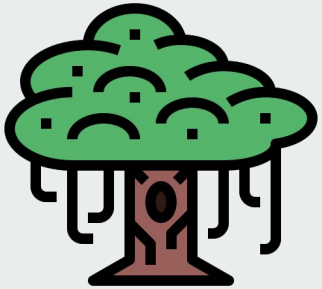
Banyan: Fast Rotating Leader BFT

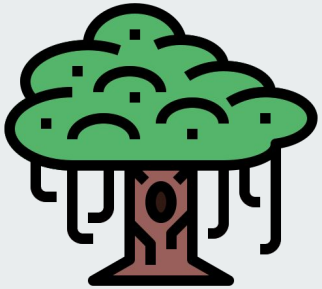
Yann Vonlanthen, Jakub Sliwinski, Massimo Albarello, and Roger Wattenhofer



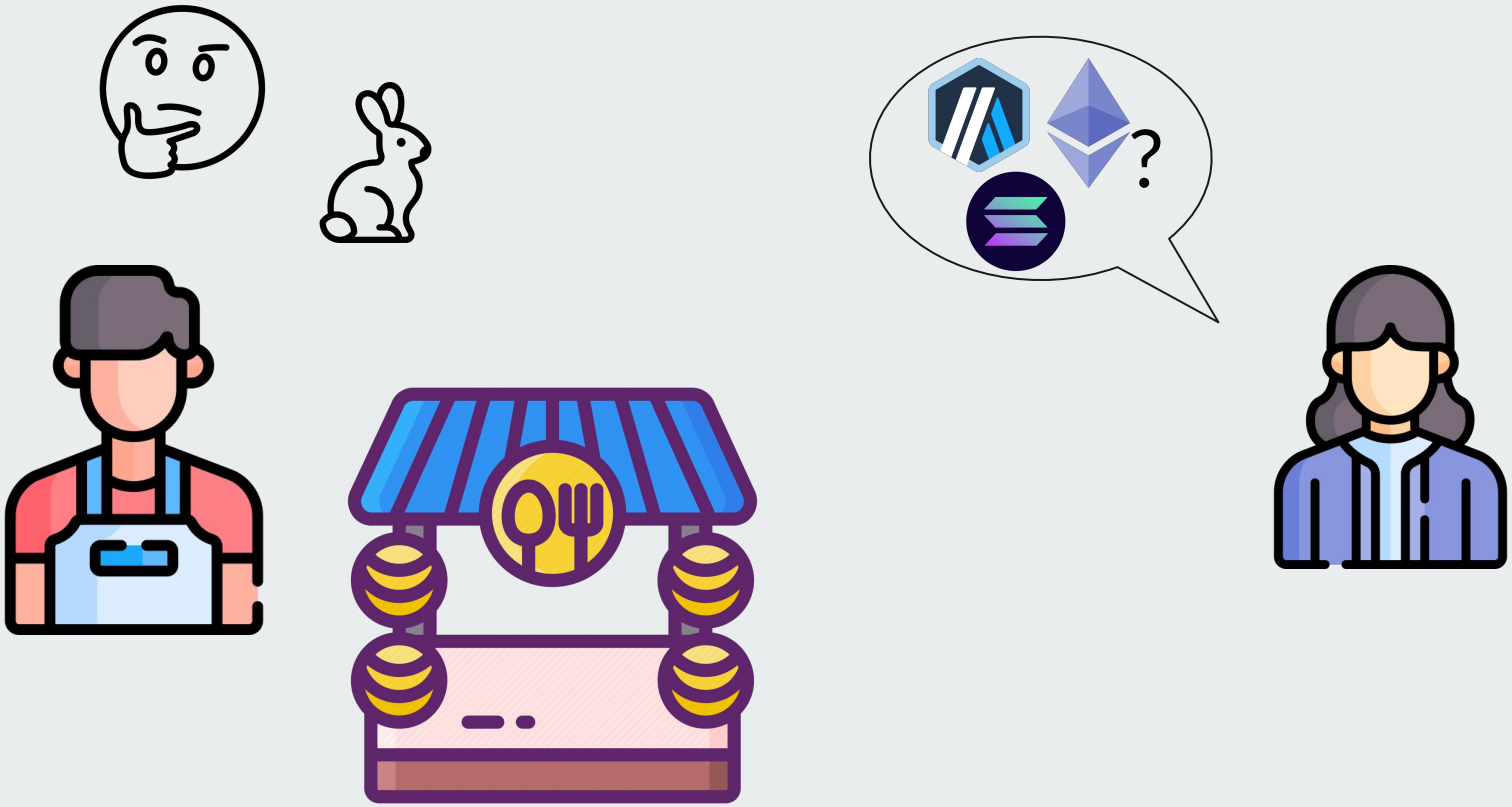
*ETH Zurich - **Dis**tributed **Co**mputing Group*

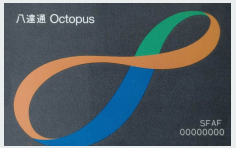
Banyan: Fast Rotating Leader BFT

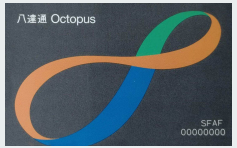
Yann Vonlanthen, Jakub Sliwinski, Massimo Albarello, and Roger Wattenhofer



*ETH Zurich - **Dis**tributed **Co**mputing Group*

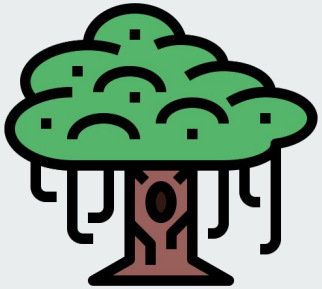



八達通 Octopus
SFAF
00000000

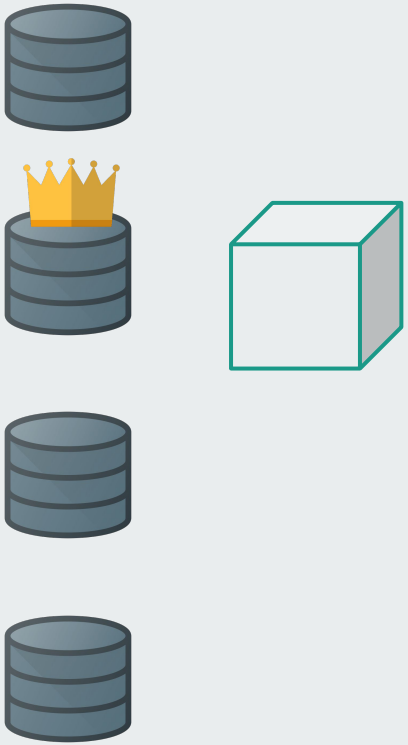

八達通 Octopus
SFAF
00000000

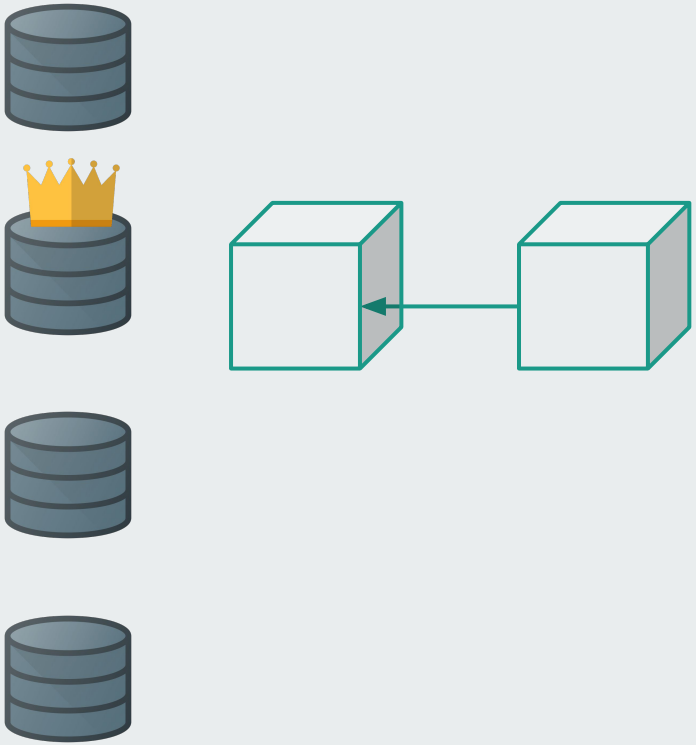
Banyan: Fast Rotating Leader BFT

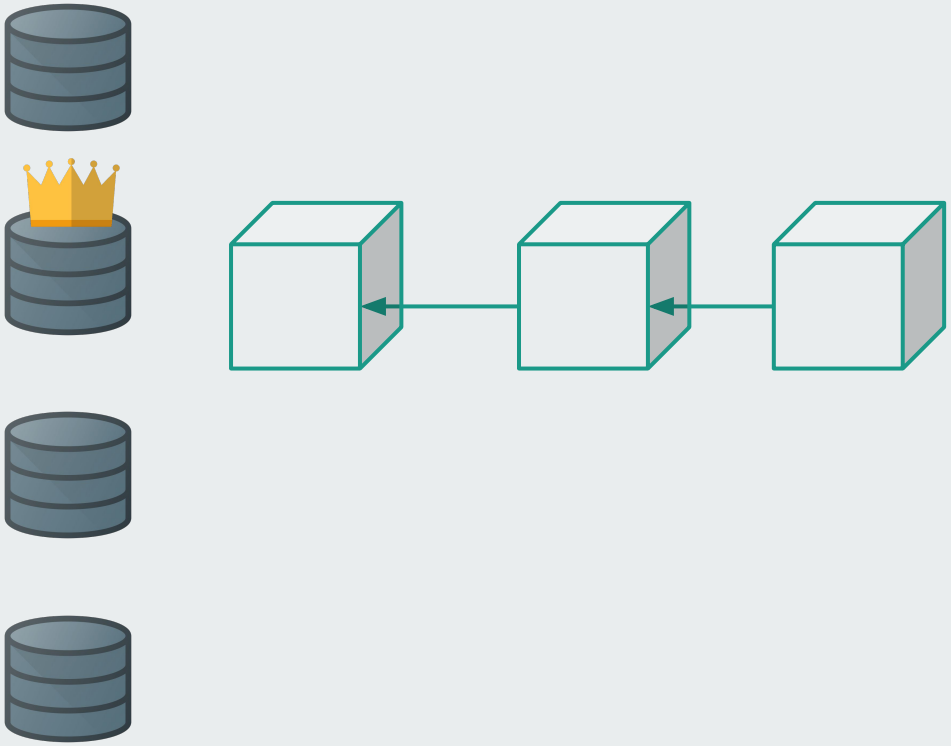
Yann Vonlanthen, Jakub Sliwinski, Massimo Albarello, and Roger Wattenhofer



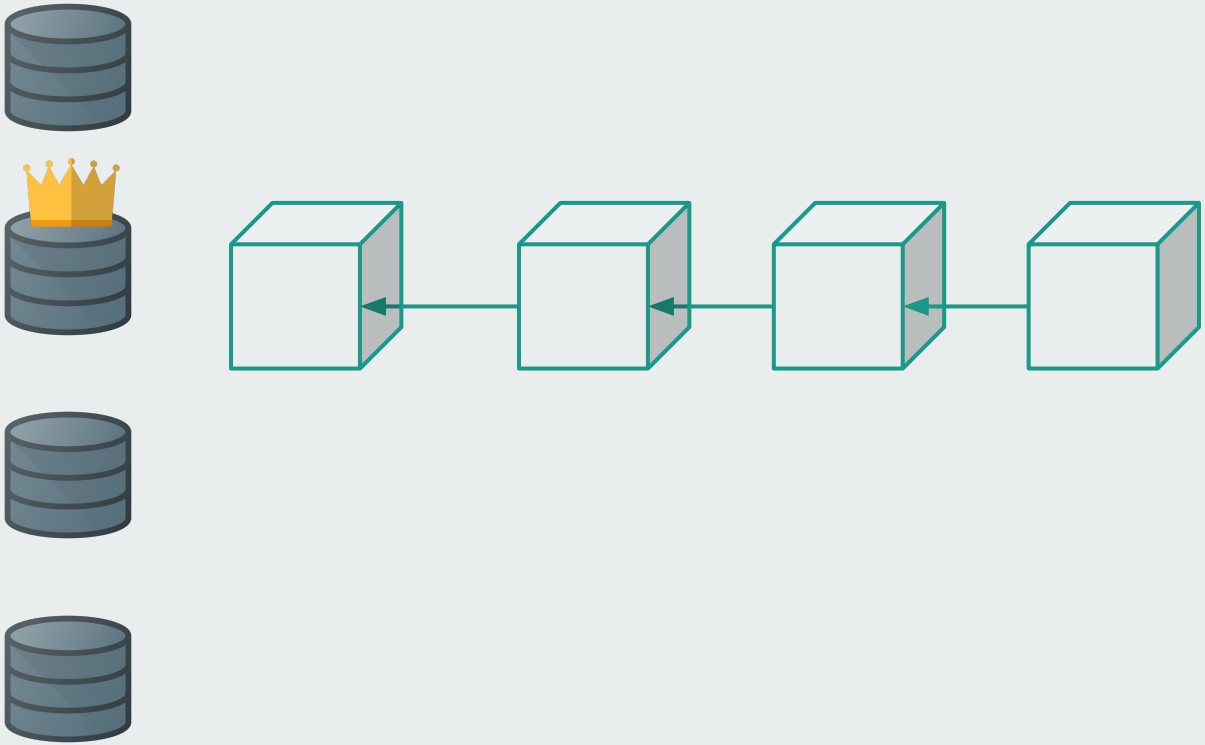
*ETH Zurich - **Dis**tributed **Co**mputing Group*

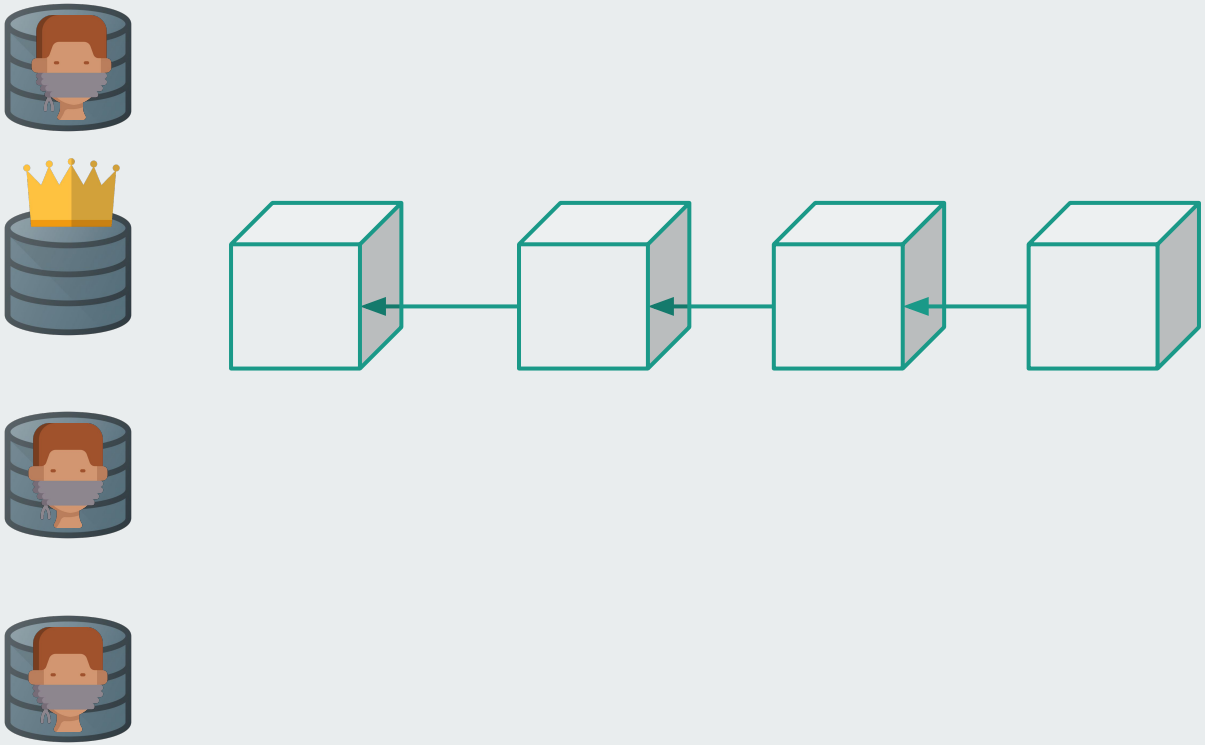


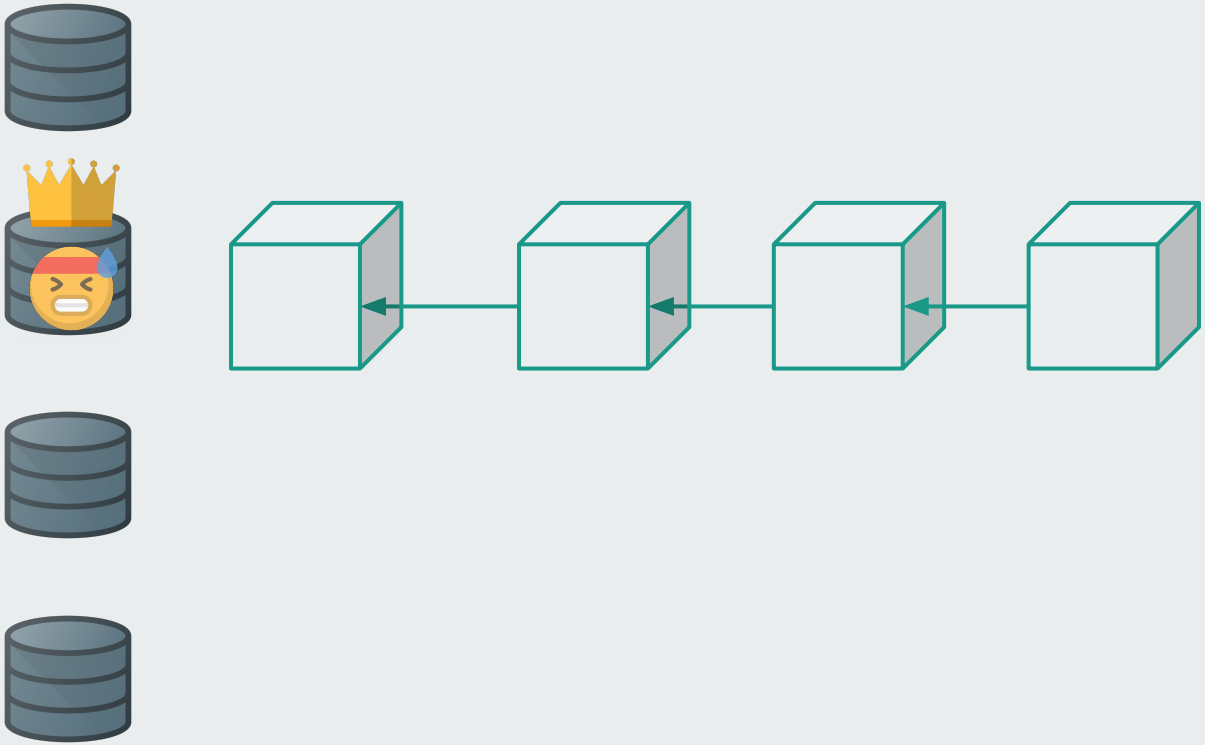


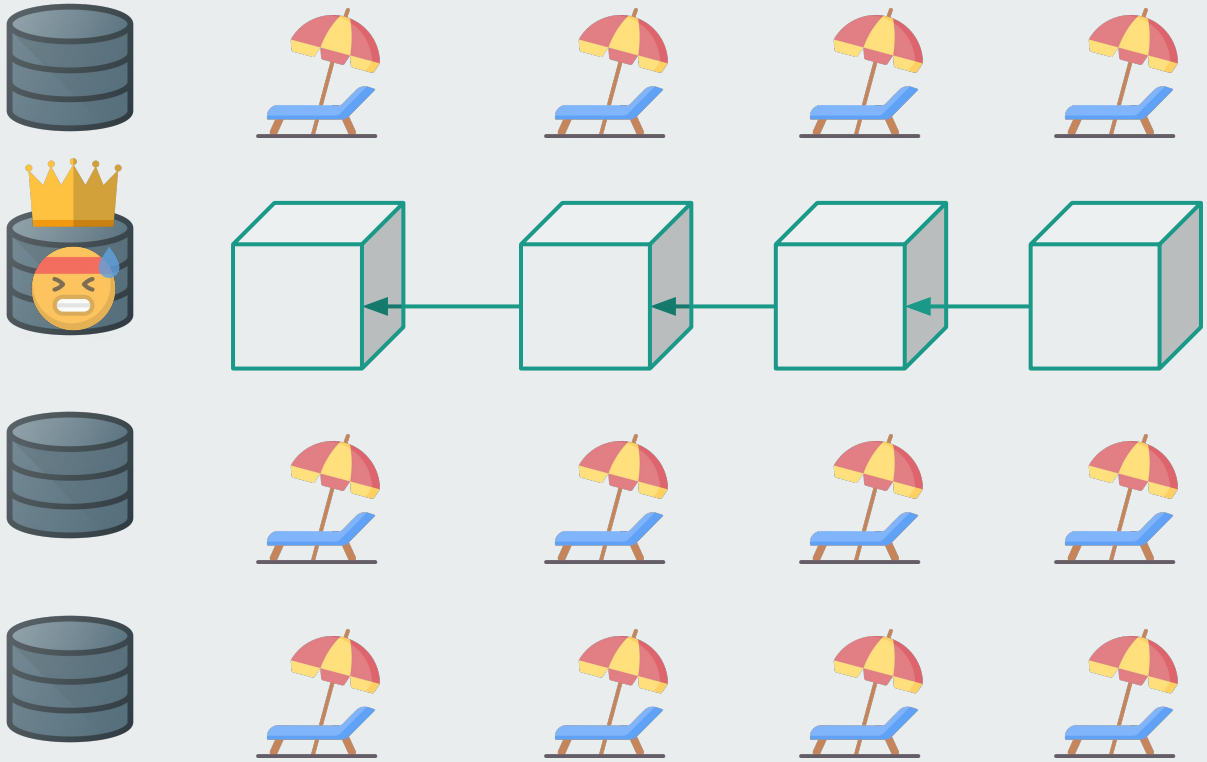


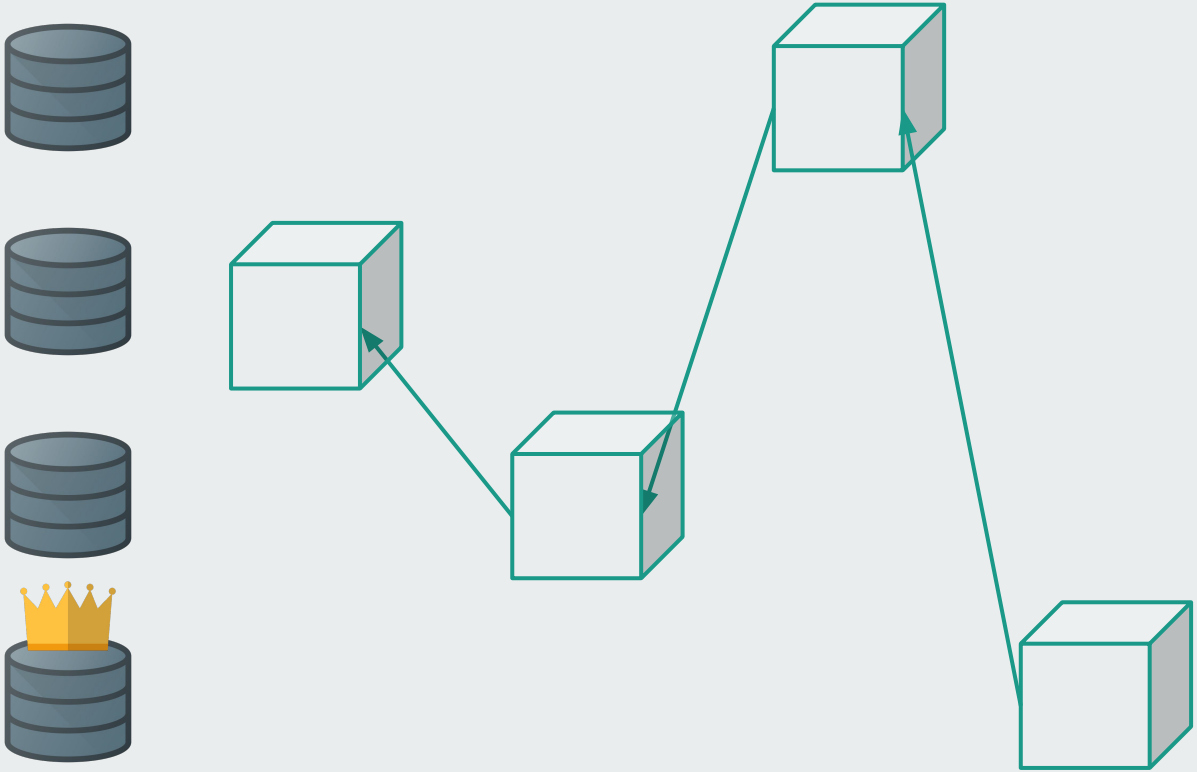
Steady Leader











Why Rotating Leader BFT Protocols?

