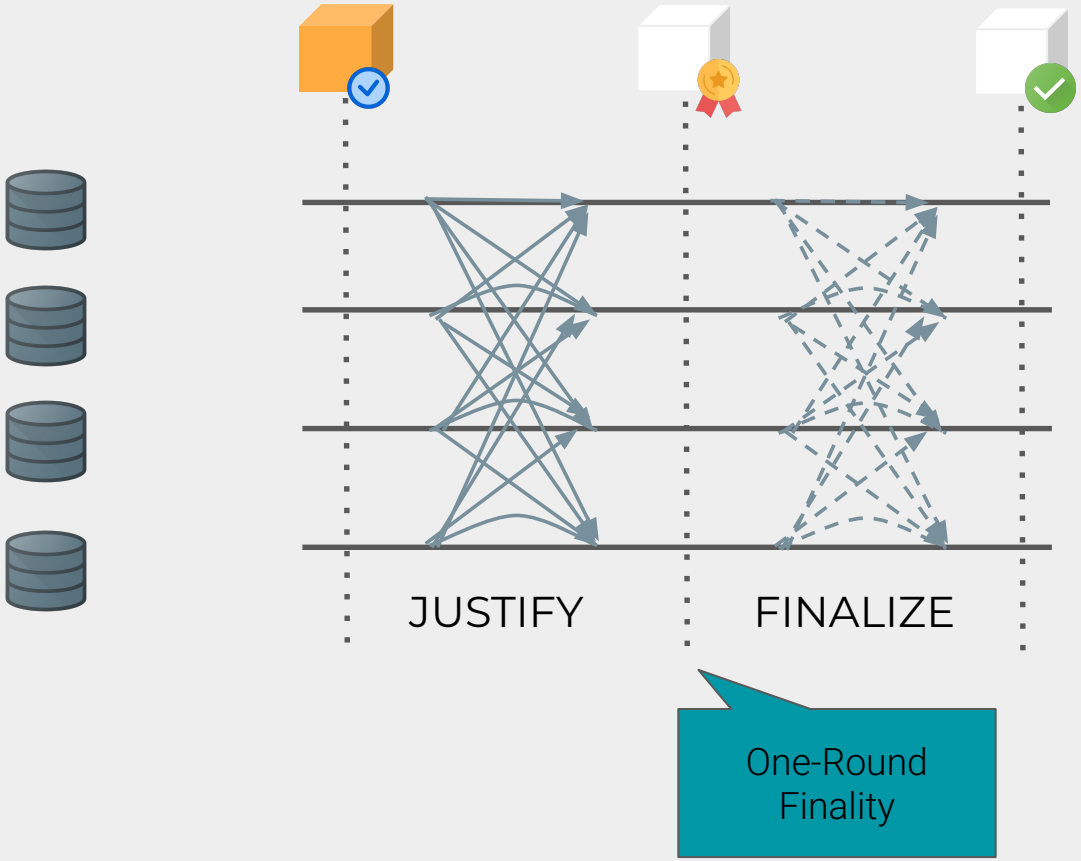
strawmap.org

One-Round Finality

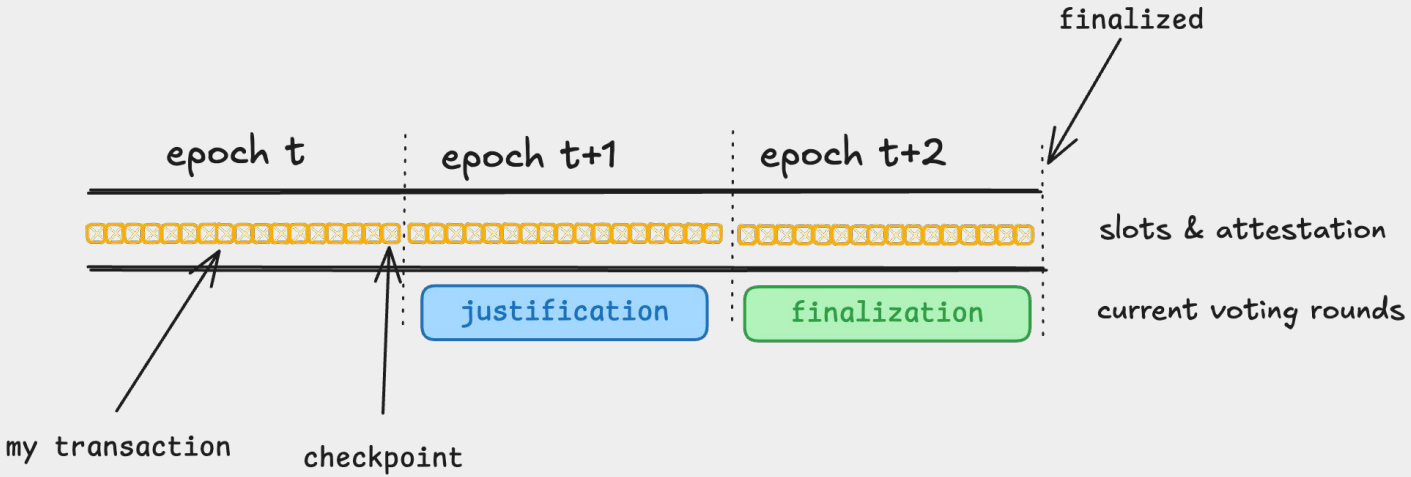



JUSTIFY
FINALIZE

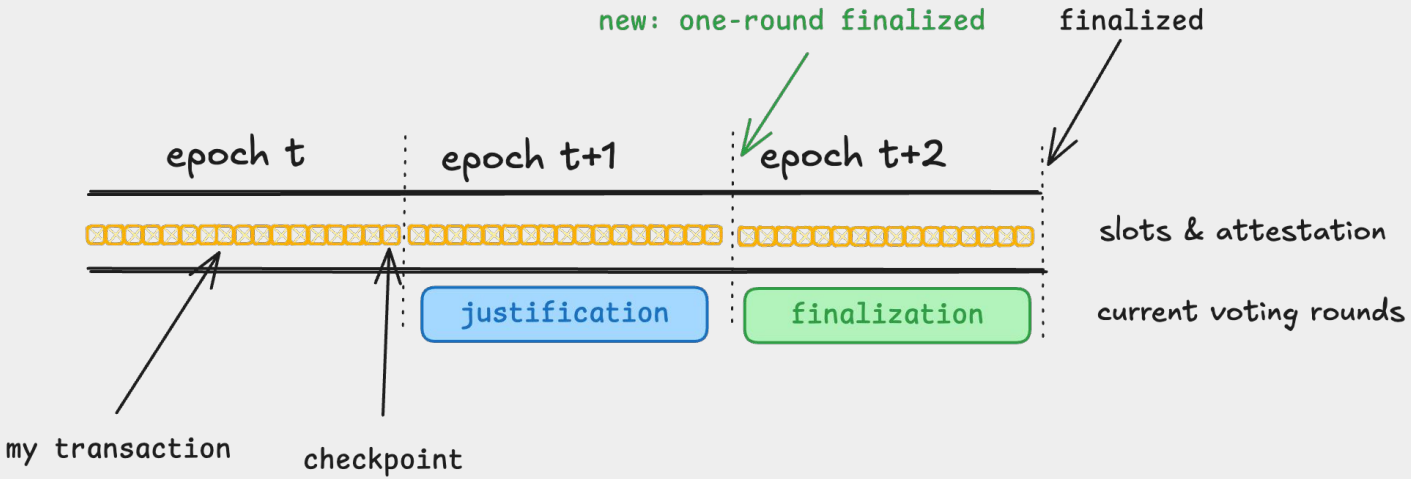

JUSTIFY
