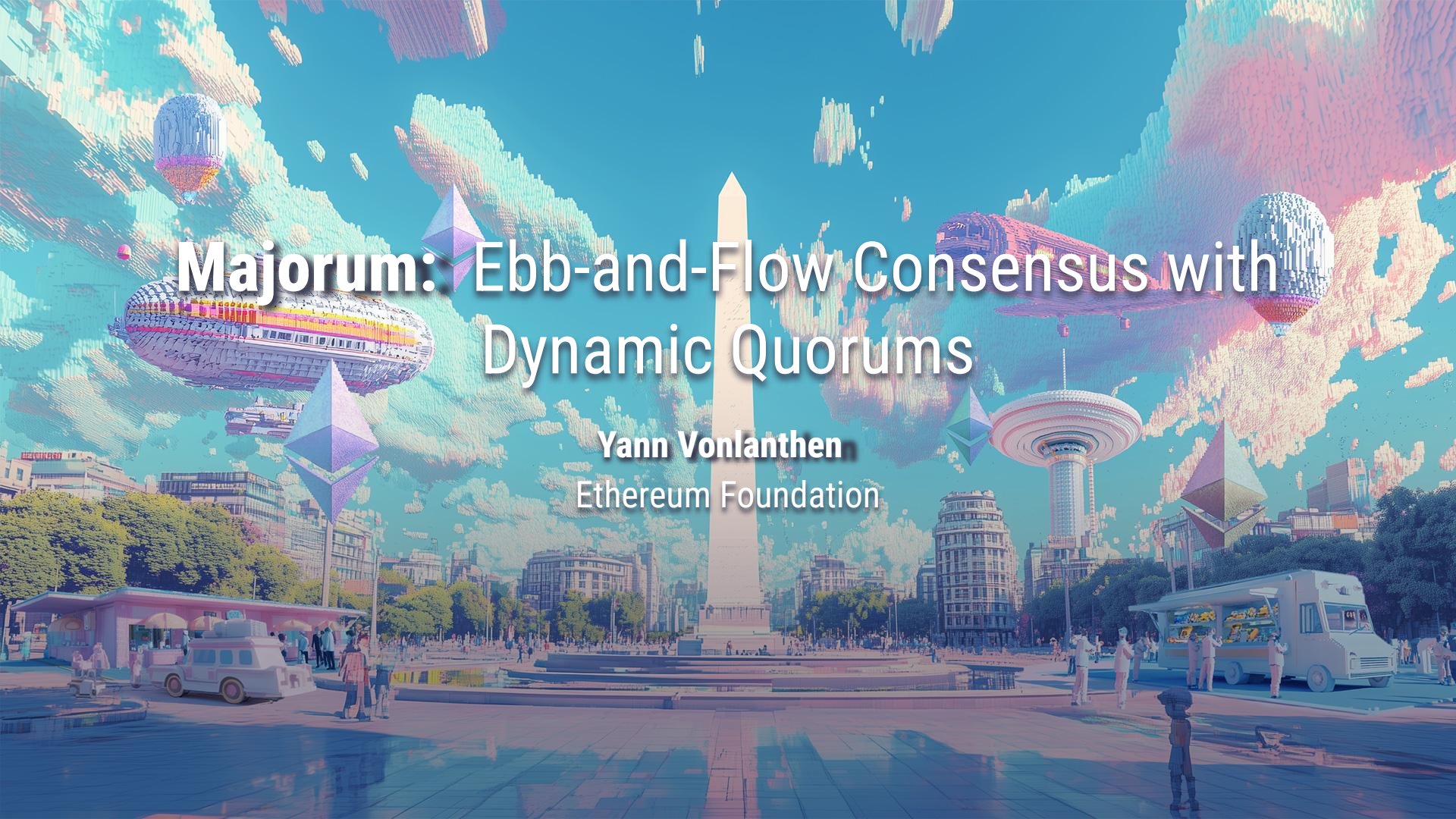


Majorum: Ebb-and-Flow Consensus with Dynamic Quorums

Yann Vonlanthen
Ethereum Foundation

Majorum: Ebb-and-Flow Consensus with Dynamic Quorums

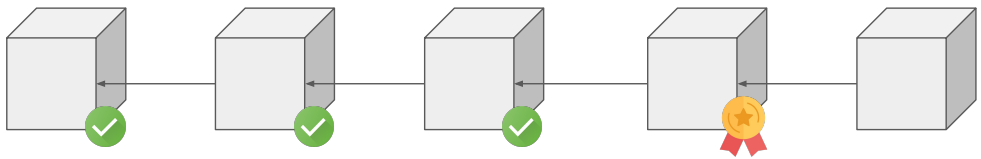
Yann Vonlanthen
Ethereum Foundation

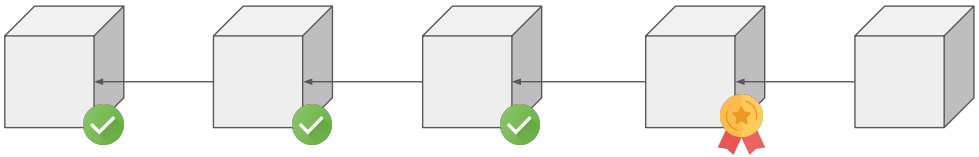
Francesco D'Amato
Ethereum Foundation

Roberto Saltini
Ethereum Foundation

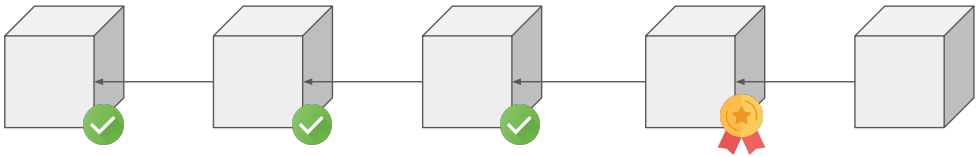
Thanh-Hai Tran
Independent

Luca Zanolini
Ethereum Foundation

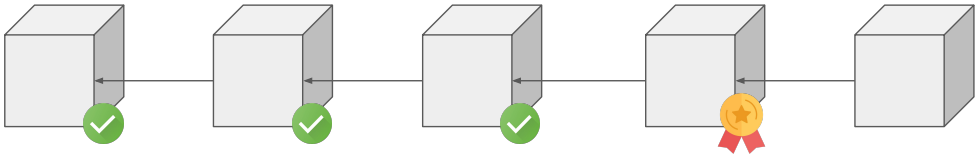




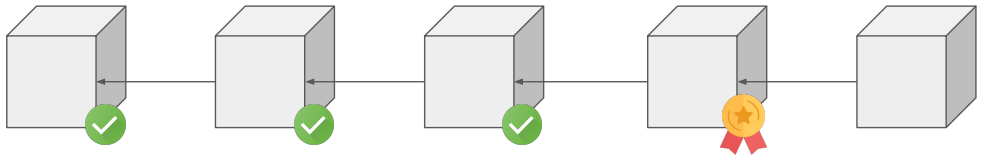

aws

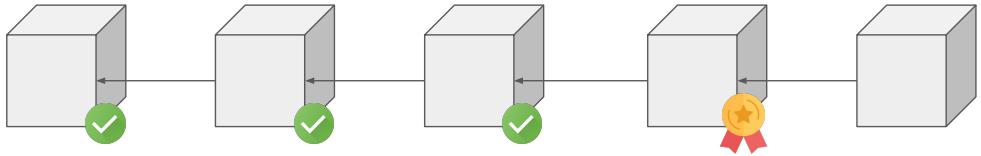


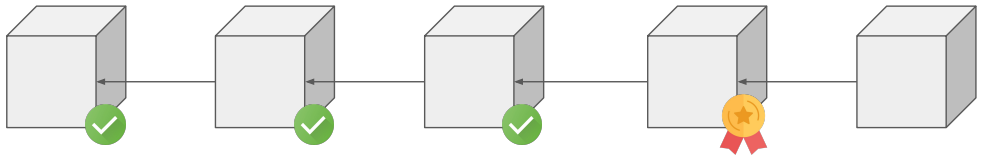

aws

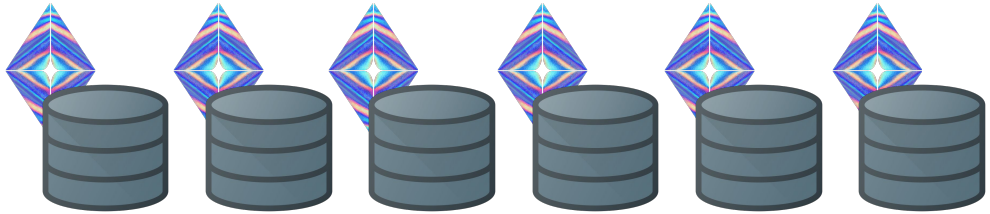


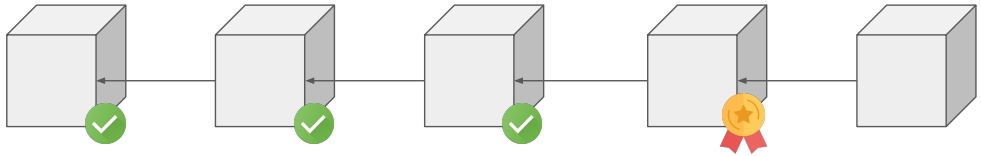

aws

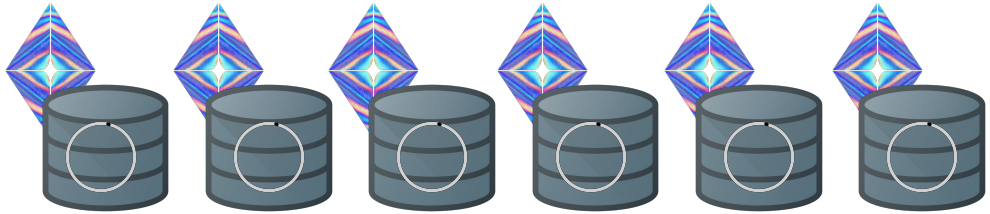


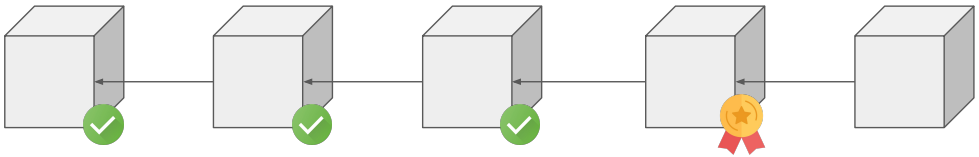


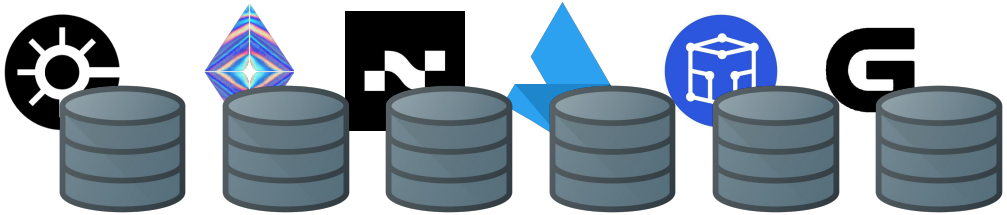


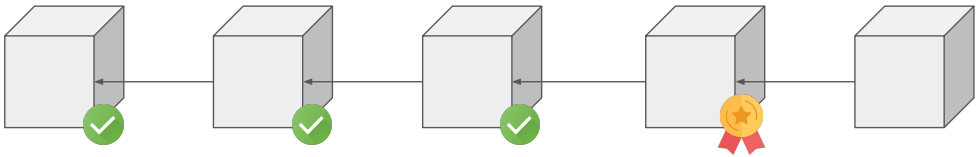


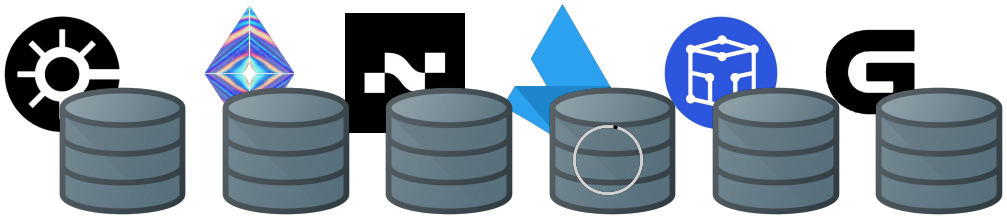


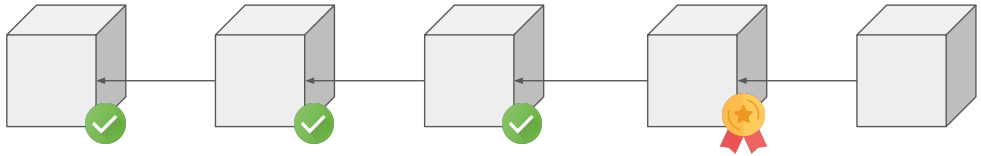


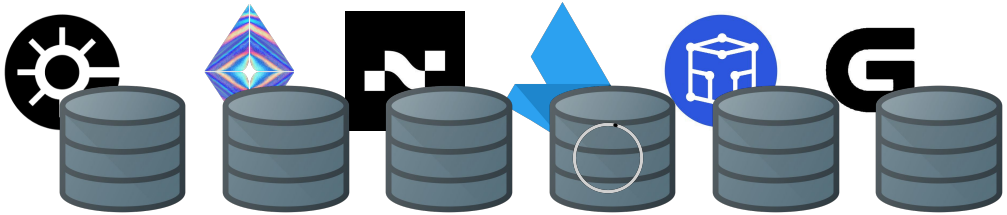










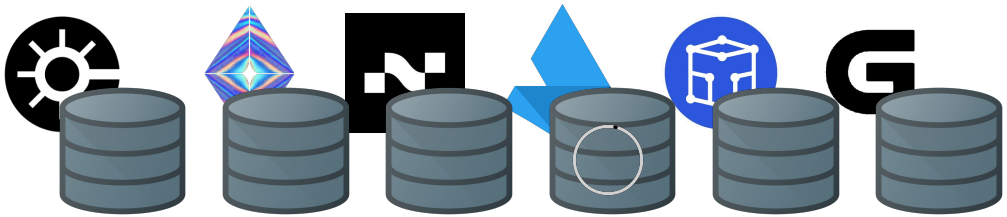


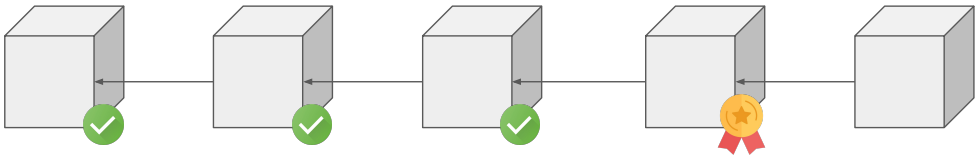
Validator Activity in Ethereum Drops 25% Following Fusaka Deployment