- Censorship resistance.
- Uniform distribution of work.
- Fairness.

Internet Computer Consensus [PODC'22]

Internet Computer Consensus

Jan Camenisch
DFINITY Foundation
Zurich, Switzerland
jan.camenisch@dfinity.org

Manu Drijvers
DFINITY Foundation
Zurich, Switzerland
manu.drijvers@dfinity.org

Timo Hanke
DFINITY Foundation
Zurich, Switzerland
timo.hanke@dfinity.org

Yvonne-Anne Pignolet
DFINITY Foundation
Zurich, Switzerland
yvonneanne.pignolet@dfinity.org

Victor Shoup
DFINITY Foundation
Zurich, Switzerland
victor.shoup@dfinity.org

Dominic Williams
DFINITY Foundation
Zurich, Switzerland
dominic.williams@dfinity.org

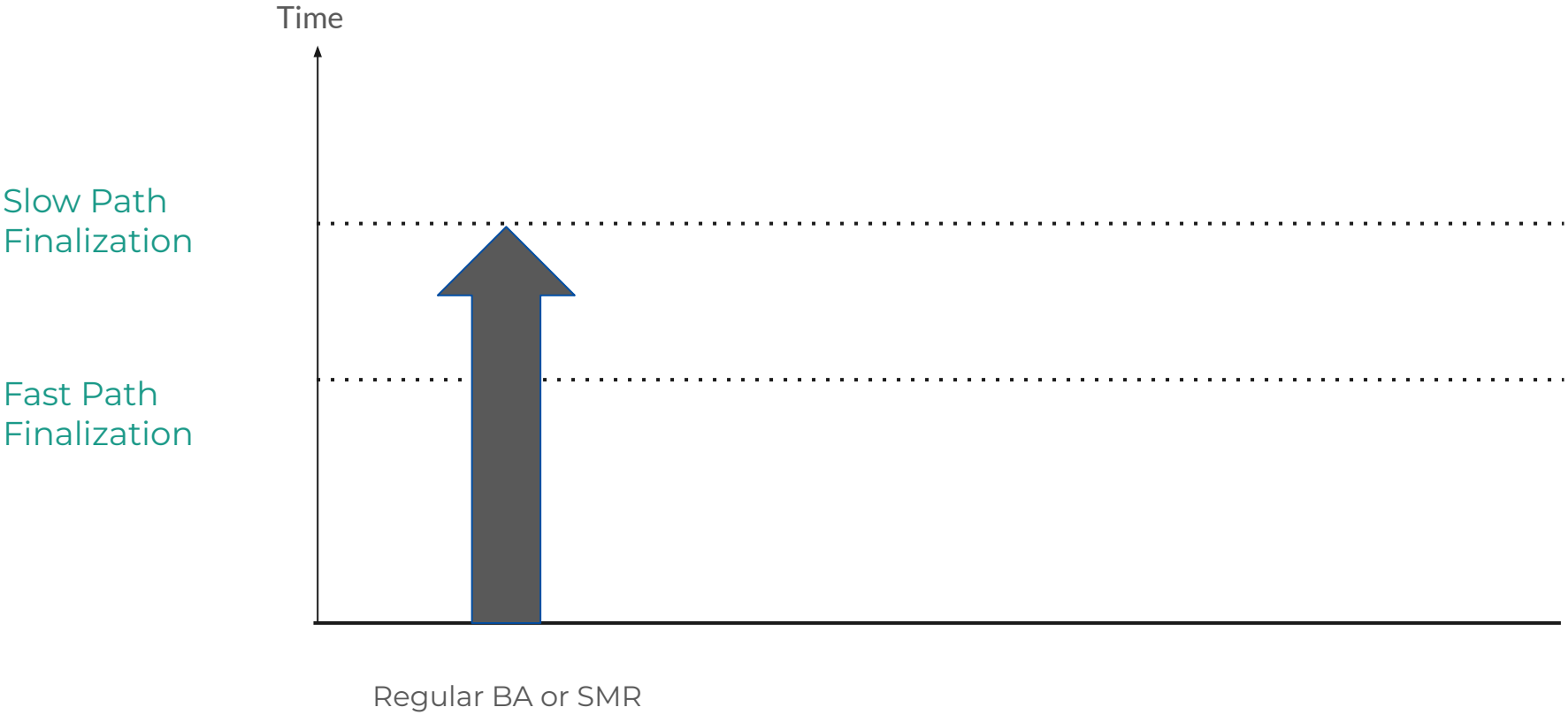
ABSTRACT

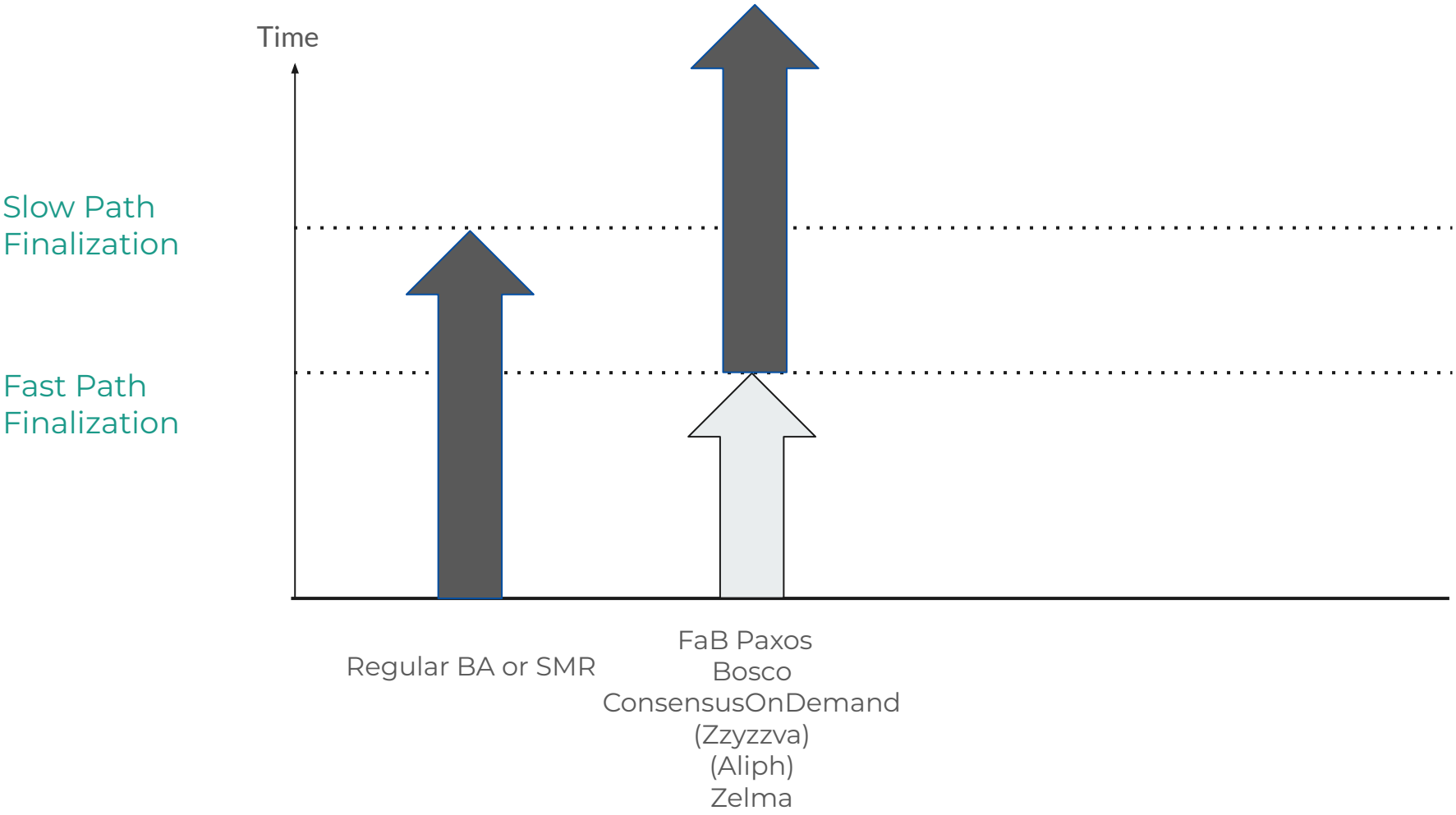
We present the Internet Computer Consensus (ICC) family of protocols for atomic broadcast (a.k.a., consensus), which underpin the Byzantine fault-tolerant replicated state machines of the Internet Computer. The ICC protocols are leader-based protocols that assume partial synchrony, and that are fully integrated with a blockchain. The leader changes probabilistically in every round. These protocols are simple and robust: in any round where the leader is corrupt (which itself happens with probability less than 1/3) or the network is asynchronous, each ICC protocol will effectively allow other parties to step in and propose blocks for that round and to move the protocol forward to the next round. In case there was no agreement on a single block in a round, a decision for this round will be taken in a later round with synchronous network behavior and an honest leader. The task of reliably disseminating the blocks to all parties is an integral part the protocol. We present three different protocols, along with various minor variations on each. The first of these protocols (ICC0) illustrates the

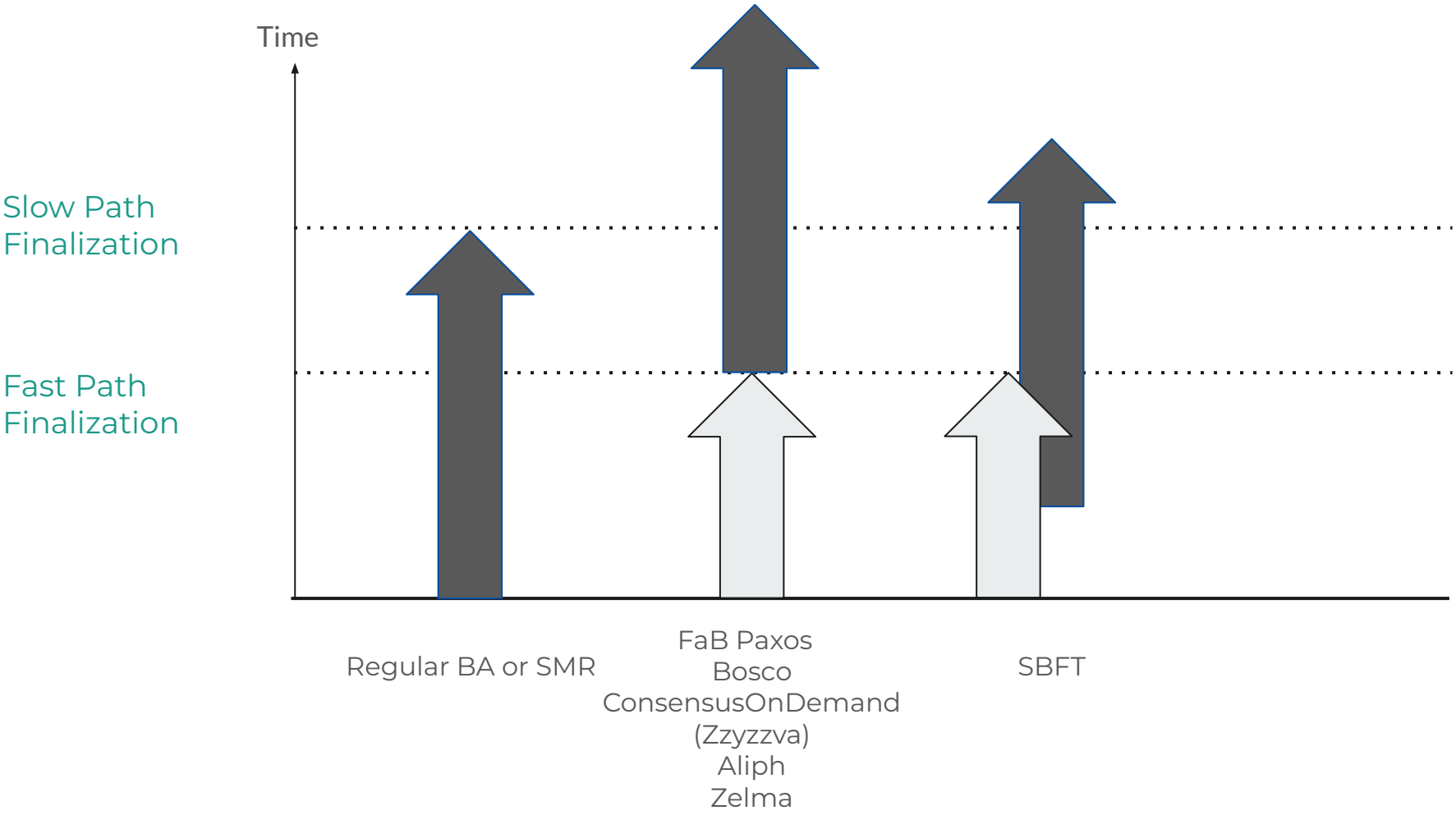
ACM Reference Format:
Jan Camenisch, Manu Drijvers, Timo Hanke, Yvonne-Anne Pignolet, Victor Shoup, and Dominic Williams. 2022. Internet Computer Consensus. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing (PODC '22), July 25–29, 2022, Salerno, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3519270.3538430

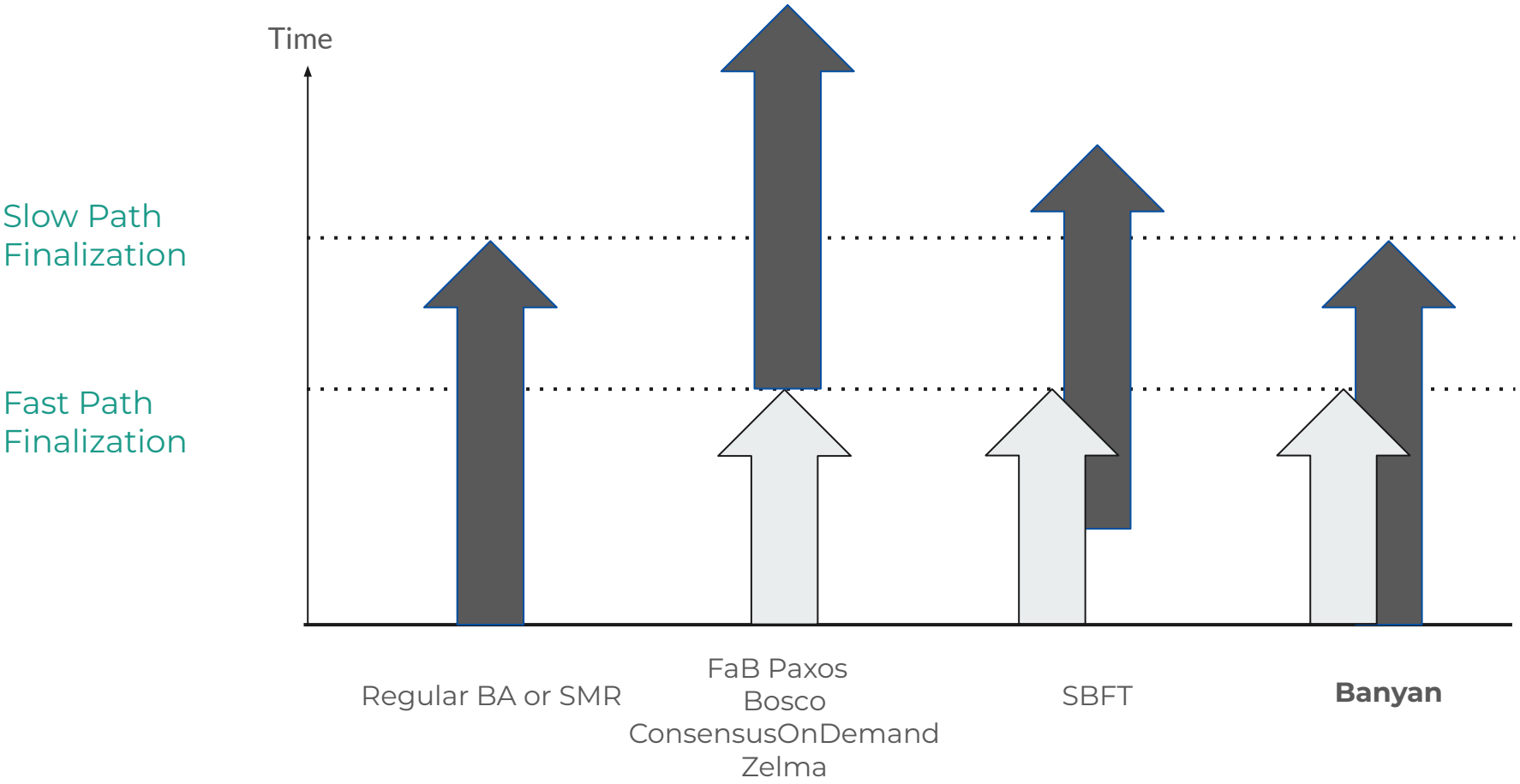
1 INTRODUCTION

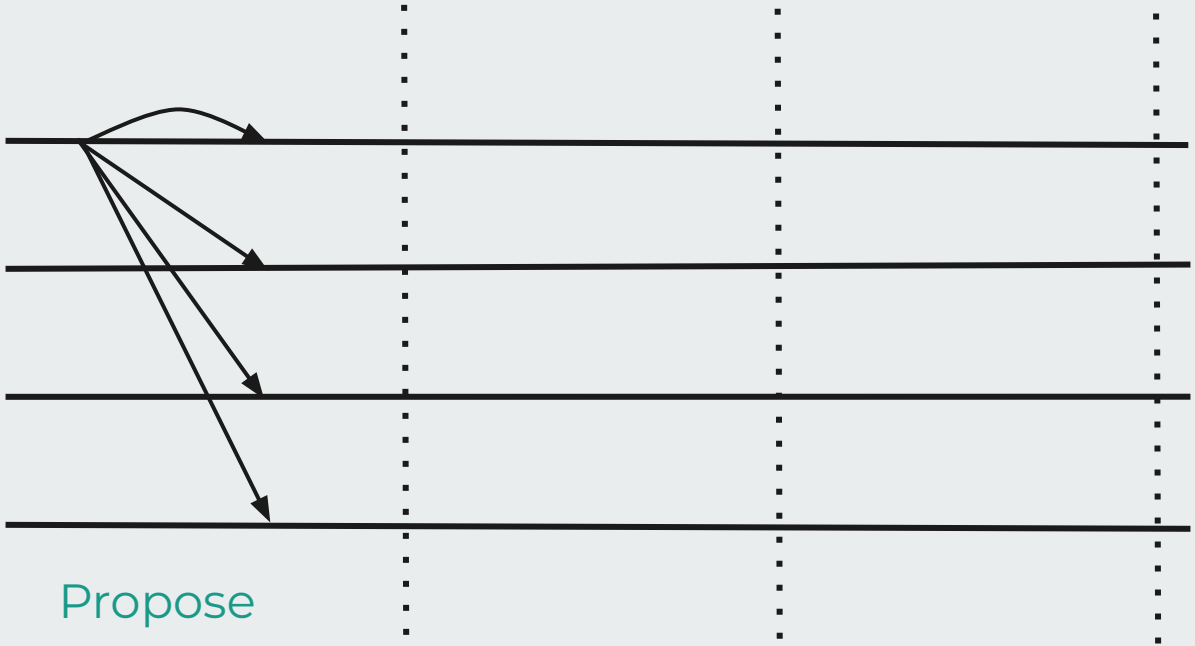
Byzantine fault tolerance (BFT) is the ability of a computing system to endure arbitrary (i.e., Byzantine) failures of some of its components while still functioning properly as a whole. One approach to achieving BFT is via *state machine replication* [33]: the logic of the system is replicated across a number of machines, each of which maintains state, and updates its state is by executing a sequence of *commands*. In order to ensure that the non-faulty machines end up in the same state, they must each deterministically execute the same sequence of commands. This is achieved by using a protocol for *atomic broadcast* [9, 16, 33].

