Multi-Proposer Construction on top of Tendermint

Malachite

(rhymes with "dynamite")

/ AdiSeredinschi
https://informal.systems/malachite

Roadmap of this talk

1. Concrete

Malachite

2. Abstract

Tendermint

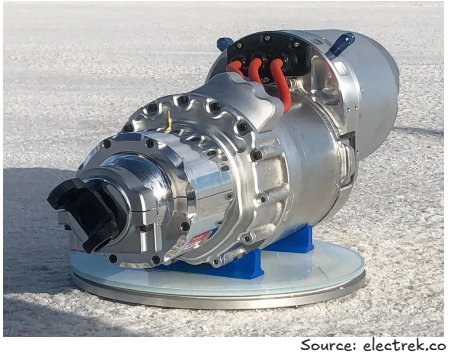
Muppet: Multi-proposer construction

3. Discussion

Decentralized Application

By Vauxford - Own work, CC BY-SA 4.0

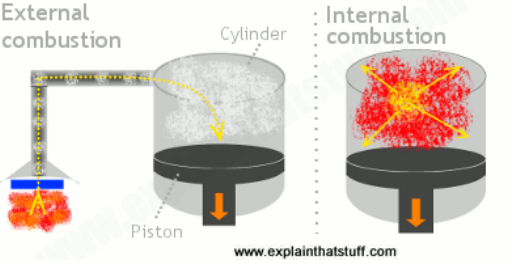
Malachite



Source: electrek.co

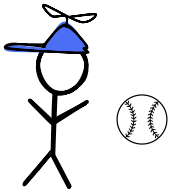
Tendermint

Buchman, Kwon, Milosevic. "The latest gossip on BFT consensus" arXiv preprint arXiv:1807.04938 (2018).

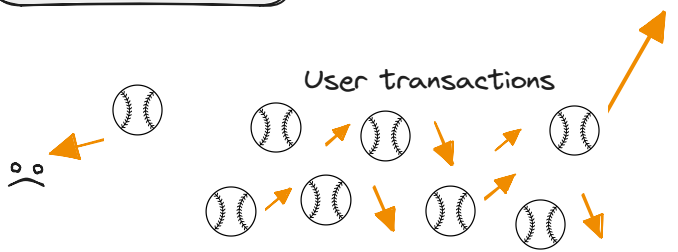


Malachite: Origins

Malachite came about from the need to run Tendermint in a strange new way, for Starknet decentralized sequencer



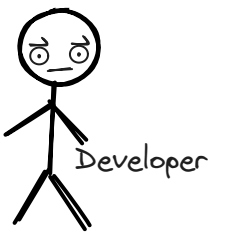
Execution and proofs

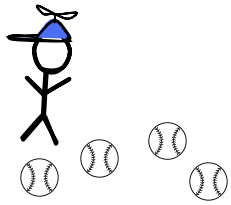


L1

Decentralized Sequencer

chaotic system:
- bugs
- lost tx
- inconsistent state
- liveness problems
- efficiency & costs
- maintenance





* Typical protocols needed for a decentralized rollup

* Inspired from Starknet here

Execution and proofs

Proof Scheduling

L1

L1

Proof Submission

Forced Staking Update

Decentralized Sequencer

Malachite

Malachite: Origins

 <> informal SYSTEMS

Requirements for the decentralized sequencer protocol

- Battle-tested
- Simple

  well-understood — specs exist
  fast to production — thoroughly researched

  
- Performance
- Optimistic responsiveness

...

Rich, novel architectures

Madara, Pathfinder

Snapchain

Crosslink

Malachite

Objectives

- facilitate growth and stability
- drive application differentiation

Strategy

- Exceptional expressivity and power to enable richer architectures
- Exceptional performance

Roadmap of this talk

1. Concrete

2. Abstract

- Tendermint
- Muppet: Multi-proposer construction

3. Discussion

Tendermint
And the art of simplifying

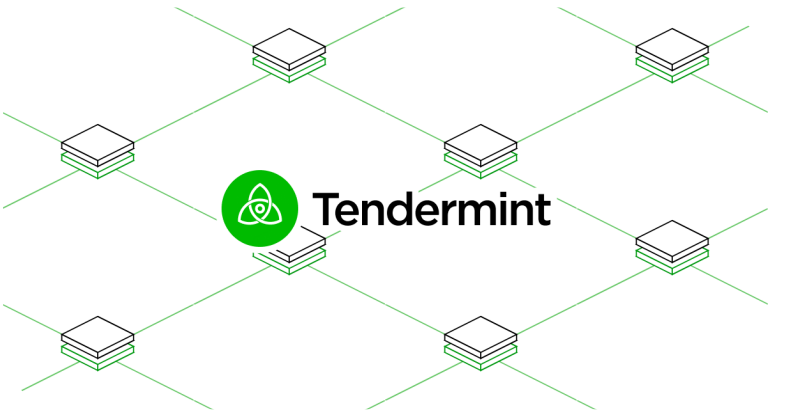
- the most widely-used BFT consensus protocol

- represents the meeting of theory <> practice circa 2013

- some constraints brought from practice:
* builds on a gossip network
* rotate leaders all the time