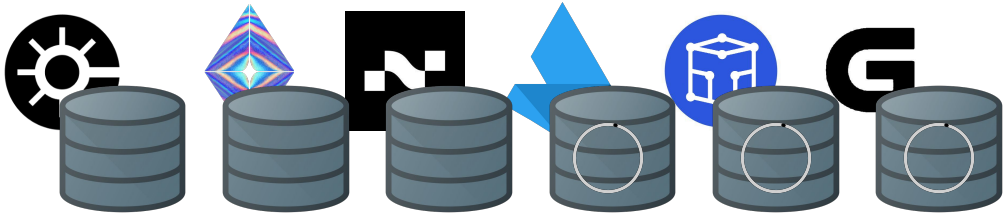
Validator activity in Ethereum drops 25% following Fusaka deployment.

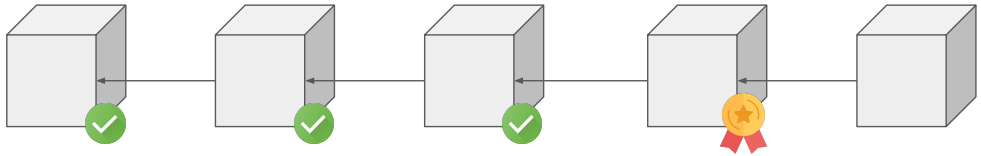
05.12.2025 • ForkLog

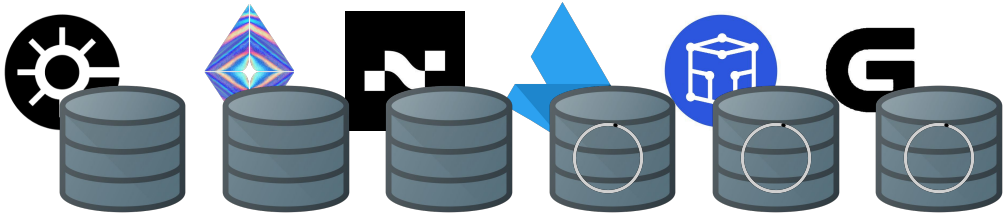
Shortly after the deployment of Fusaka, a malfunction occurred in the popular consensus client Prysm, disabling a portion of Ethereum validators.



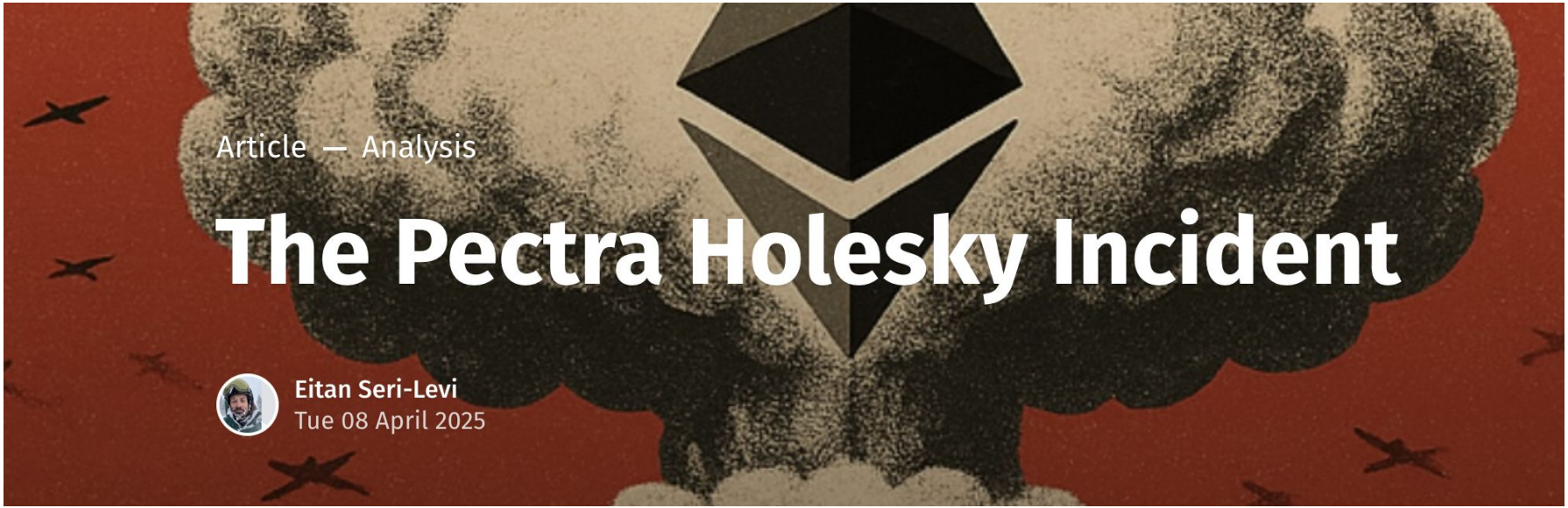


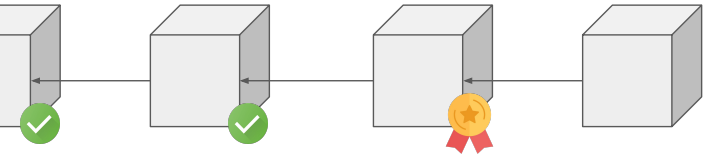


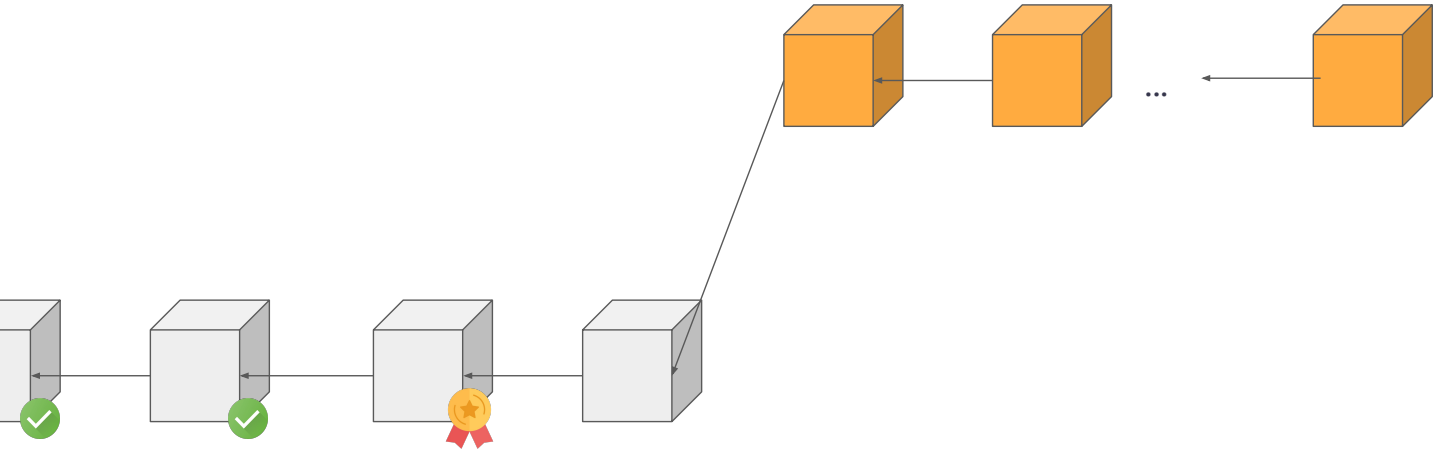


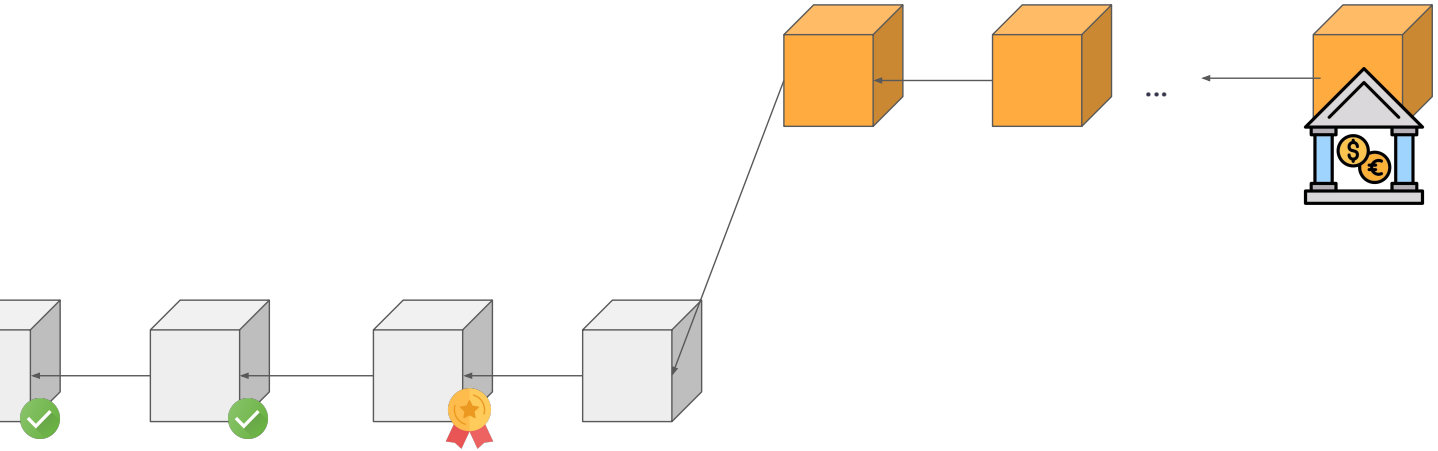


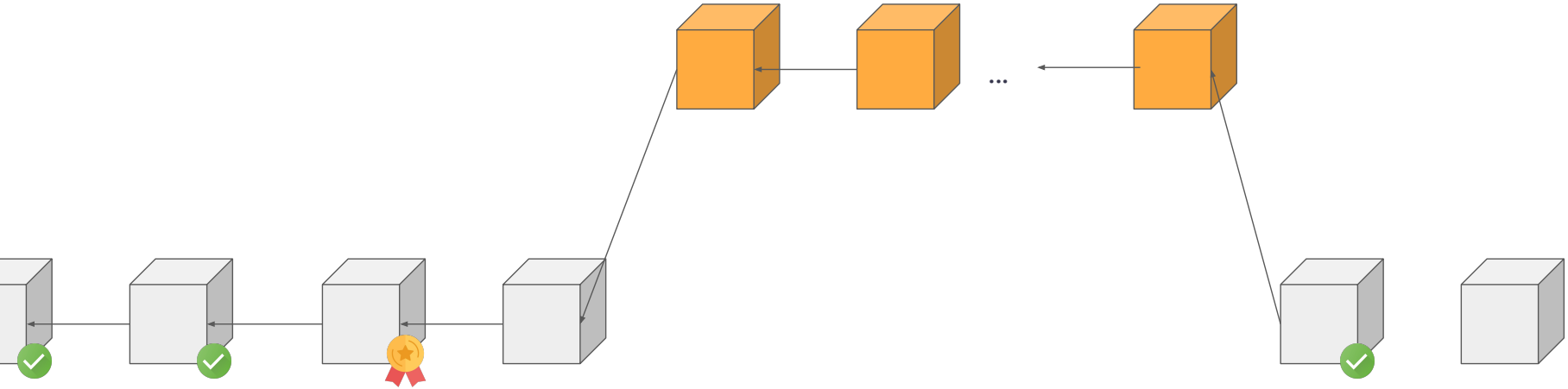
https://blog.sigmaprime.io/pectra-holesky-incident.html