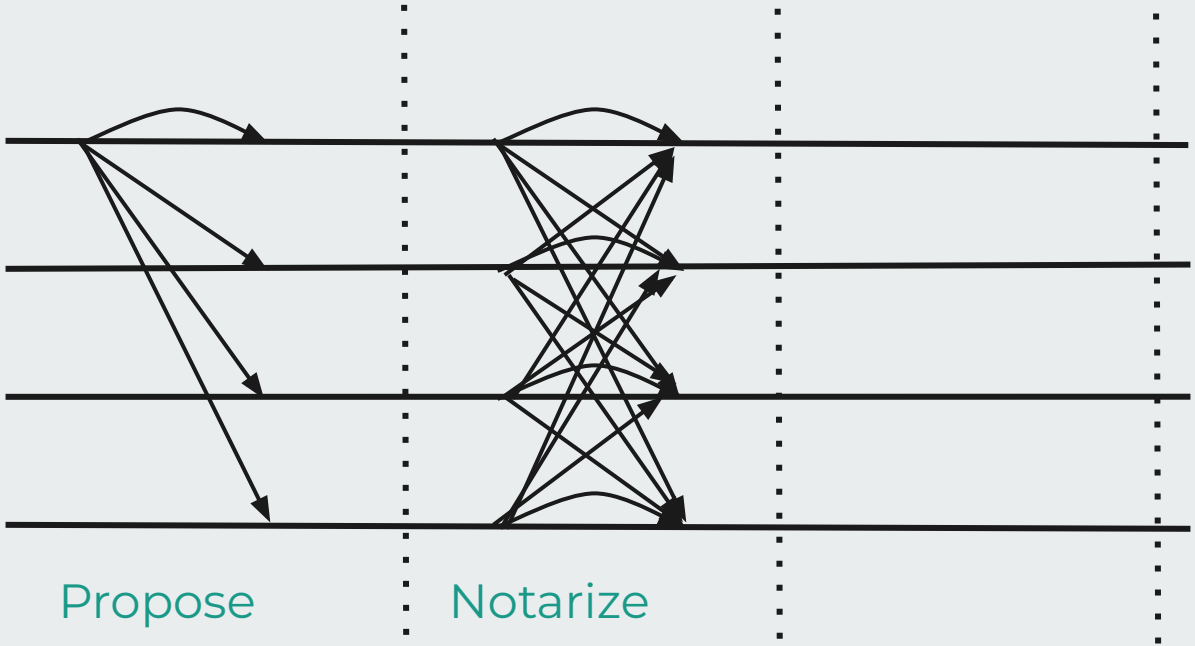

Time
Slow Path Finalization
Fast Path Finalization
Regular BA or SMR

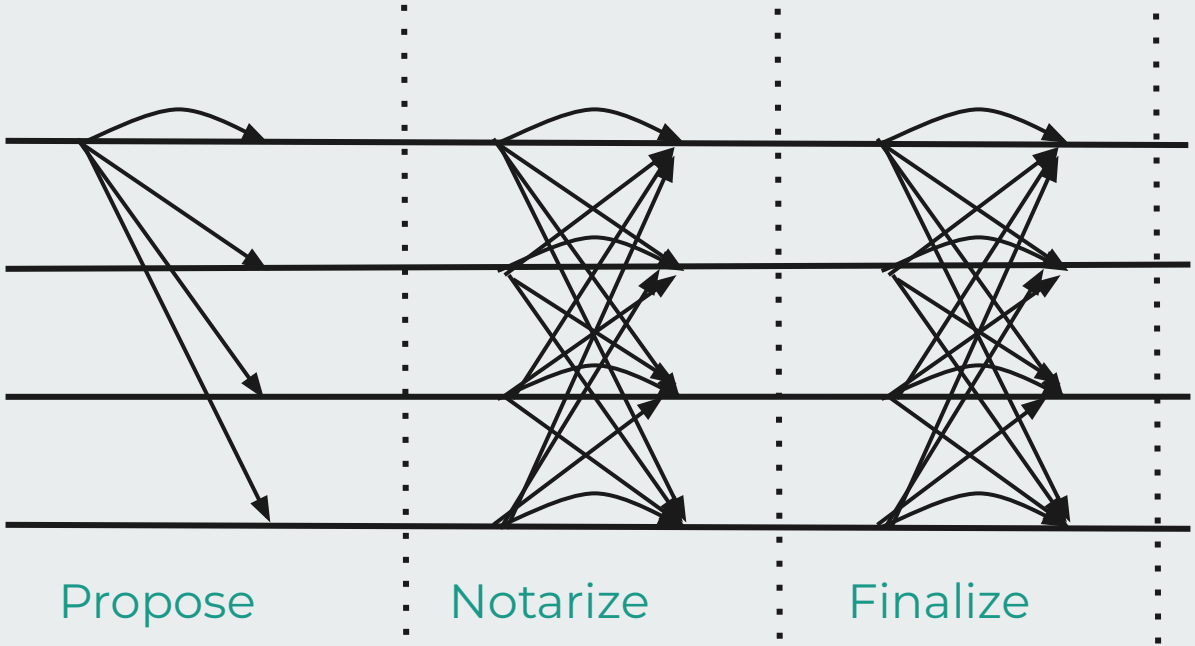

Time
Slow Path Finalization
Fast Path Finalization
Regular BA or SMR
FaB Paxos
Bosco
ConsensusOnDemand
(Zzyzzva)
(Aliph)
Zelma

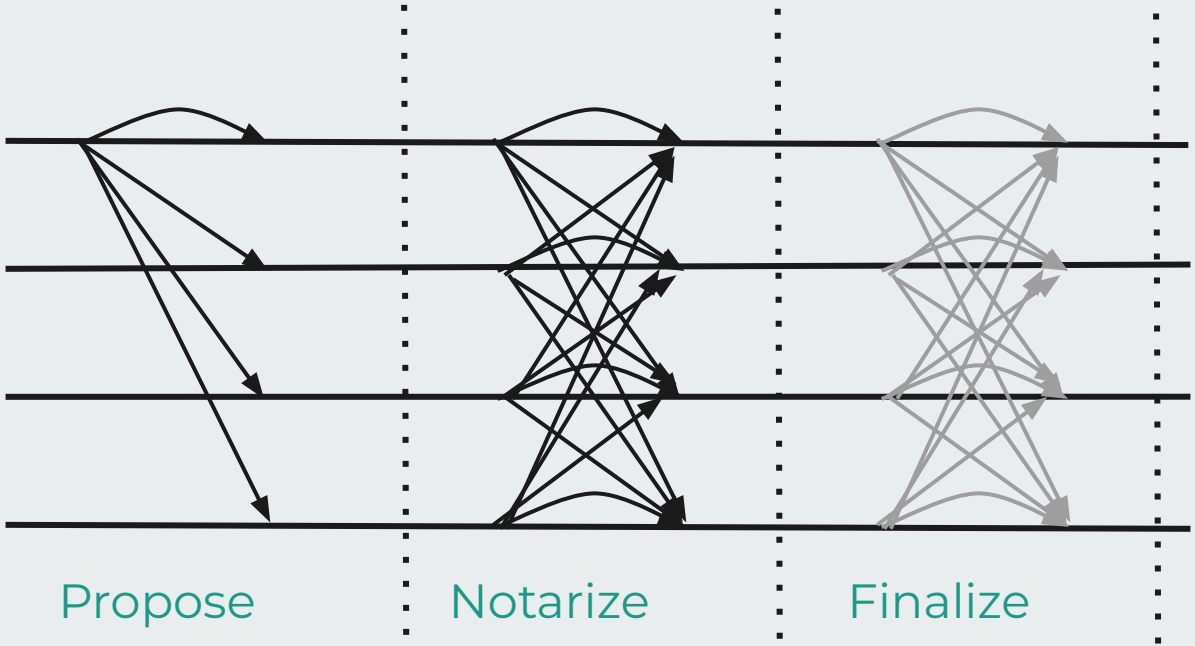

Time
Slow Path Finalization
Fast Path Finalization
Regular BA or SMR
FaB Paxos
Bosco
ConsensusOnDemand
(Zzyzzva)
Aliph
Zelma
SBFT


Time
Slow Path Finalization
Fast Path Finalization
Regular BA or SMR
FaB Paxos
Bosco
ConsensusOnDemand
Zelma
SBFT
Banyan


Propose

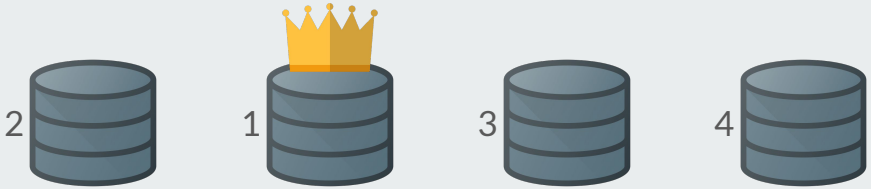

Propose
Notarize


Propose
Notarize
Finalize

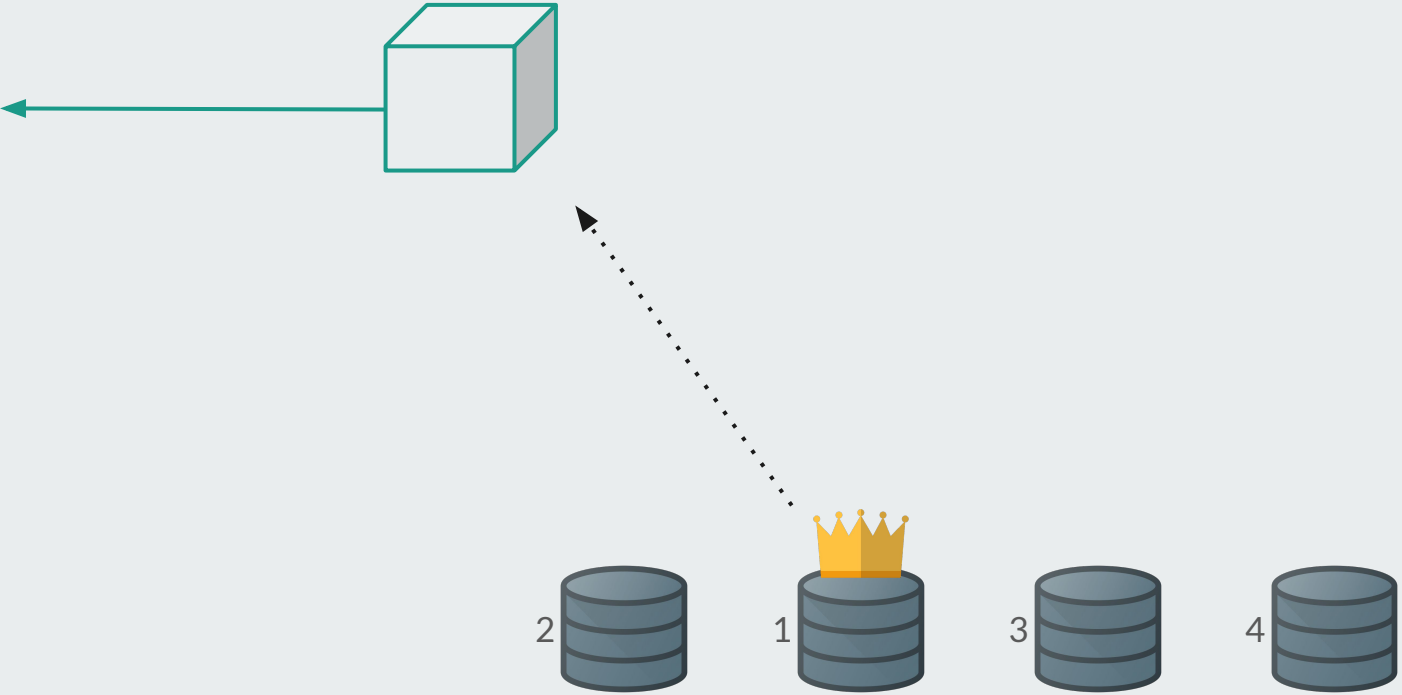

Propose
Notarize
Finalize

Internet Computer Consensus - Optimistic Case

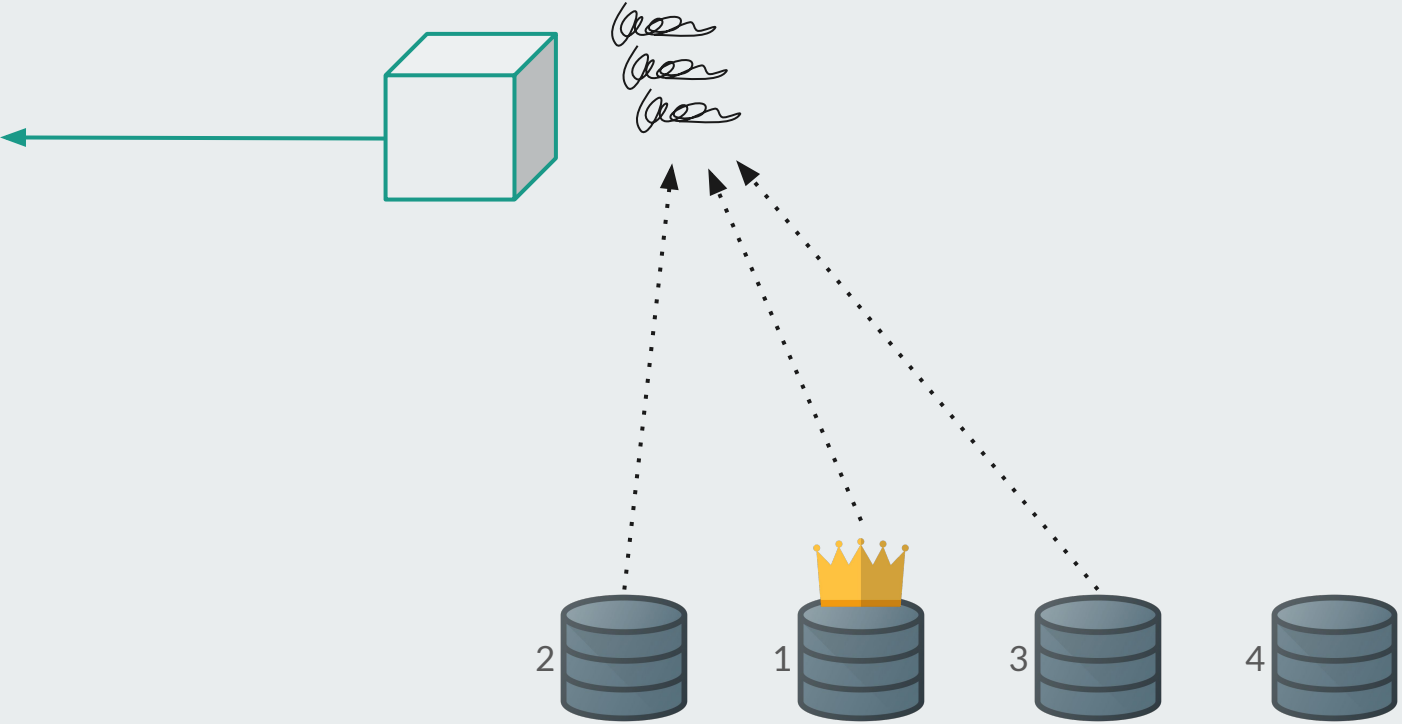
Internet Computer Consensus - Optimistic Case



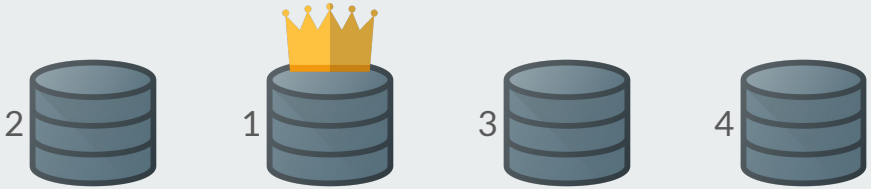
Internet Computer Consensus - Optimistic Case



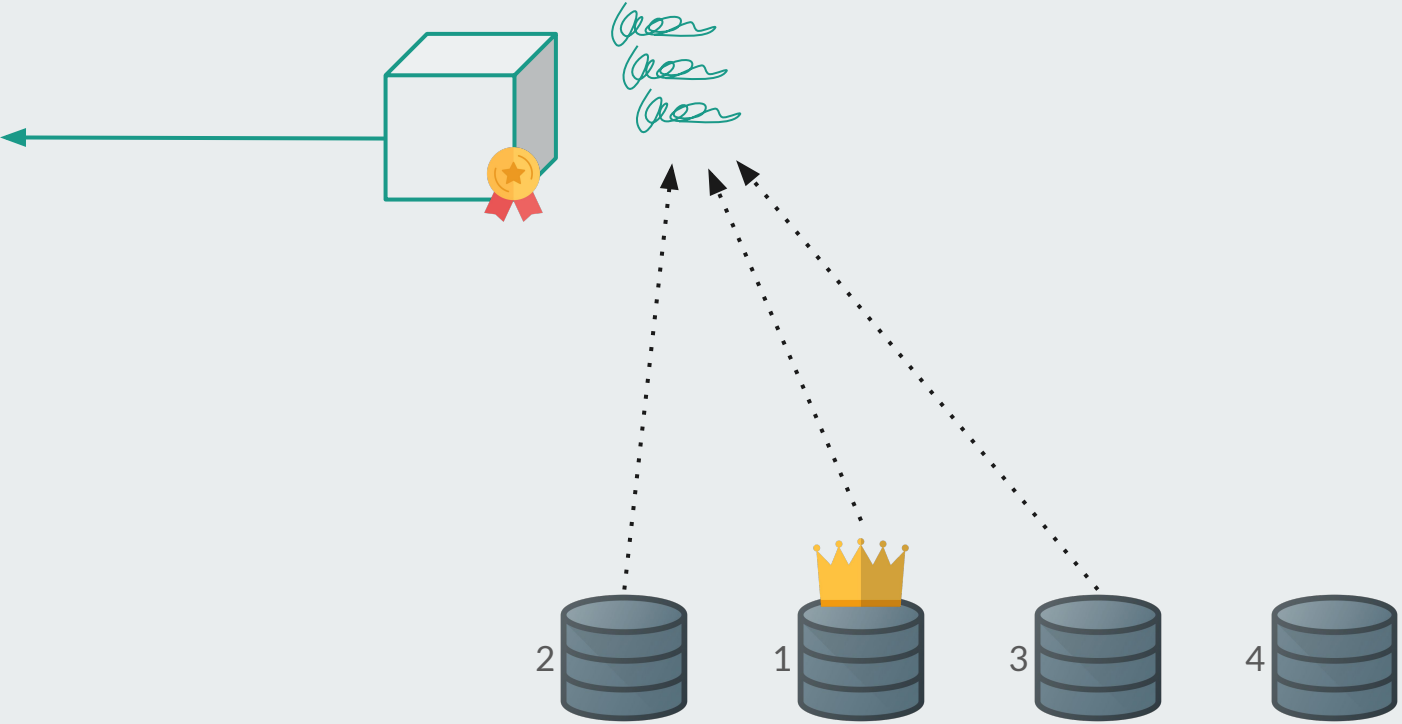
Internet Computer Consensus - Optimistic Case



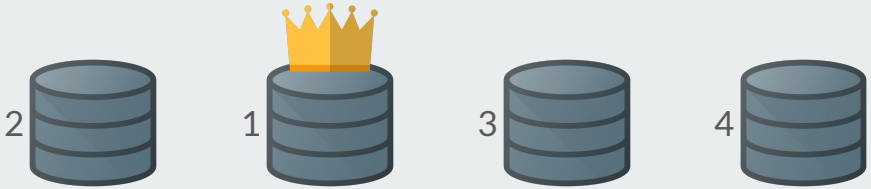
Internet Computer Consensus - Optimistic Case



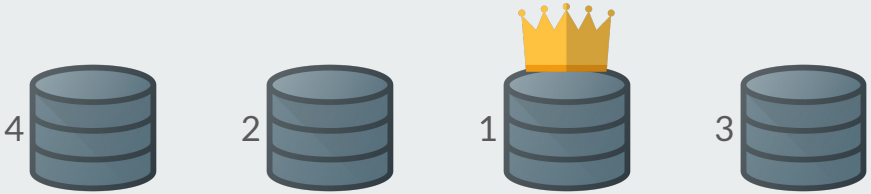
Internet Computer Consensus - Optimistic Case