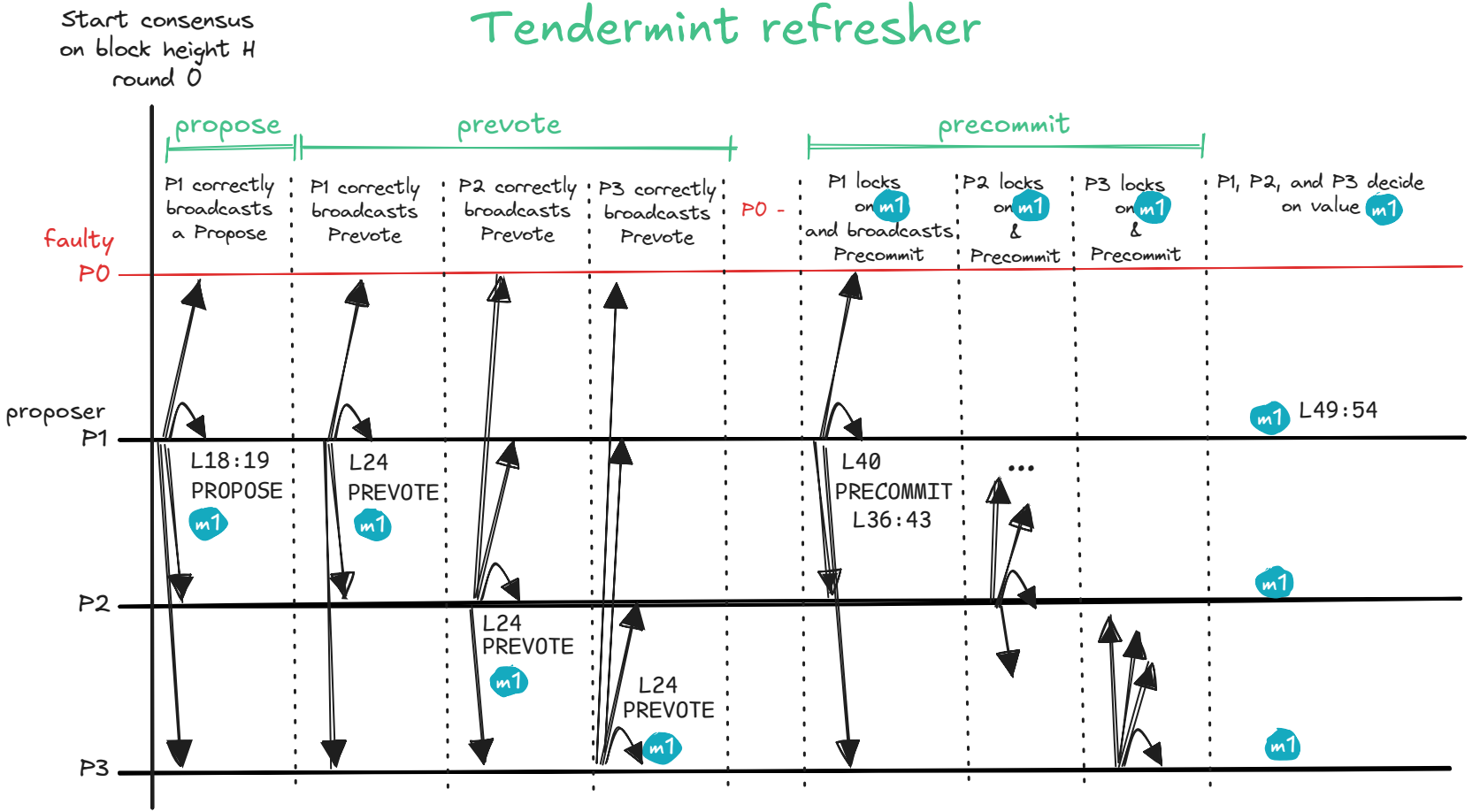
- main novelty:
* the simplified termination mechanism
* unified graceful & degraded paths

- The protocol was wrong the first couple of times

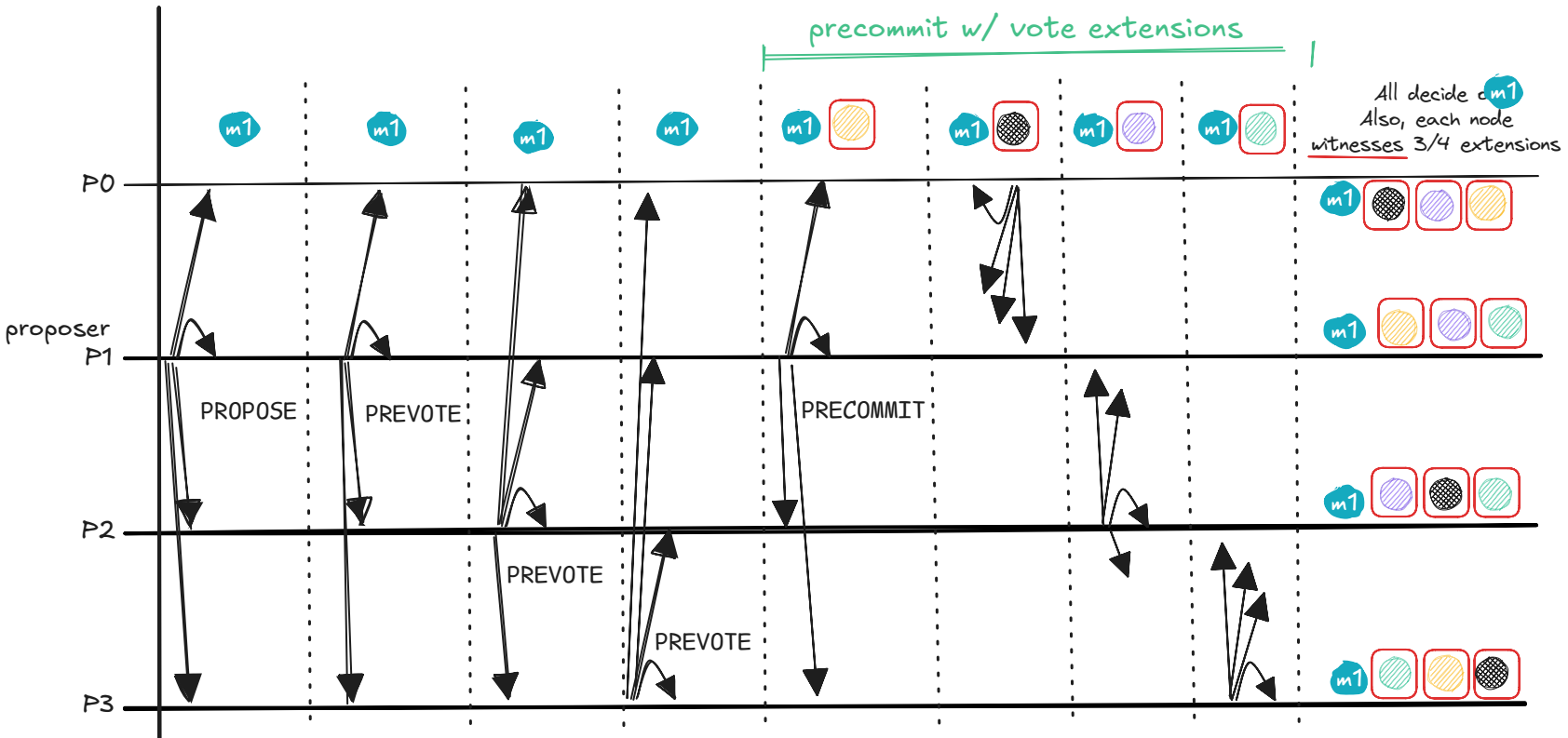


Buchman, Kwon, Milosevic. "The latest gossip on BFT consensus" arXiv preprint arXiv:1807.04938 (2018)