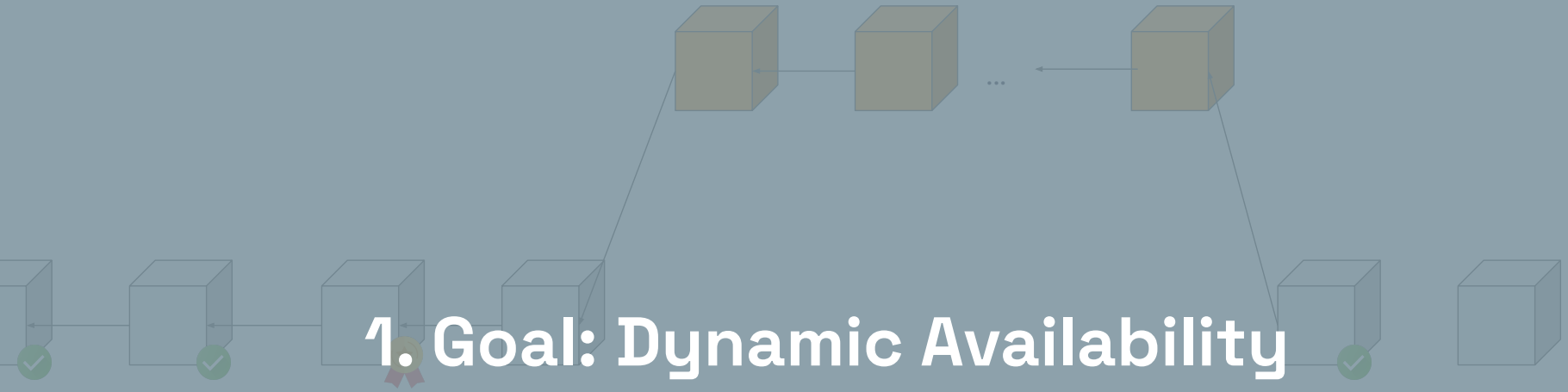






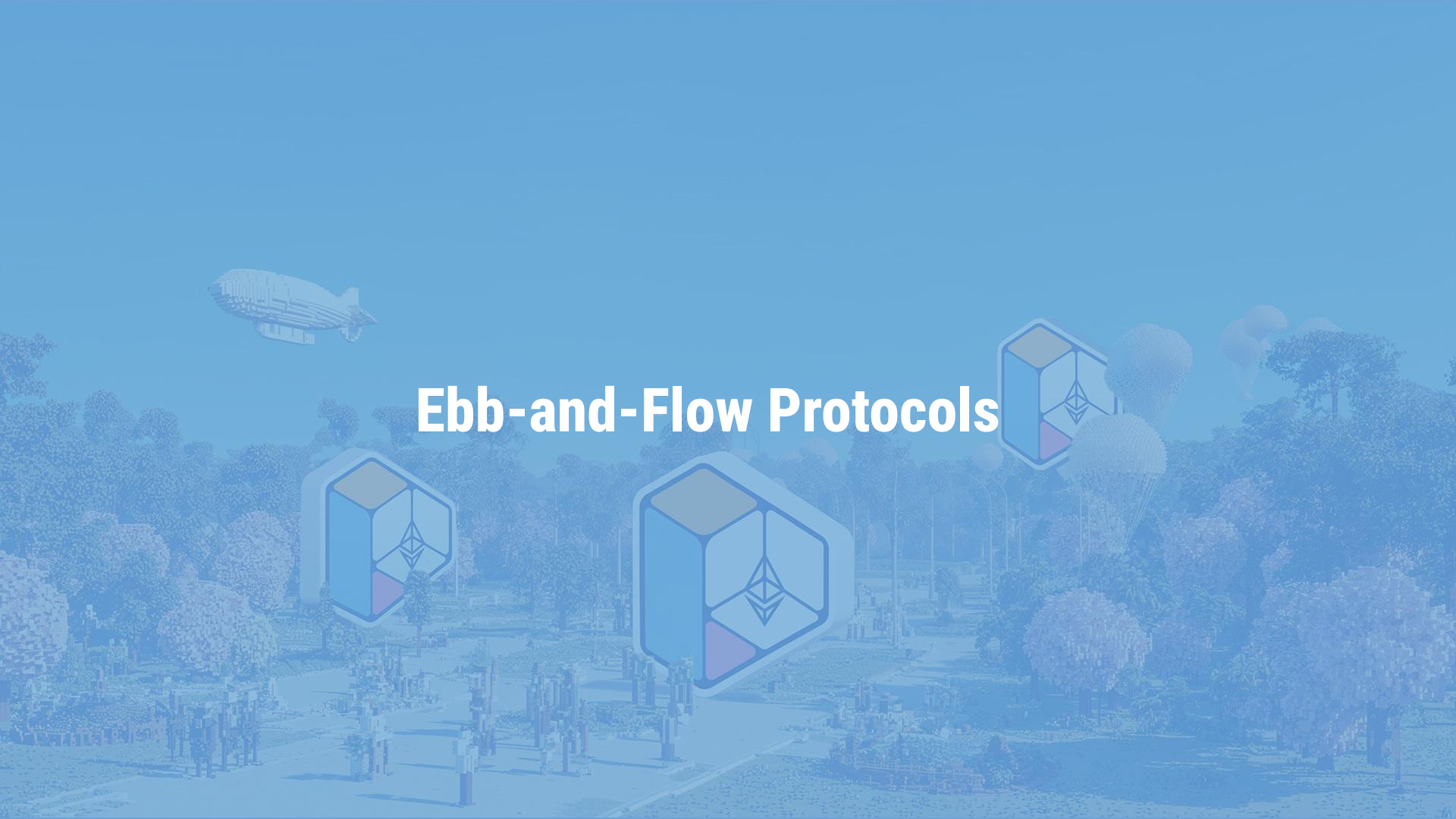






1. Goal: Dynamic Availability

Ebb-and-Flow Protocols



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

Joachim Neu
jneu@stanford.edu

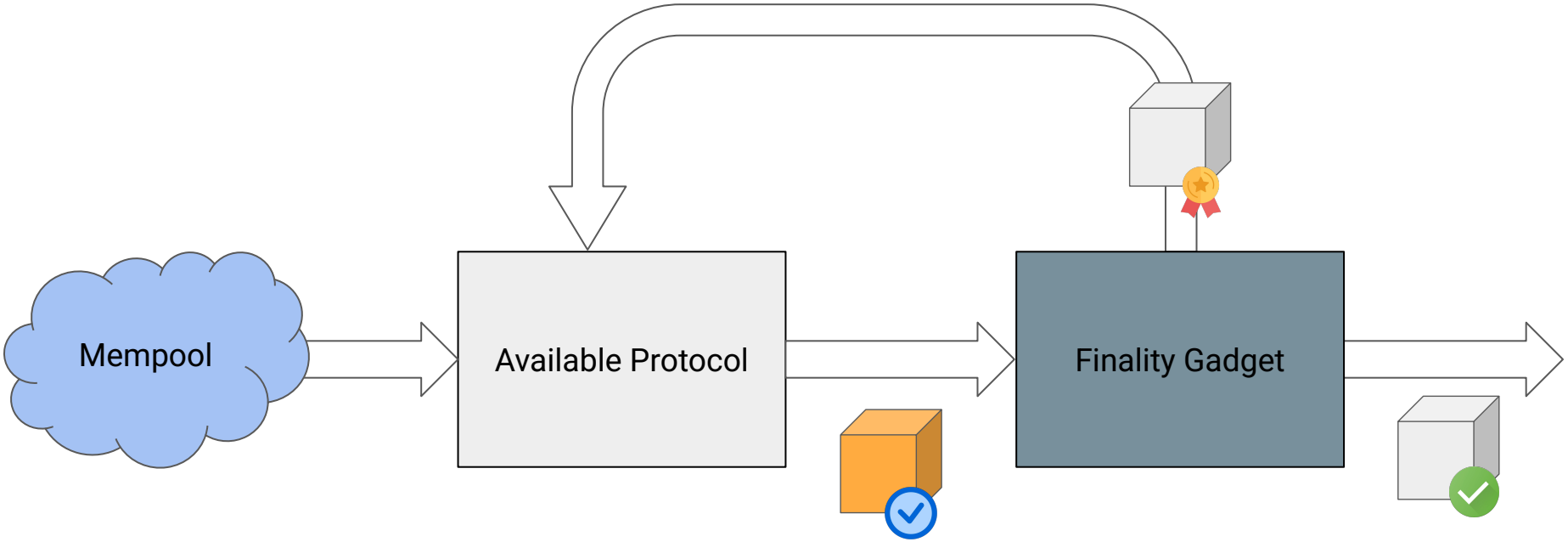
Ertem Nusret Tas
nusret@stanford.edu

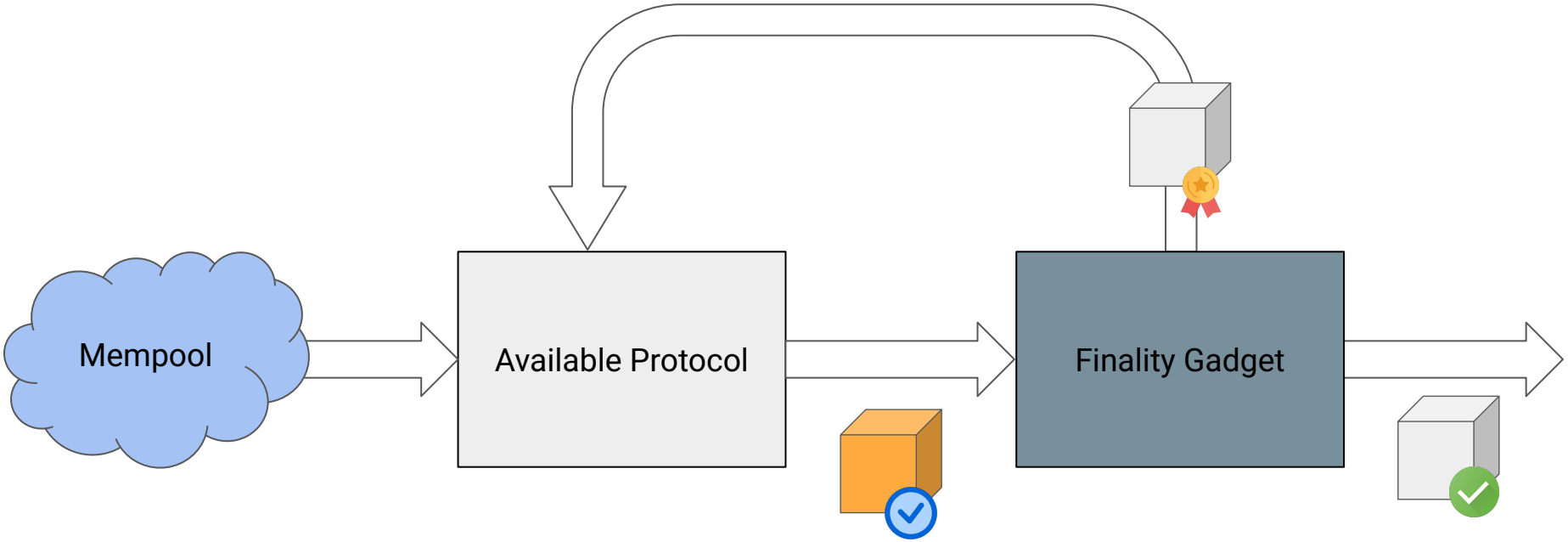
David Tse
dntse@stanford.edu

***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



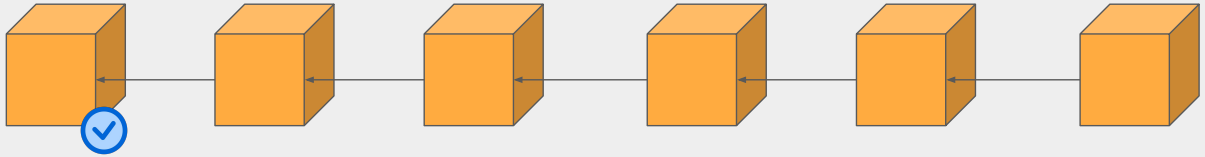
Prefix Property: The available chain is a prefix of the final chain.