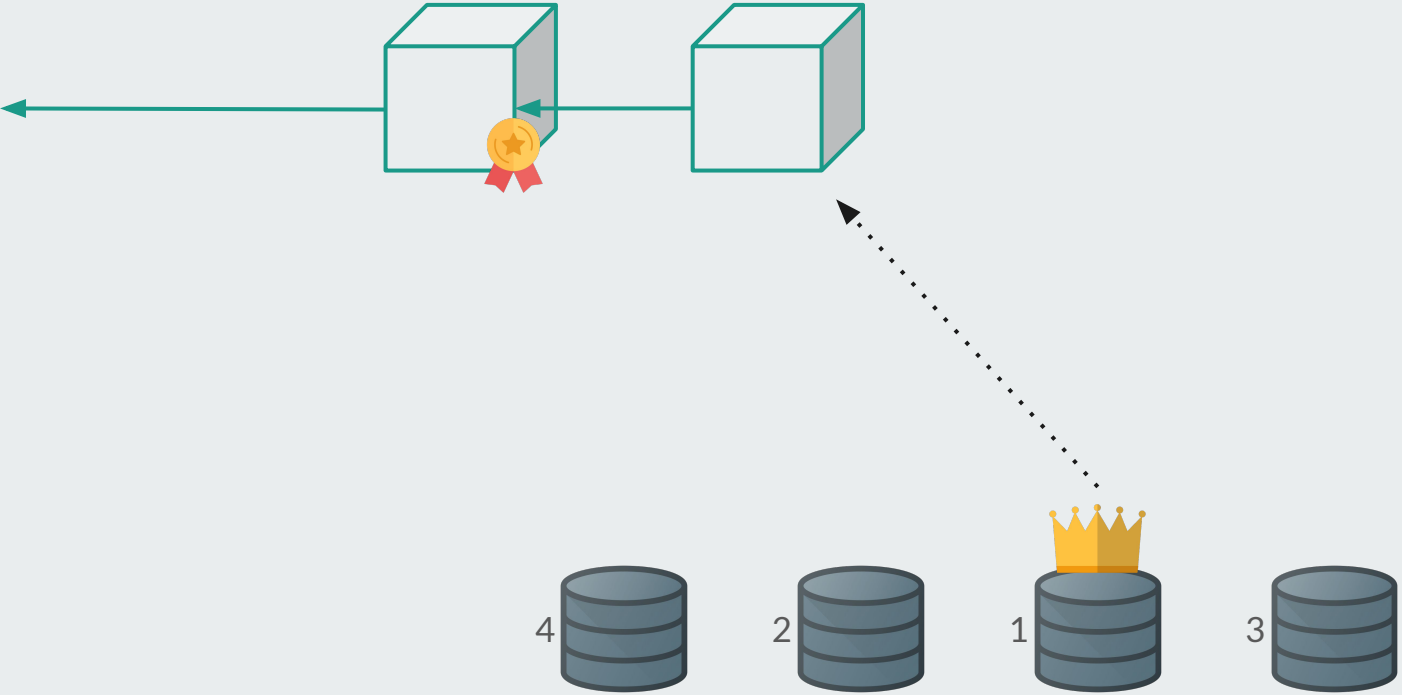
Internet Computer Consensus - Optimistic Case



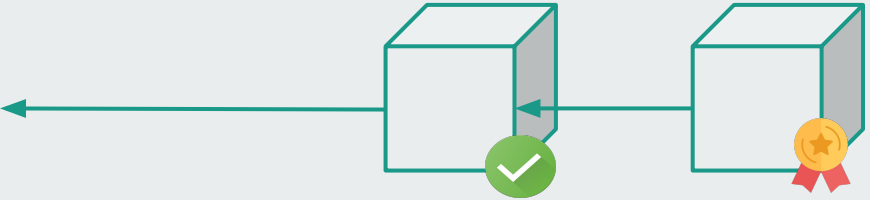
Internet Computer Consensus - Optimistic Case

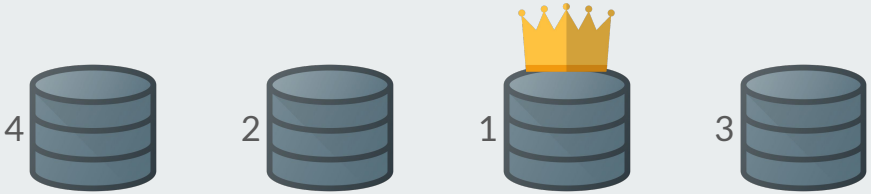


Internet Computer Consensus

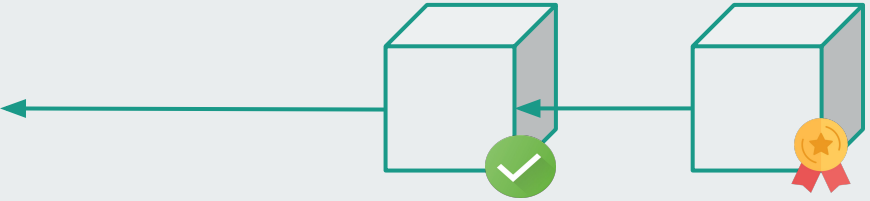


Internet Computer Consensus

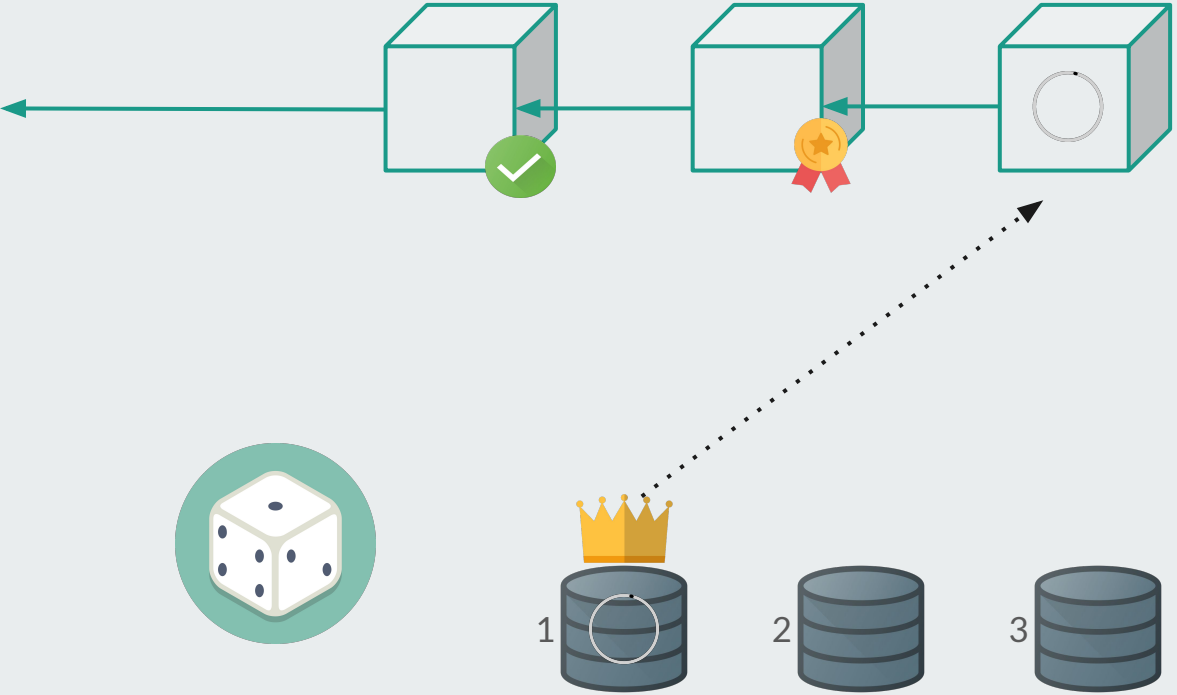




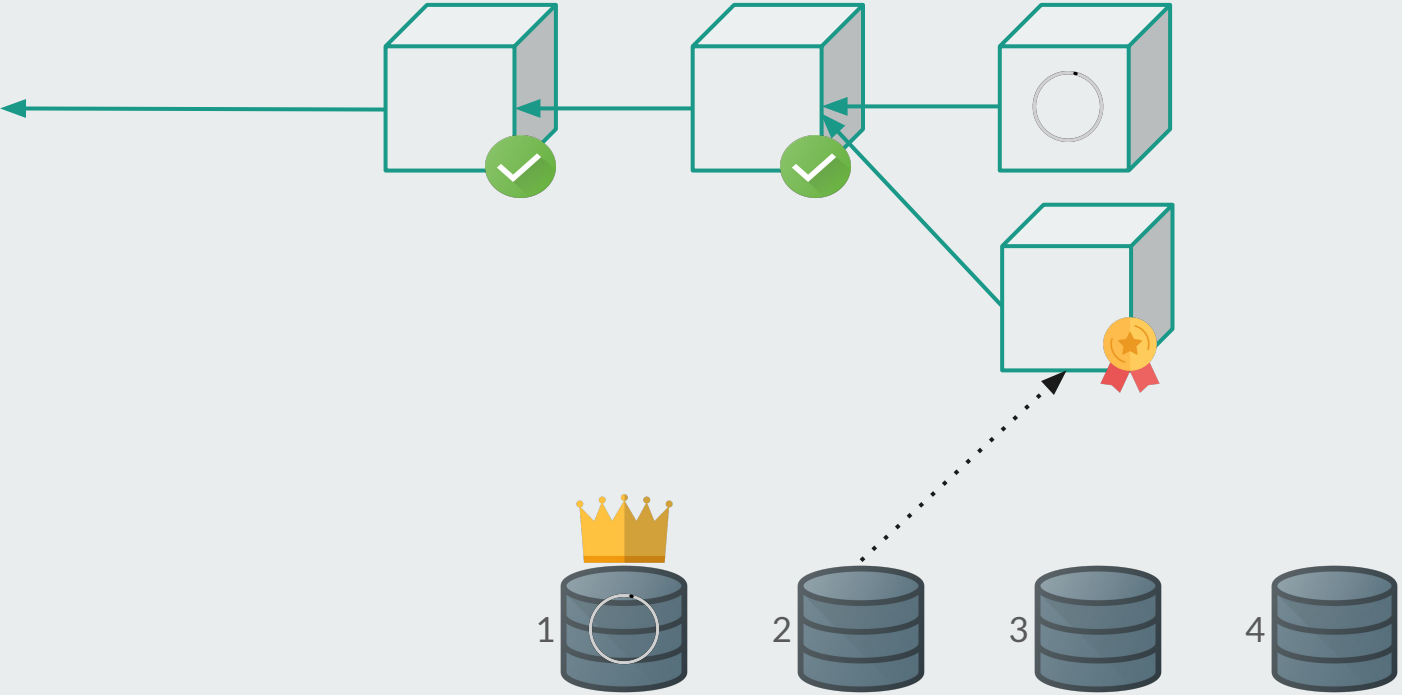
Internet Computer Consensus



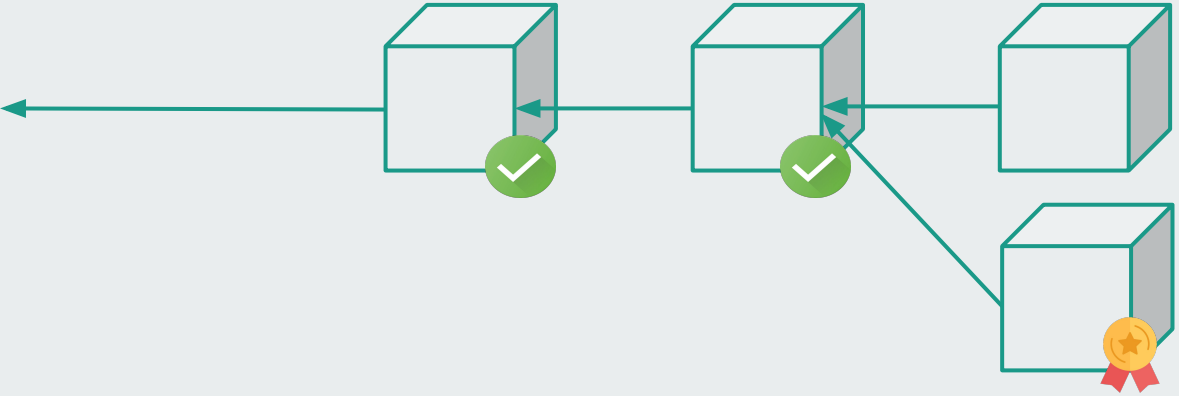
Internet Computer Consensus



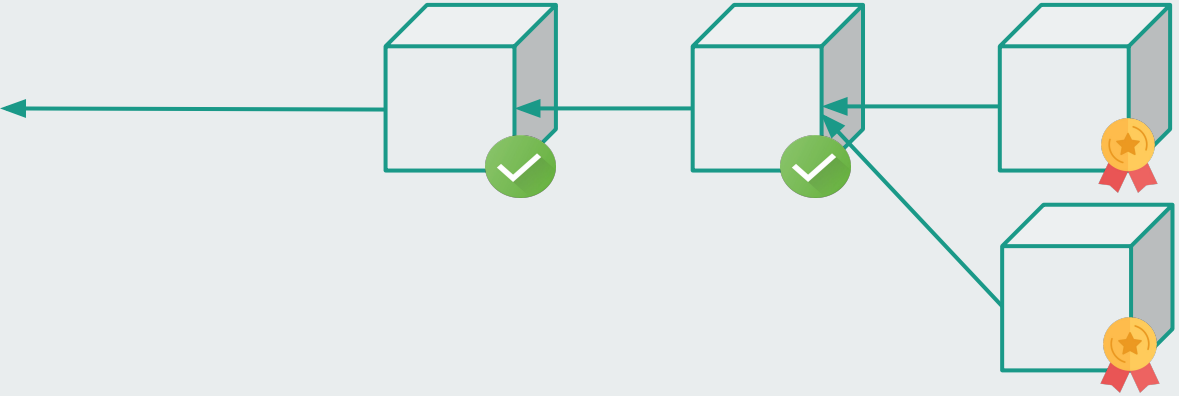
Internet Computer Consensus



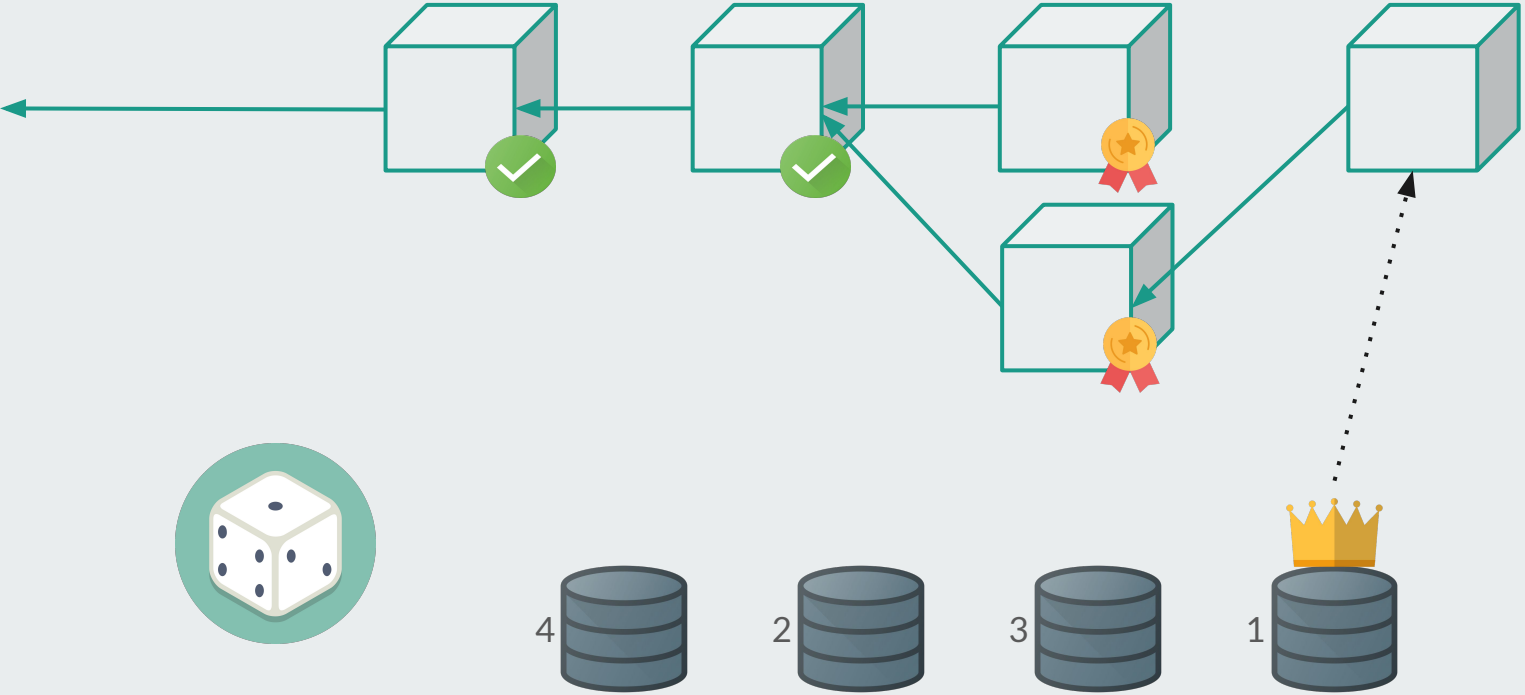
Internet Computer Consensus



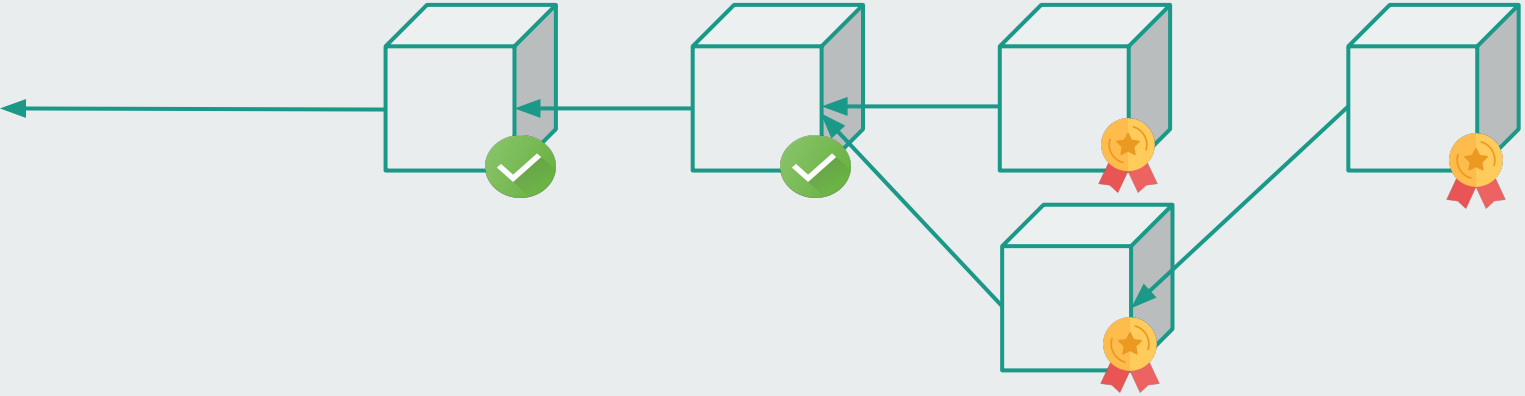
Internet Computer Consensus



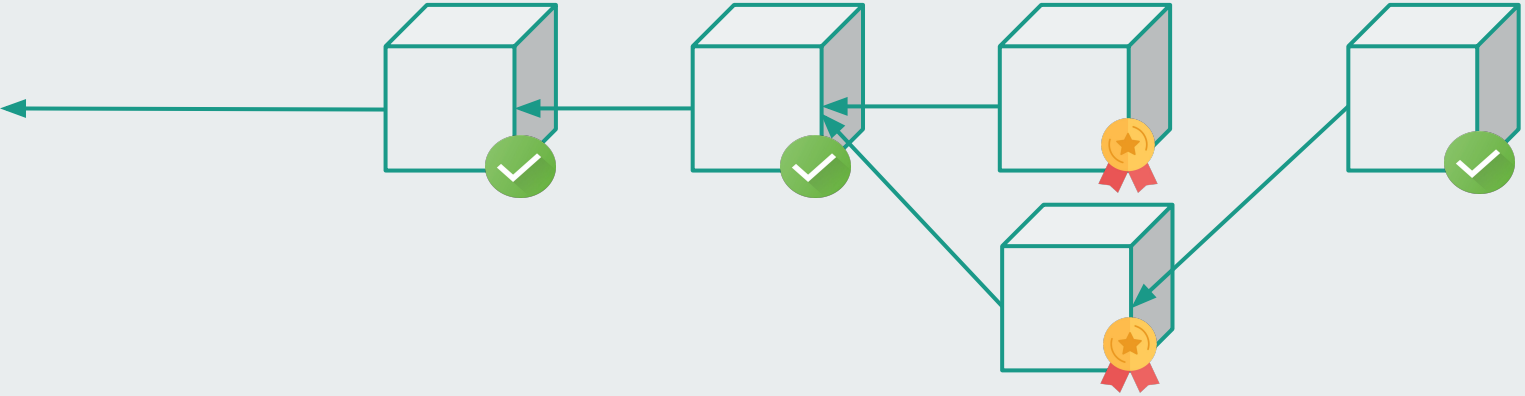
Internet Computer Consensus



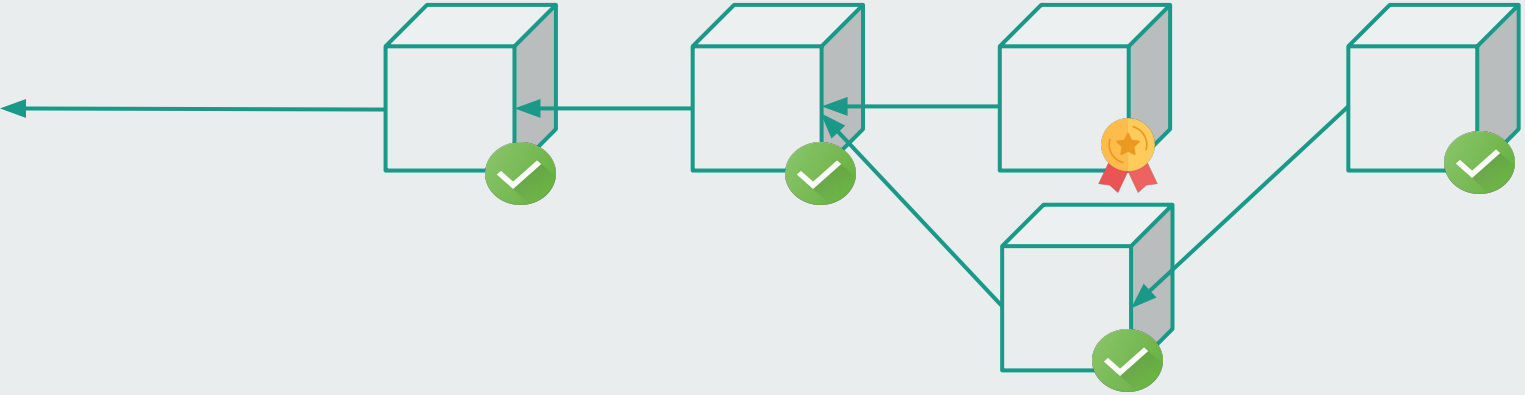
Internet Computer Consensus



Internet Computer Consensus



Internet Computer Consensus

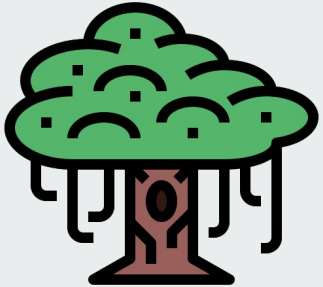


Internet Computer Consensus

notarization signatures

finalization signatures

Banyan



first notarization signature

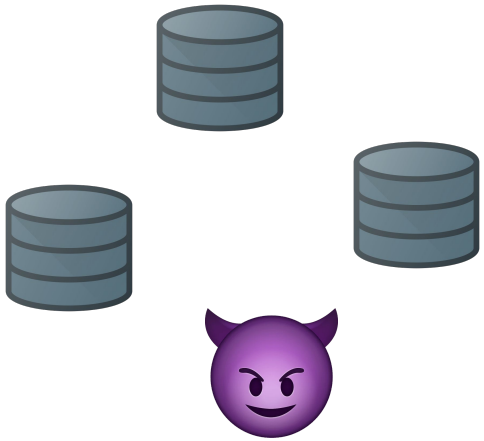
other notarization signatures

finalization signatures

Model – Banyan

We get p=1 for free!!