Tendermint refresher



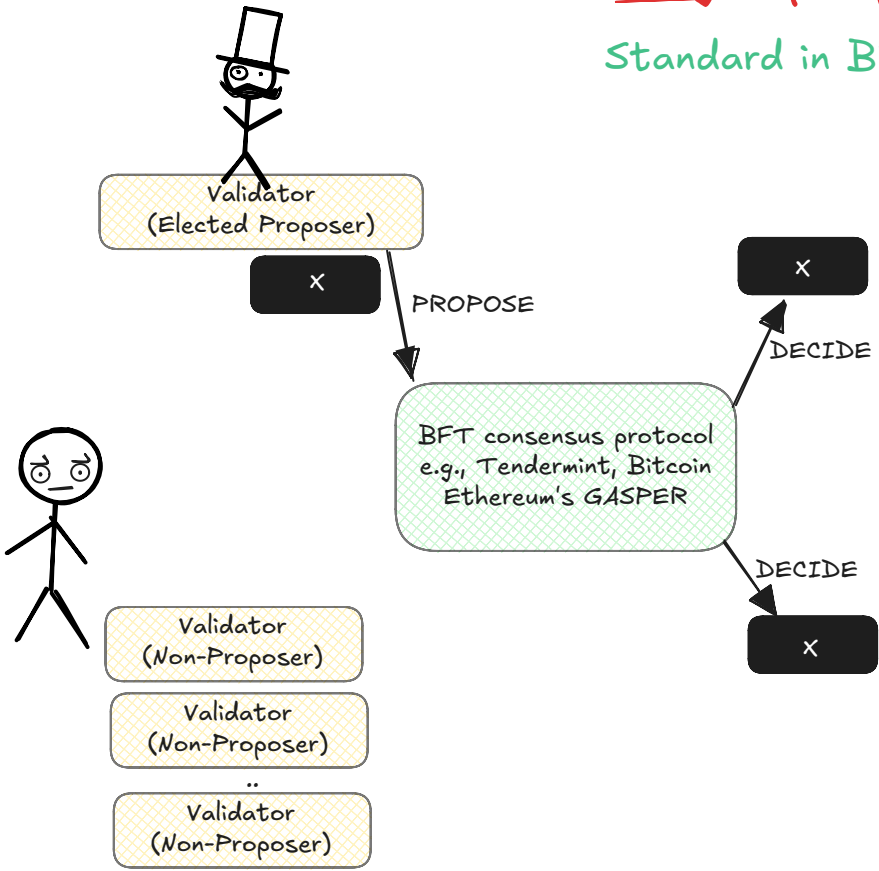
Vote Extensions



Each vote extension is signed, arbitrary data

Single-proposal API

Standard in BFT consensus



Abstract view:

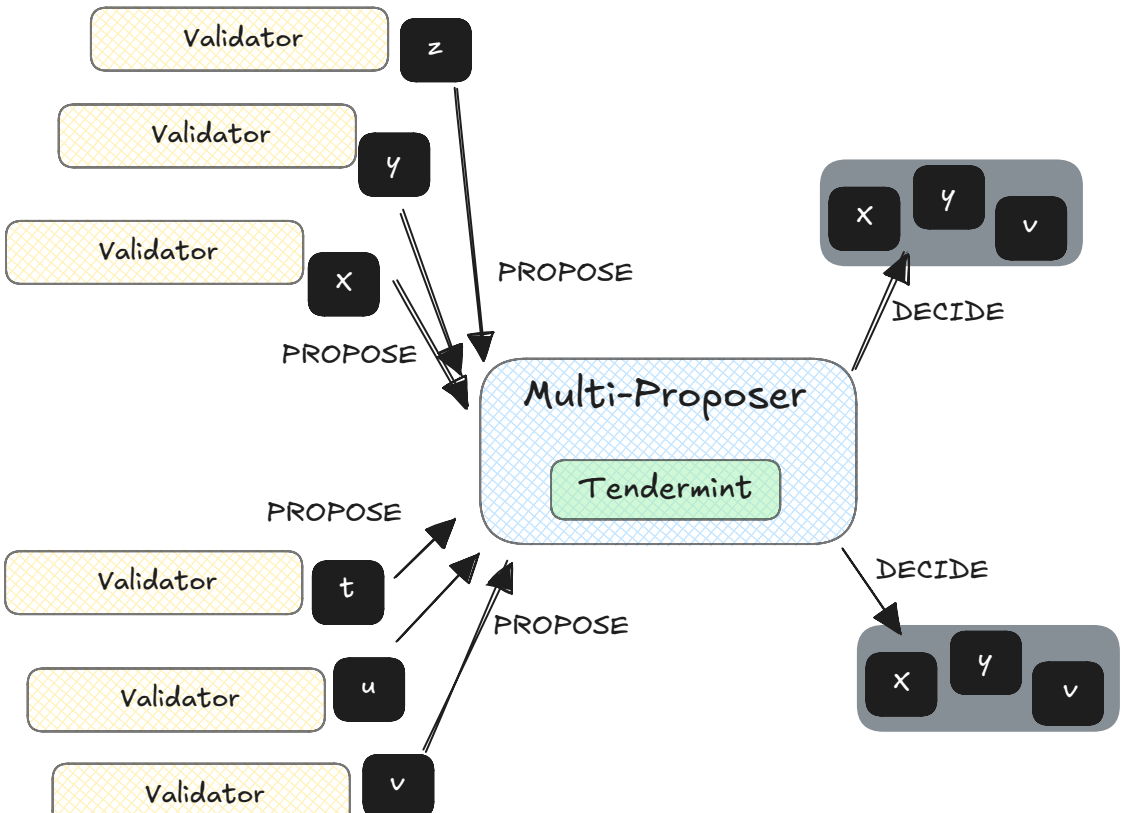
- 1 proposal at a time
- 1 block = 1 decision = 1 proposal
- rotating leader
- total order over decisions