Available Chain

$f < n / 2$

Δ (Synchronous)

1 honest awake

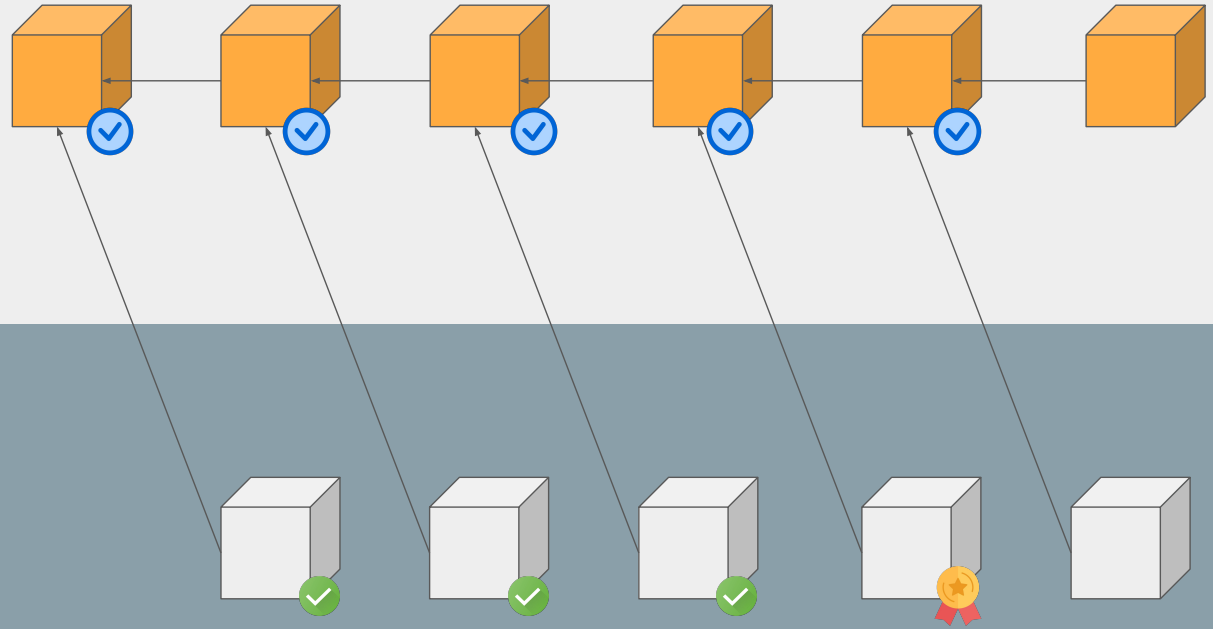


Available Chain

f < n / 2

Δ (Synchronous)

1 honest awake



Finality Chain

f < n / 3

Δ after **GST** (part. sync)

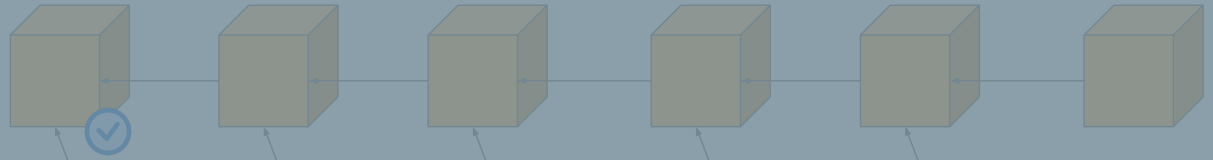
All online after **GAT**

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

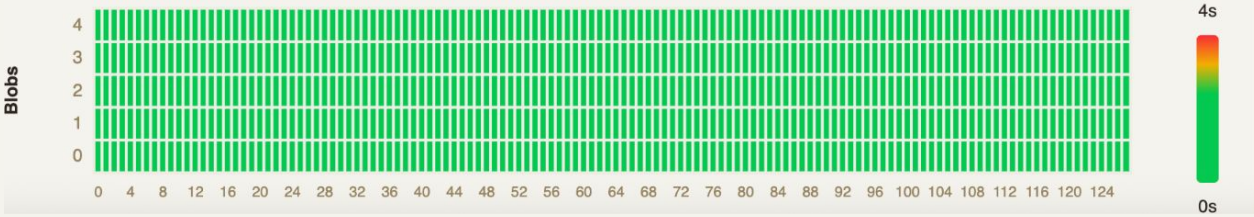
Block | Attestations | Aggregations

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

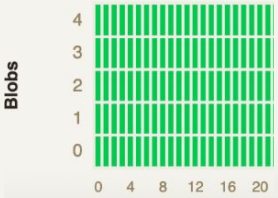
Block | Attestations | Aggregations

0s 4s 8s 12s

6.0s Attest 3 validators

Data from nodes contributing

DATA COLUMN AVAILABILITY



ATTESTATION ARRIVALS

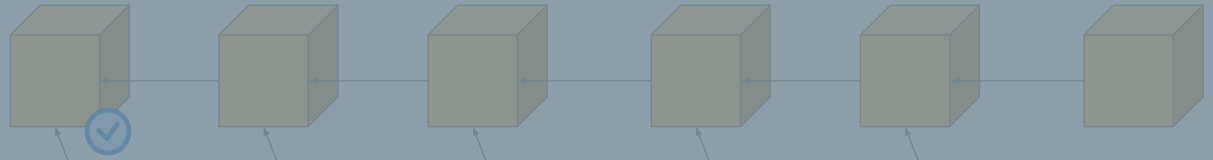
lab.ethpandaops.io

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

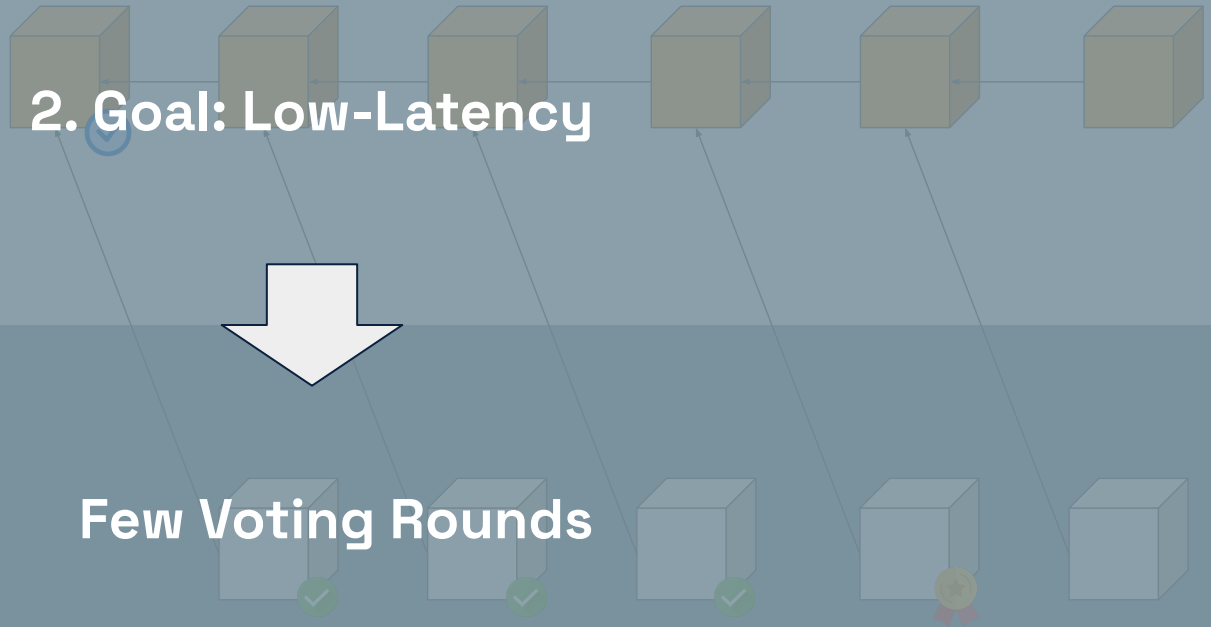
Finality Chain

f < n / 3

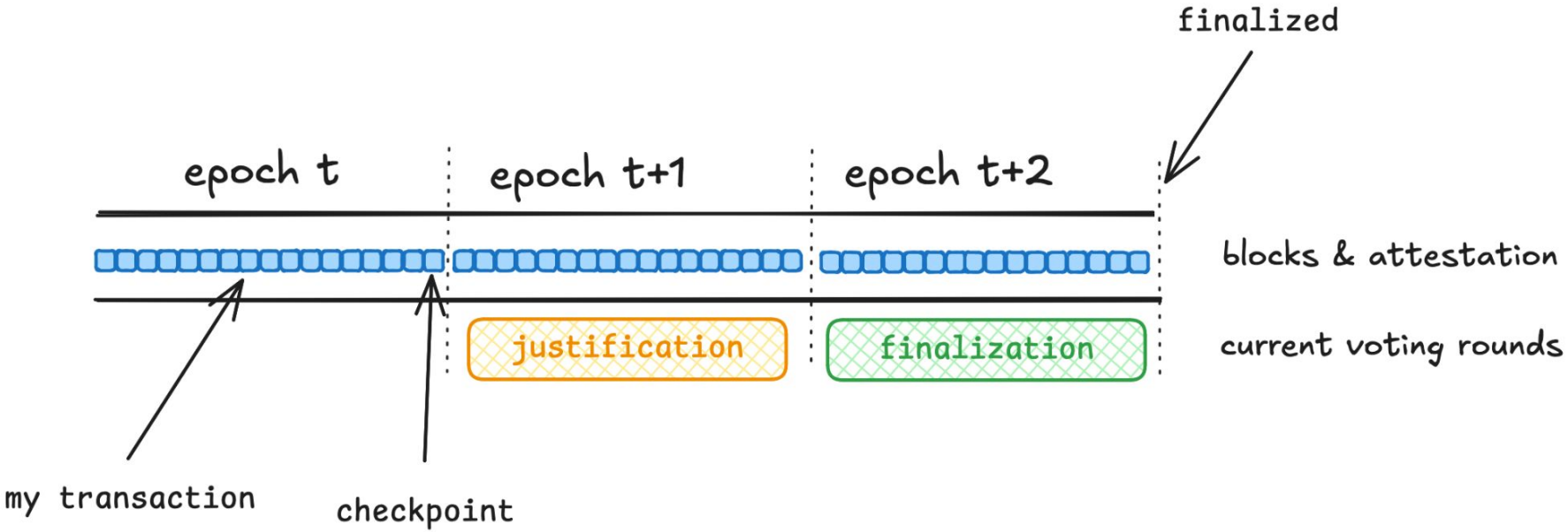
Δ after **GST** (part. sync)

All online after **GAT**

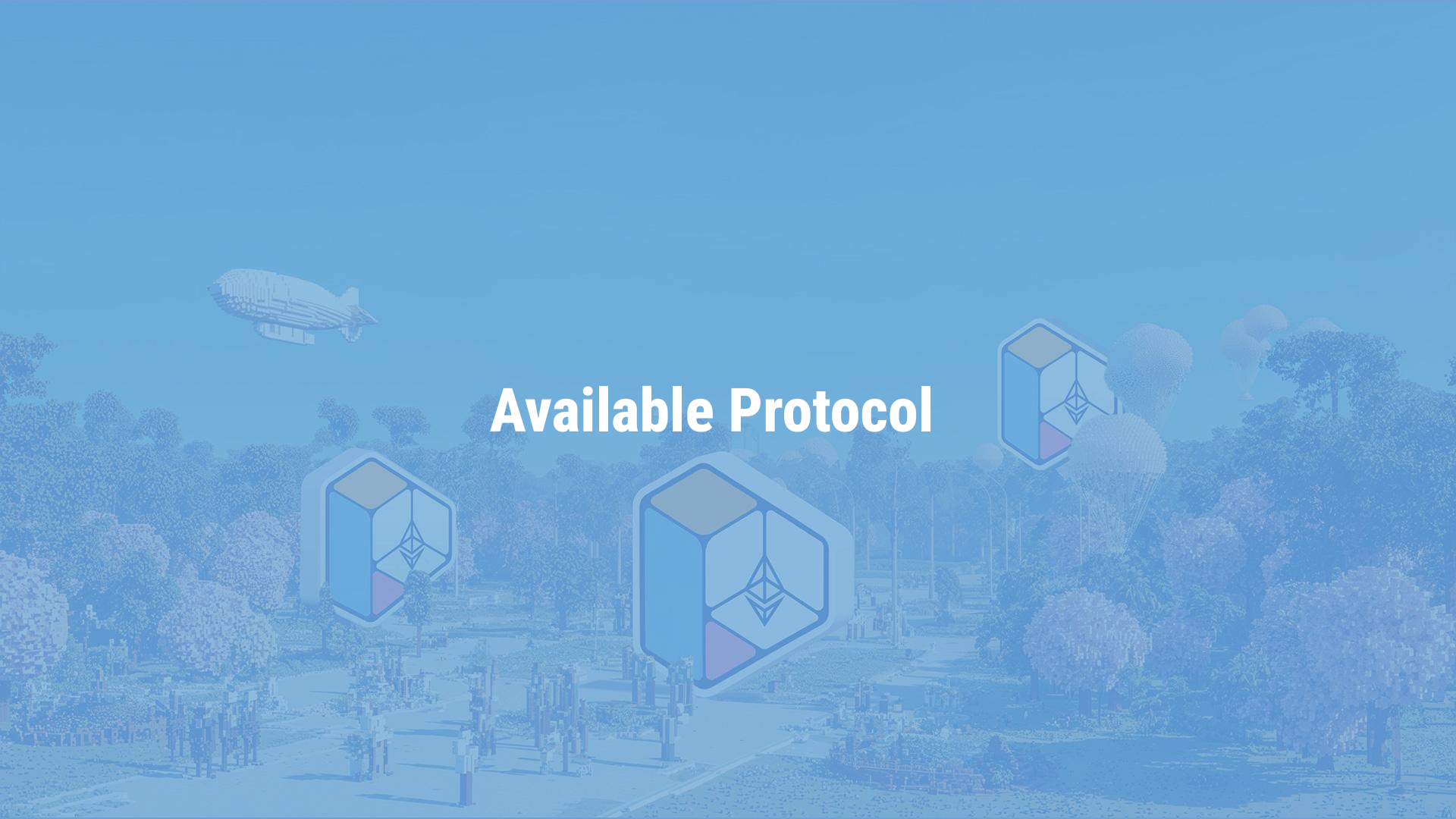
2. Goal: Low-Latency



Few Voting Rounds


finalized
epoch t
epoch t+1
epoch t+2
blocks & attestation
justification
finalization
current voting rounds
my transaction
checkpoint