FINALIZE


JUSTIFY
FINALIZE
One-Round
Finality

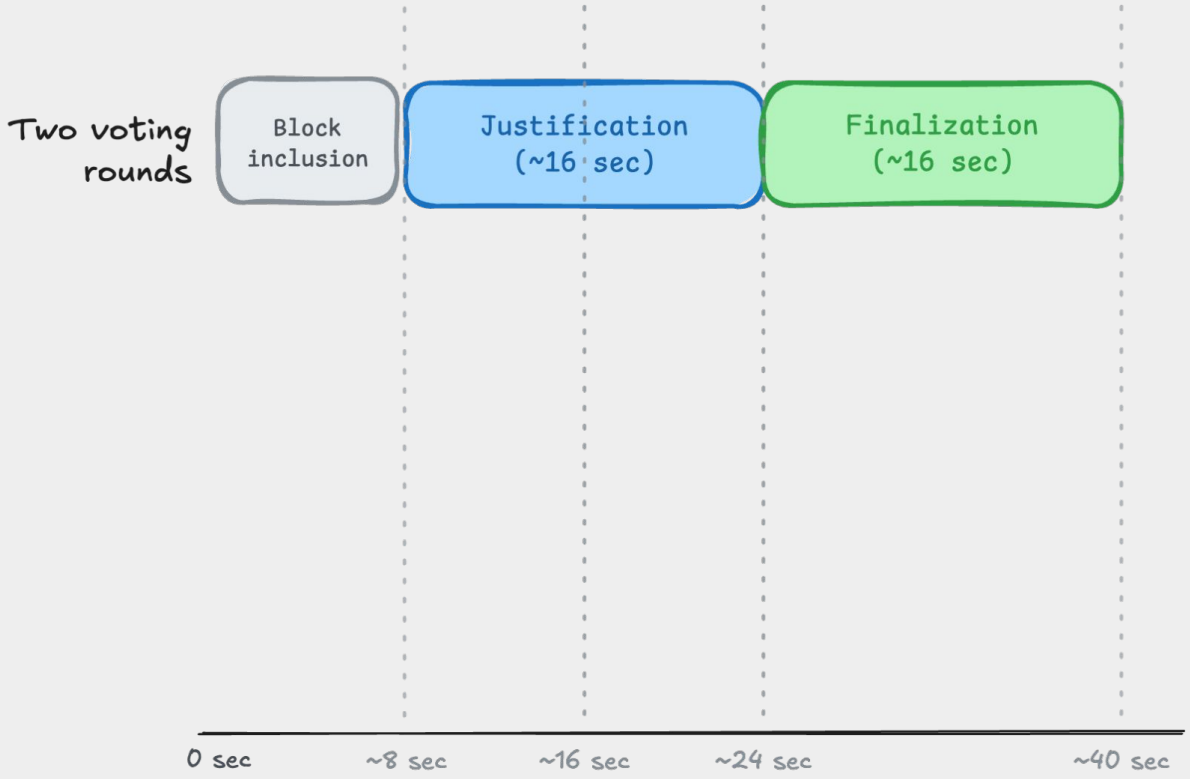
One-Round Casper FFG



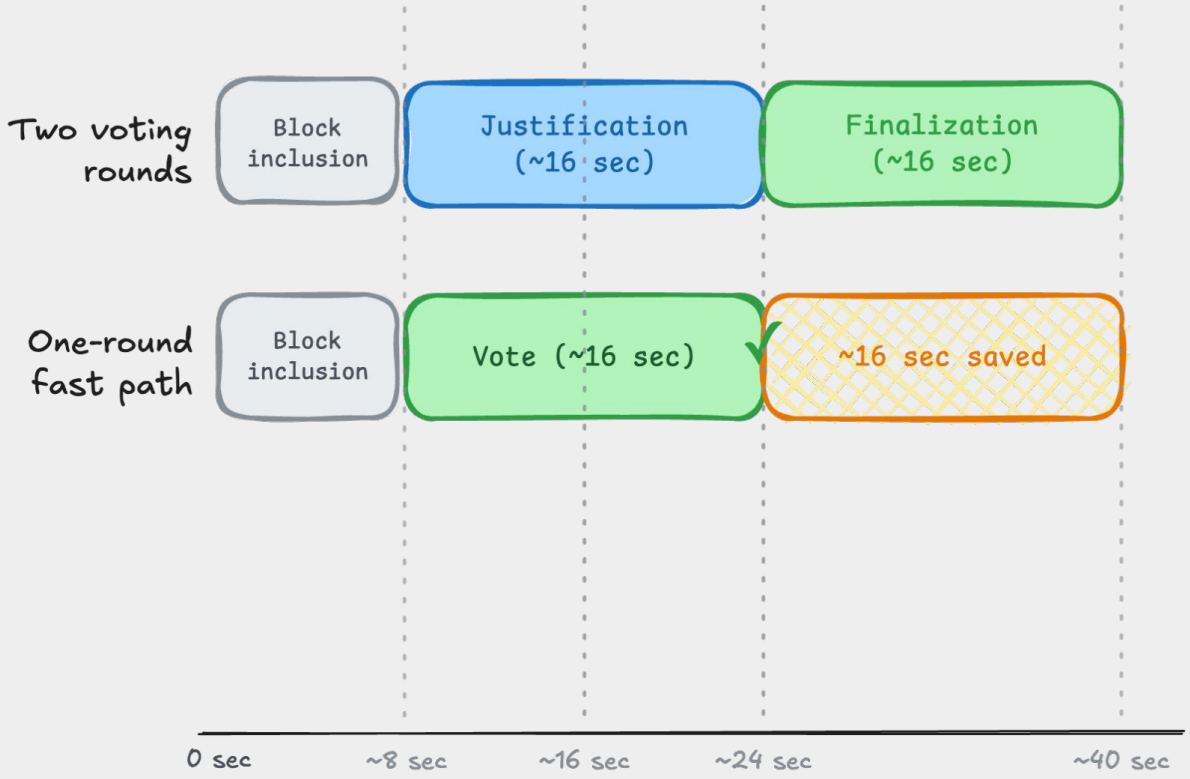
One-Round Casper FFG