Tendermint

Comet

Multi-proposal API

New API support in Malachite (via Muppet)

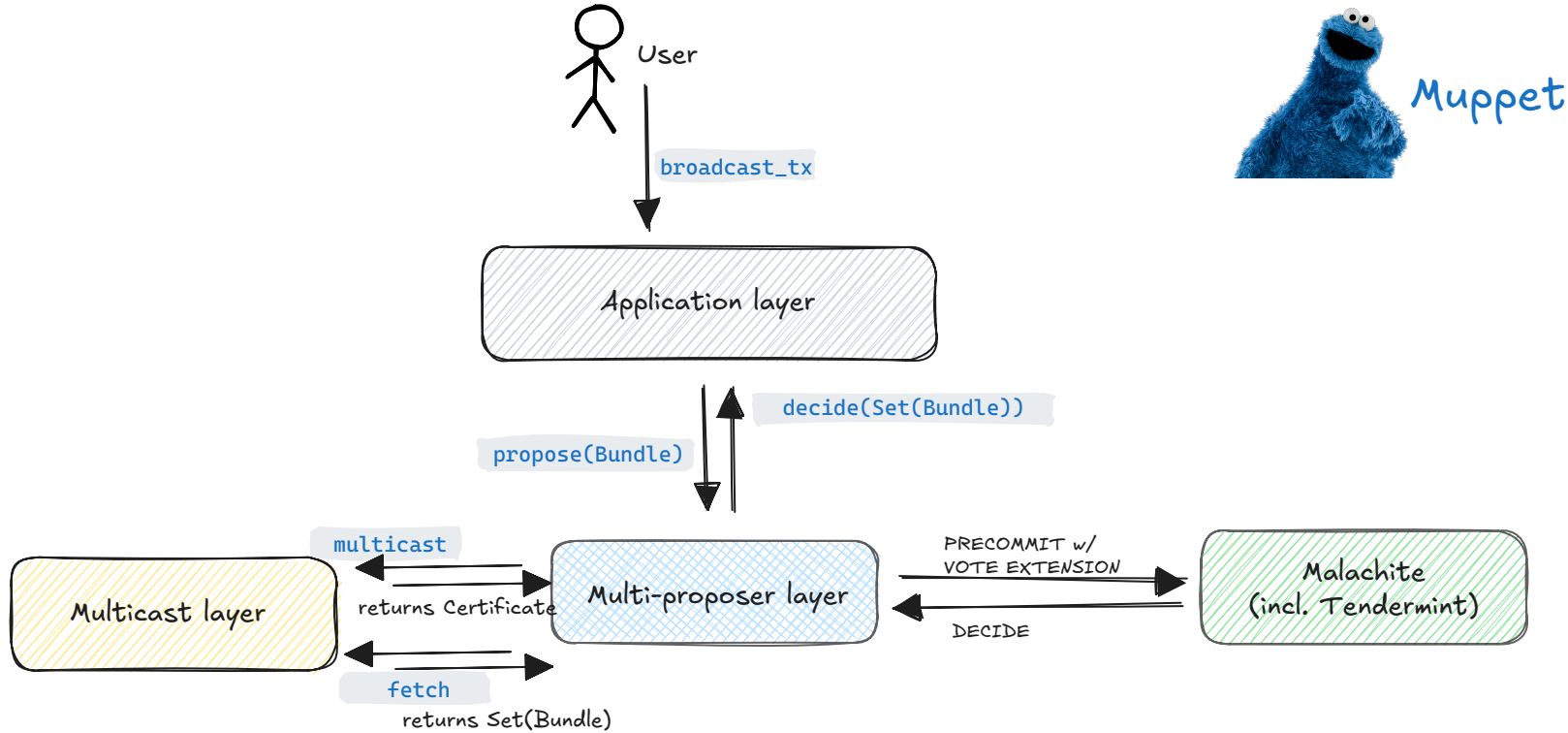


Abstract view:

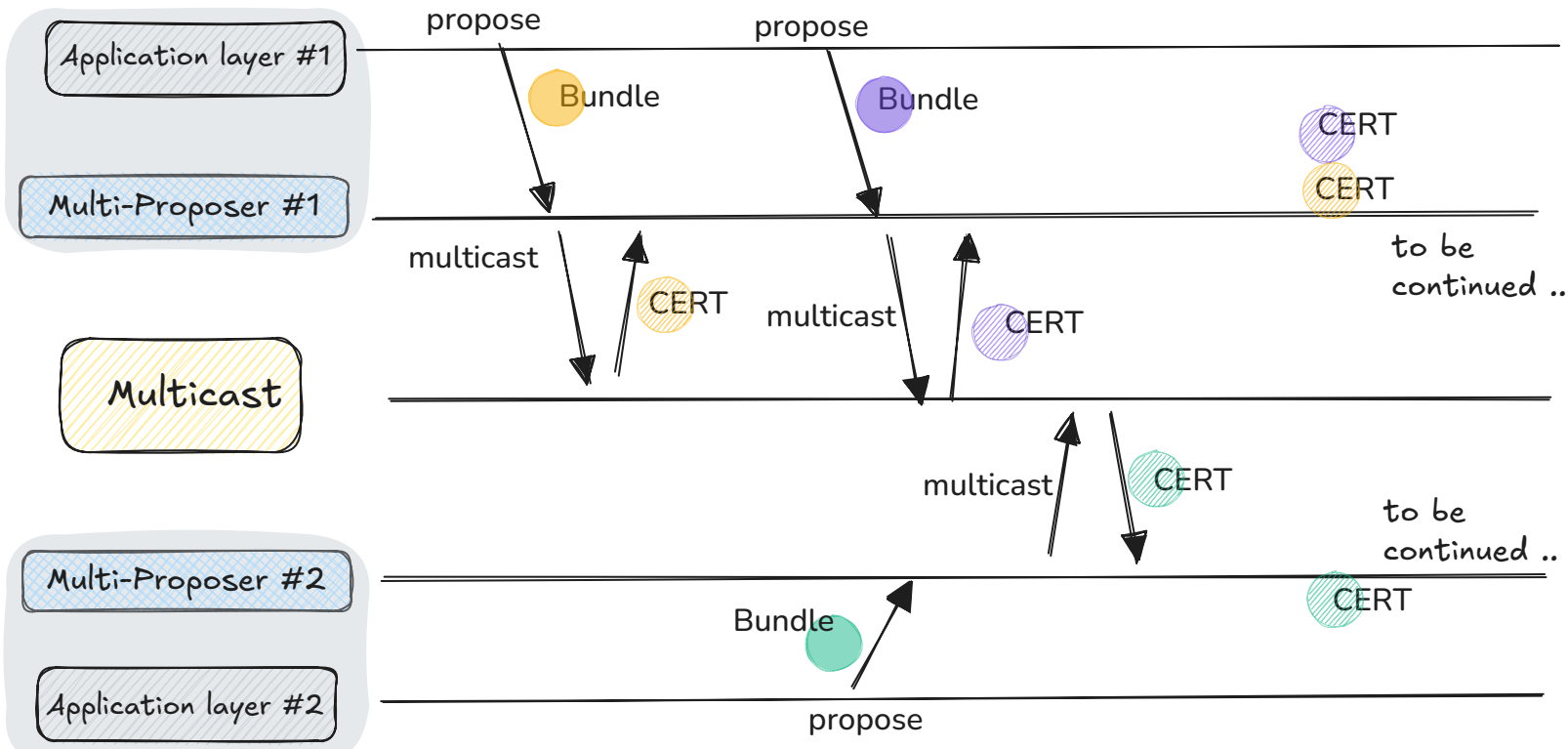
- many proposals at a time
- 1 block = 1 decision = many proposals
- (same) rotating leader
- (same) total order over decisions

API is symmetric

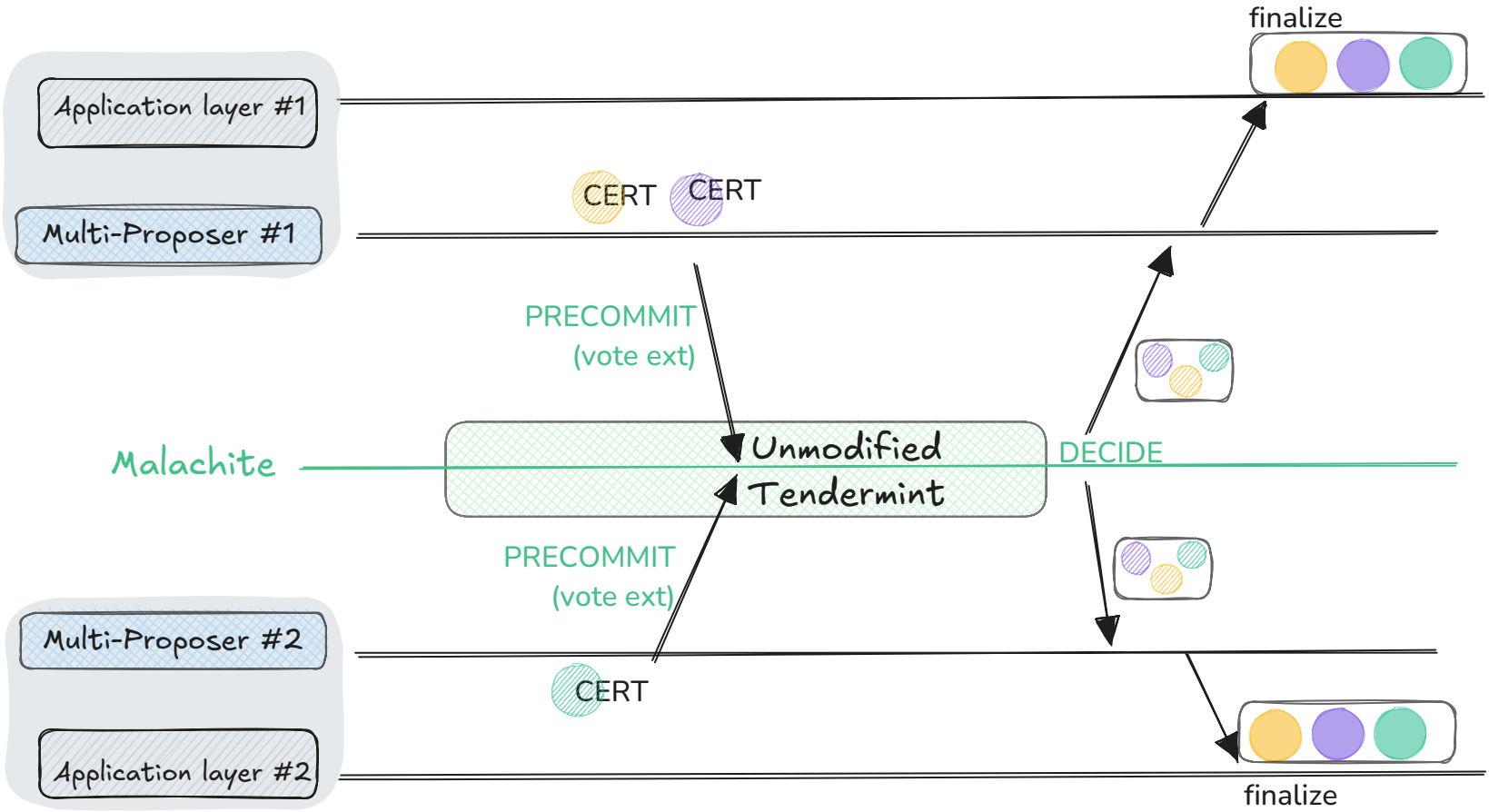
Malachite


User
broadcast_tx
Application layer
decide(Set(Bundle))
propose(Bundle)
multicast
Multicast layer
returns Certificate
Multi-proposer layer
PRECOMMIT w/
VOTE EXTENSION
DECIDE
Malachite
(incl. Tendermint)
fetch
returns Set(Bundle)
Muppet