Available Protocol

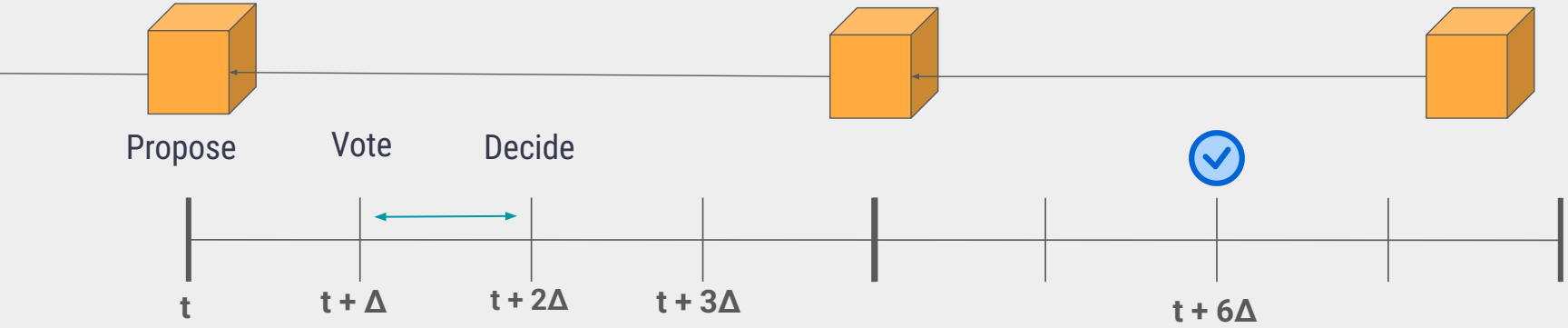


TOB-SVD

$f < n / 2$

Δ (Synchronous)

1 honest awake



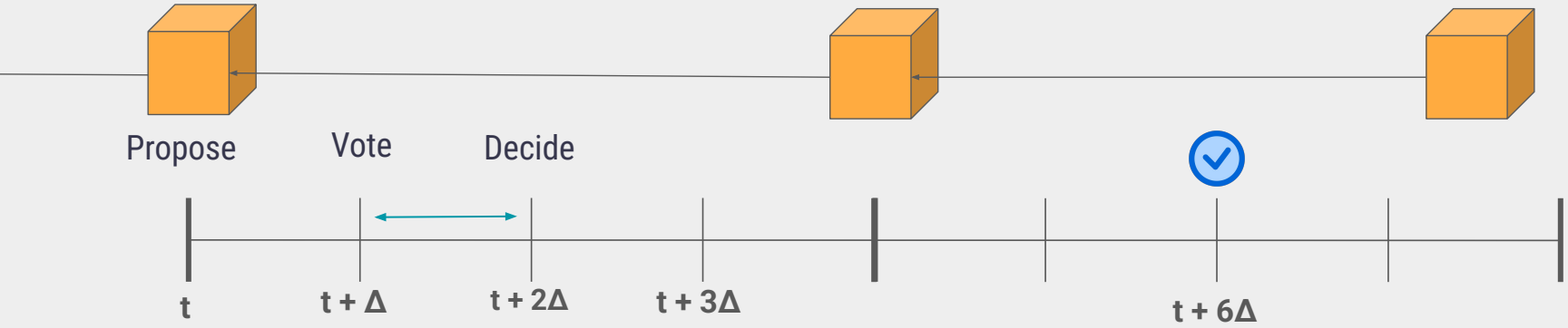
TOB-SVD

Single vote decision!

$f < n / 2$

Δ (Synchronous)

1 honest awake

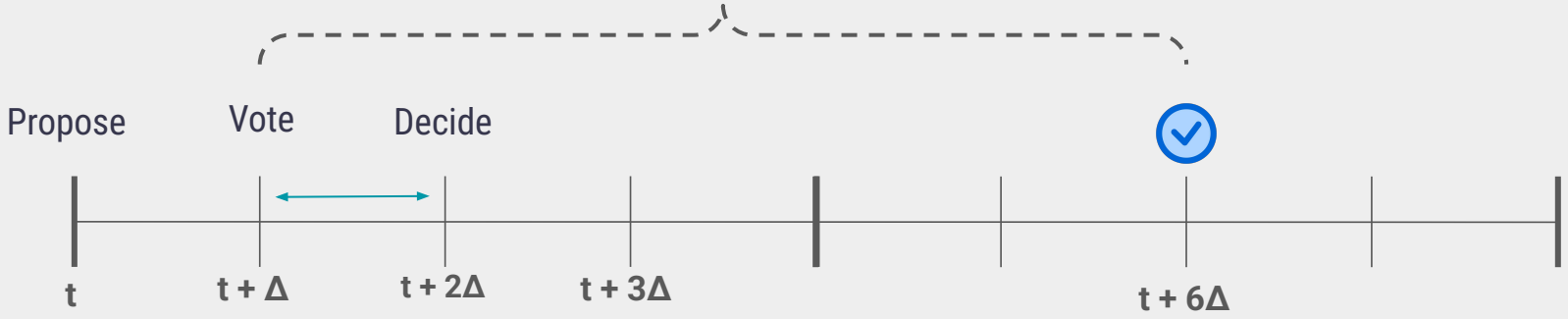


TOB-SVD

f < n / 2

Δ (Synchronous)

1 honest awake

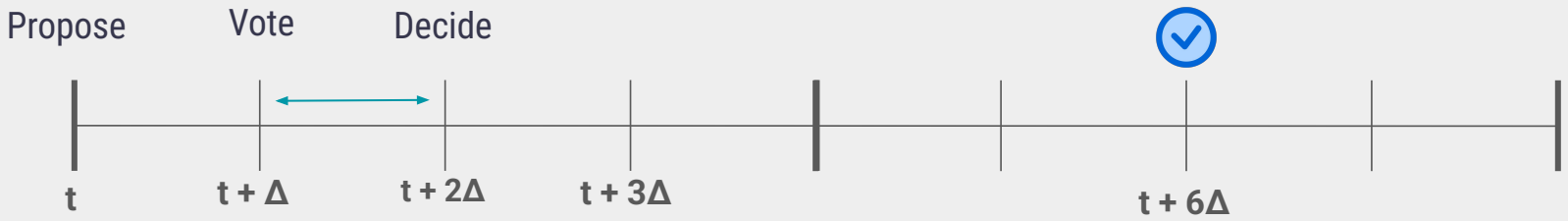


TOB-SVD: Dynamic Quorums

f < n / 2

Δ (Synchronous)

1 honest awake

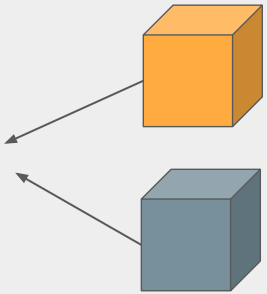


TOB-SVD: Dynamic Quorums

$f < n / 2$

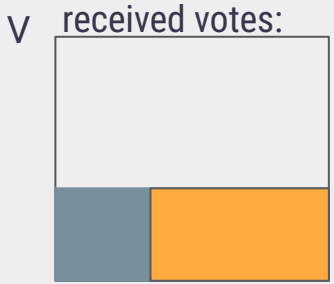
Δ (Synchronous)

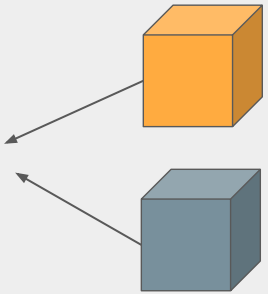
1 honest awake



Propose Vote Decide

TOB-SVD

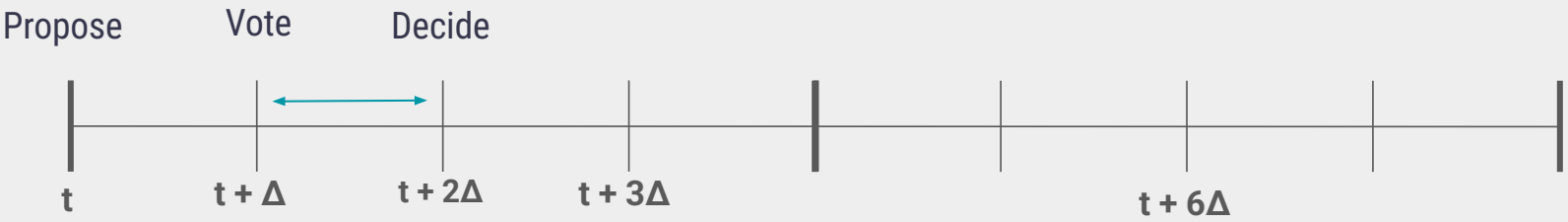




f < n / 2

Δ (Synchronous)

1 honest awake



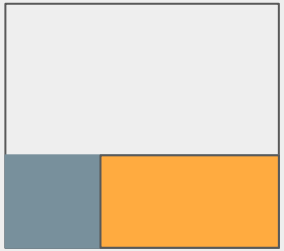
TOB-SVD

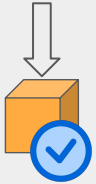
$f < n / 2$

Δ (Synchronous)

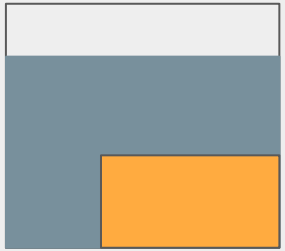
1 honest awake

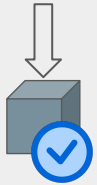
V received votes:

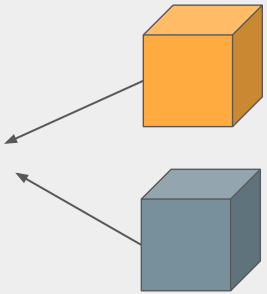




U received votes:







Propose Vote Decide

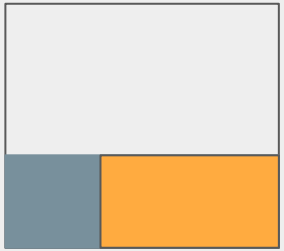
TOB-SVD

f < n / 2

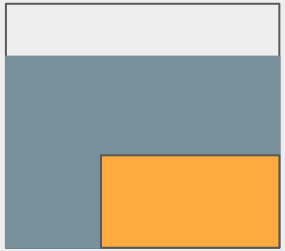
Δ (Synchronous)

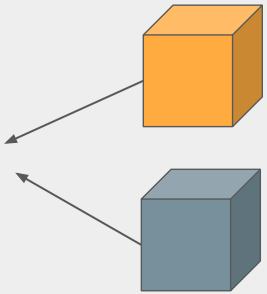
1 honest awake

V received votes:



U received votes:





Solution: Time shifted quorums!

Propose Vote Decide

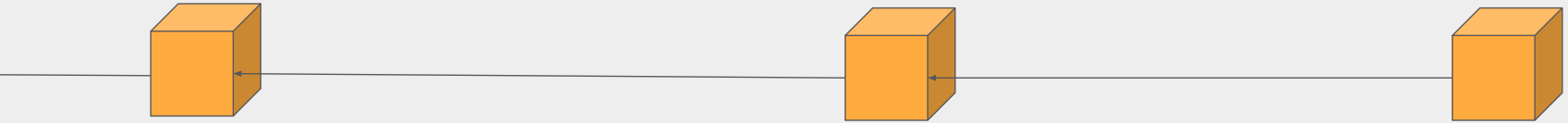
t t + Δ t + 2Δ t + 3Δ t + 6Δ

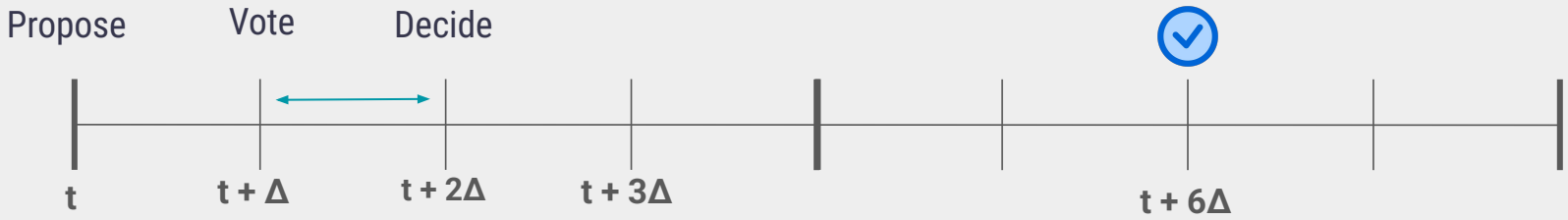
TOB-SVD

$f < n / 2$

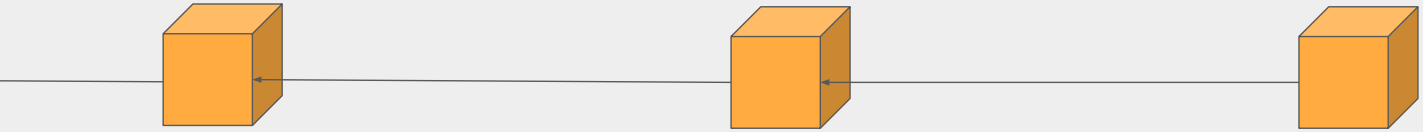
Δ (Synchronous)

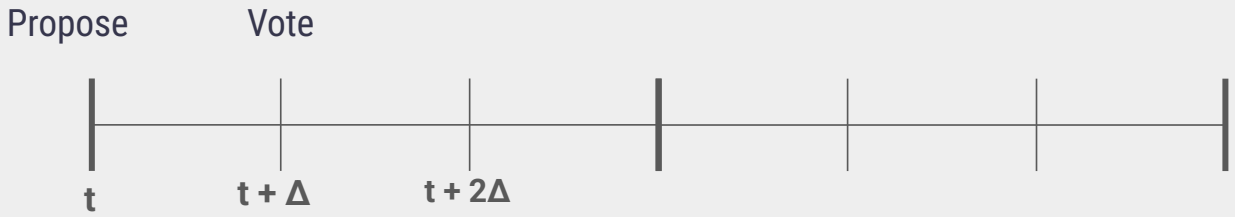
1 honest awake





Step 1: Probabilistic TOB-SVD





$f < n / 2$

Δ (Synchronous)

