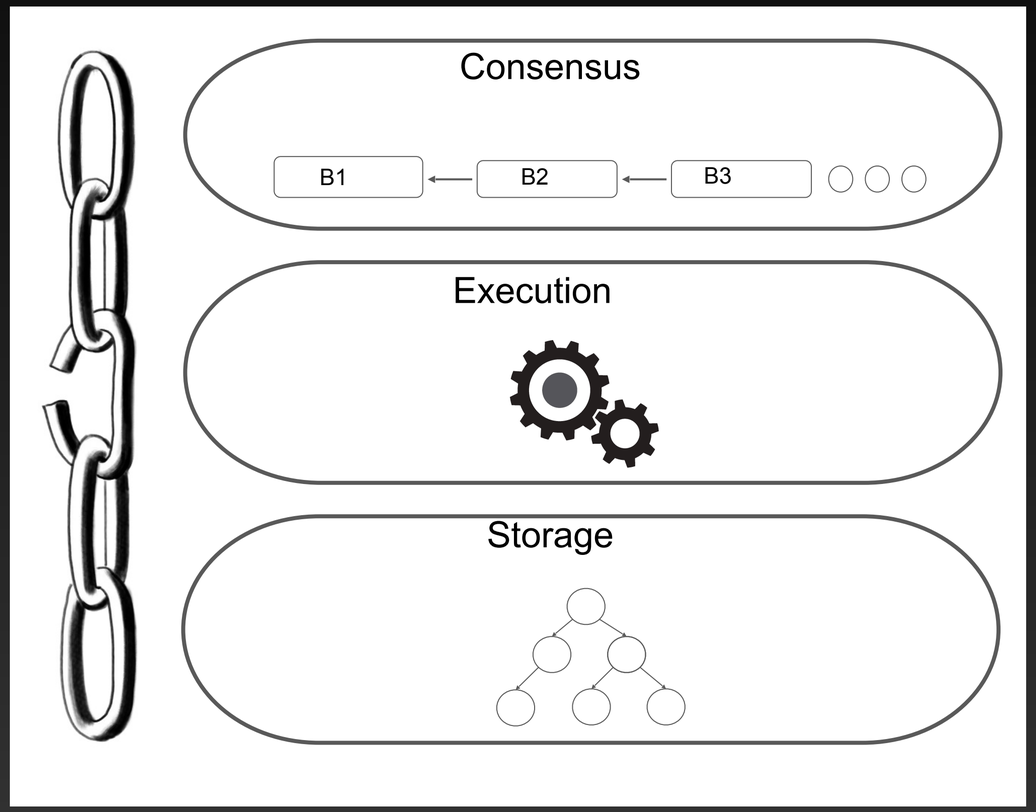
DAG Meets BFT

Theory Meets Practice

Alexander Spiegelman

Scaling Blockchains



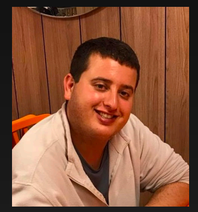
Joint Work

George Danezis

Neil Giridharan

Idit Keidar

Lefteris Kokoris-Kogias



Oded Naor

Alberto Sonnino

Alexander Spiegelman

Our DAG-Based BFT Approach

All You Need Is DAG (DAG-Rider)

PODC 2021

Narwhal&Tusk

EuroSys (Best paper) 2022

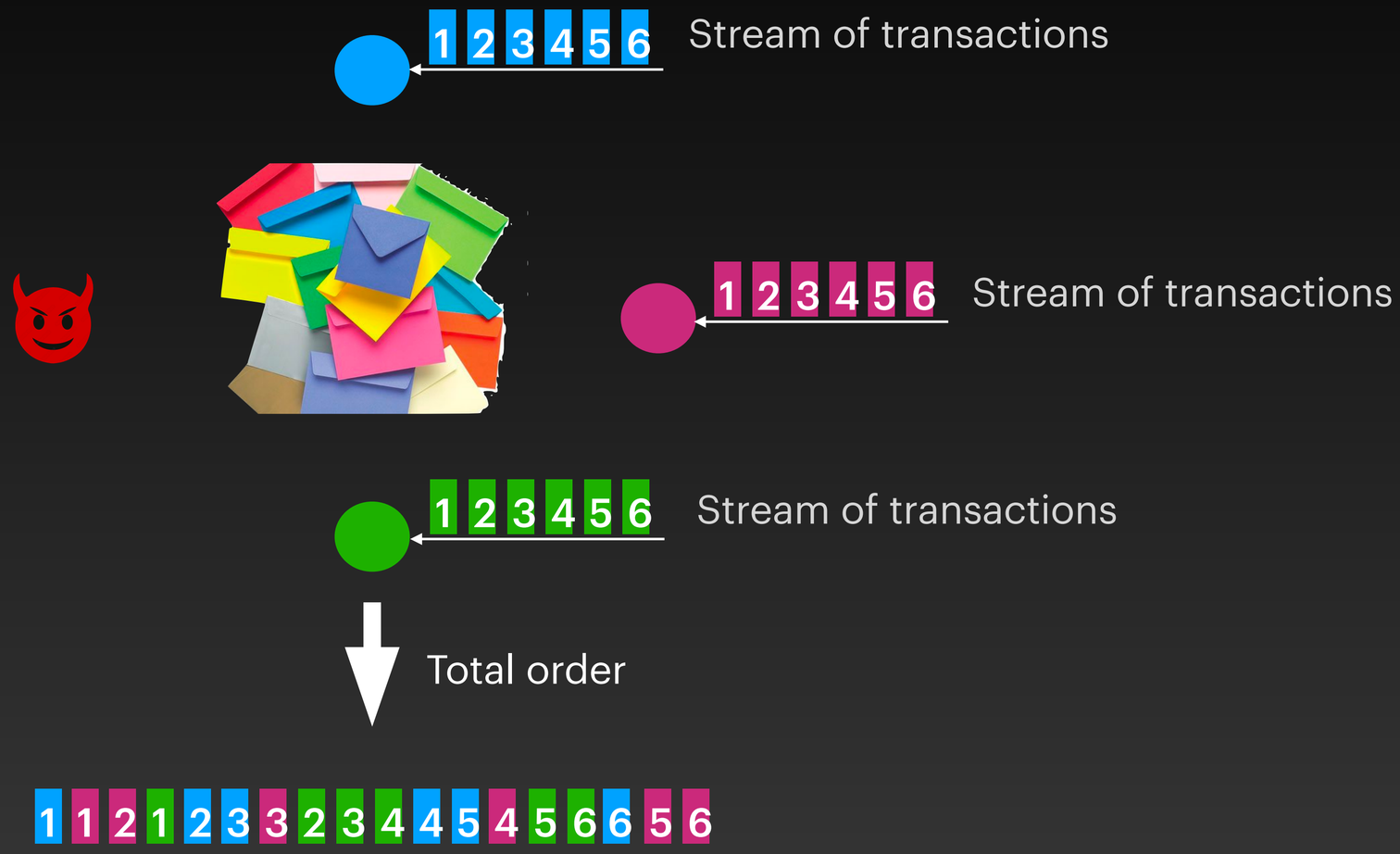
Bullshark

CCS 2022

Being implemented by several Blockchain companies, e.g., Aptos, Celo, Mysten Labs, and Somelier.

The Goal

n validators
f < n/3 Byzantine



The Challenge

- Performance
  - Throughput
  - Latency
- Scalability
  - Number of validators
- Simplicity
  - Easy to deploy, maintain, debug

The Agenda

- Performance
  - Throughput
  - Latency
- Scalability
  - Number of validators
- Simplicity
  - Easy to deploy, maintain, debug

Narwhal

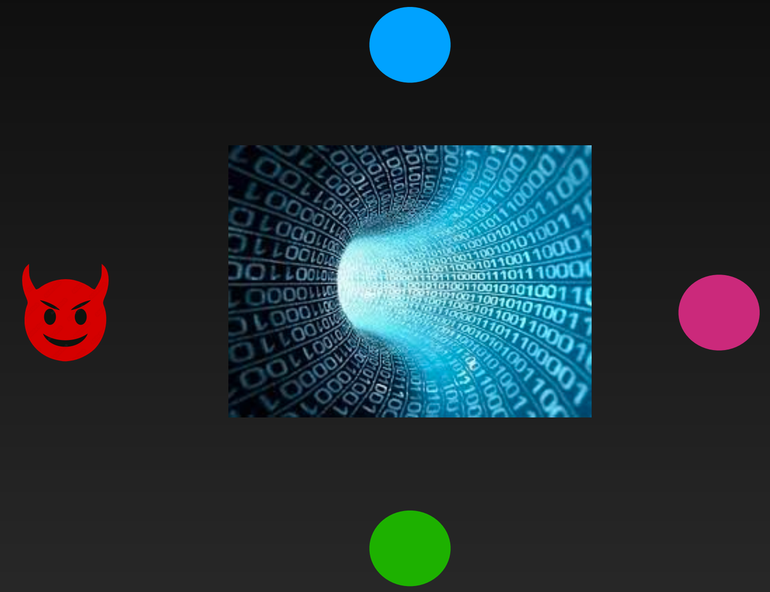
Bullshark

Scaling Consensus

Current Designs

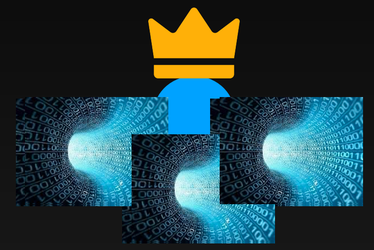
Optimize overall message complexity of the consensus protocol