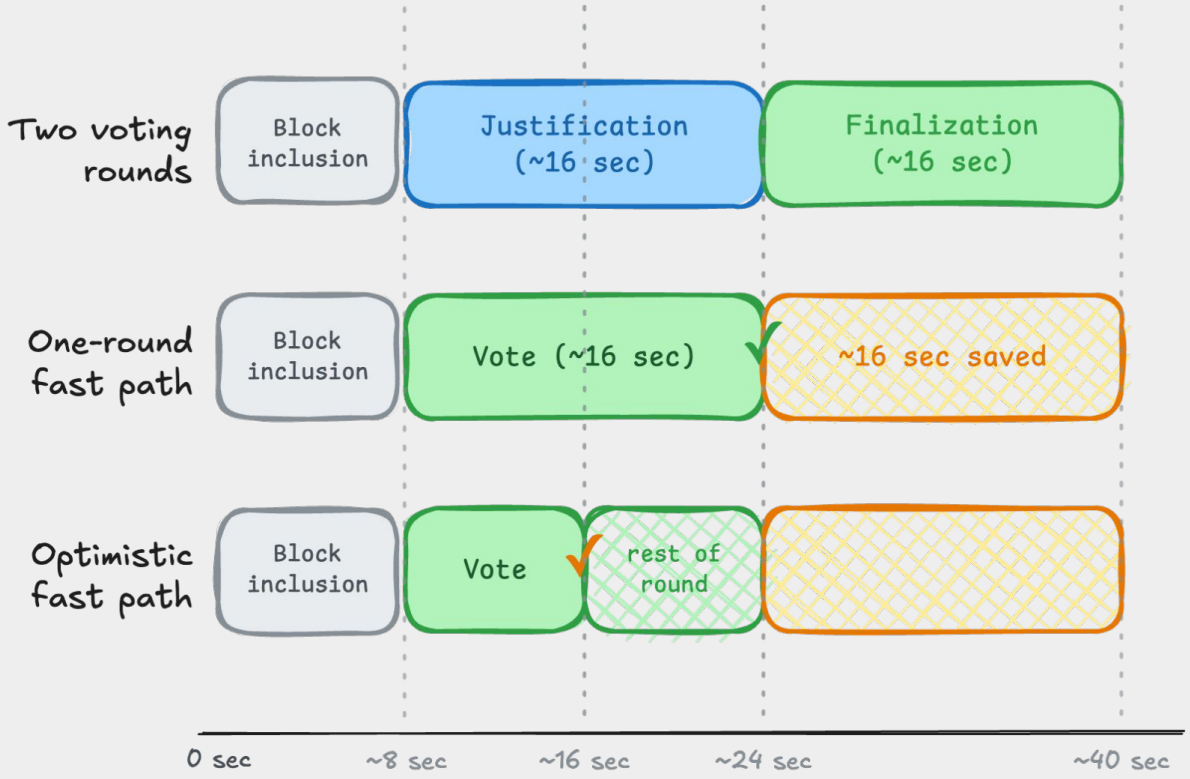
One-Round Decoupled Consensus



One-Round Decoupled Consensus

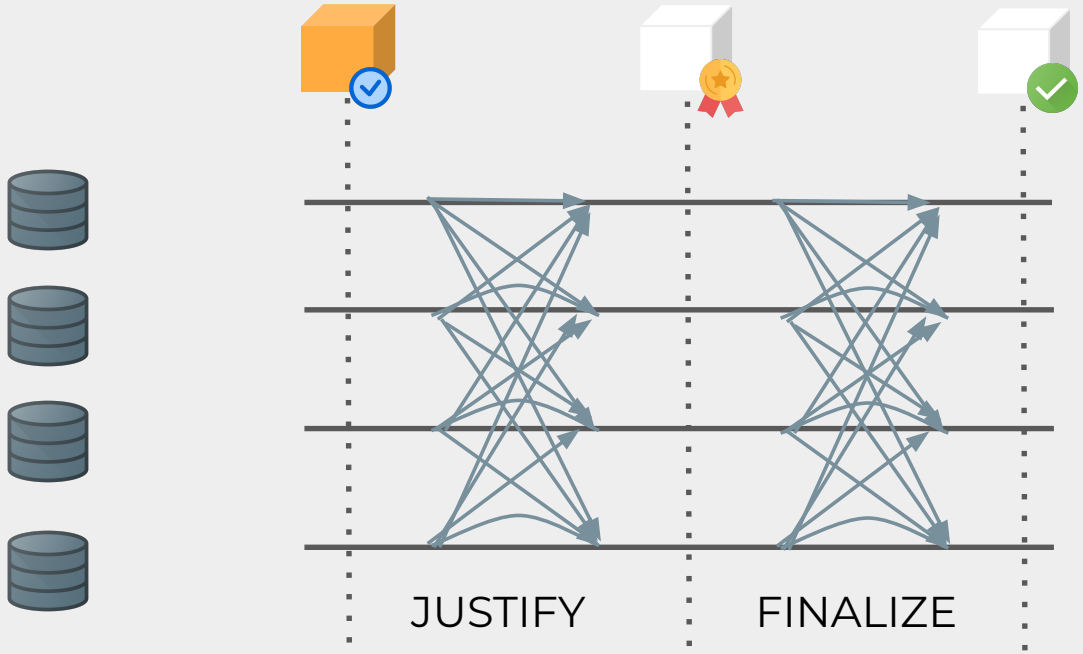


One-Round Decoupled Consensus



Optimistic

f = 33%