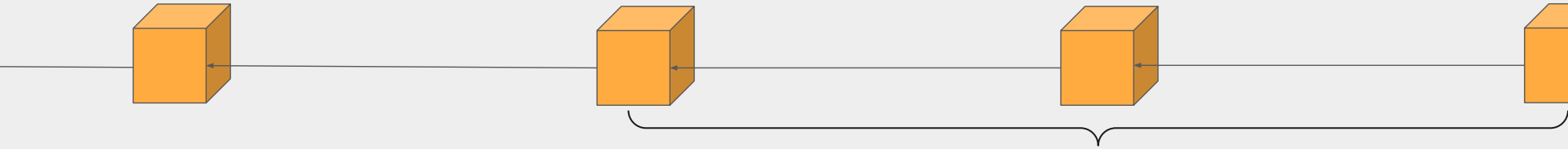
1 honest awake

Step 1: Probabilistic TOB-SVD

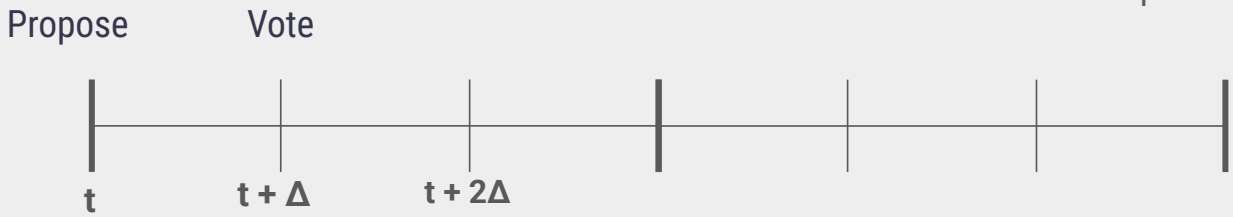
f < n / 2

Δ (Synchronous)

1 honest awake



k-deep ⇒ *confirmed*

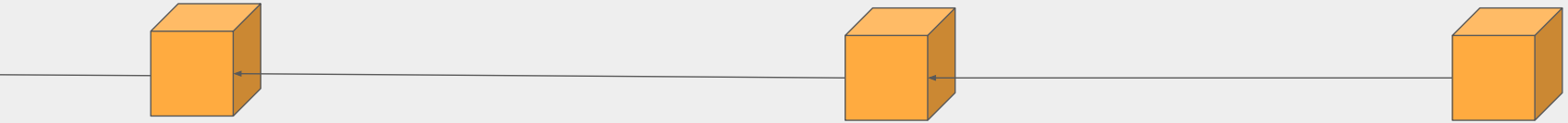


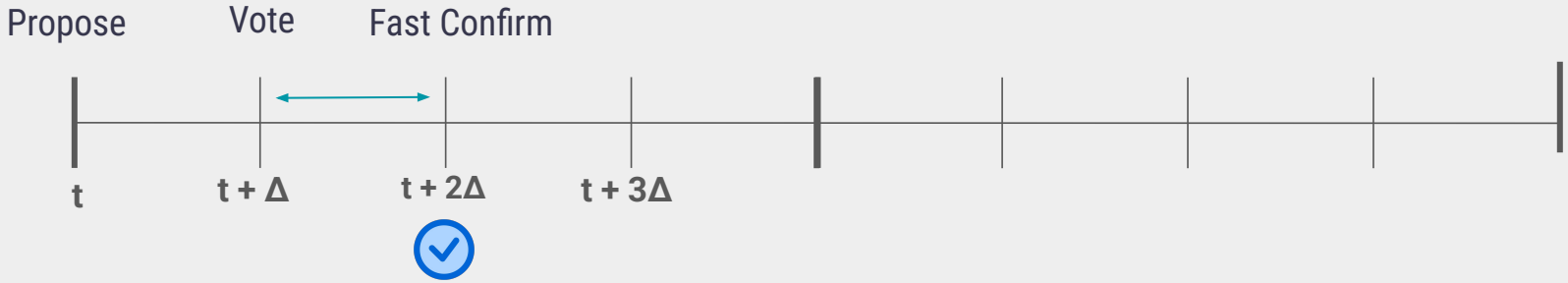
Step 2: Fast Confirmation

f < n / 2

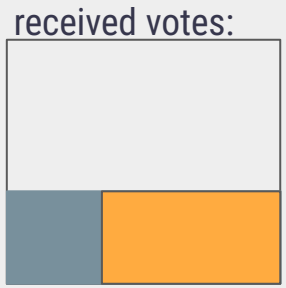
Δ (Synchronous)

1 honest awake





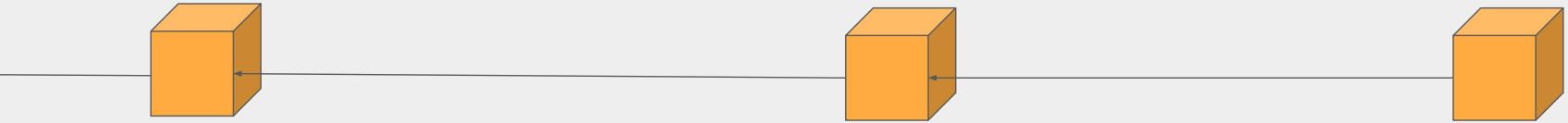
Step 2: Fast Confirmation



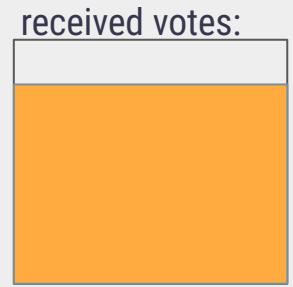
f < n / 2

Δ (Synchronous)

1 honest awake



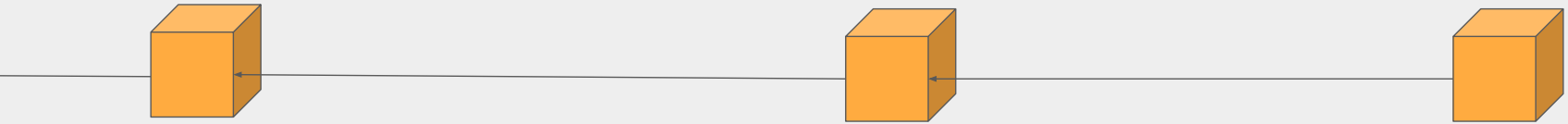
Step 2: Fast Confirmation

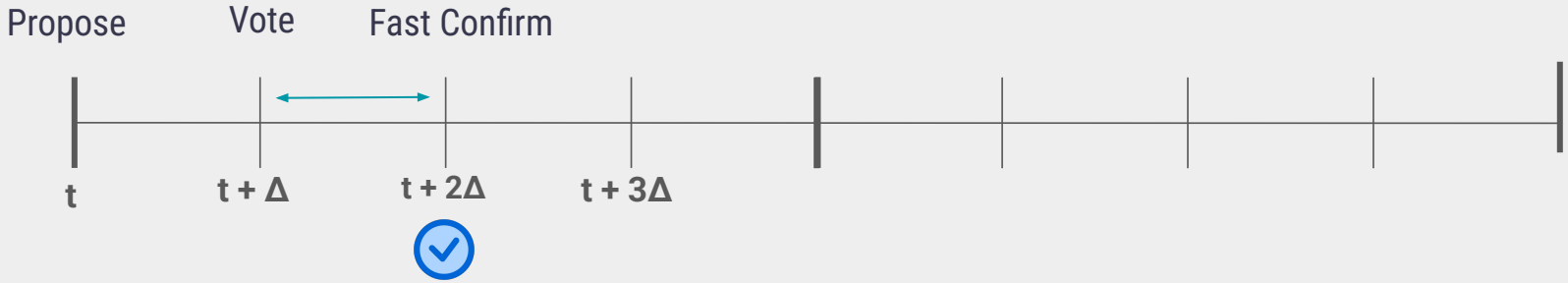


f < n / 2

Δ (Synchronous)

1 honest awake



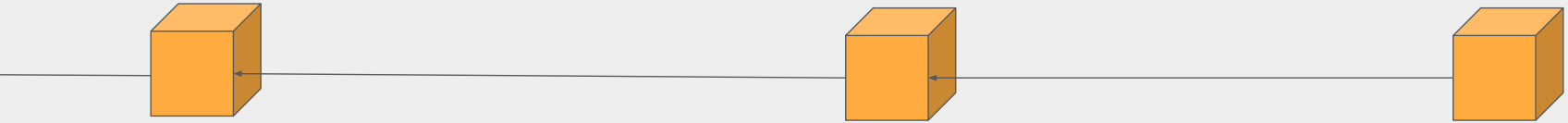


Probabilistic TOB-SVD with Fast Confirmation

$f < n / 2$

Δ (Synchronous)

1 honest awake

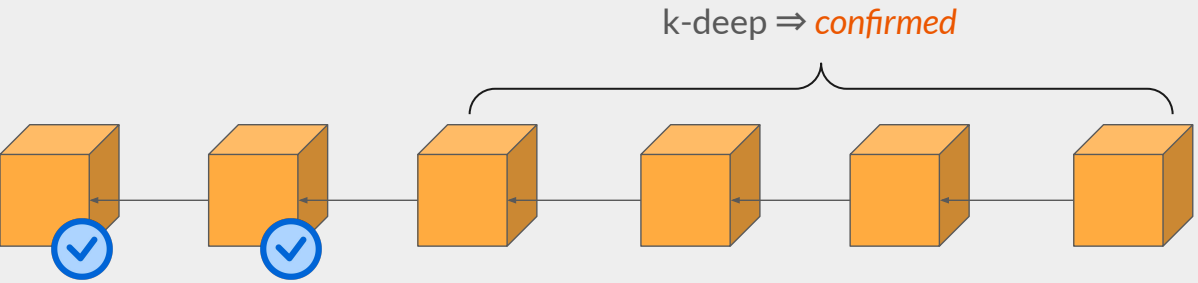


TOB-SVD: Dynamic Quorums

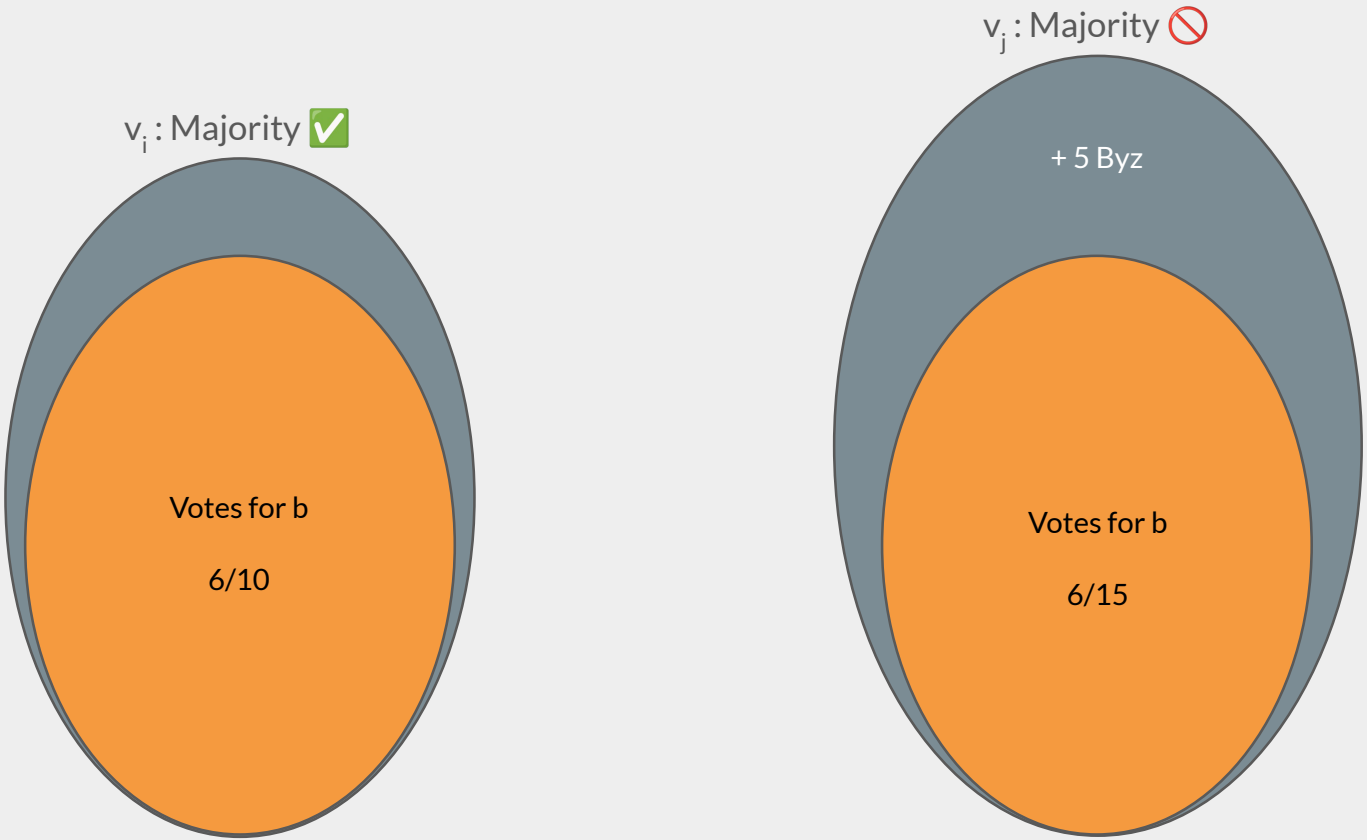
$f < n / 2$

Δ (Synchronous)