Multi-Proposer Design (1)

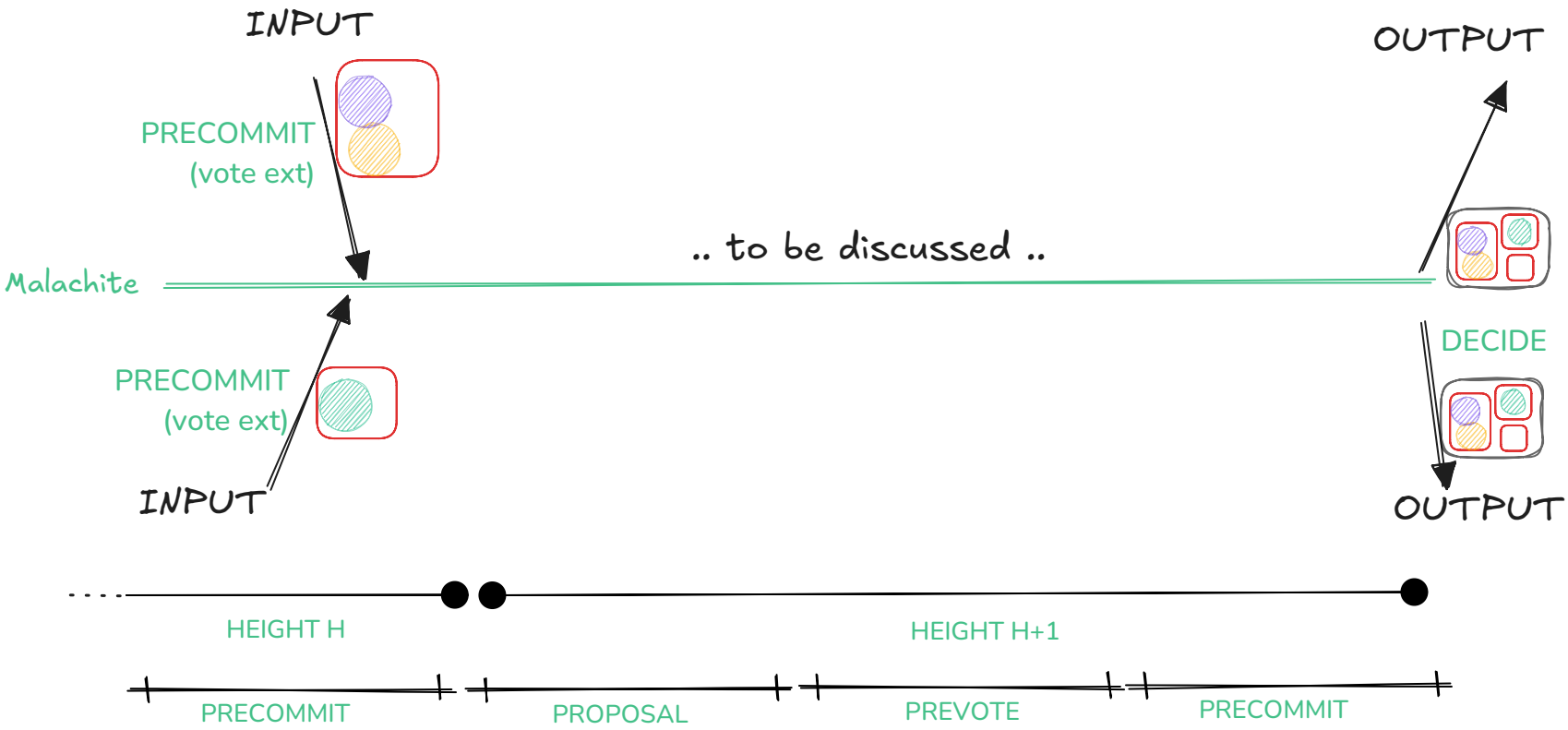


Multi-Proposer Design (2)



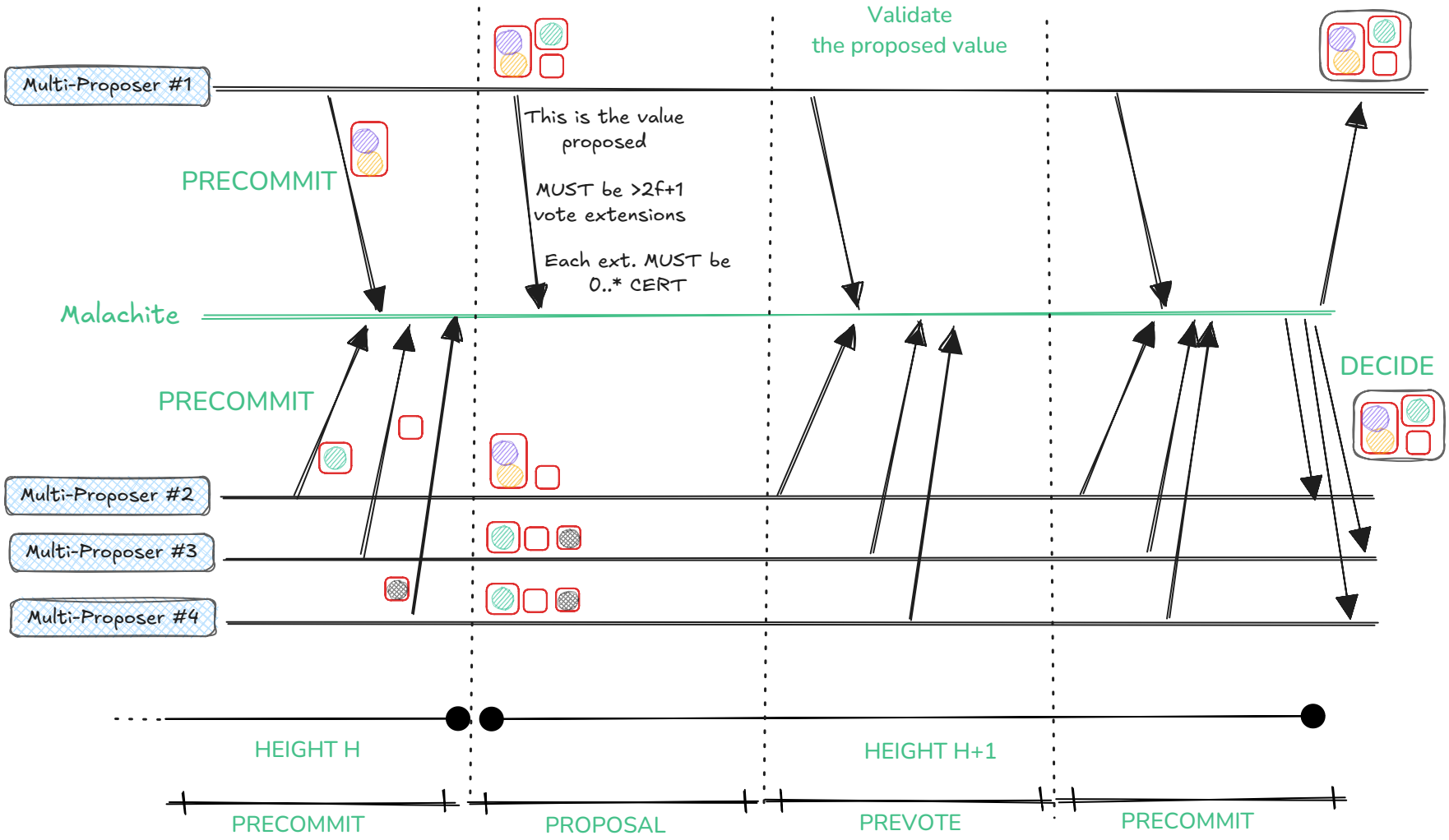
Ordering of concise certificates (not raw Bundles) -> Higher throughput