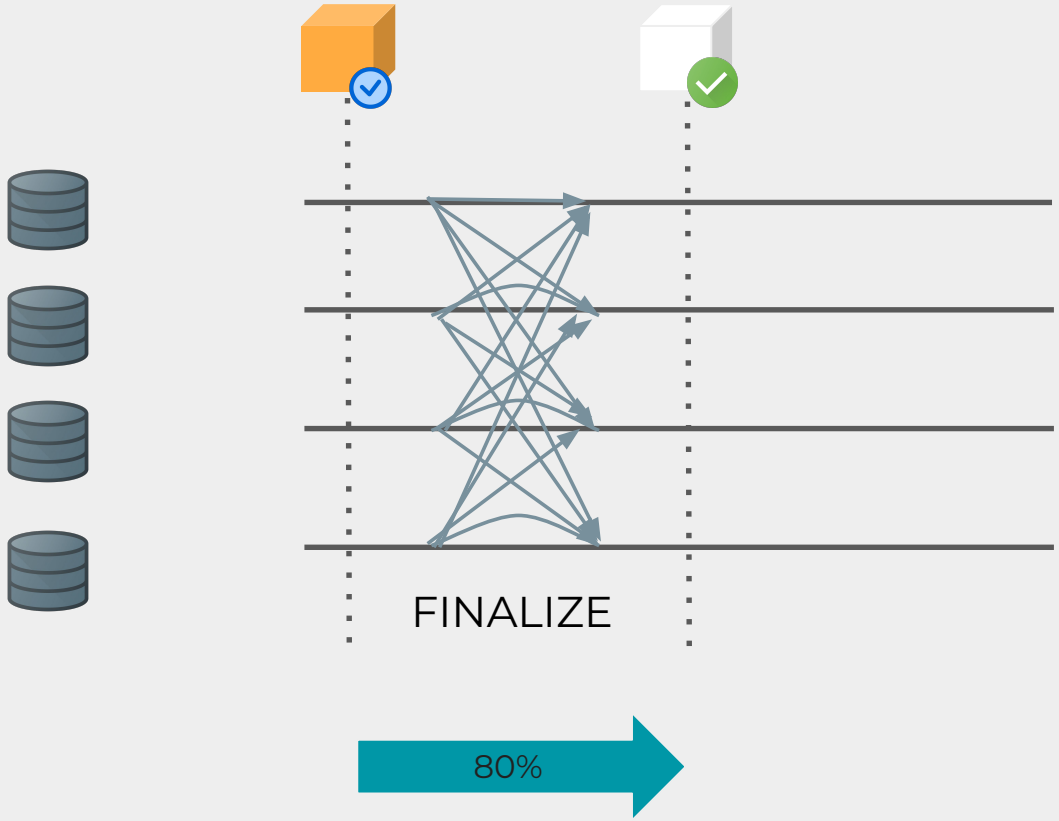


Fast Path: 100%

Slow Path: 67% 67%

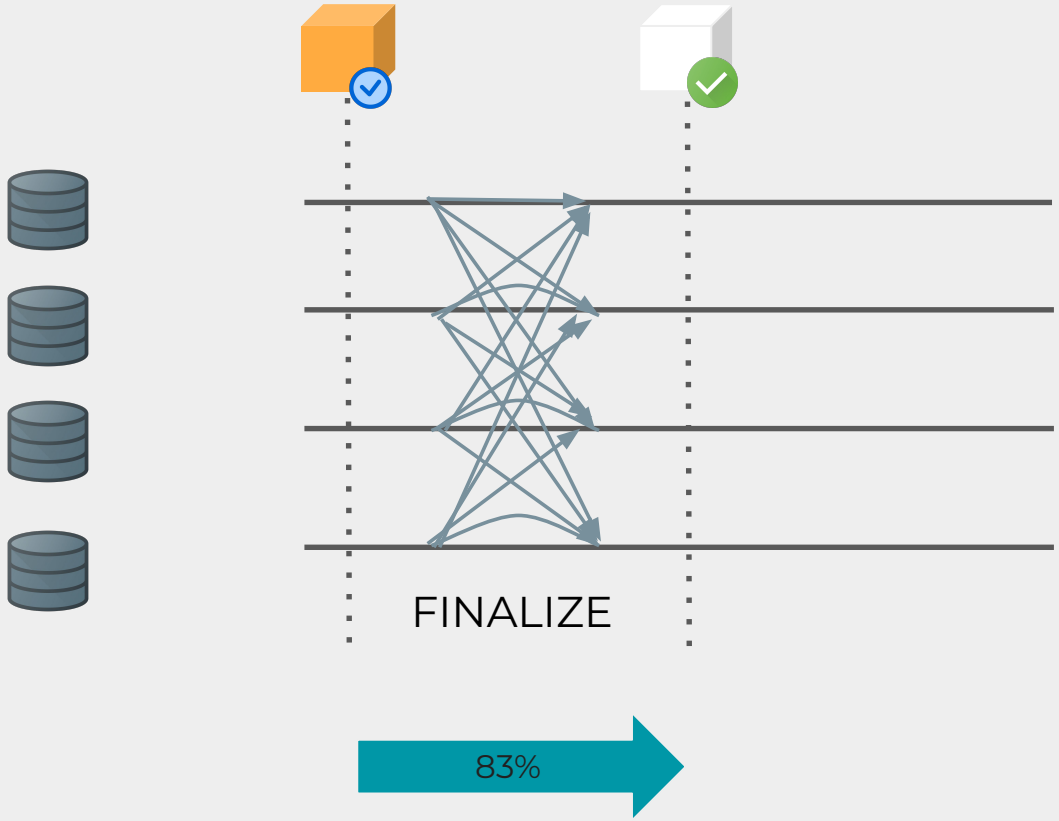
Pure

f = 20%



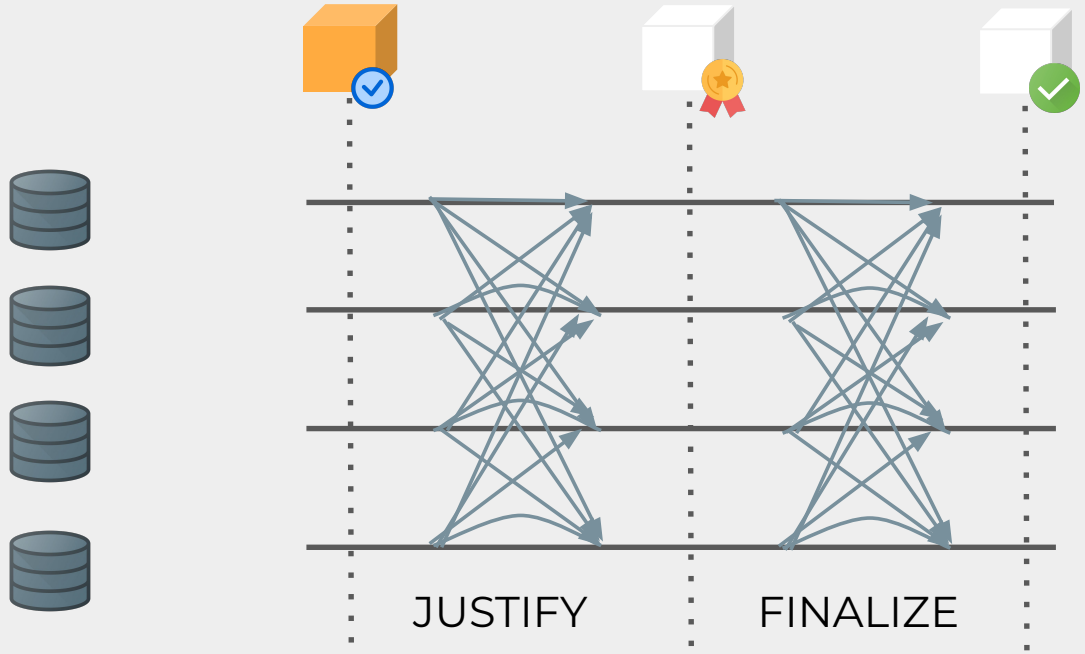
Pure