1 honest awake

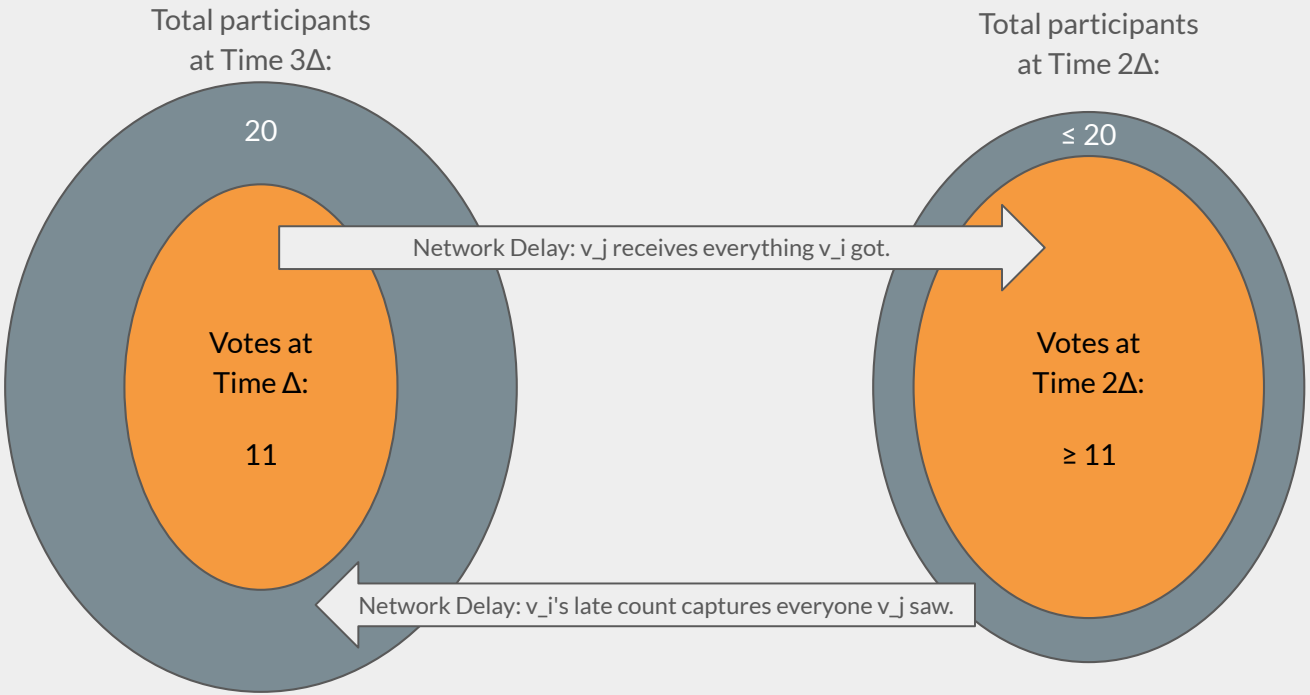


The Problem: Dynamic Quorum Non-Transferability



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 ✅

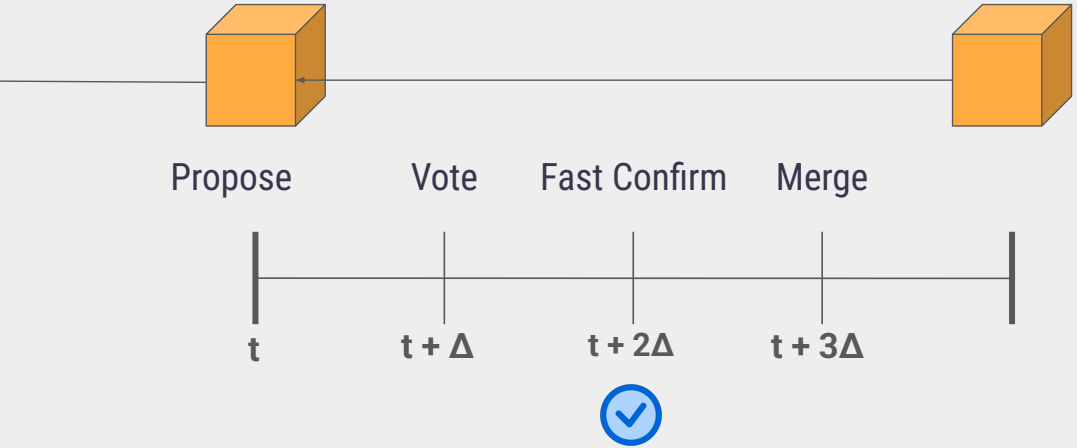
11 > 20/2

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).

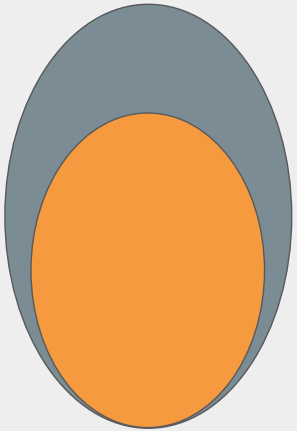
Probabilistic TOB-SVD

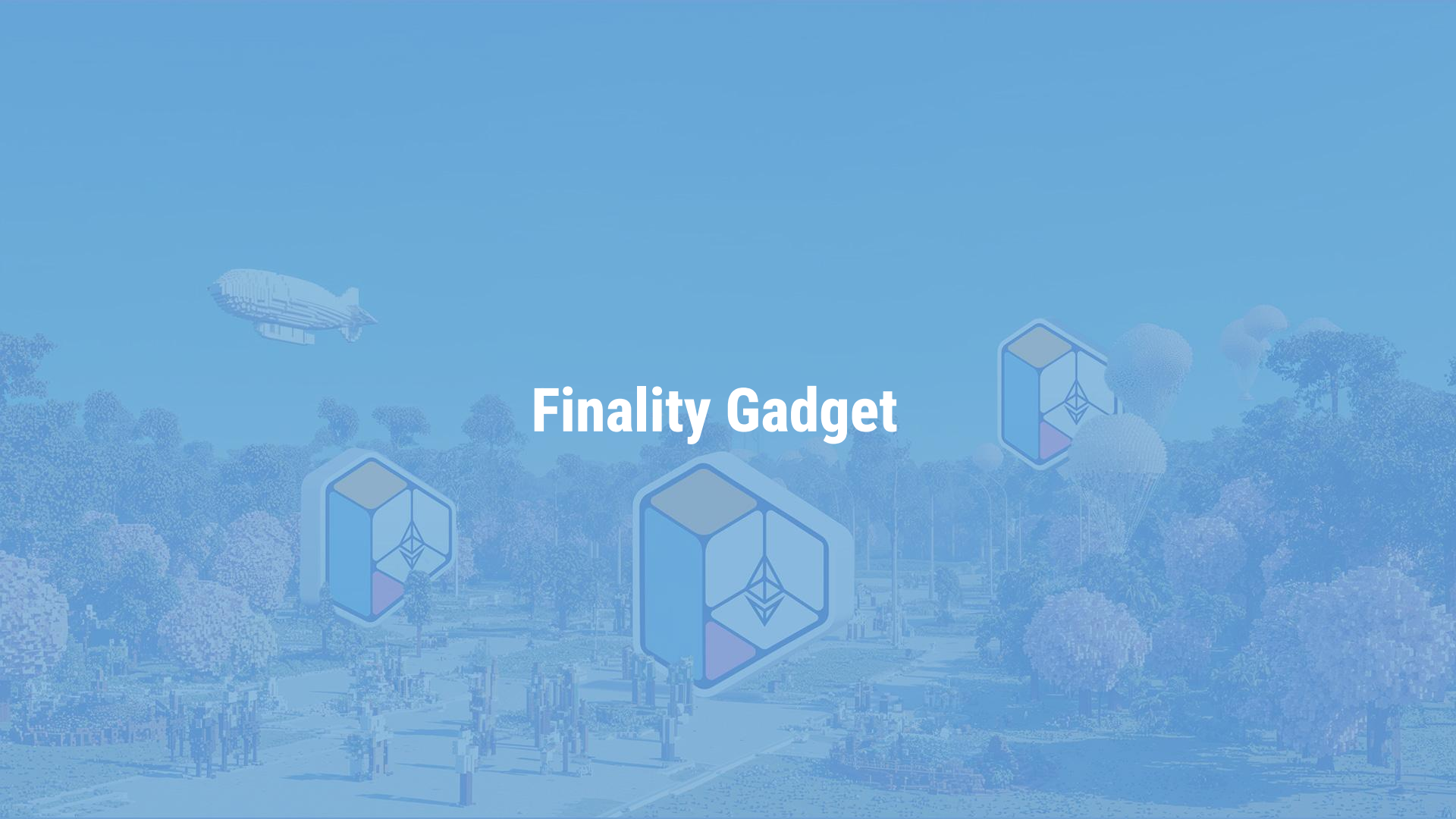


f < n / 2

Δ (Synchronous)

1 honest awake



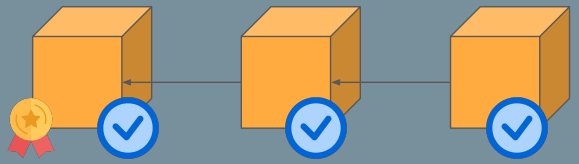

Finality Gadget

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



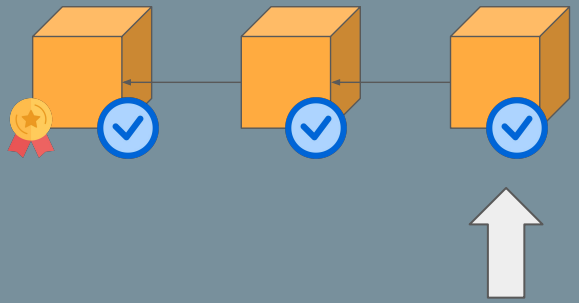
Pick voting target:
Confirmed head of available chain (respecting justification)

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



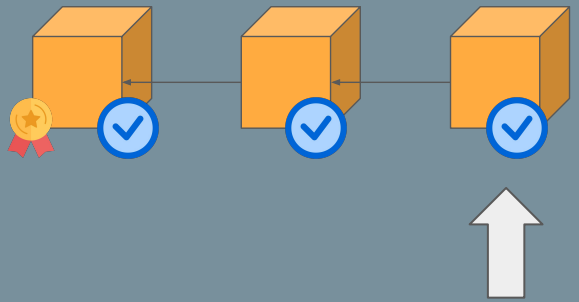
Pick voting target:
Confirmed head of available chain (respecting justification)

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



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

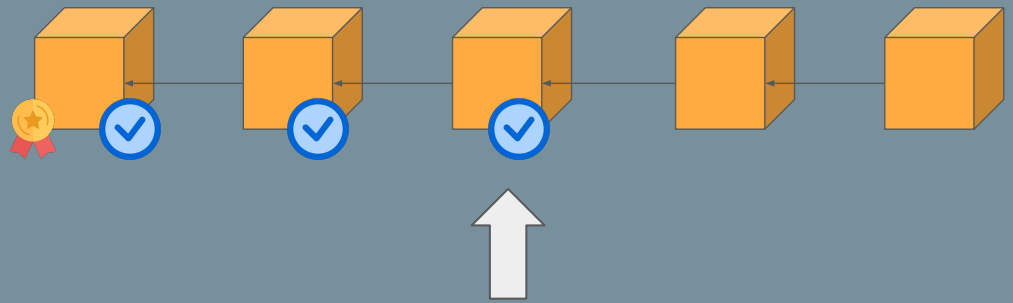
⇒ fast confirmed

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**



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

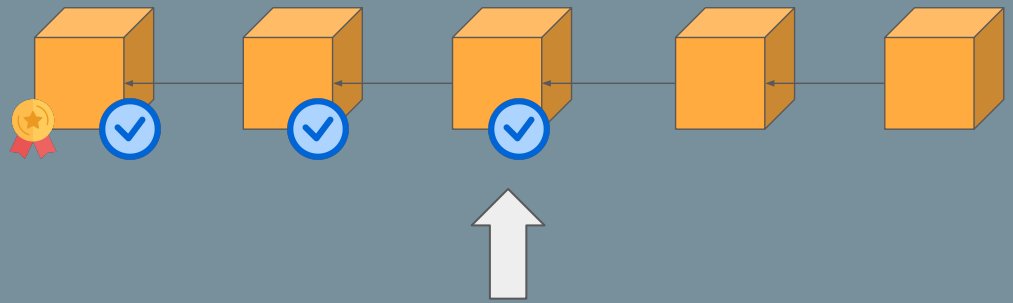
⇒ fast confirmed
⇒ k - deep

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



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

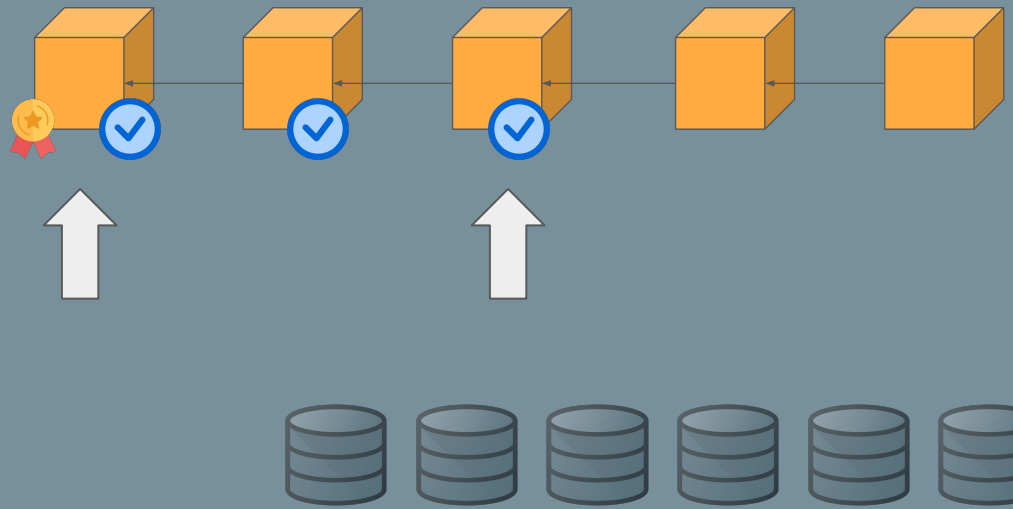
⇒ fast confirmed
⇒ k - deep
(under sleepiness, or asynchrony)