Multi-Proposer Design (3)

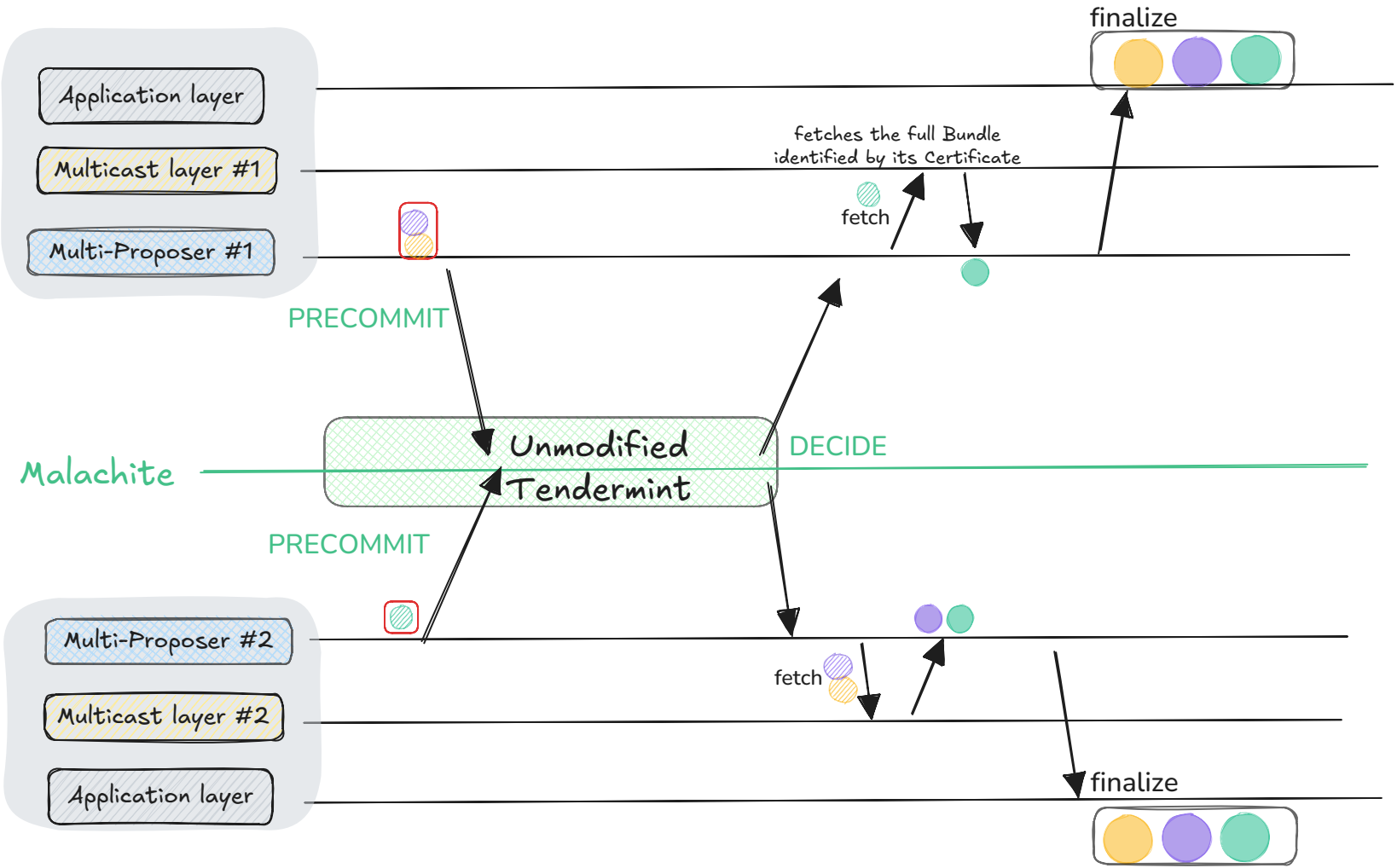


Finalization Latency: 4 one-way message delays

Multi-Proposer Design (4)