Current Designs



Data has to be shared

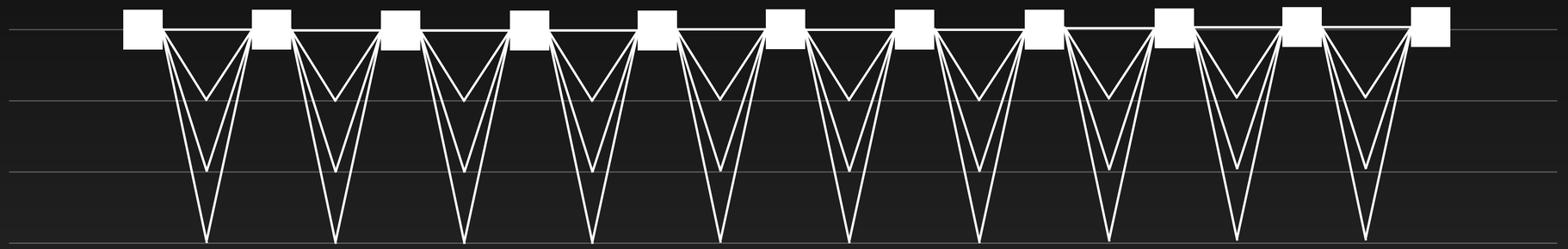
Current Designs

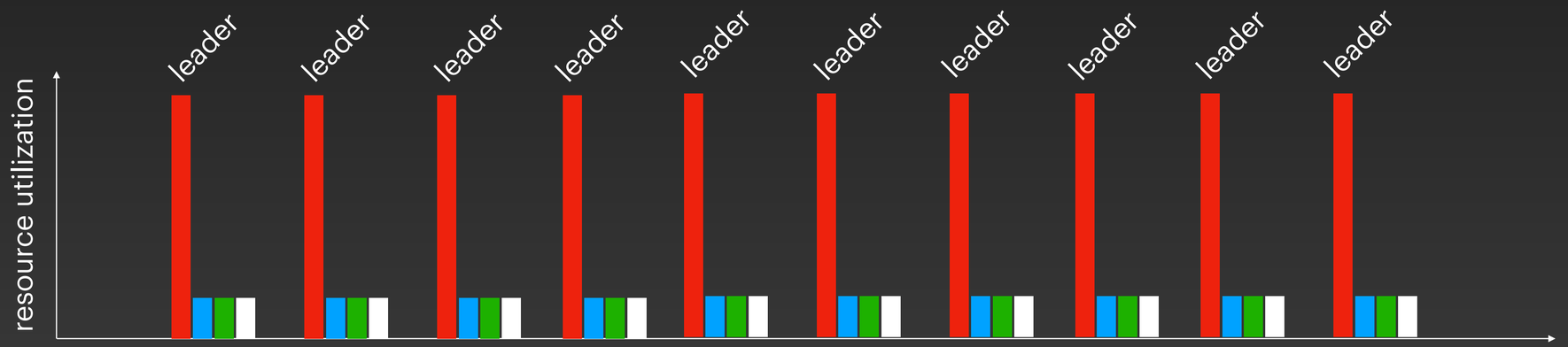


Monolithic protocol sharing transaction data as part of the consensus

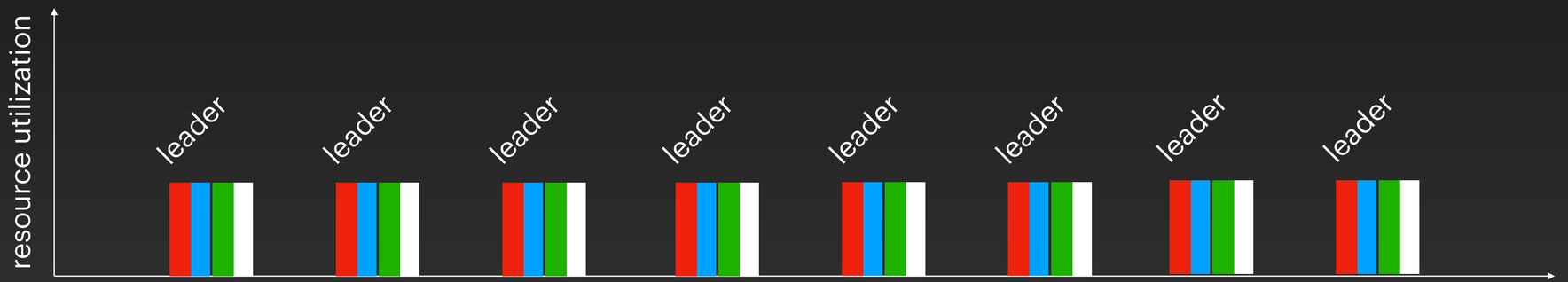
Current Designs

Typical leader-based protocols





Load Balance



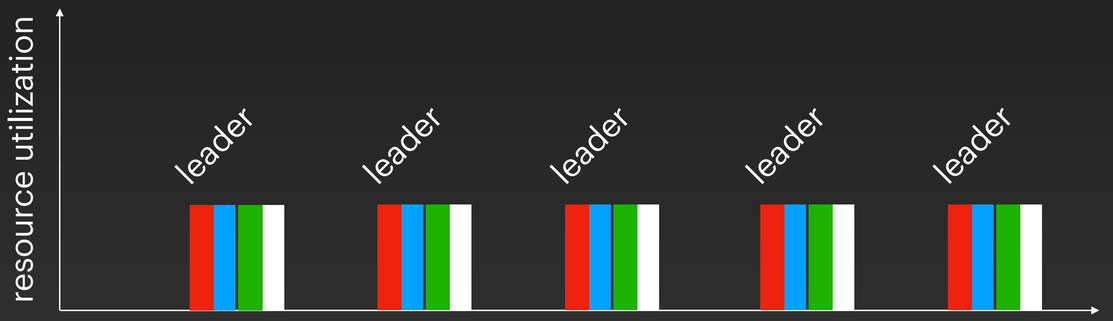
Decoupling data dissemination from metadata ordering is the key to performance

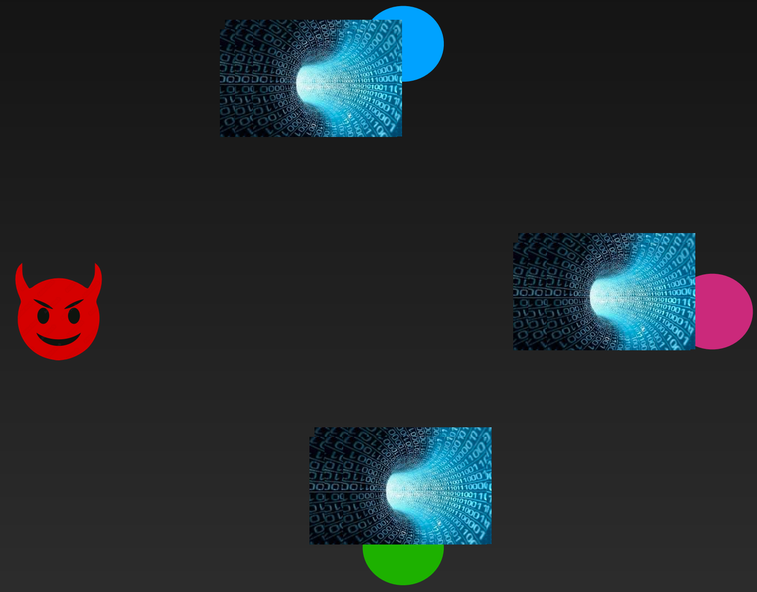
Data Dissemination

Metadata Ordering

Data Dissemination

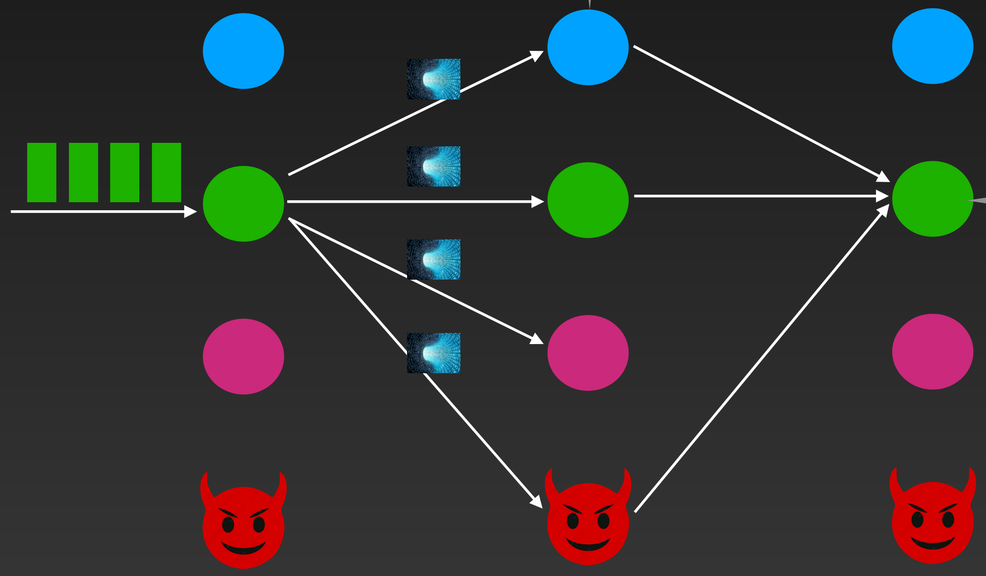
Each validator asynchronously disseminates its data





Proof of Availability

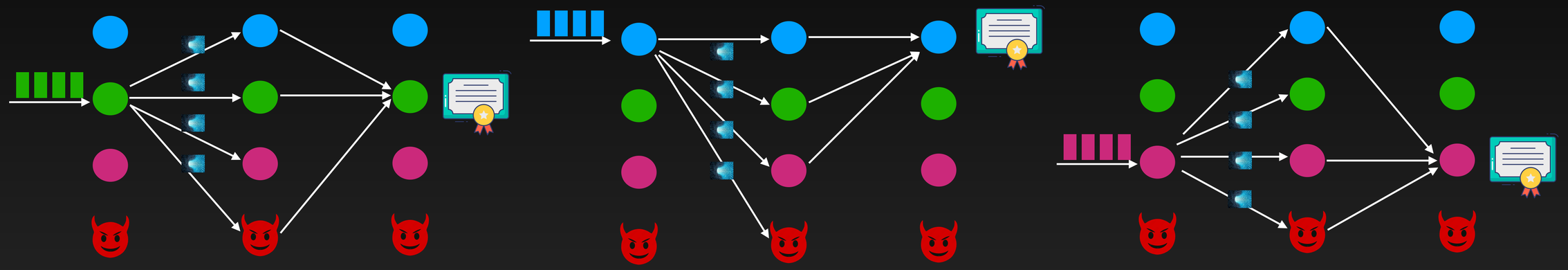
Persist data and send back a signature

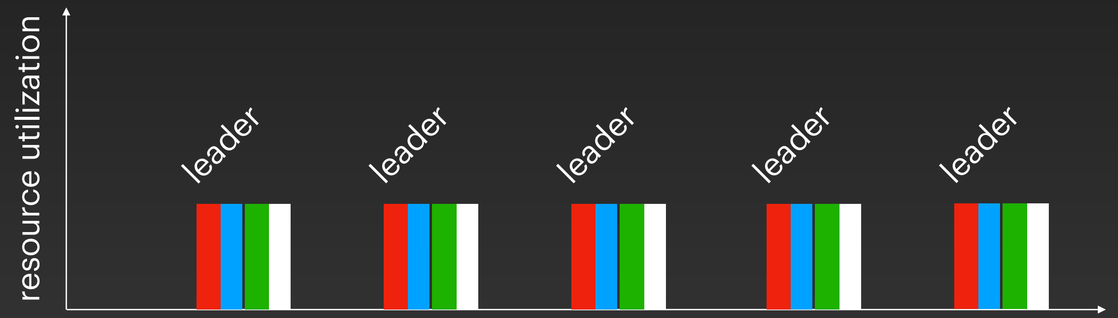


Collect quorum certificate

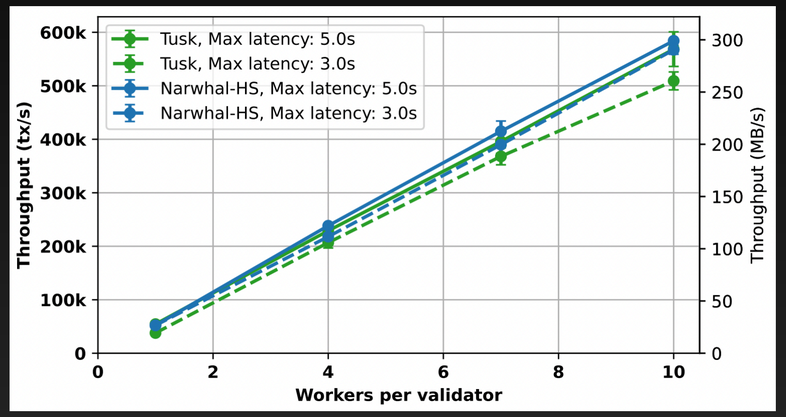
Data can be retrieved before execution

Proof of Availability



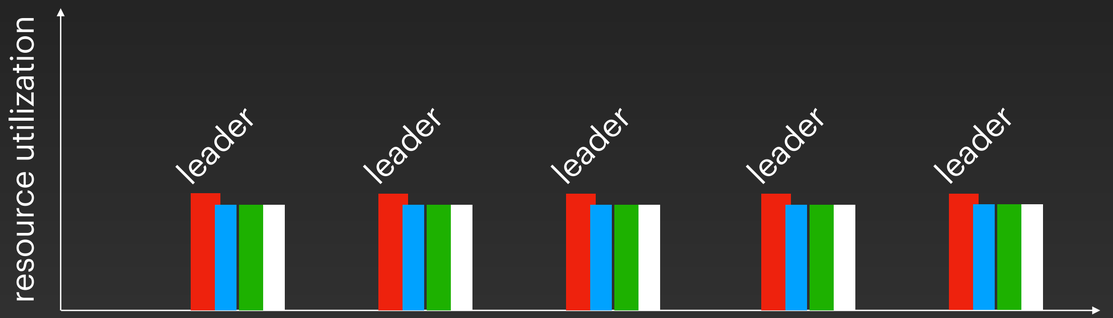


Scaling Out


Tusk, Max latency: 5.0s
Tusk, Max latency: 3.0s
Narwhal-HS, Max latency: 5.0s
Narwhal-HS, Max latency: 3.0s
Throughput (tx/s)
Throughput (MB/s)
Workers per validator
0
100k
200k
300k
400k
500k
600k
0
50
100
150
200
250
300
0
2
4
6
8
10