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)

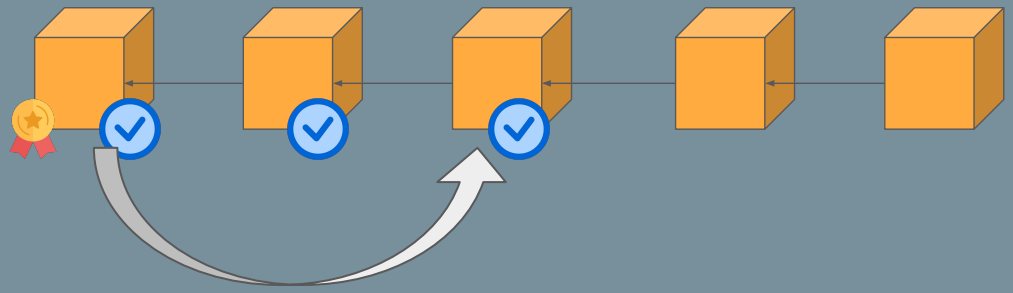
Pick voting source:
Latest justified block

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)

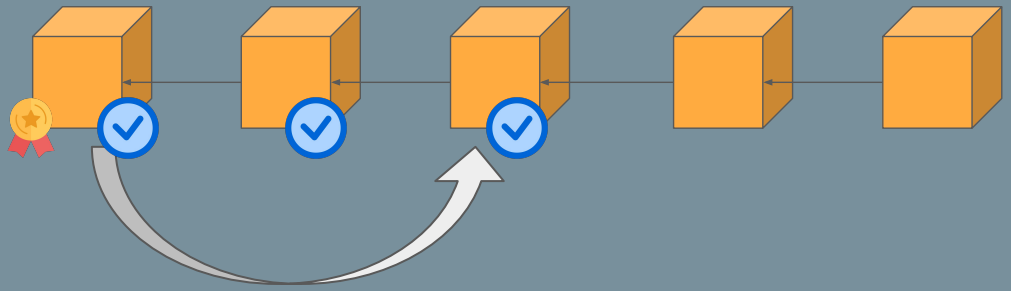
Pick voting source:
Latest justified block

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



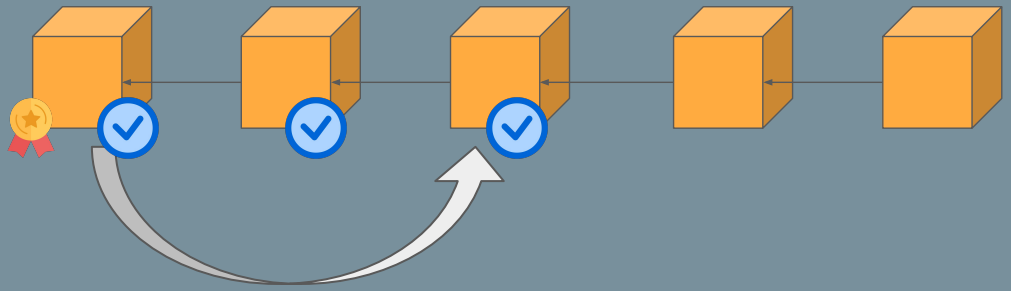
Justify block B once:
≥ 67% votes with target B

Finality Gadget

$f < n / 3$

Δ after **GST** (part. sync)

All online after **GAT**



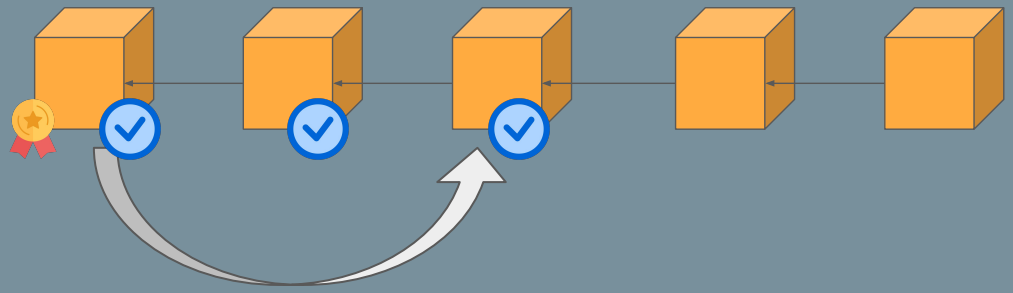
Justify block B once:
≥ 67% votes with target B

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

All online after **GAT**



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

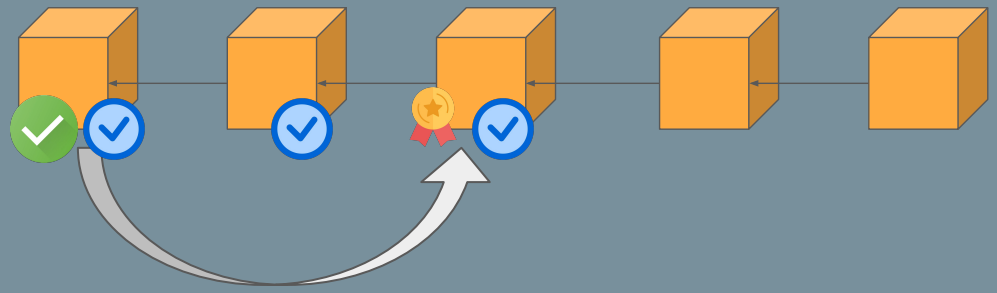
Finalize block B once:
≥ 67% votes with source B and target B' > B

Finality Gadget

f < n / 3

Δ after **GST** (part. sync)

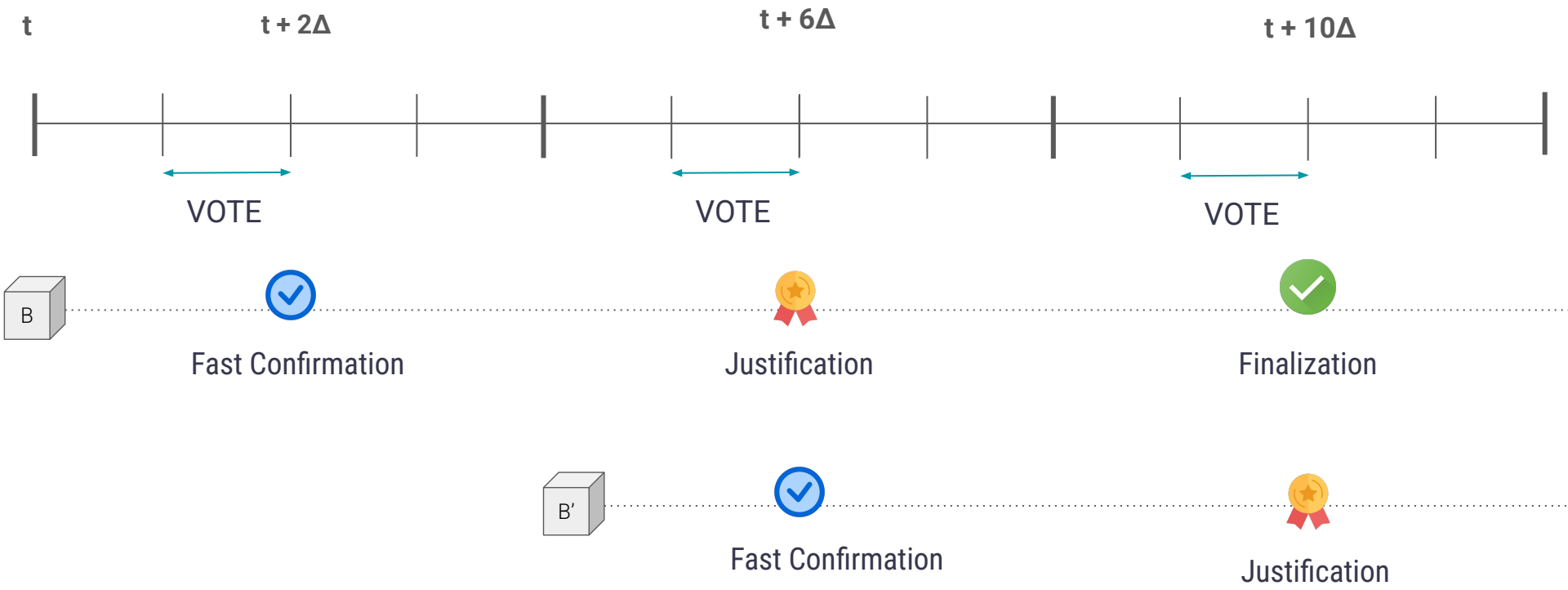
All online after **GAT**

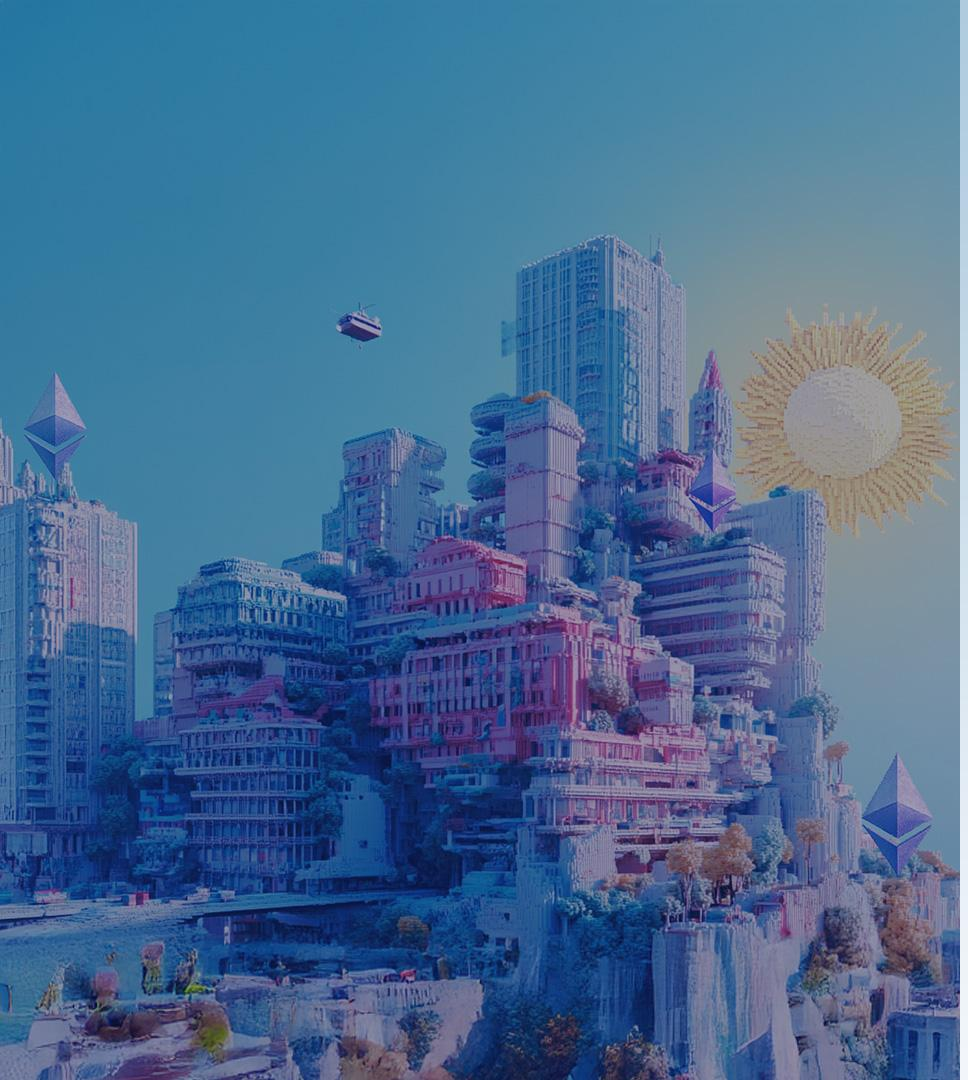


Justify block B once:
≥ 67% votes with target B

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

Thank you!

@yannvon

yann.vonlanthen@ethereum.org