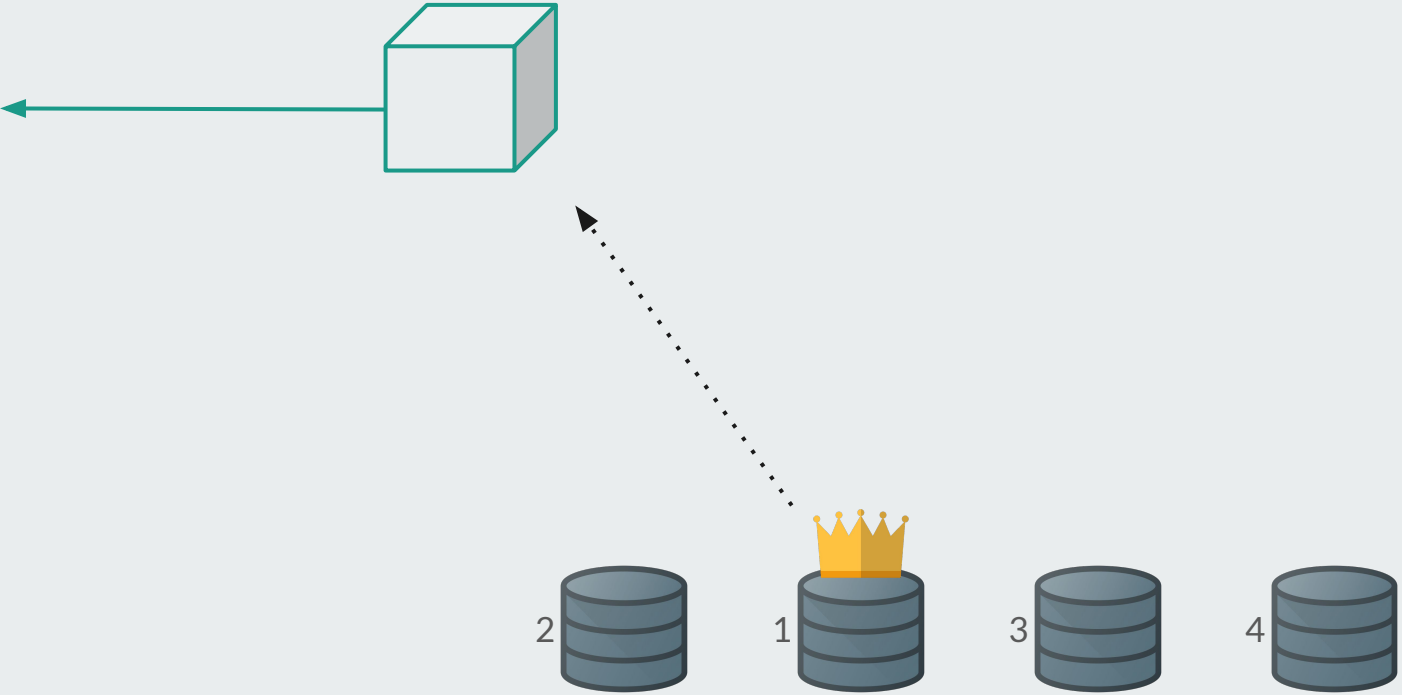
$n \geq 3f + 2p^* - 1$



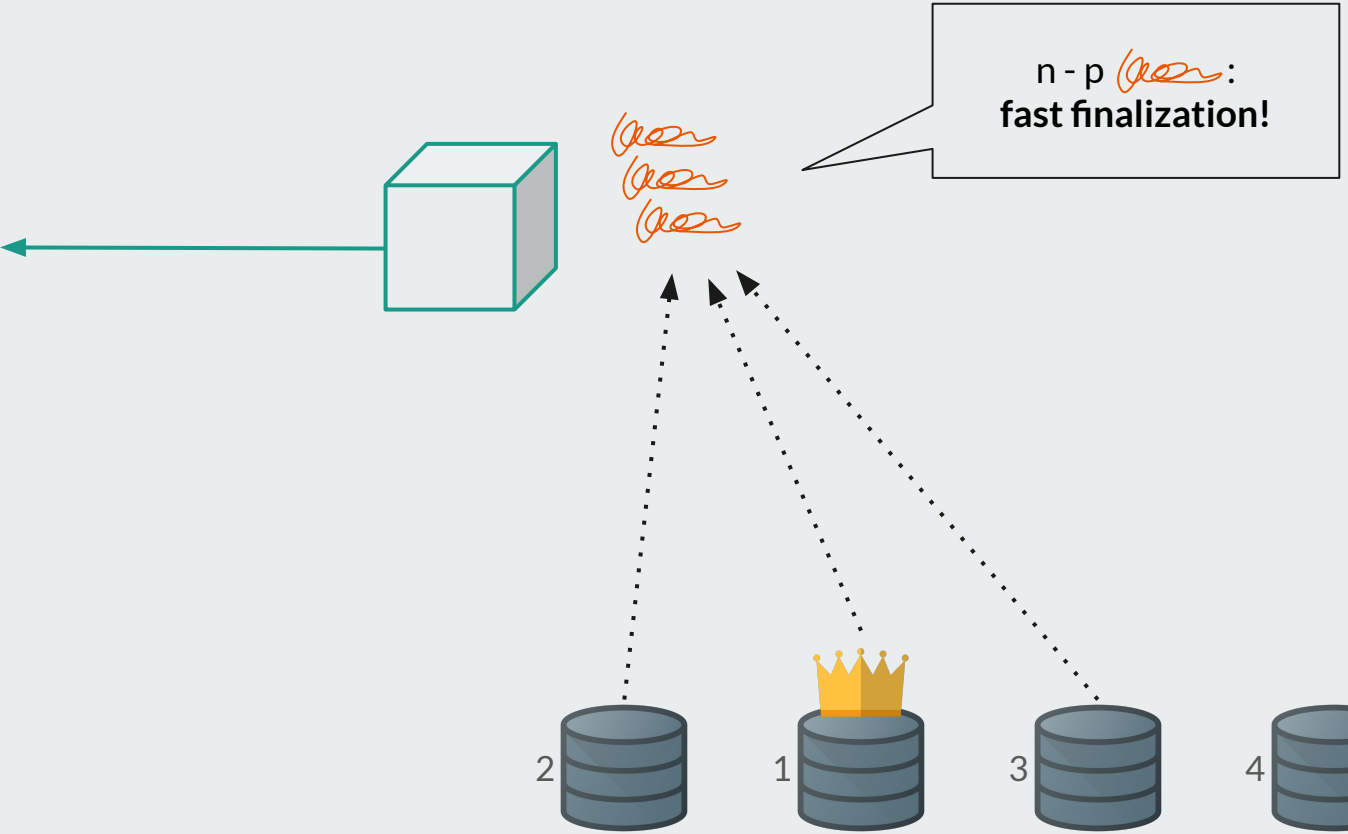
Rule: Commit when **(n - p)**

notarization signatures received

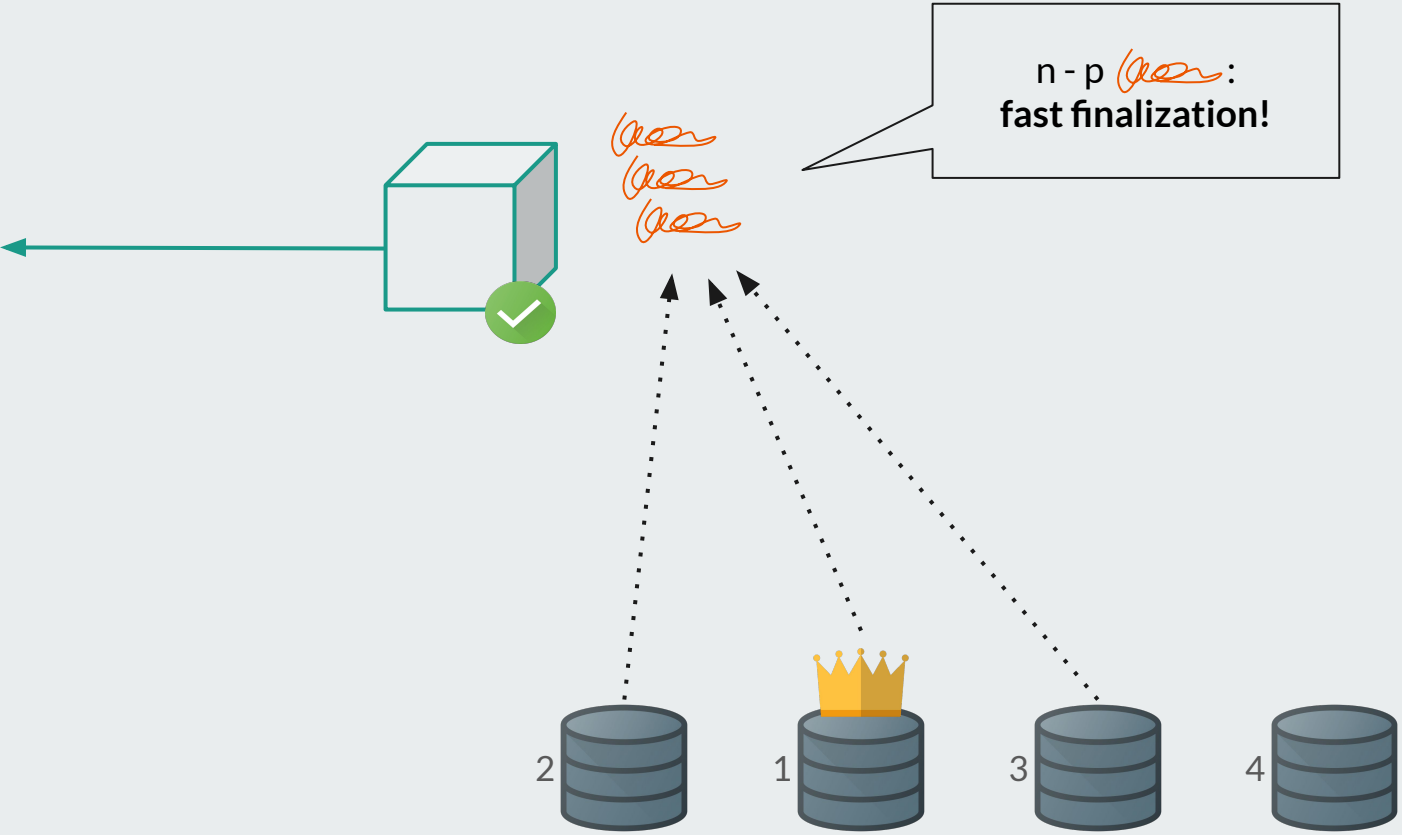
Banyan - Optimistic Case



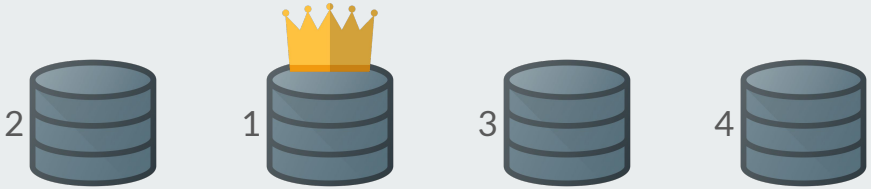
Banyan - Optimistic Case



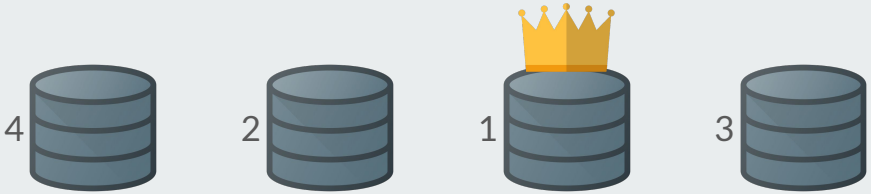
Banyan - Optimistic Case



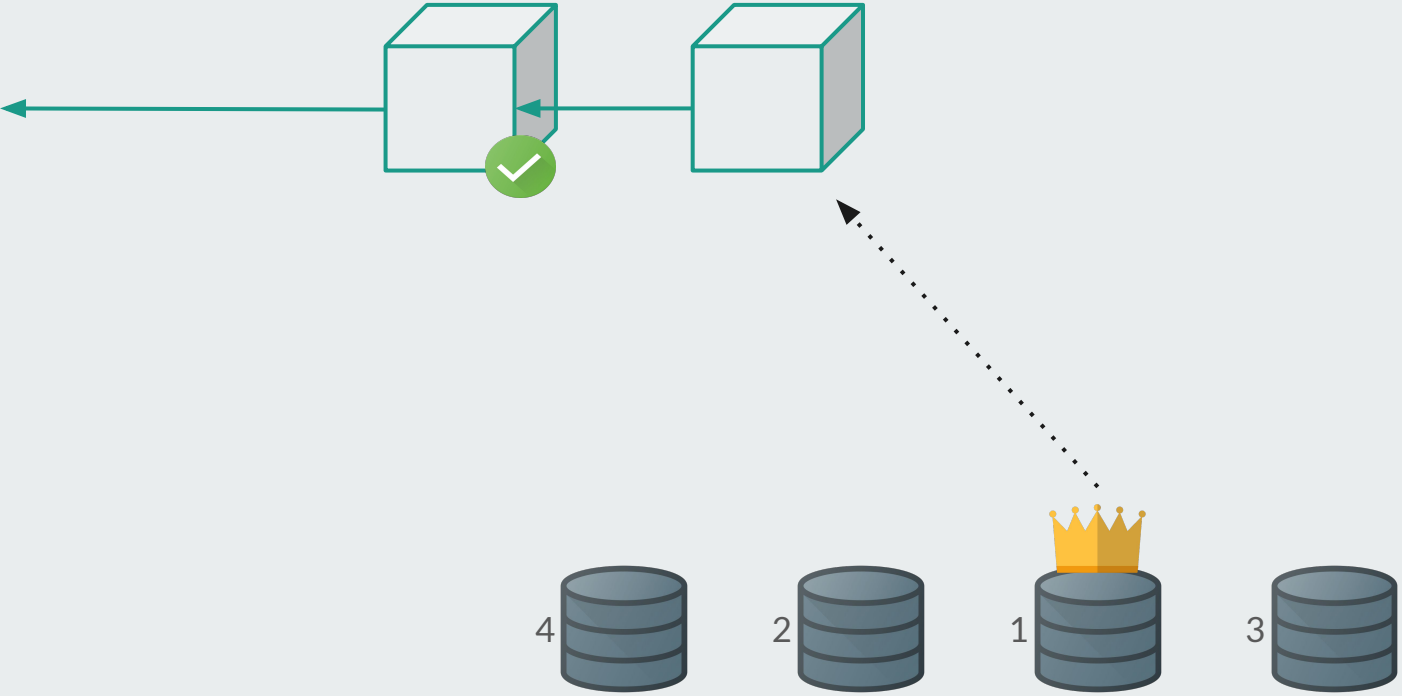
Banyan - Optimistic Case



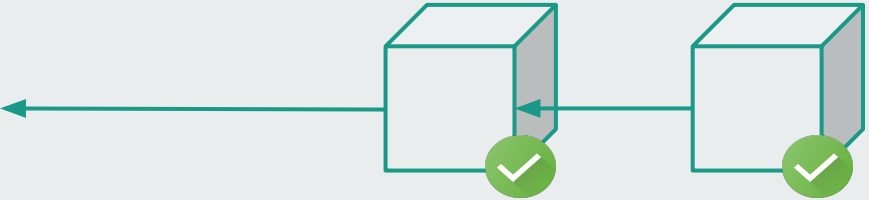
Banyan - Optimistic Case

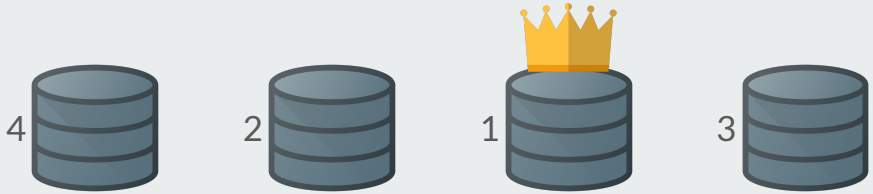


Banyan - Optimistic Case

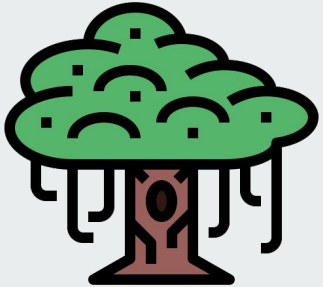


Banyan - Optimistic Case





Banyan - Recap



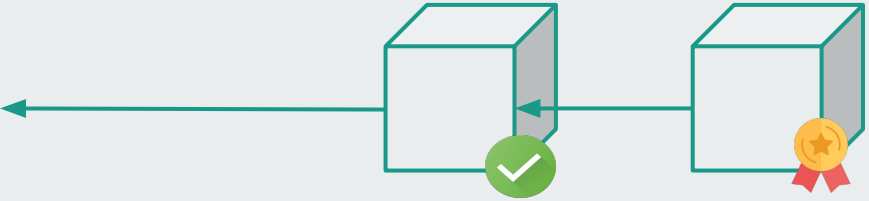
Fast path: Commit when **n - p**

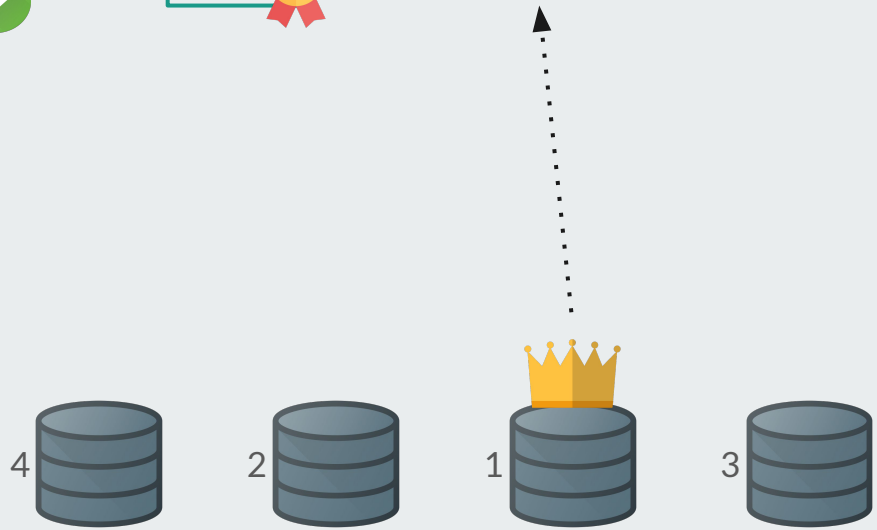
notarization signatures received

Slow path: ICC + new rule:

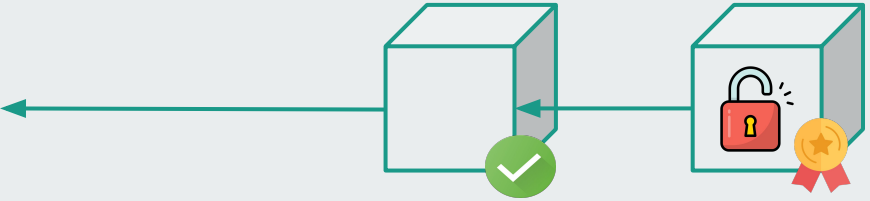
only extend and finalize *unlocked* blocks