Metadata Ordering

- The leader is no longer a major bottleneck
- Consensus communication complexity is much less important
- Use low latency protocol

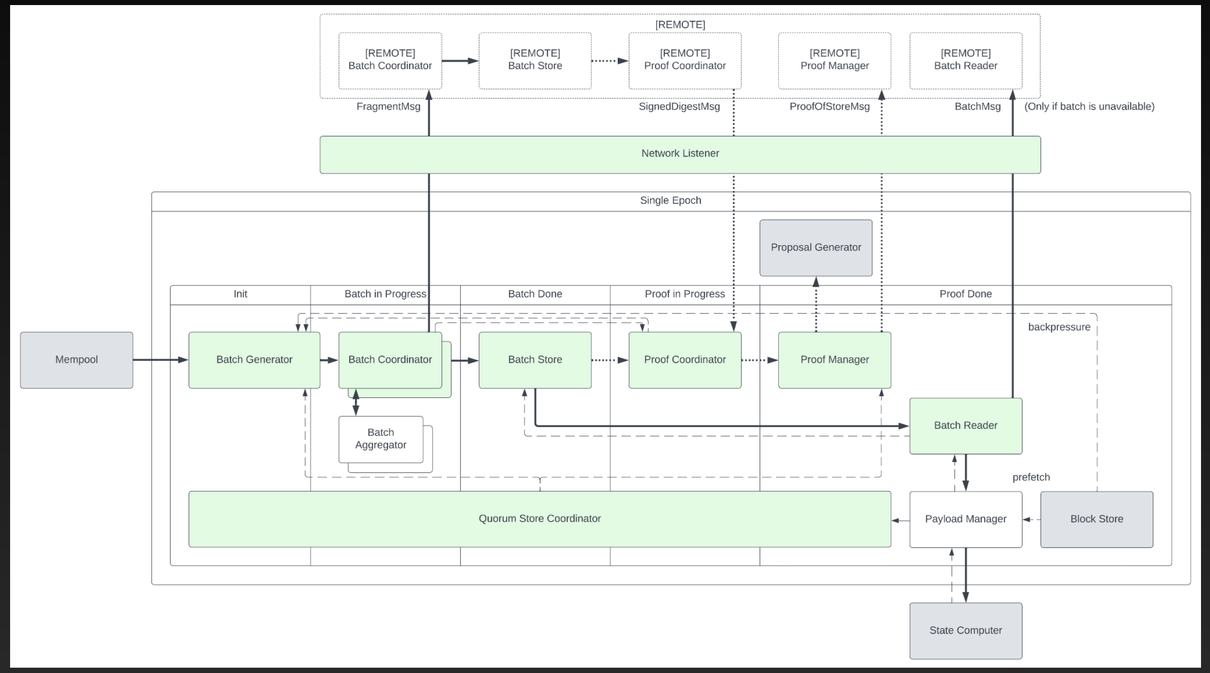


Quorum Store

- The Aptos implementation of Narwhal
- Aptos uses Jolteon for consensus
  - *Jolteon and Ditto: Network-Adaptive Efficient Consensus with Asynchronous Fallback* [GKSSX]
- Over 100 validators on 3-regions setup
- 10x performance gain for consensus only test
- Latency was (surprisingly) reduced!

Jolteon

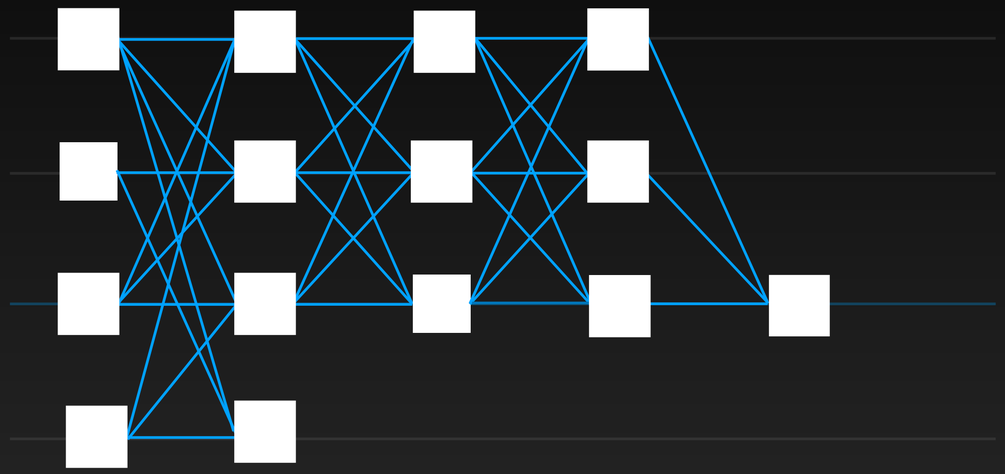
Production Code Is Complex

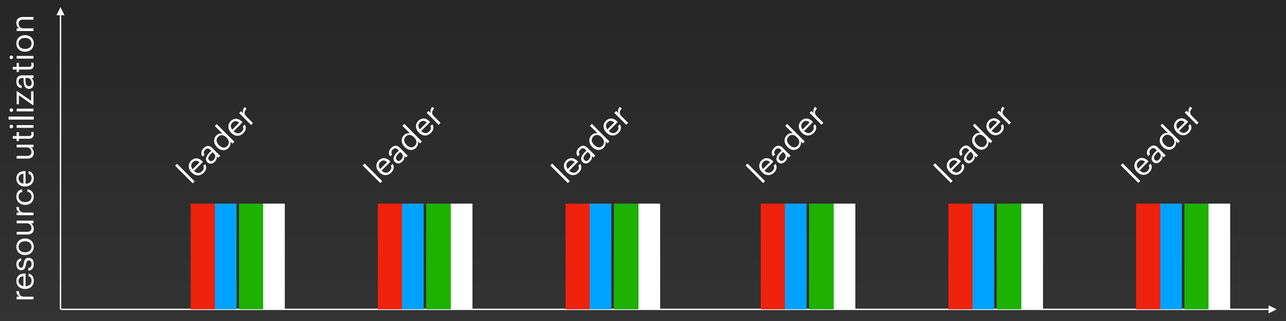


- Storage, cache, quota, expiration and garbage collection, state sync, network prioritization, back pressure, etc
- Maintain and debug

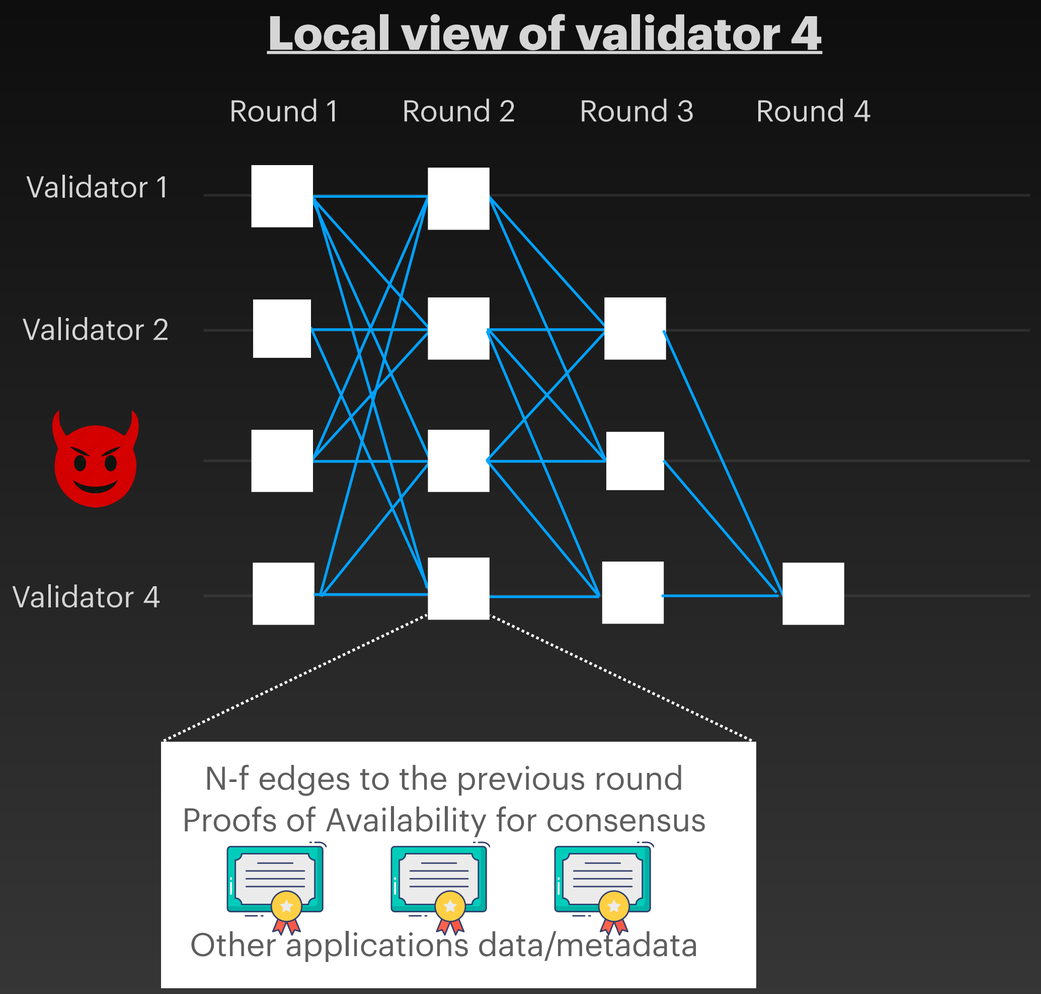
DAG Meets BFT

- Network abstraction
- Simple consensus
  - No view-change, view-synchronization
  - Way to switch between partial synchrony and asynchrony
- Building one system
  - Piggyback more protocols (e.g., ADKG, MPC)
- Perfect load balance
  - Leaderless