f = 25%



Hybrid

f = 33%



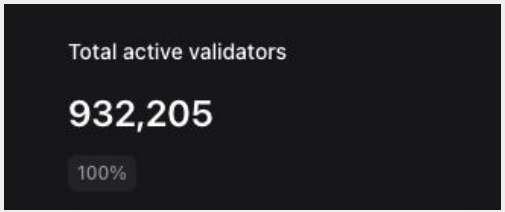
Fast Path: 92%

Slow Path: 51% 83%

Why Ethereum Is Uniquely Positioned

Large Validator Set

→ removing a round is meaningful



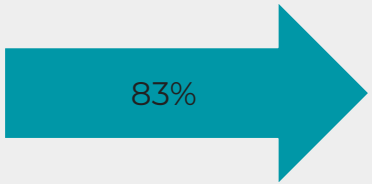
High Participation

→ we can benefit from optimistic conditions

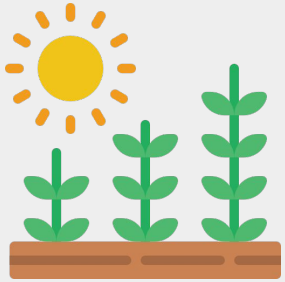
Participation %
99.81%
99.57%

Synergies with High Finality Threshold

→ we benefit from raising the finality threshold from 67% to e.g., 83%



Synergies with High Finality Threshold



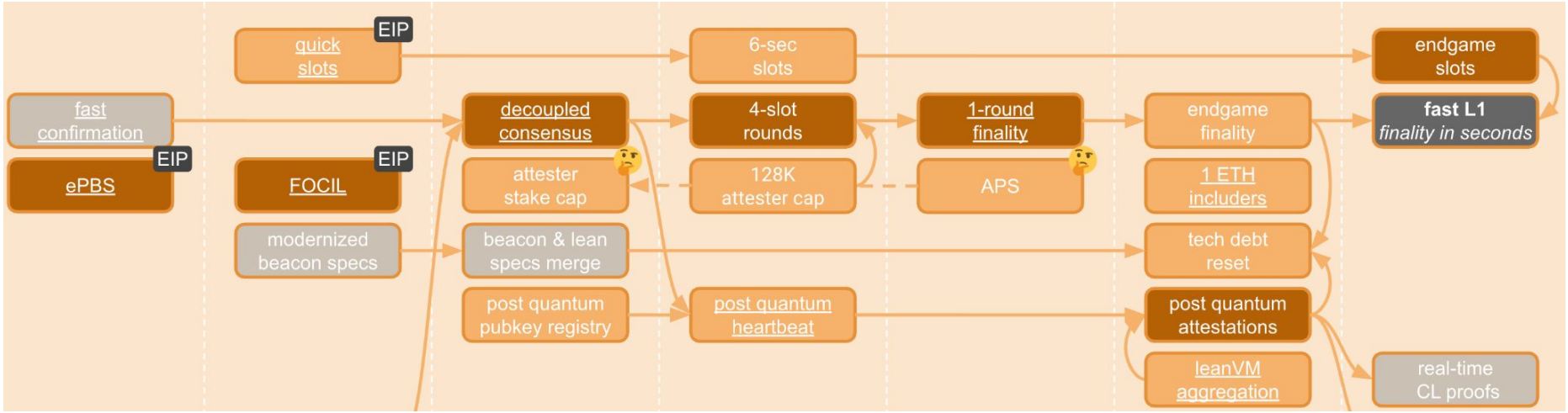
CROPS

Hardness

Stronger Finality

Consequences:

1. Longer periods of non-finality
2. Add economic safety on top?



strawmap.org

Thank you!

@yannvon

yann.vonlanthen@ethereum.org