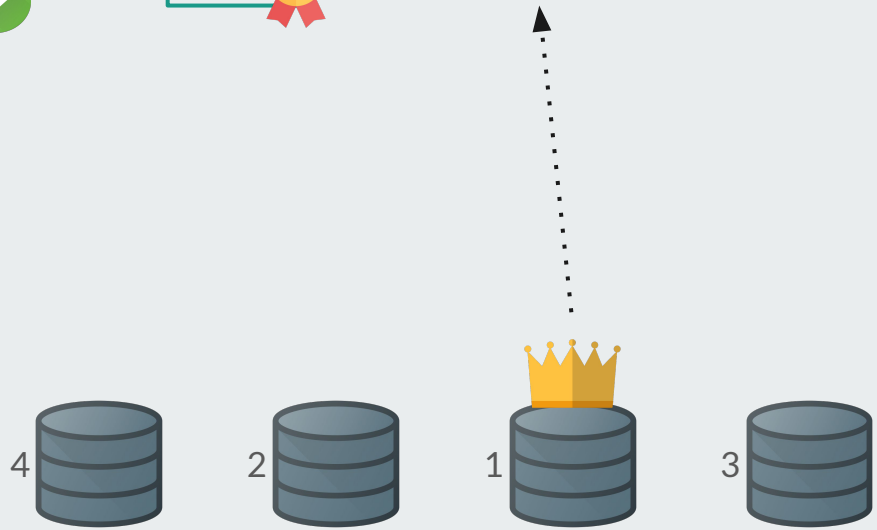
Banyan - Unlocking Blocks



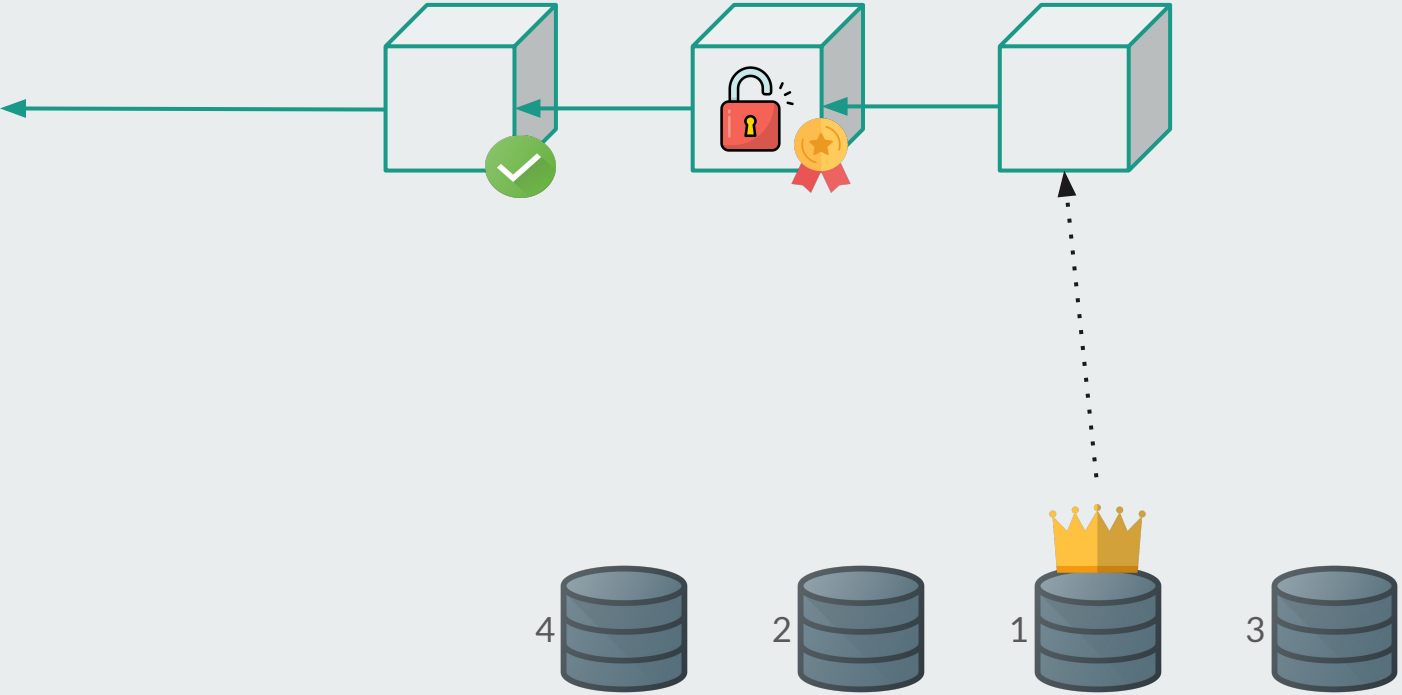


Banyan - Unlocking Blocks

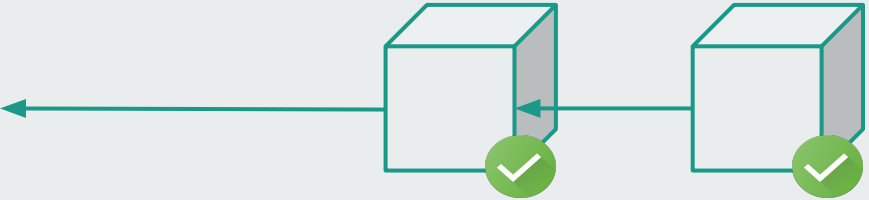


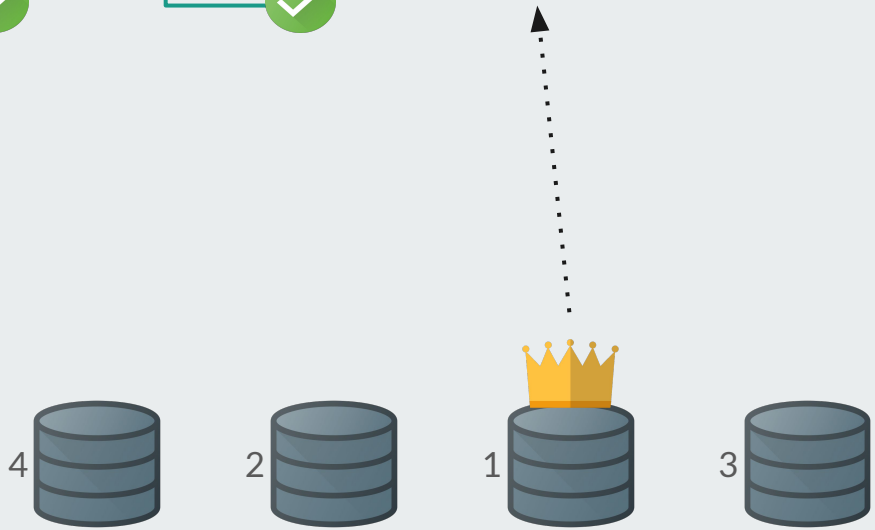


Banyan - Unlocking Blocks

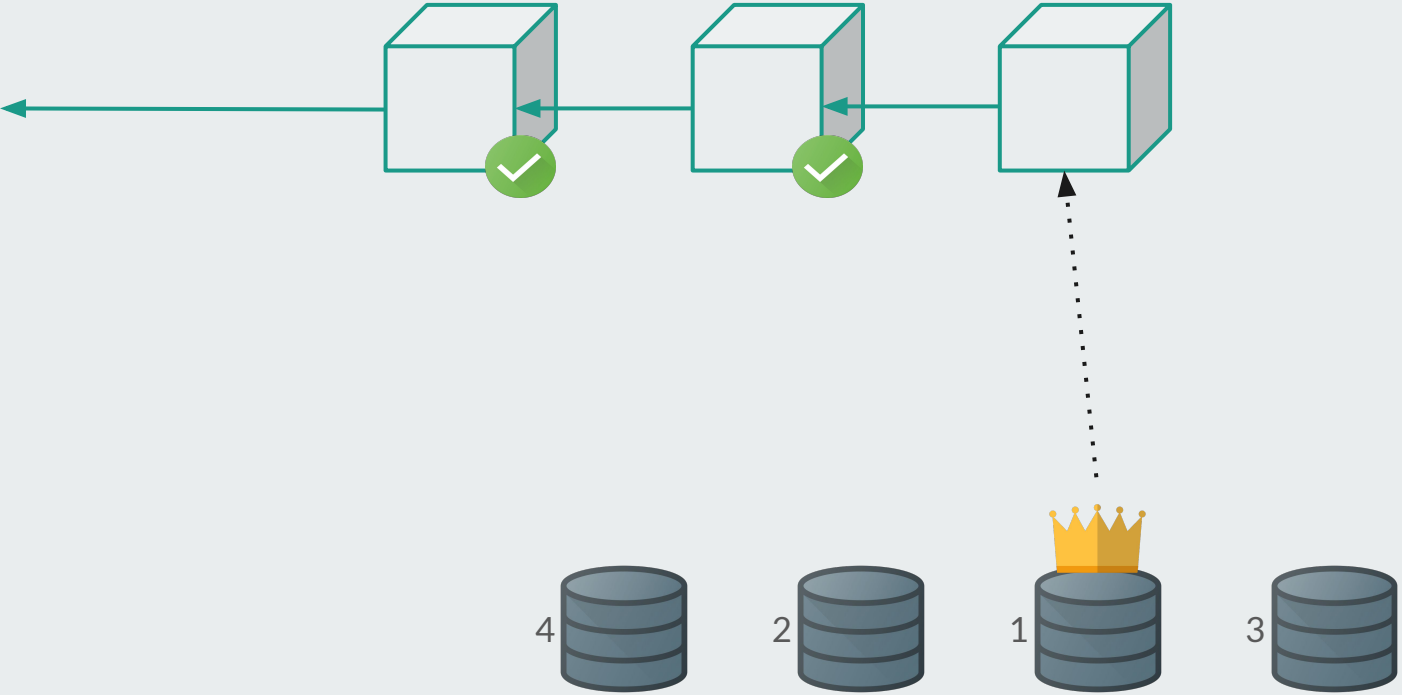


Banyan - Unlocking Blocks

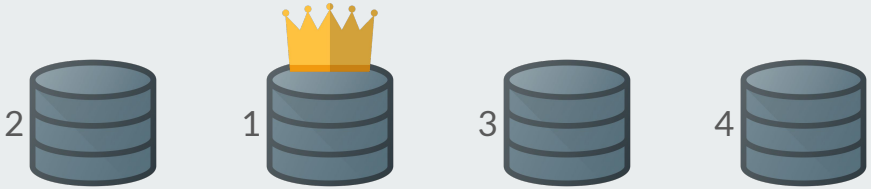




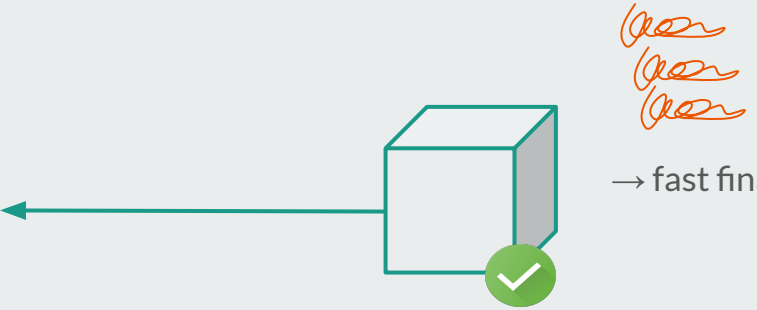
Banyan - Unlocking Blocks



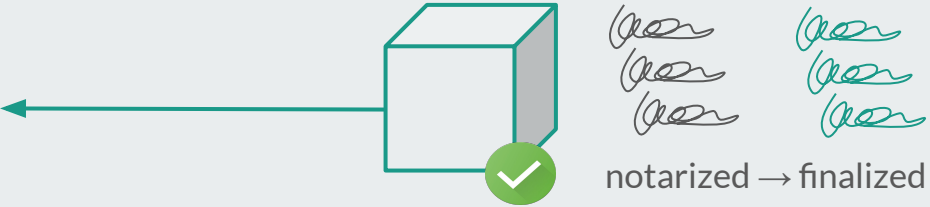
Unlocked Block: Will not conflict with fast path



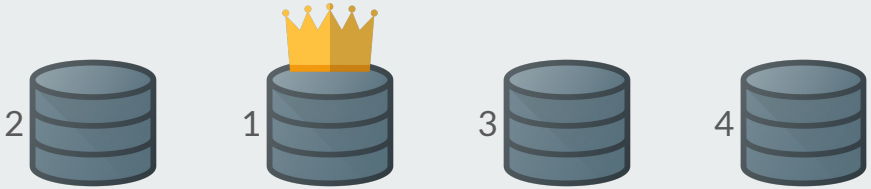
Correctness



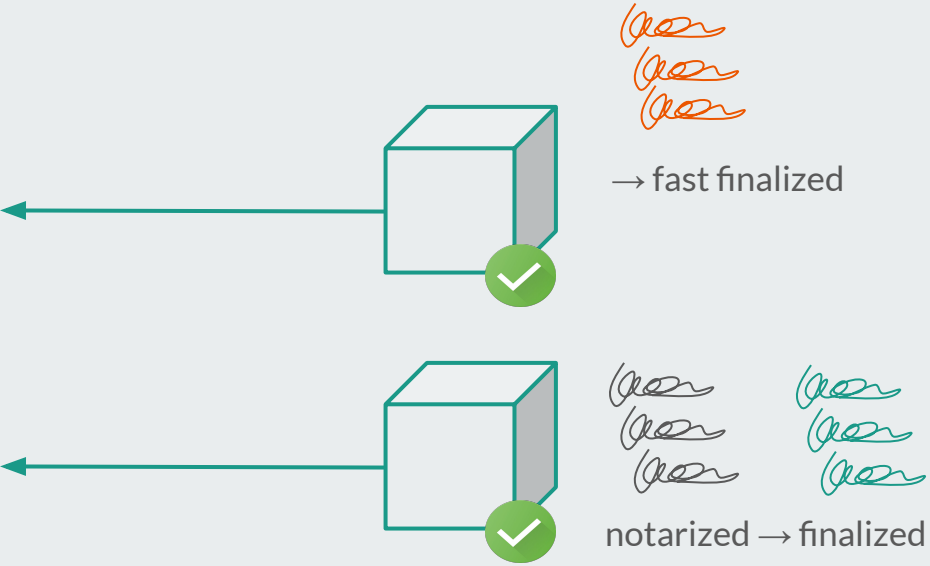
→ fast finalized

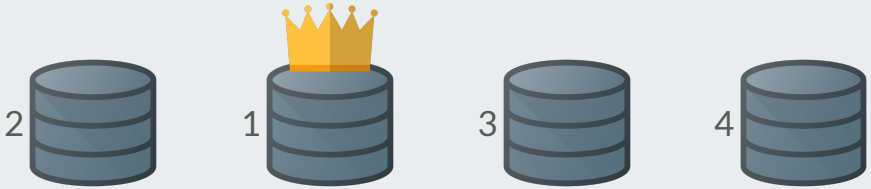


notarized → finalized

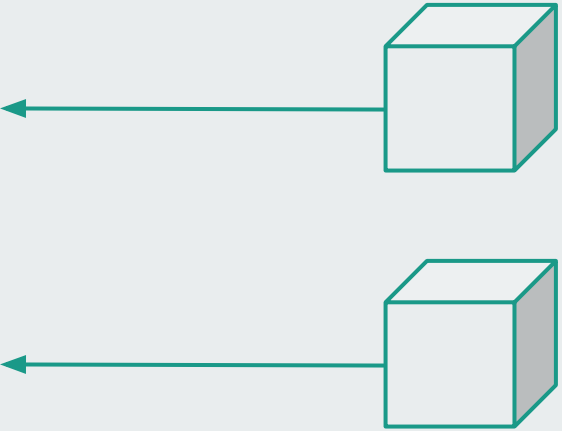


Correctness

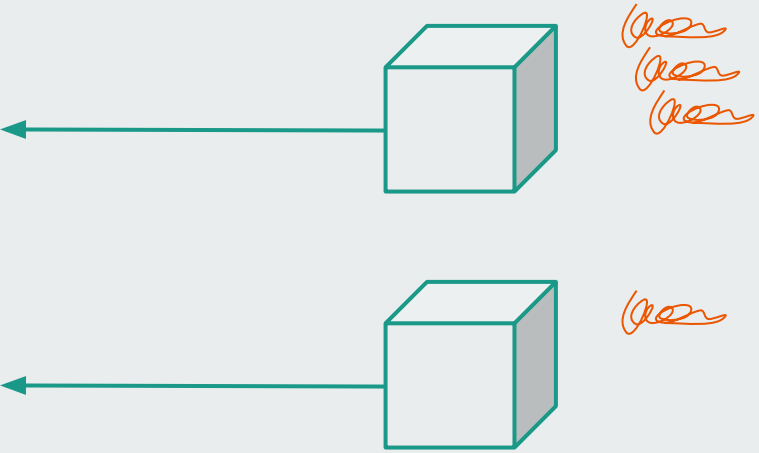




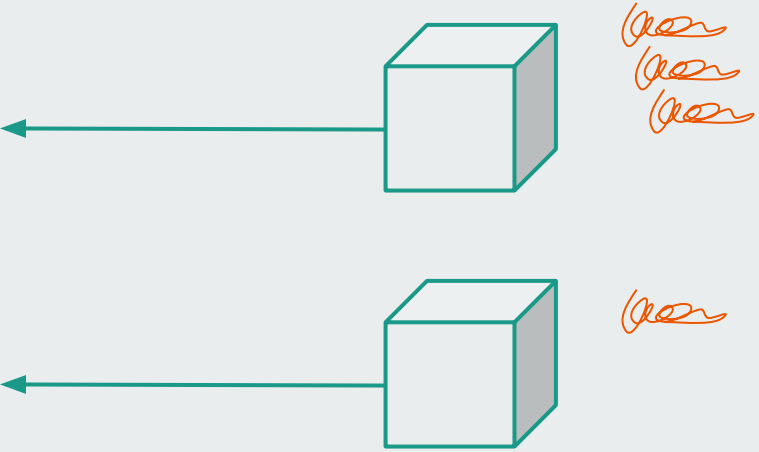
Unlocked Block - Case 1



Unlocked Block - Case 1



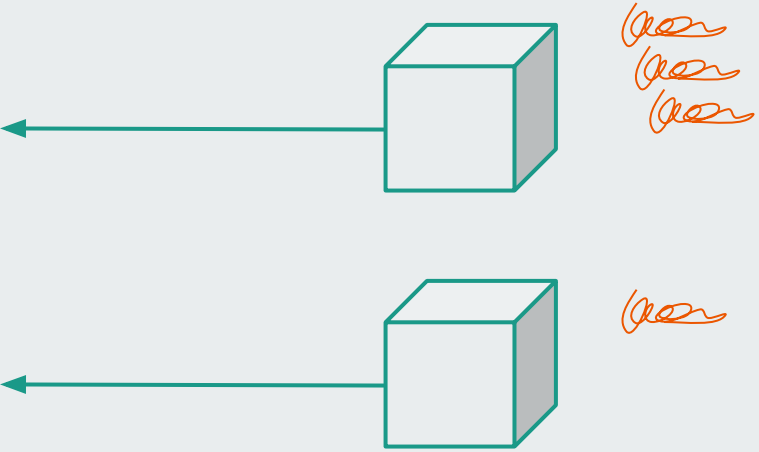
Unlocked Block - Case 1



number of *for block B* > ***f + p***

→ B unlocked

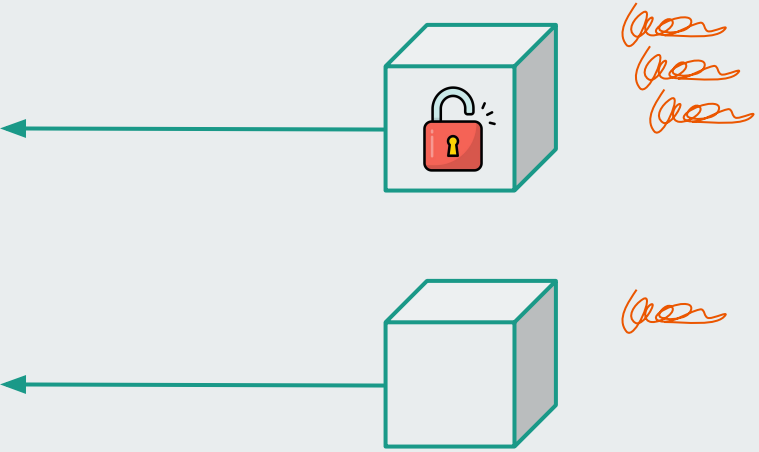
Unlocked Block - Case 1



number of *for block B*
+ number of *for non leader blocks* > ***f + p***

→ B unlocked

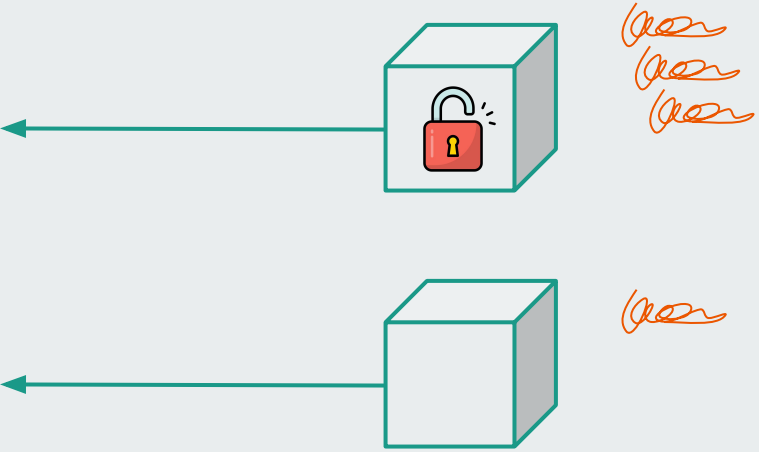
Unlocked Block - Case 1



number of *for block B*
+ number of *for non leader blocks* > ***f + p***

→ B unlocked

Unlocked Block - Case 1

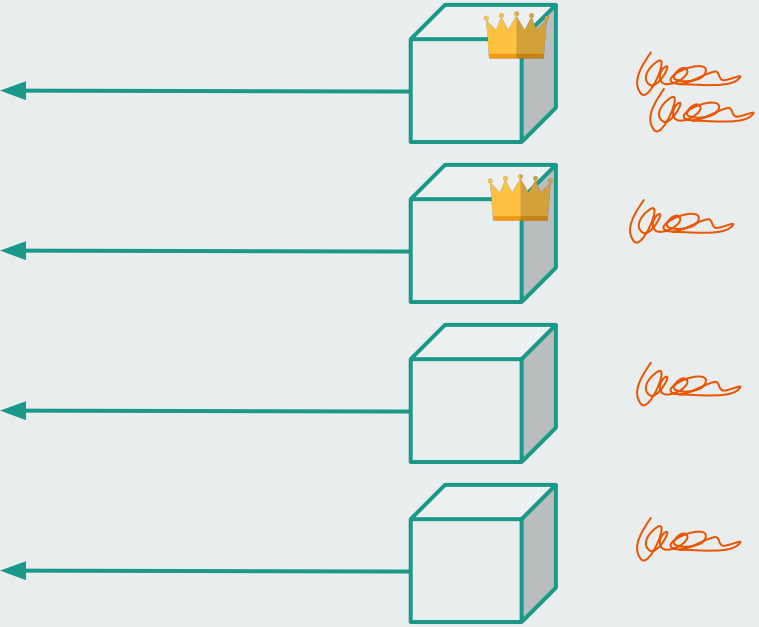


number of *for block B*
+ number of *for non leader blocks* > ***f + p***

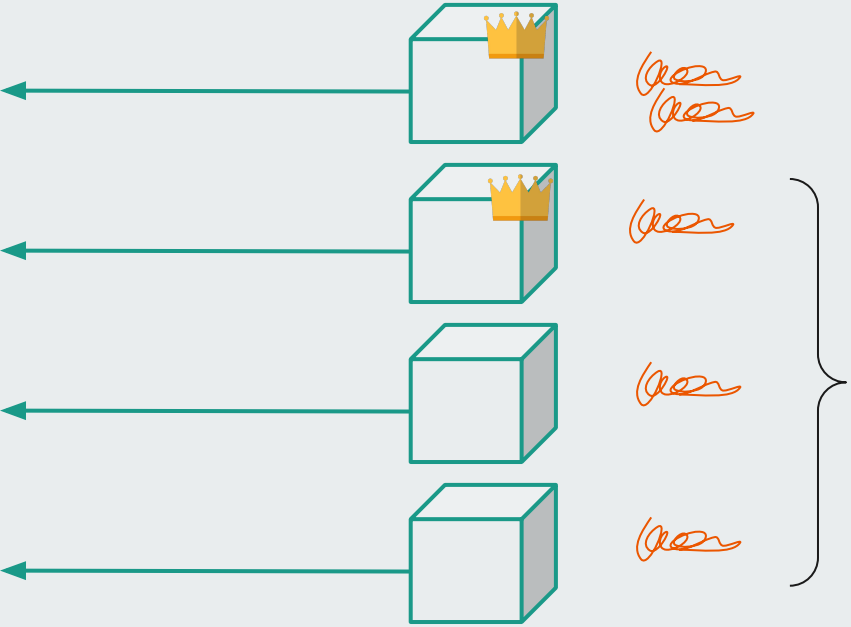
→ B unlocked

Can never be fast finalized.