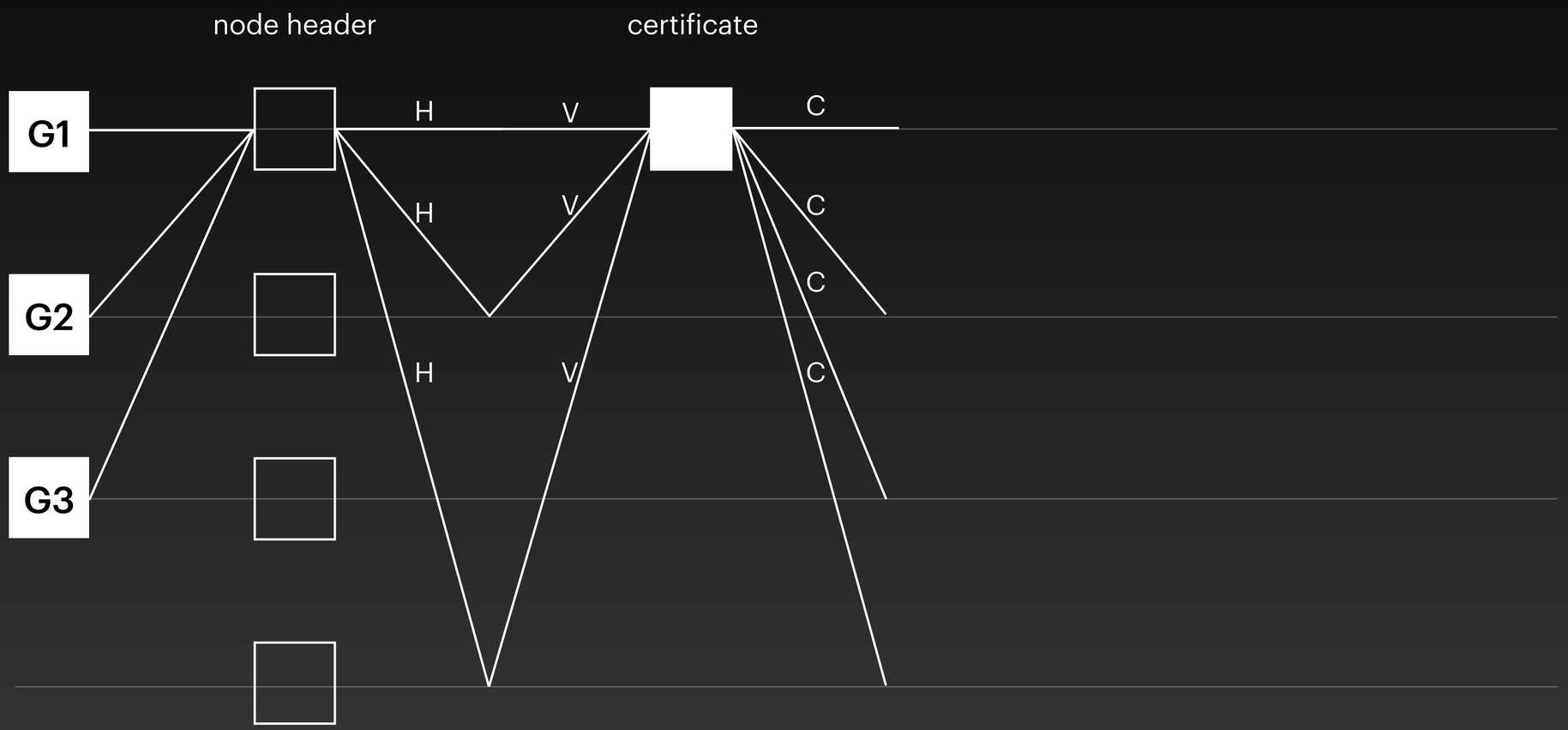
Building a DAG



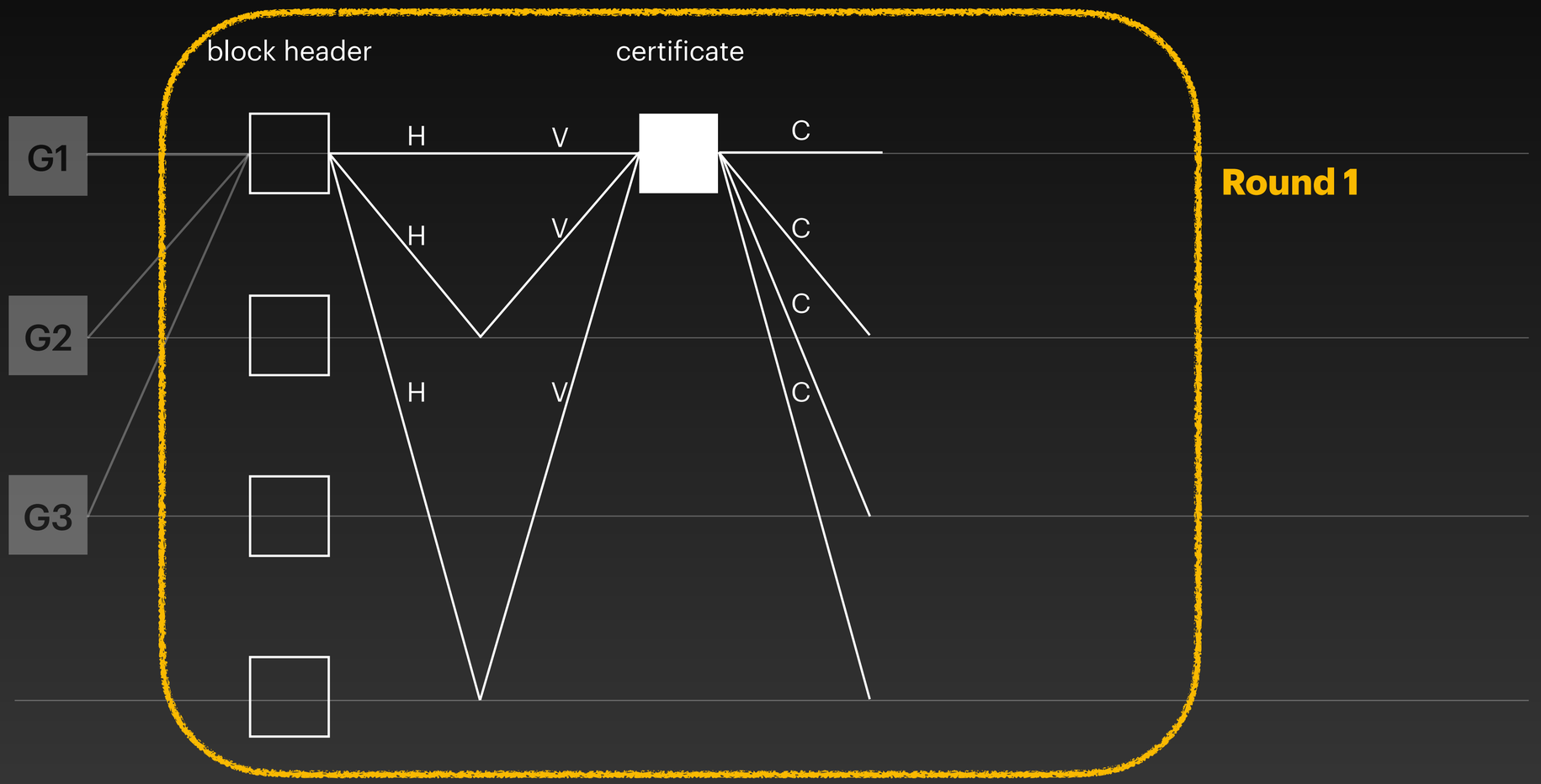
Challenges

- Reliable communication
  - Need to deliver n-f nodes to advance rounds
- Causal history
- Non-equivocation

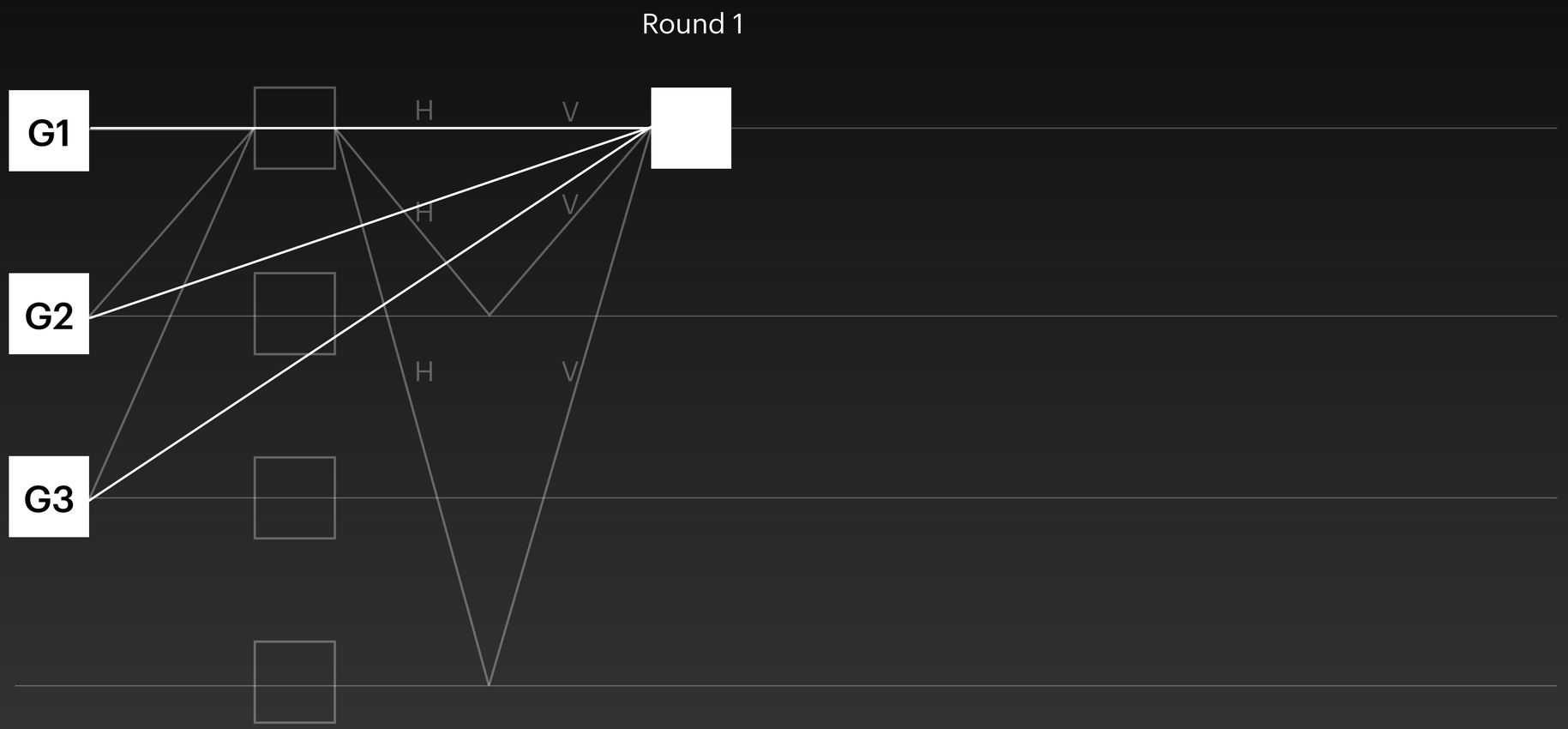
Non-Equivocation



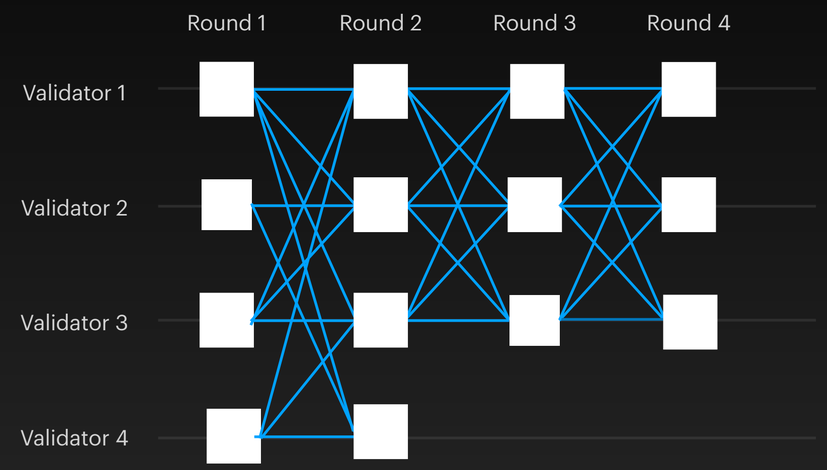
Non-Equivocation



Non-Equivocation



Zero Communication Overhead Consensus



Total order

Aleph

DAG-Rider

Tusk

Bullshark

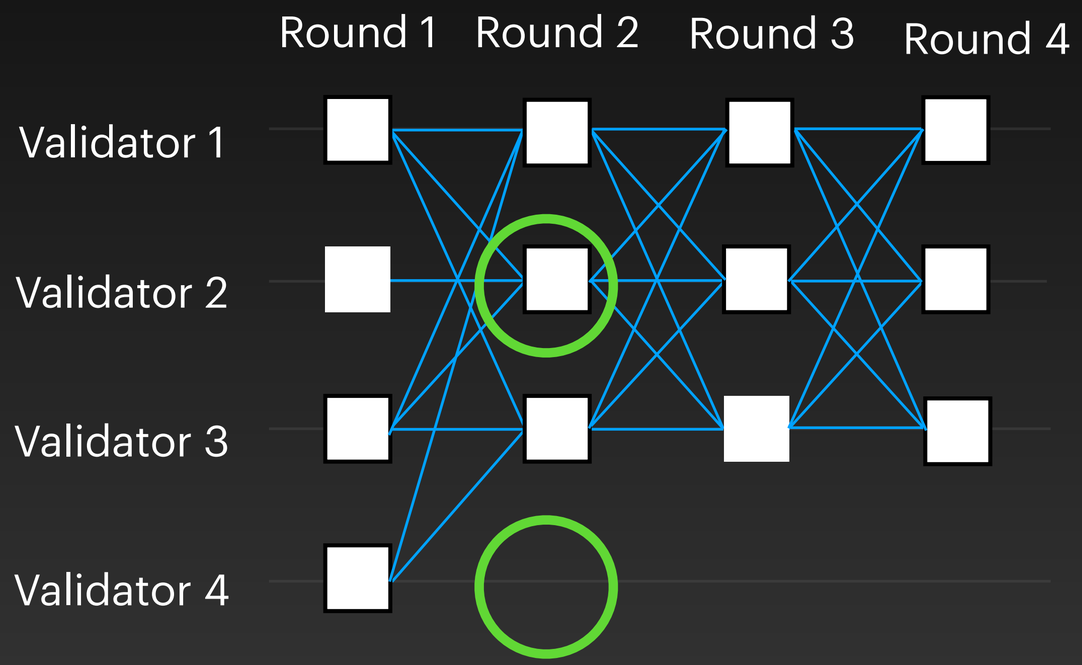
Bullshark

1. Partially synchronous version
2. Asynchronous Version (DAG-Rider)
3. Best of both worlds

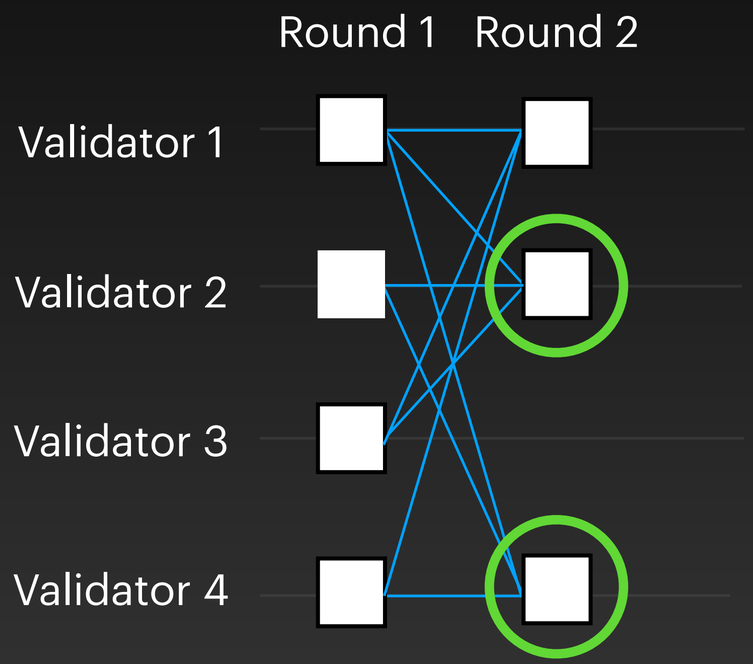
Bullshark

Non-Equivocation

Local view of validator 1



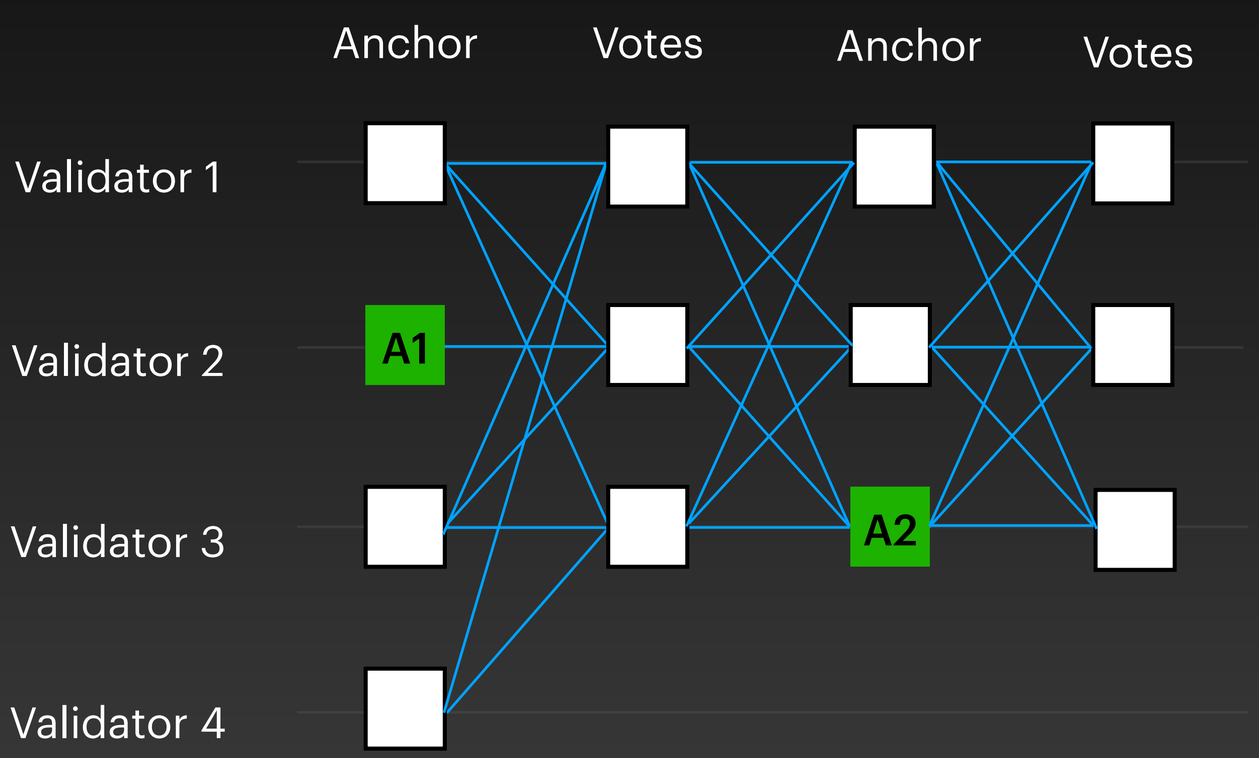
Local view of validator 4



Bullshark

Interpreting The DAG

Local view of validator 1



Anchors

A priori known -> Partial synchrony

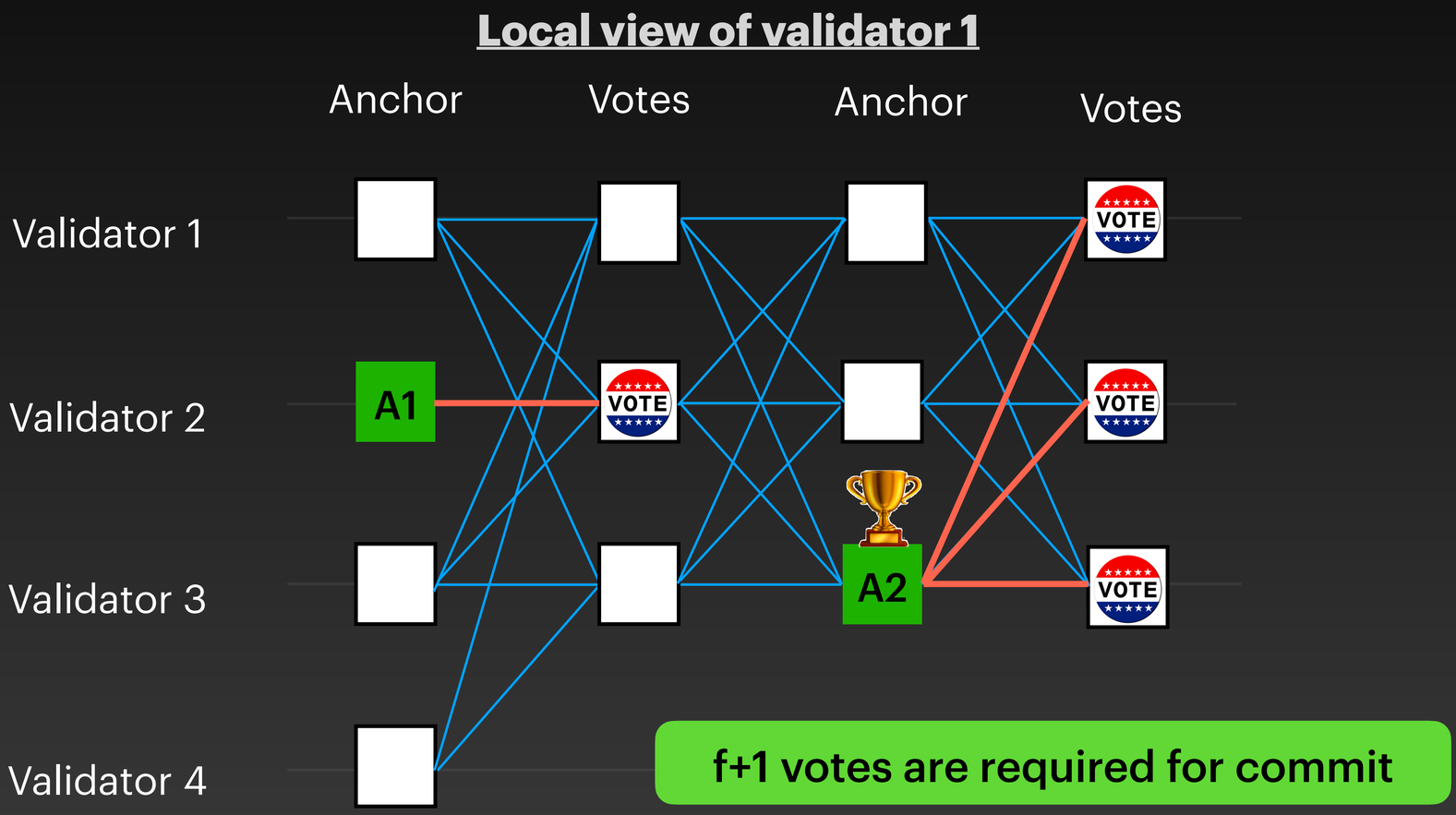
Elected in retrospect -> Asynchrony

Goal

1. Decide which anchors to commit.
2. Deterministically order their causal histories.

Bullshark

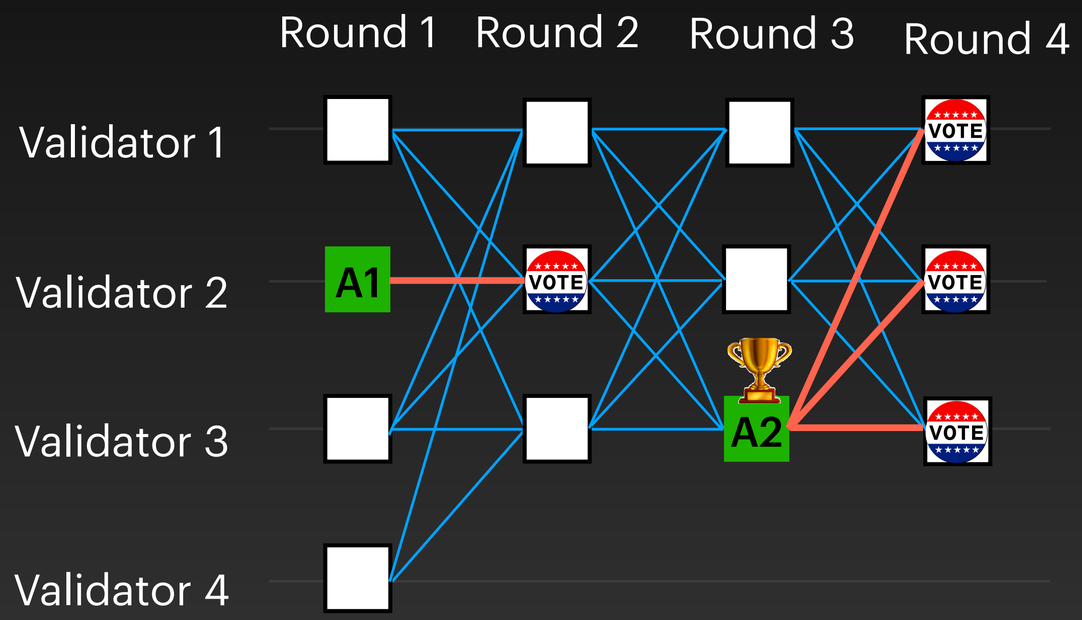