Unlocked Block - Case 2



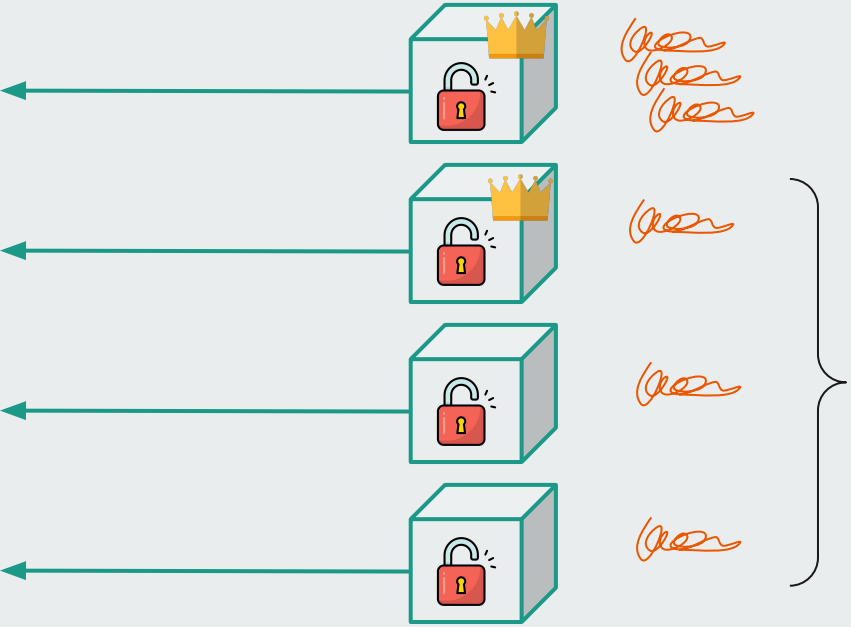
Unlocked Block - Case 2



number of *for non-max blocks* > ***f + p***

→ all unlocked

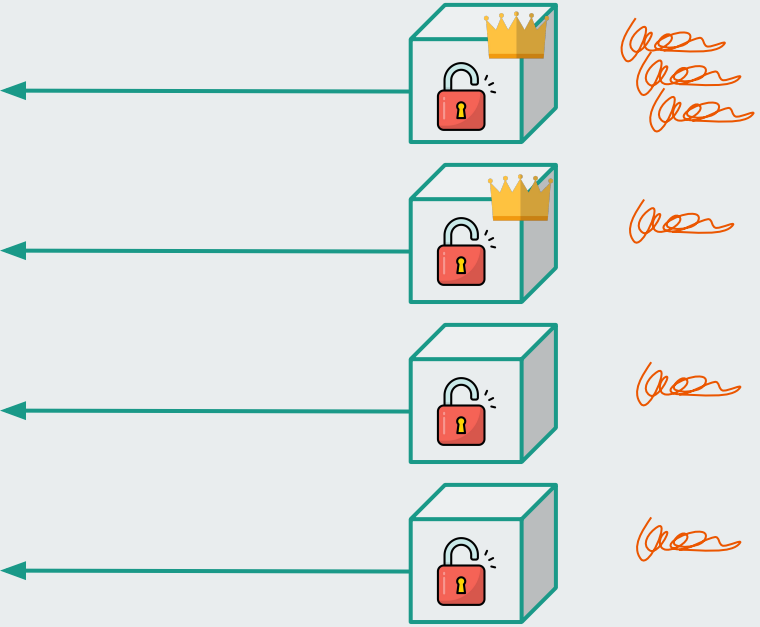
Unlocked Block - Case 2



number of *for non-max blocks* > ***f + p***

→ all unlocked

Unlocked Block - Case 2



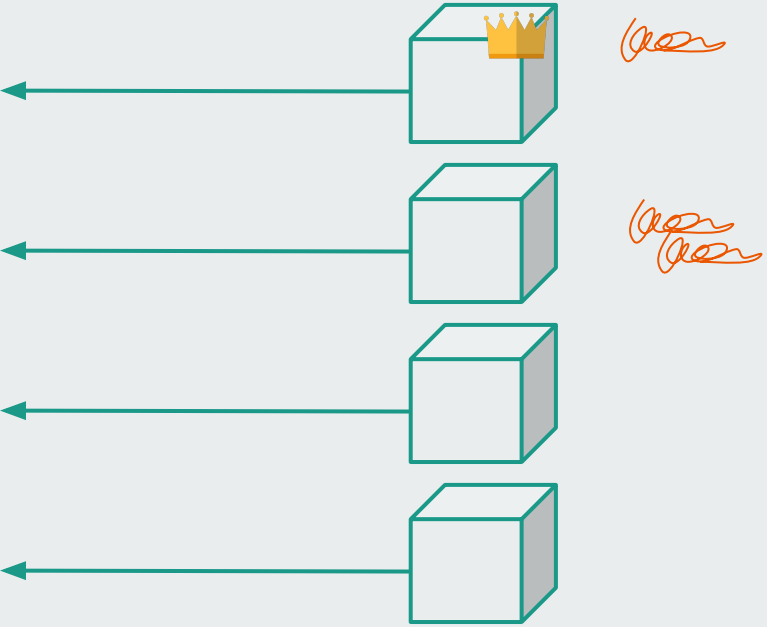
No block will be fast finalized.

number of *for non-max blocks* > ***f + p***

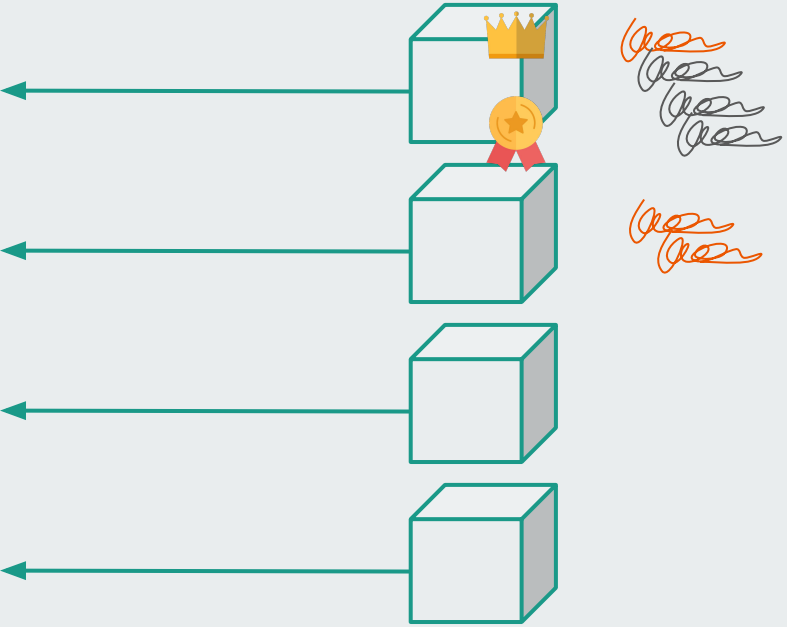
→ all unlocked

New Rule = Restriction ?

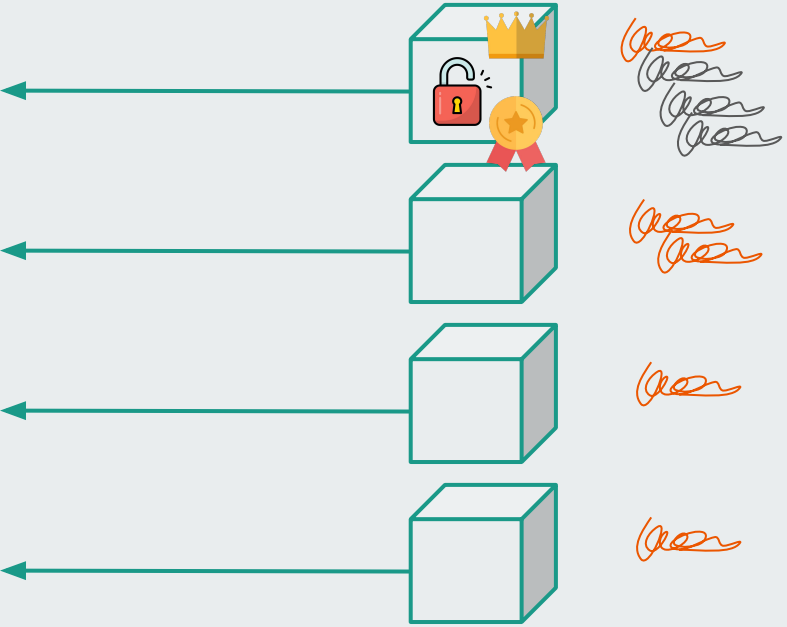
New Rule = Restriction ?



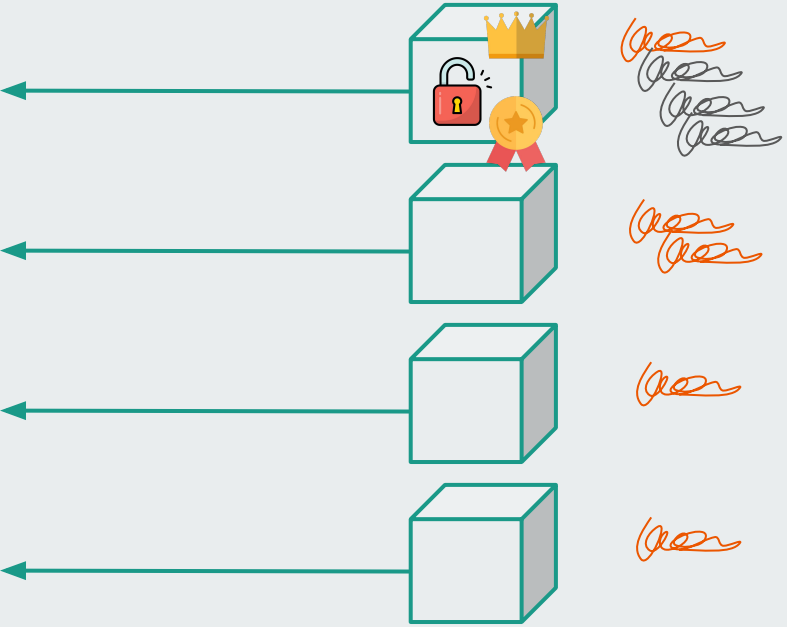
New Rule = Restriction ?



New Rule = Restriction ?



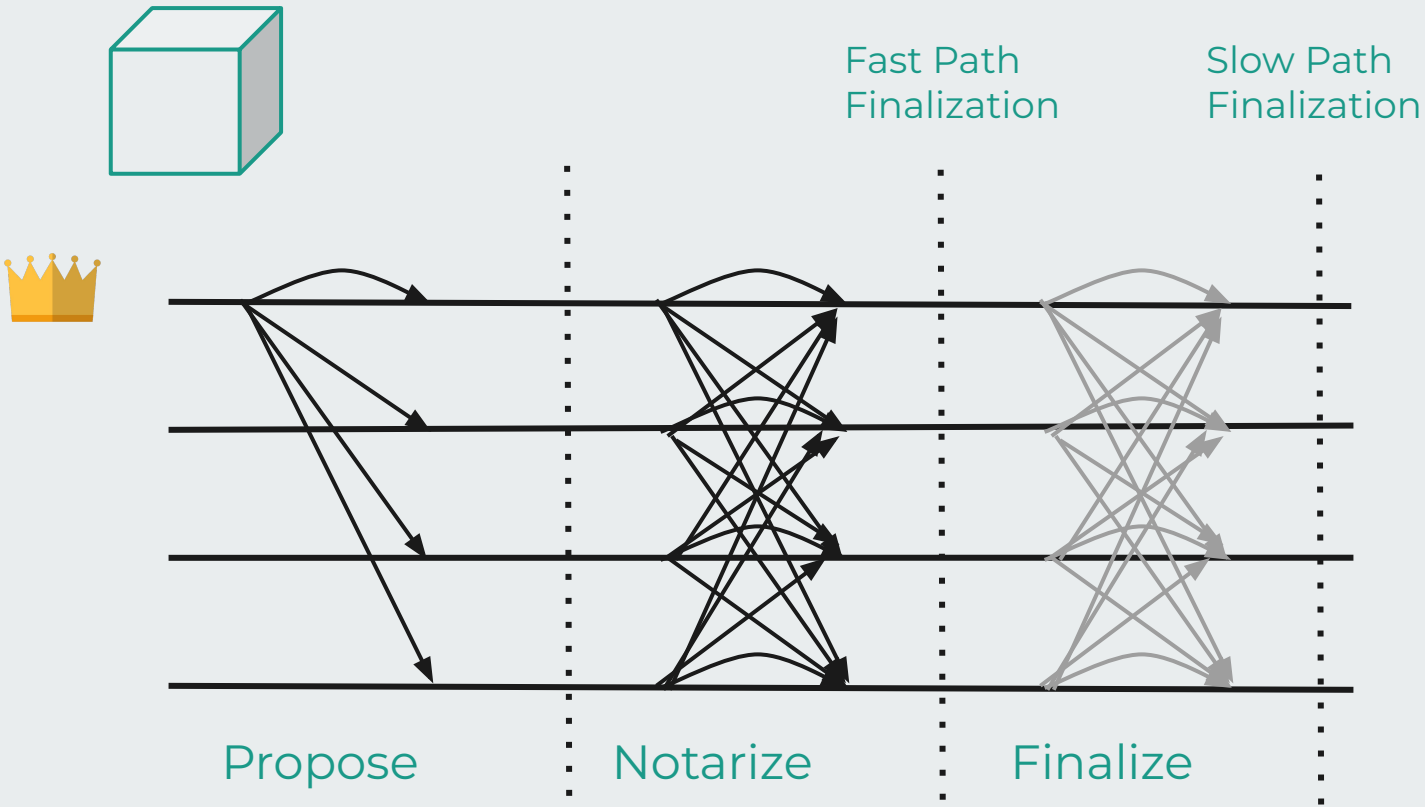
New Rule = Restriction ?



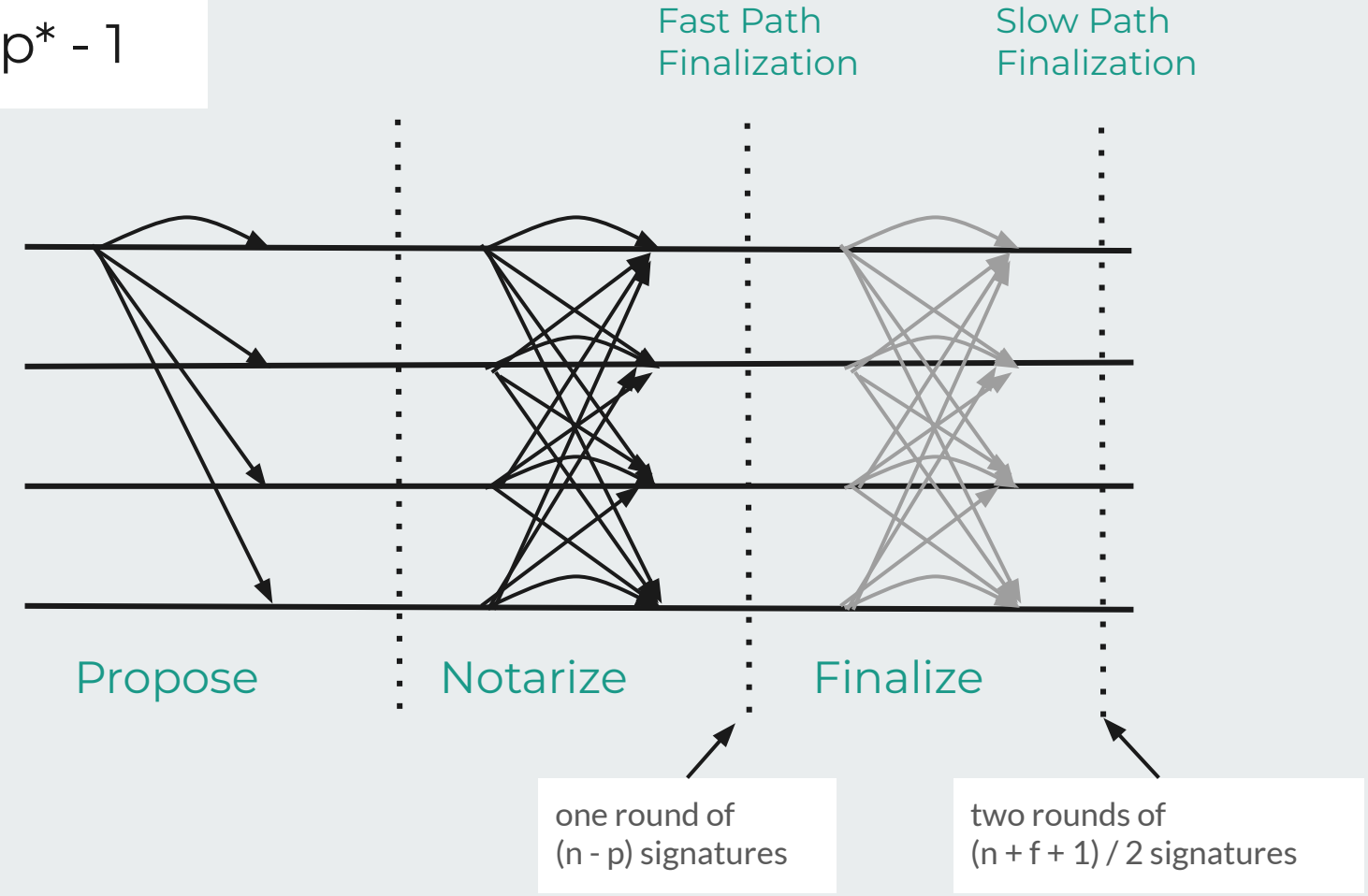
Worst case: waiting for

(n - f)

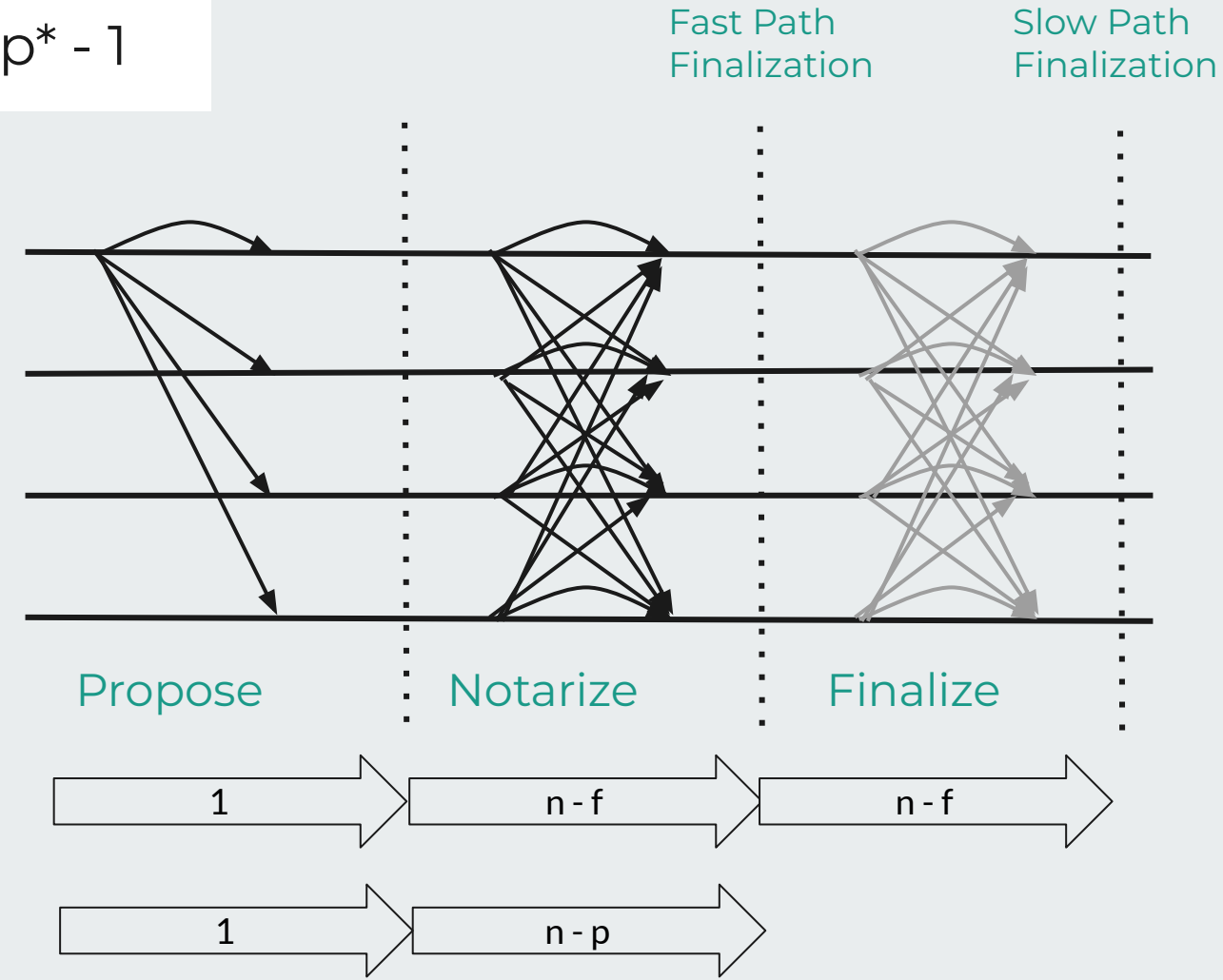
while a block is already notarized.