Commit Rule



Bullshark

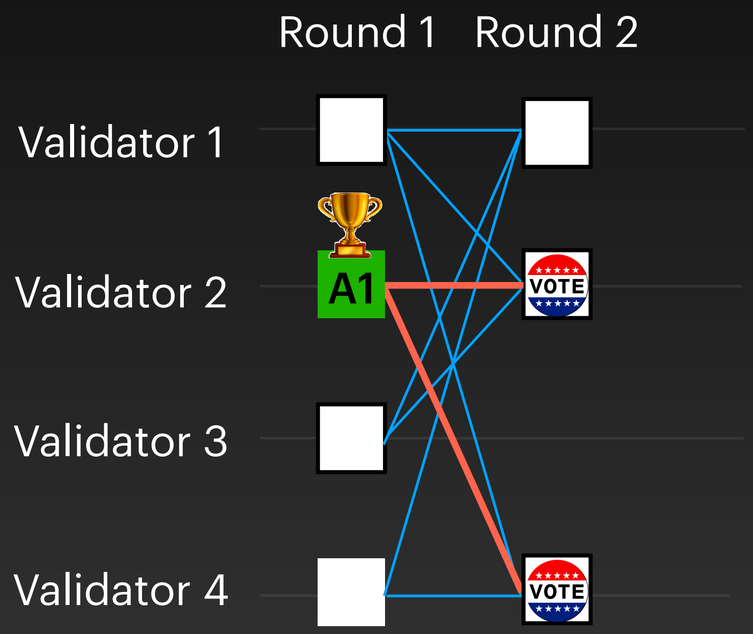
Different Local Views

Local view of validator 1



Has to order A1 before A2!

Local view of validator 4



A1 is committed

Bullshark

Quorum-Intersection

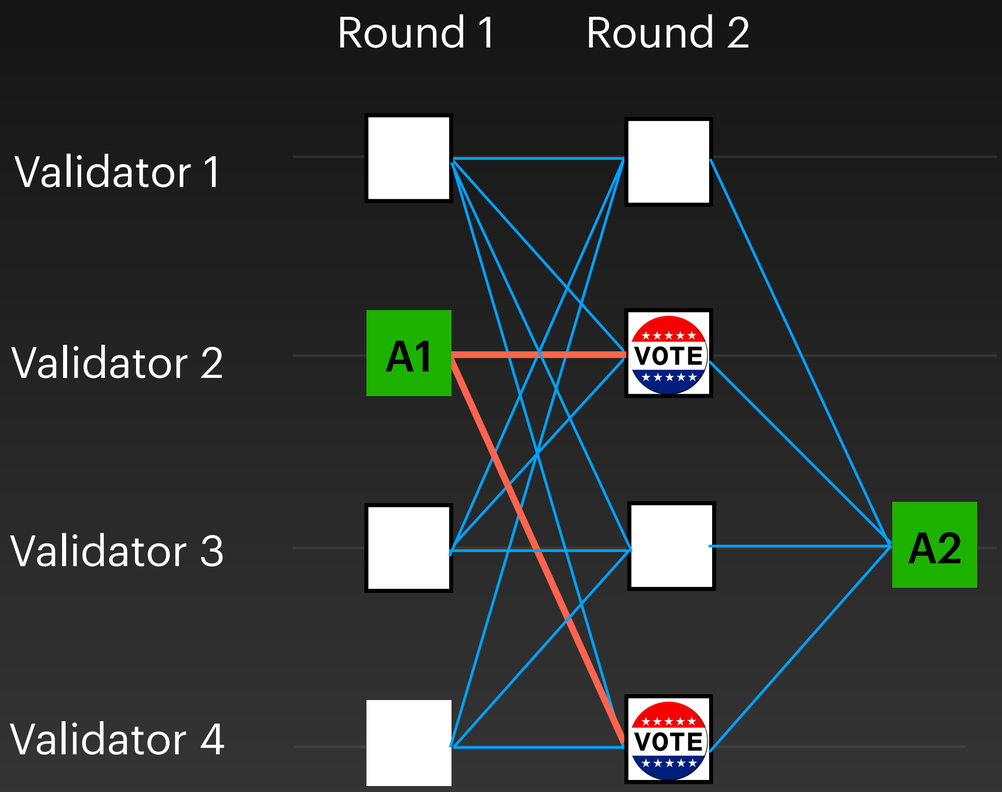
f+1 votes required to commit + Each node refers to n-f nodes from the previous round

If an anchor A is committed → All future anchors will have a path to at least 1 vote for A → All future anchors will have a path to A

Bullshark

Quorum-Intersection

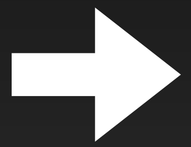
Local view of validator 4



Bullshark

Quorum-Intersection

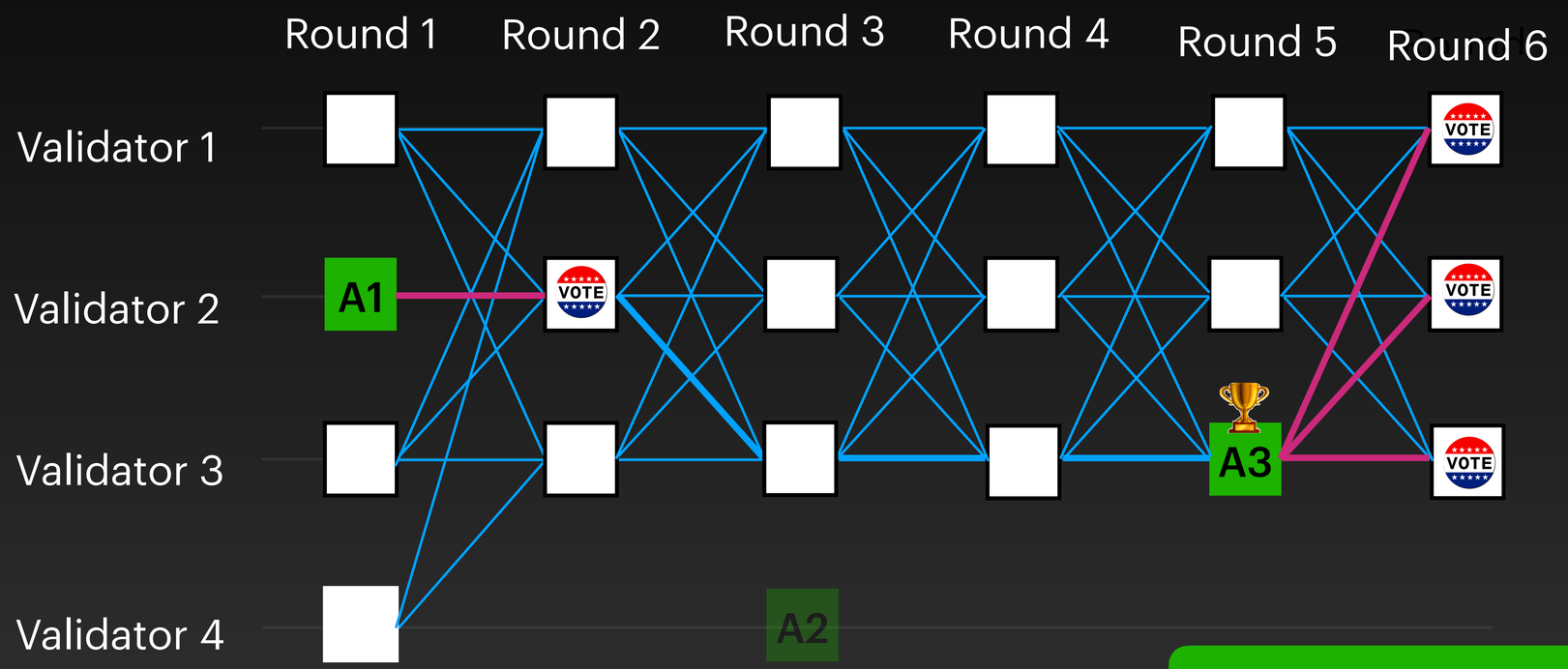
No path from future anchor A’ to A



No honest Validator committed A

Bullshark

The Full Protocol

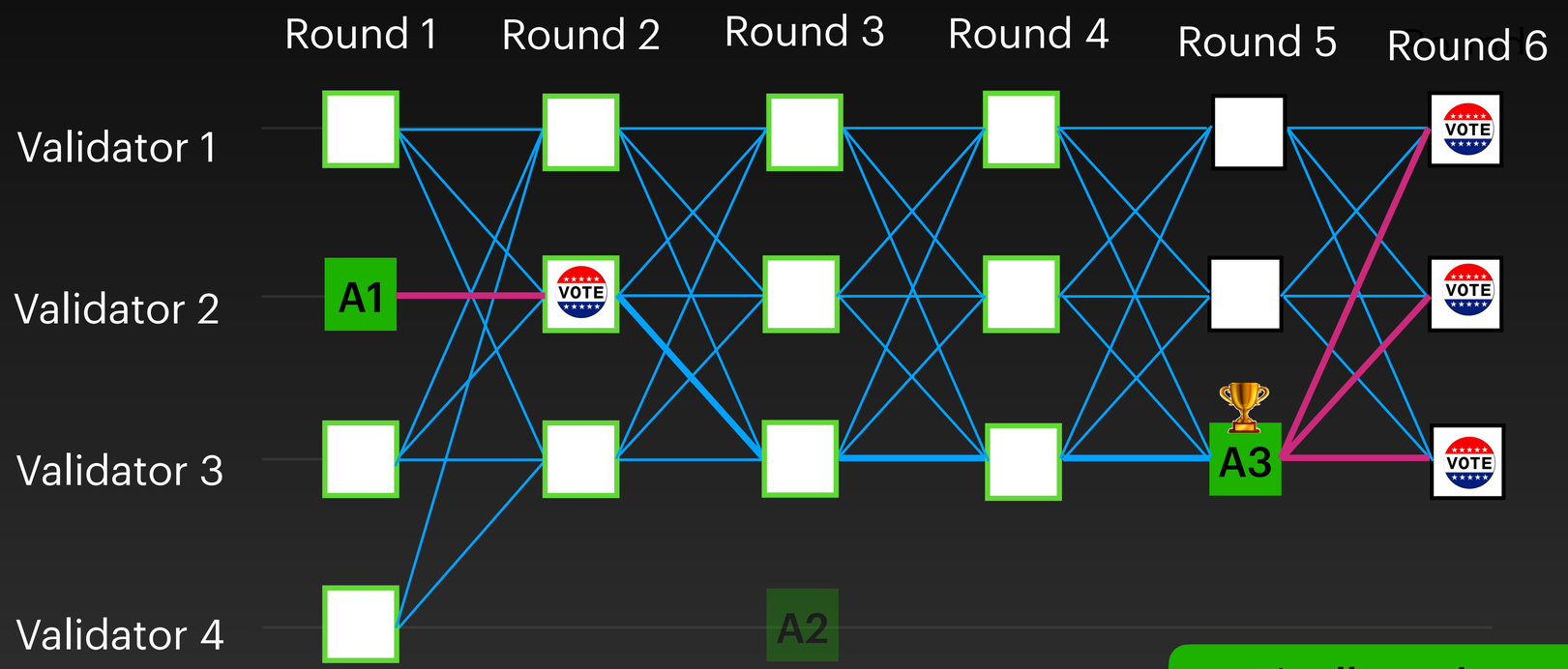


Continue recursively from A1 until an anchor that was previously ordered is reached

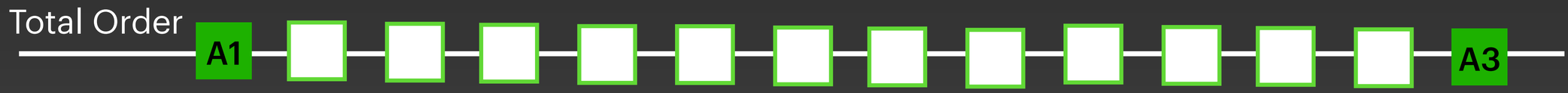
Total Order

Bullshark

The Full Protocol

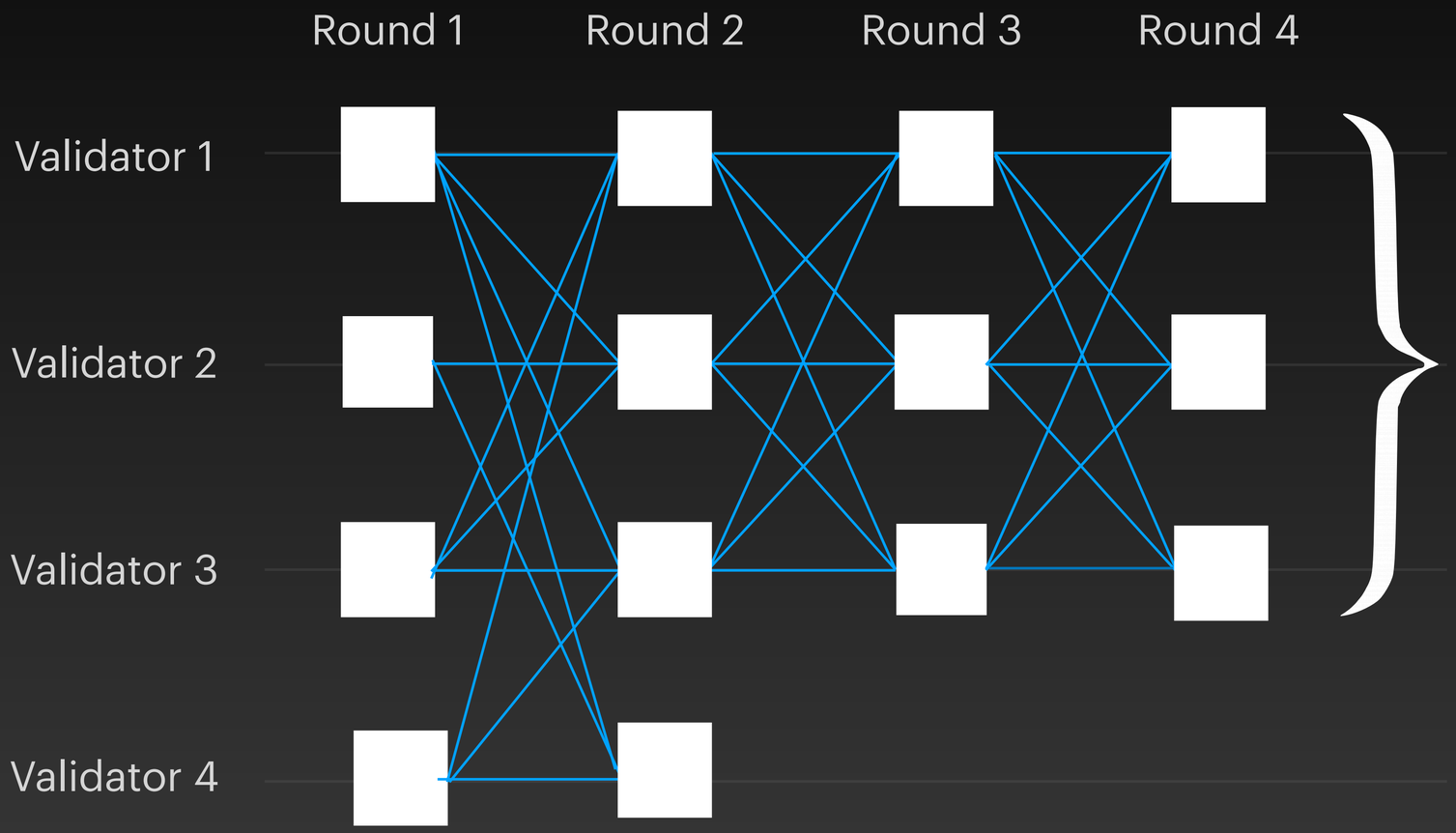


Finally, order anchors' causal histories one by one by some deterministic order



Bullshark

Chain Quality For Free



At least 2f+1 nodes in each round out of which at most f are byzantine

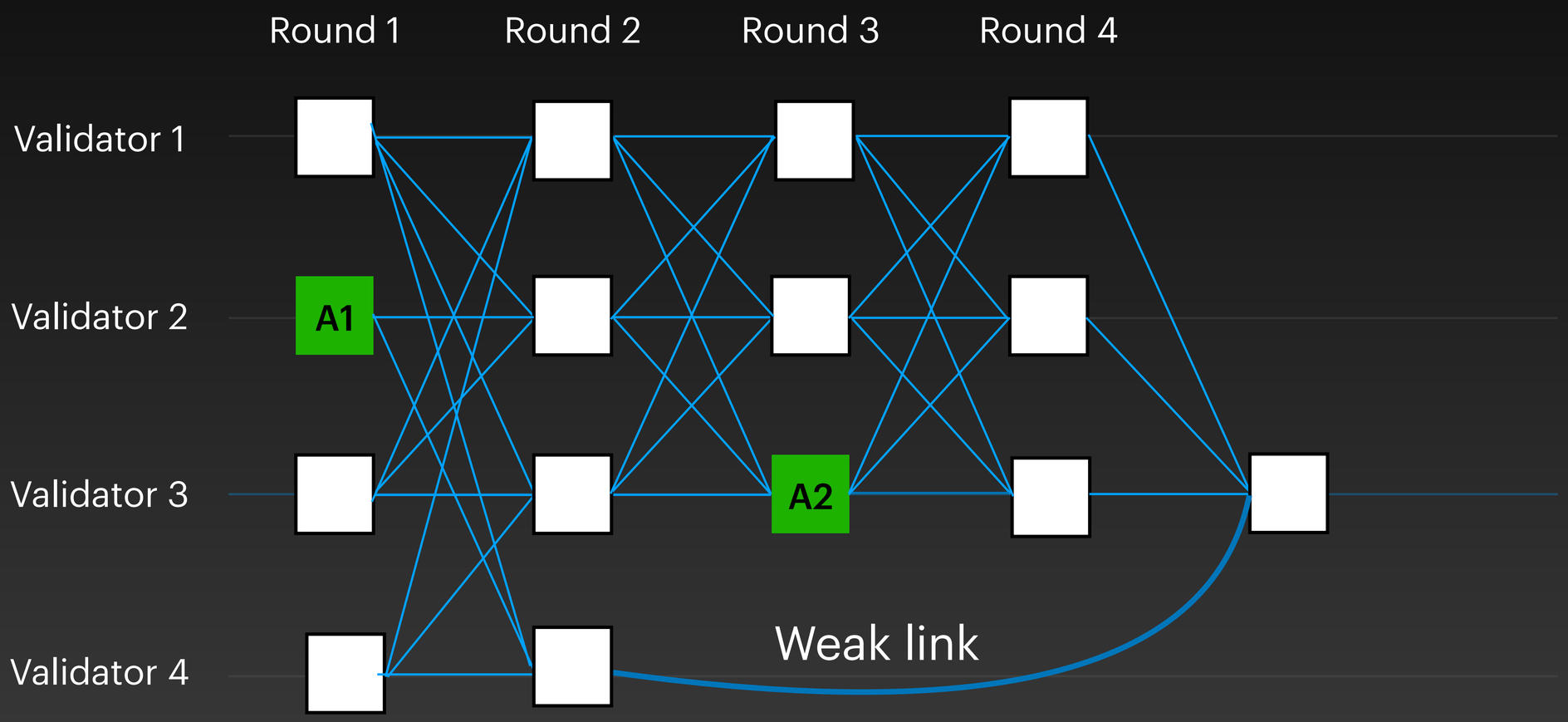
At least f+1/2f+1 > 1/2 honest

Optimal!

Probabilistic Indistinguishability and the Quality of Validity in Byzantine Agreement [AFT'22]