Fast Path
Finalization
Slow Path
Finalization
Propose
Notarize
Finalize

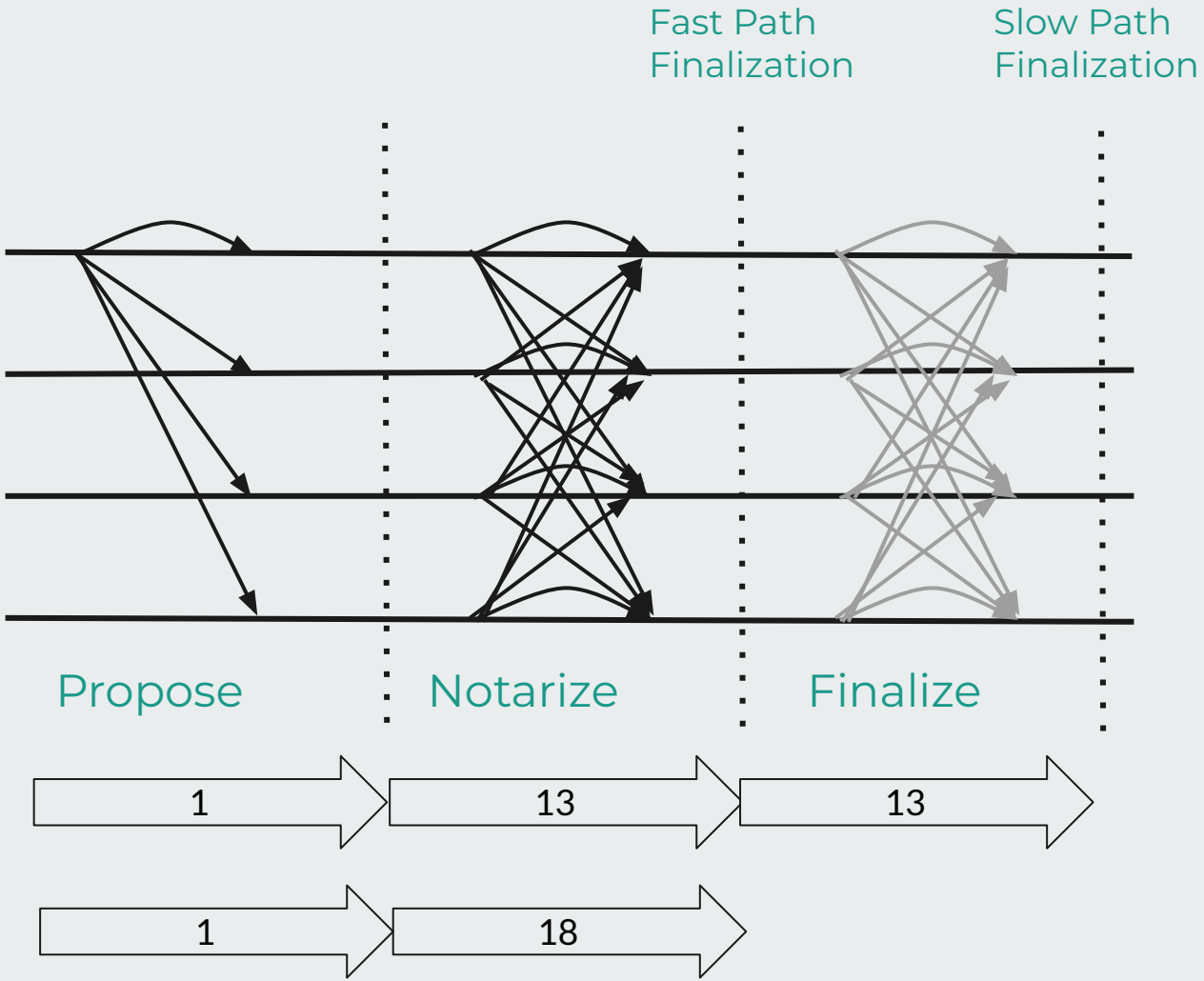
$n \geq 3f + 2p^* - 1$


Fast Path Finalization
Slow Path Finalization
Propose
Notarize
Finalize
one round of (n - p) signatures
two rounds of (n + f + 1) / 2 signatures

$n \geq 3f + 2p^* - 1$


Fast Path Finalization
Slow Path Finalization
Propose
Notarize
Finalize
1
n - f
n - f
1
n - p

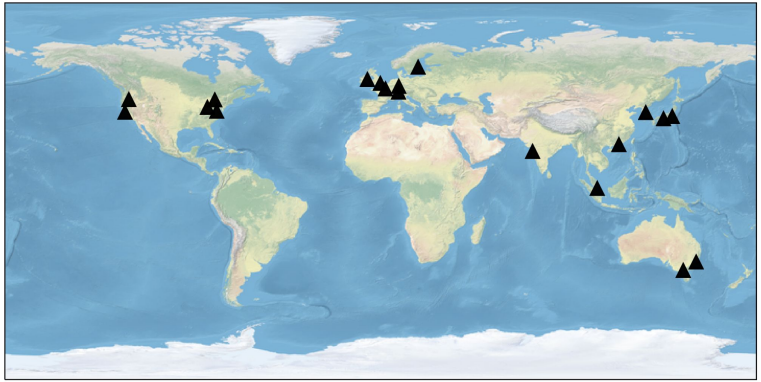
f = 6
n = 19
p = 1


Fast Path Finalization
Slow Path Finalization
Propose
Notarize
Finalize
1
13
13
1
18

n = 19 f = 6

AWS t3.large

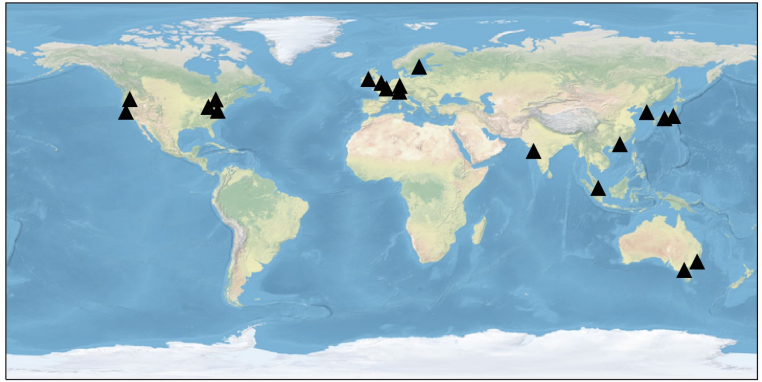
proposer latency

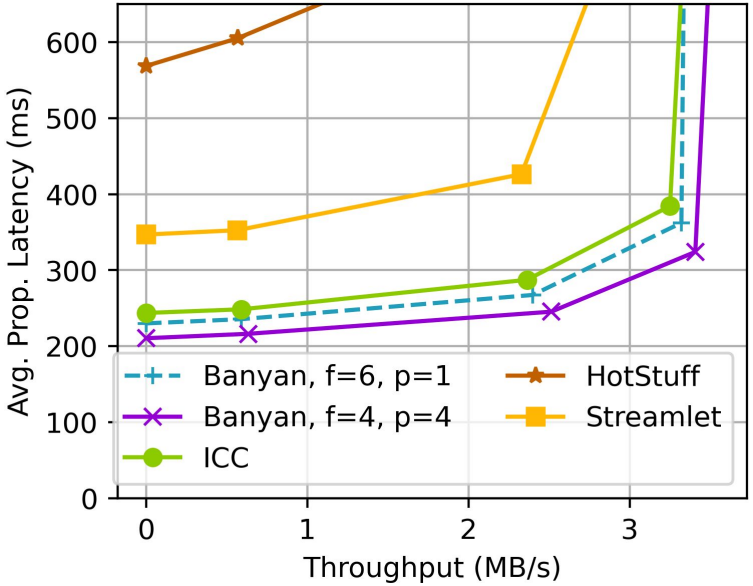


n = 19 f = 6

AWS t3.large

proposer latency



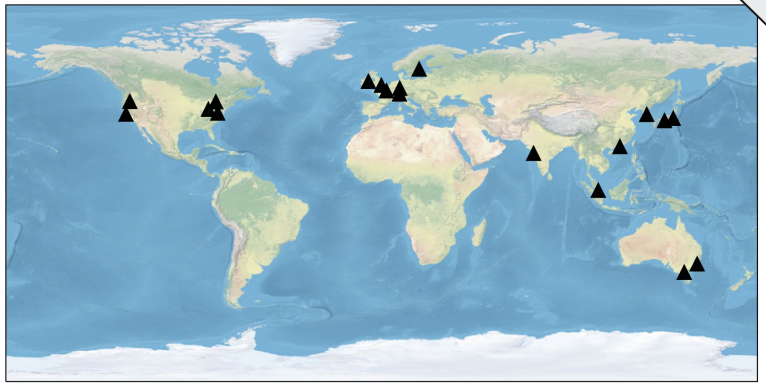


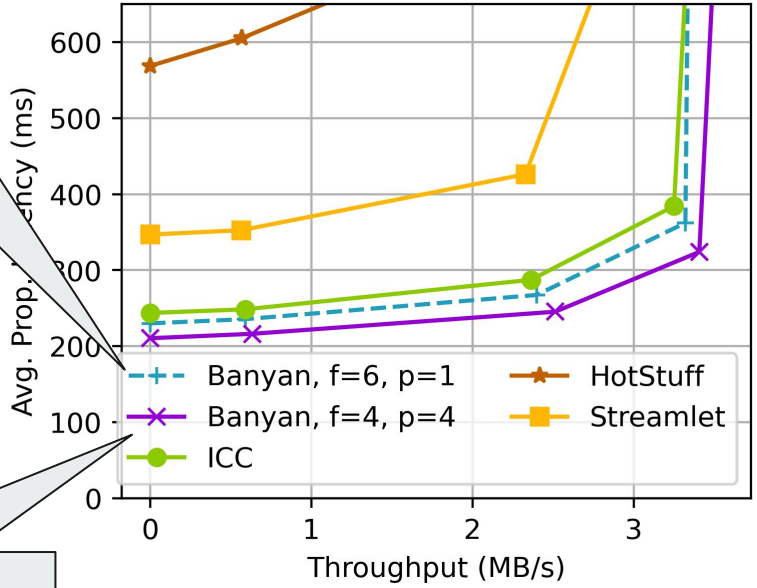
n = 19 f = 6

AWS t3.large

proposer latency

5.81% improvement



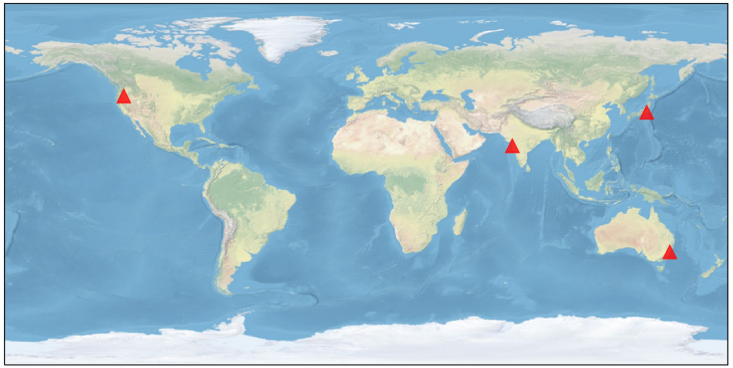


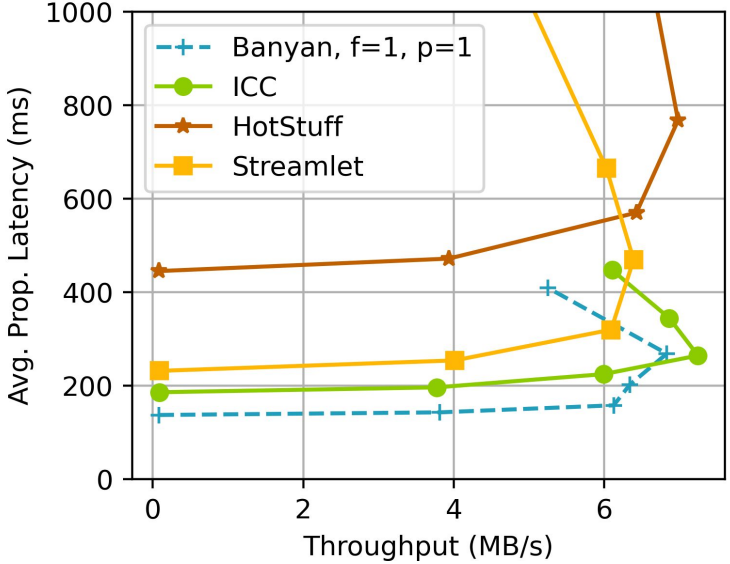
16.0% improvement

n = 19 f = 6

AWS t3.large

proposer latency

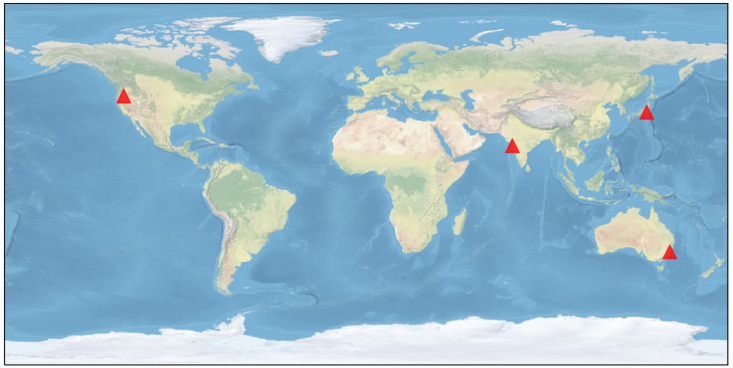


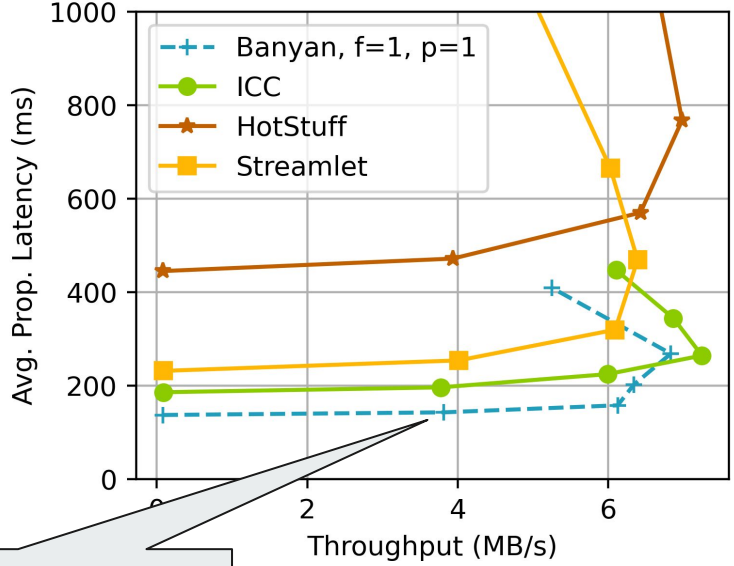


n = 4 f = 1

AWS t3.large

proposer latency



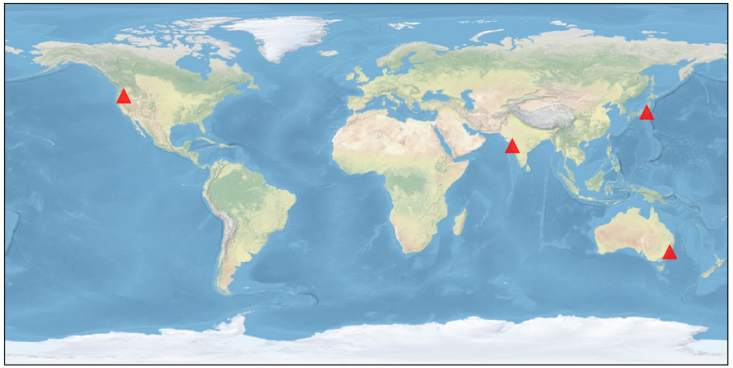


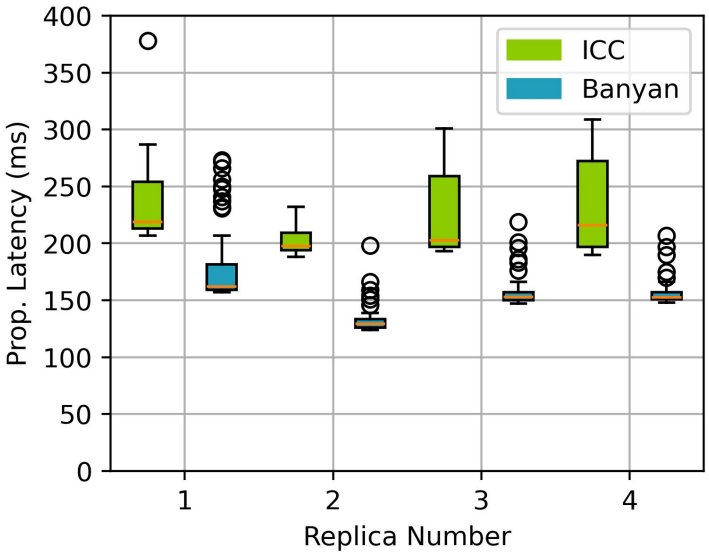
29.9% improvement

n = 4 f = 1

AWS t3.large

proposer latency

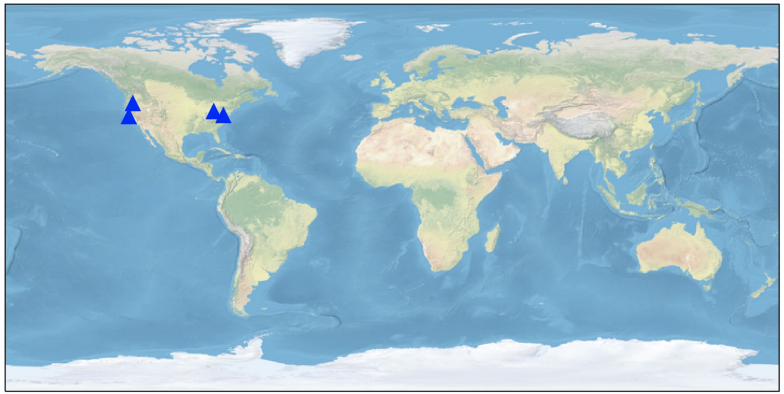


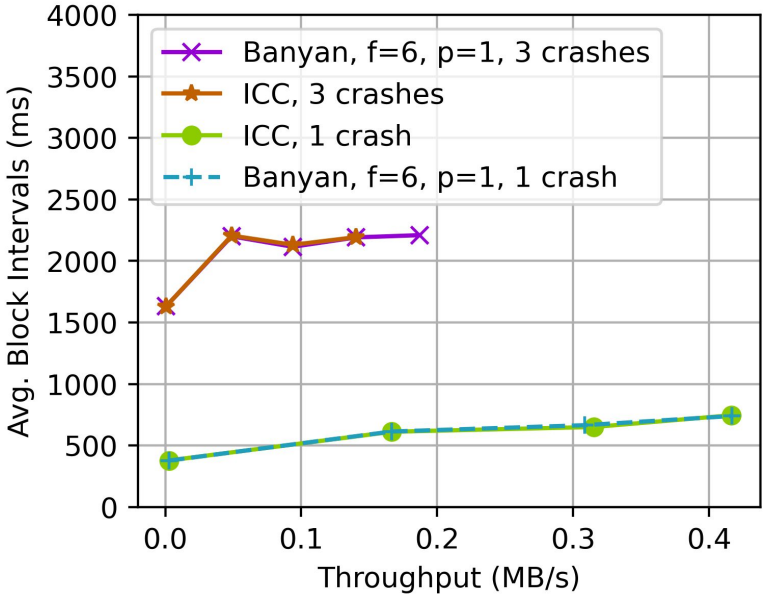


n = 4 f = 1

AWS t3.large

proposer latency





n = 19 f = 6

AWS t3.large

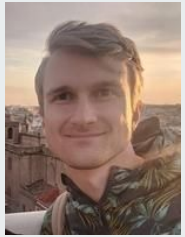
proposer latency

Main Takeaways

1. Banyan is faster than state-of-the-art (*optimistically*)

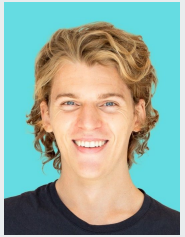
2. Banyan is never slower than ICC

Thank you !



Yann Vonlanthen
ETH Zurich

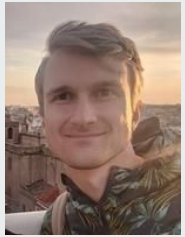
Dr. Jakub Sliwinski
ETH Zurich



Massimo Albarello
ETH Zurich

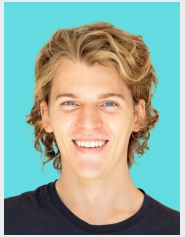
Prof. Roger Wattenhofer
ETH Zurich

Thank you !



Yann Vonlanthen
ETH Zurich

Dr. Jakub Sliwinski
ETH Zurich



Massimo Albarello
ETH Zurich

Prof. Roger Wattenhofer
ETH Zurich