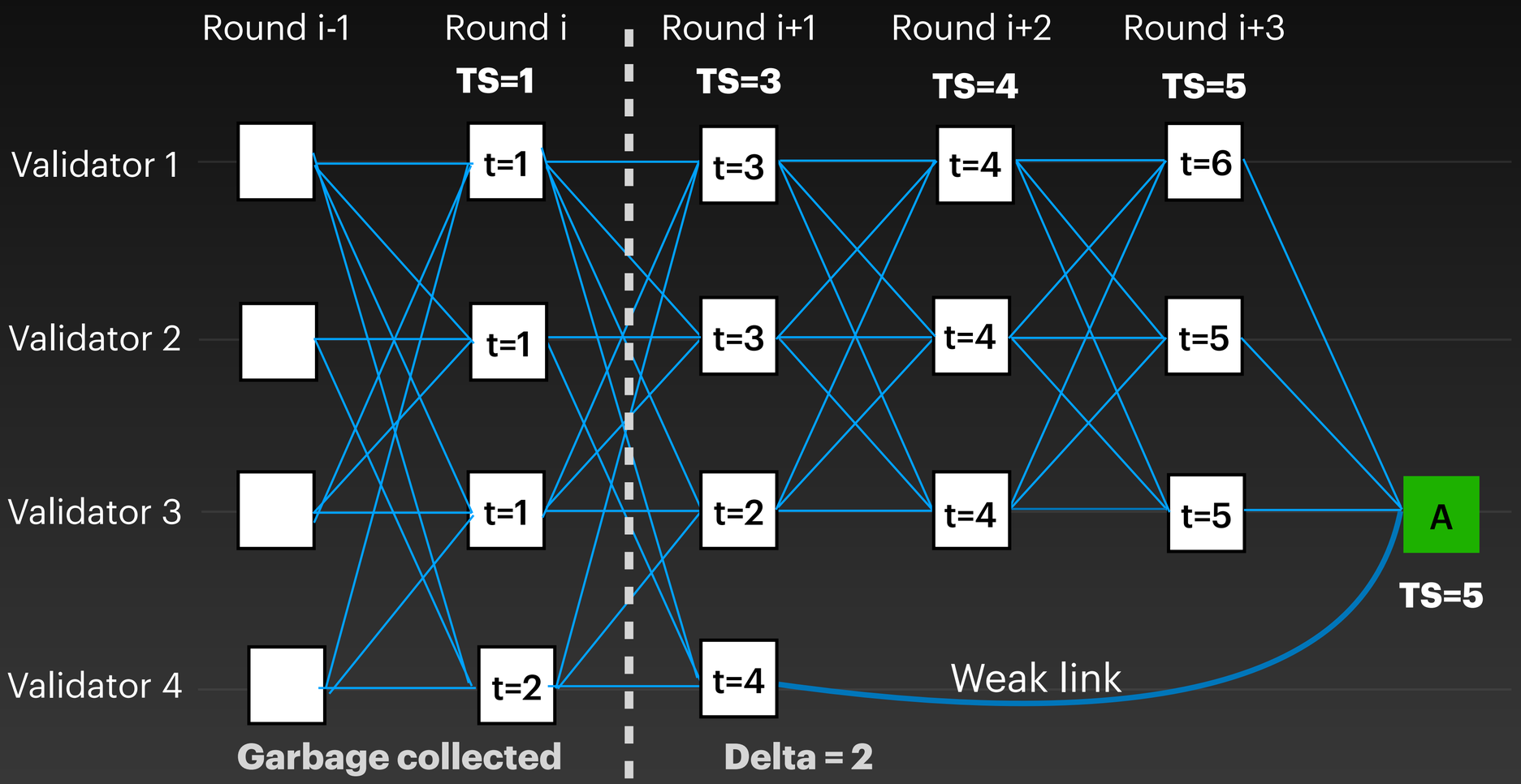
Bullshark

Fairness and Garbage Collection



Bullshark

Fairness and Garbage Collection



Round-based DAG structure

+

Consensus

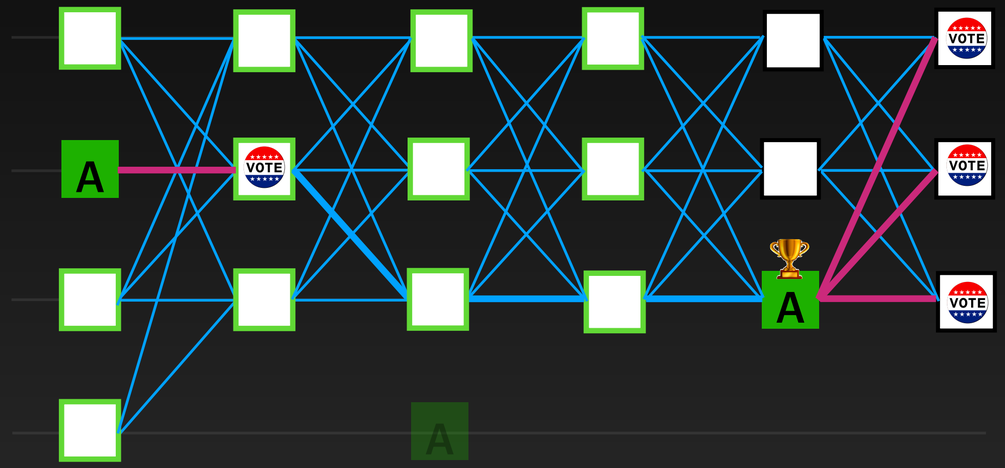
=

Easy

Bullshark

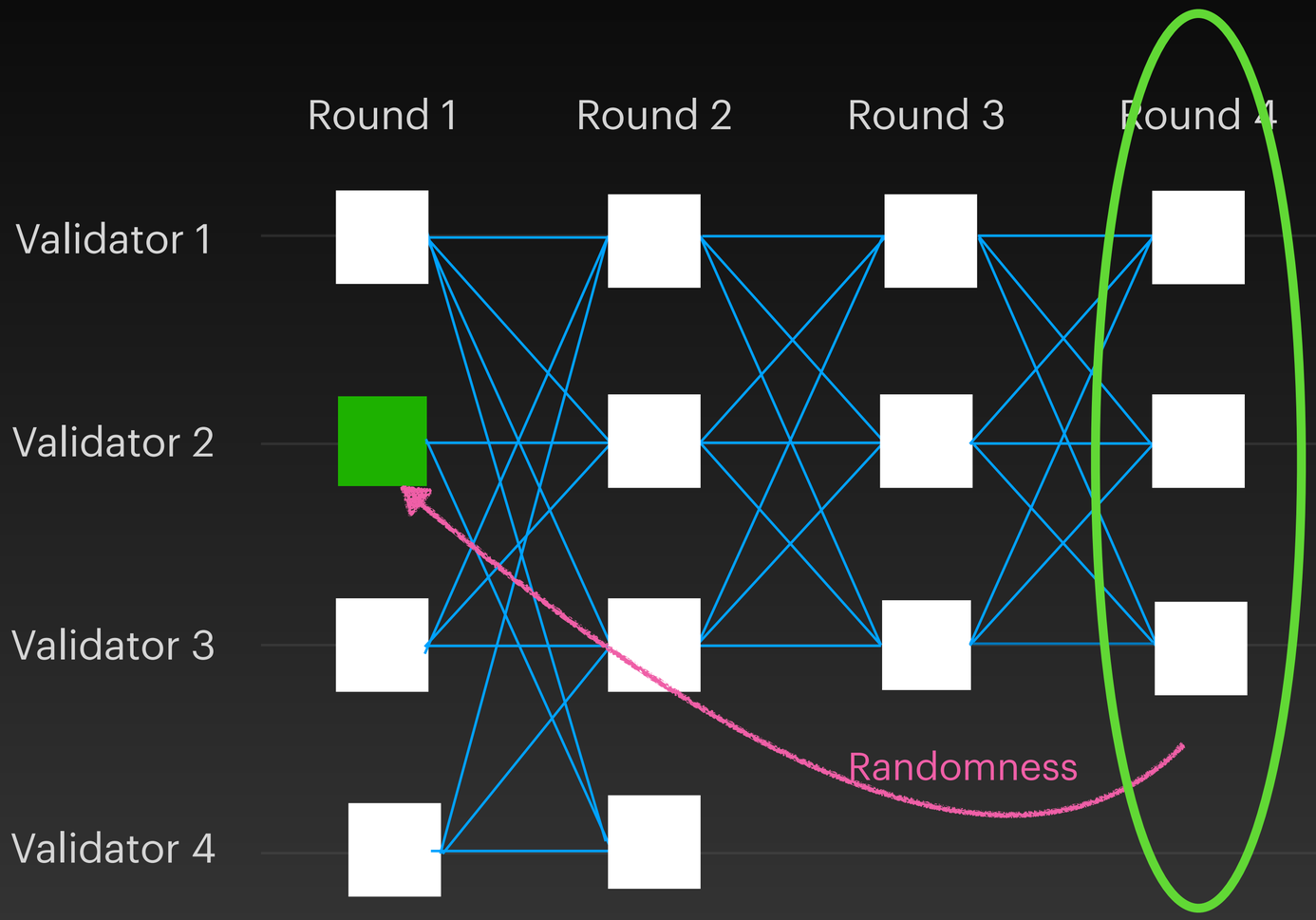
Conclusion

- Extremely simple
  - No view-change!
  - No view-synchronization!
  - Zero communication overhead
- Chain-Quality
- Fairness
- Garbage collection



Liveness

Asynchronous Liveness

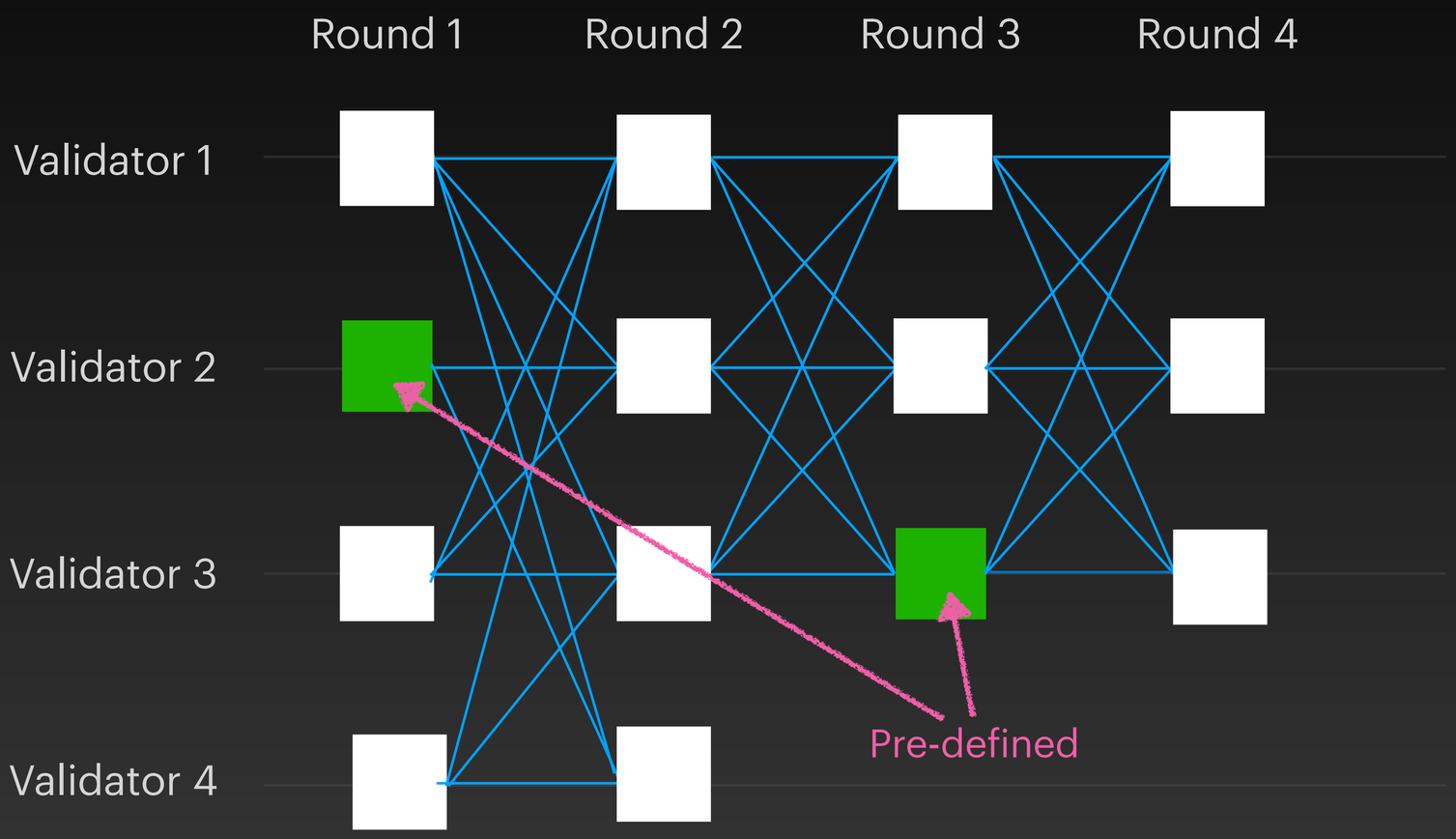


DAG-Rider

Tusk

Full Bullshark

Partially Synchronous Liveness



Bullshark

Conclusion

Papers are available online

All You Need is DAG

Idit Keidar
Technion

Eleftherios Kokoris-Kogias
IST Austria and Novi Research

Oded Naor*
Technion

Alexander Spiegelman
Novi Research

Narwhal and Tusk: A DAG-based Mempool and Efficient BFT Consensus

George Danezis
Mysten Labs & UCL

Lefteris Kokoris-Kogias
IST Austria

Alberto Sonnino
Mysten Labs

Alexander Spiegelman
Aptos

Bullshark: DAG BFT Protocols Made Practical

Alexander Spiegelman
sasha@aptoslabs.com
Aptos

Neil Giridharan
giridhn@berkeley.edu
UC Berkeley

Alberto Sonnino
alberto@mysternlabs.com
Mysten Labs

Lefteris Kokoris-Kogias
ekokoris@ist.ac.at
IST Austria

Read blogpost for extended summary

DAG Meets BFT - The Next Generation of BFT Consensus

Posted on June 28, 2022, by ***Alexander Spiegelman***

This post explains in simple words a recent development in the theory and practice of directed acyclic graph-based (DAG-based) Byzantine Fault Tolerance (BFT) consensus, published in three prestigious peer-reviewed conferences, and currently being implemented by several Blockchain companies, e.g., Aptos, Celo, Mysten Labs, and Somelier. **[Read More]**

Tags: consensus

Happy to collaborate

sasha@aptoslabs.com

Alexander Spiegelman

Consenus Is No Longer a Bottleneck
What Is Next In BFT?

- Incentives
- Randomness
- Shading