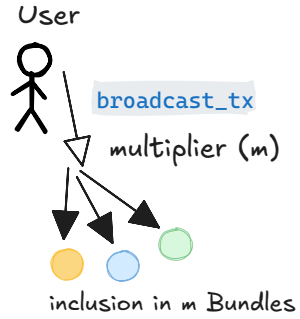


Multi-Proposer Design (5)

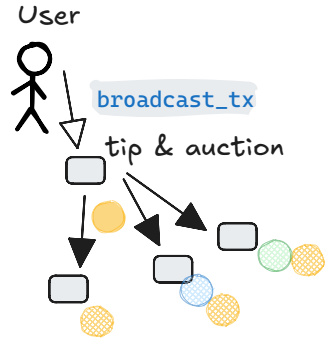


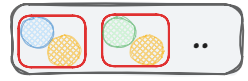
Censorship Resistance

- Value of CR unclear
- Design space available
  - **Option 1 - inclusion multiplier**
    - Each tx: User chooses an inclusion multiplier (m)
    - The multiplier determines how many validators will include the tx in their next Bundle
    - If multiplier x > f+1, then guaranteed inclusion in next block
    - Lower efficiency (high redundancy)



  - **Option 2 - dependency graph via Bundles**
    - User chooses a builder/val that guarantees inclusion, running an auction at each height (say, Flashbots)
    - Top *N* tx at Flashbots get included in each Bundle, as follows
    - At multicast layer, Flashbots pays ~F other validators to include the Bundle certificate at their next Precommit
    - Higher efficiency (but requires extra mechanism)





Roadmap of this talk

1. Concrete

2. Abstract

   Tendermint

   Muppet: Multi-proposer construction

3. Discussion

Discussion

Fees

- Inclusion
- Execution

Validation

- CheckTx
- ReCheckTx

Name for Multi-Proposer

- Muppet
- Polykite
- Malaplex
- ??

names are tough..

Free DA problem

- Require each wallet to have 5x the min fee in funds
- Don't care

informal SYSTEMS

- Lausanne, Zurich, Toronto, Berlin ..
- Core R&D & security for decentralized systems

Malachite

https://t.me/MalachiteEngine

QUINT

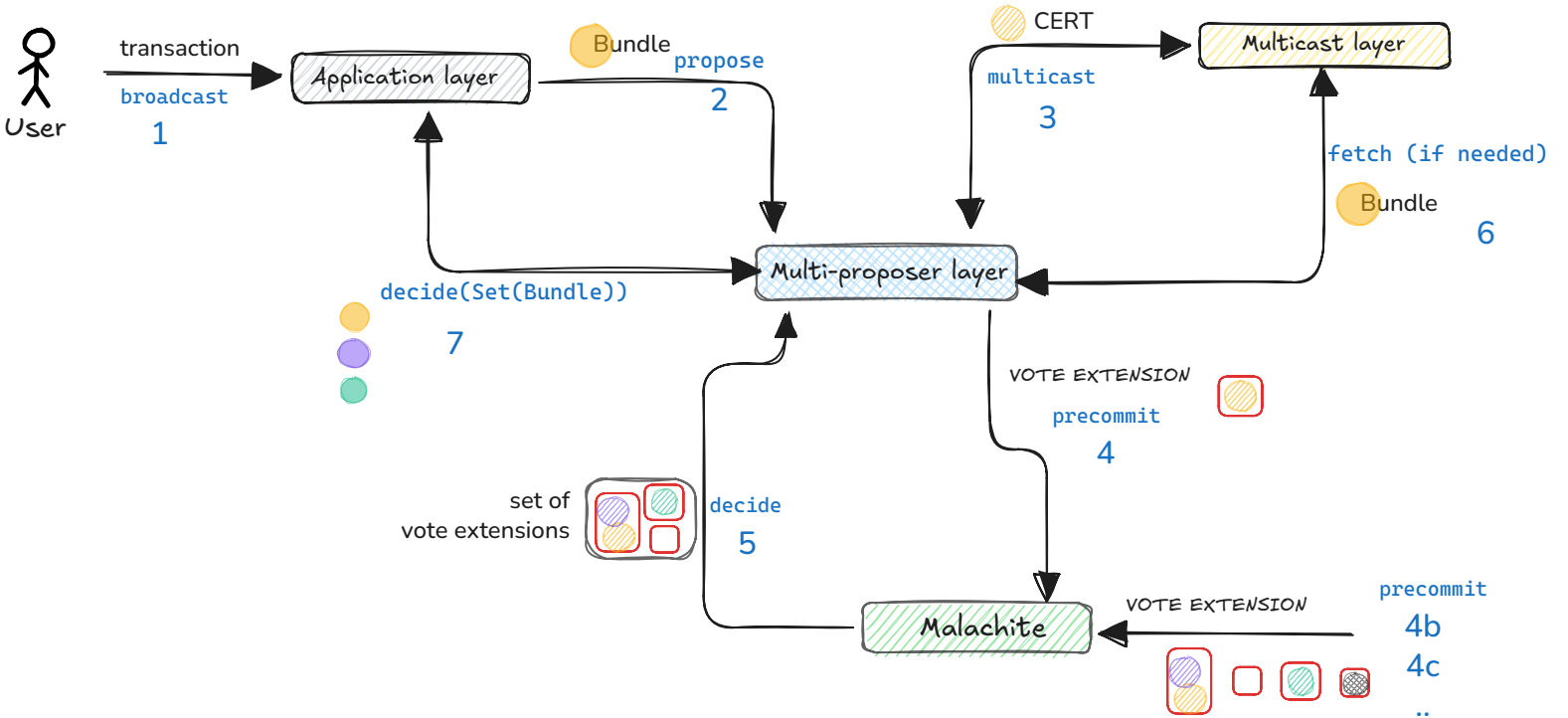
https://quint-lang.org/

CYCLES
The Open Clearing Protocol

https://cycles.money/

We are (always) hiring!
https://informal.systems/

Lifecycle of a user's transaction



Overview of cool things we're experimenting with

Multi-proposer

- throughput
- symmetric API

Two-phase Tendermint variant

- latency
- simpler base protocol
- 5F+1 fault-model

EVM integration

- reth integration
- libp2p improv & MEV protection
- engine API ergonomics

Brief genealogy

Buchman, Kwon, Milosevic. "The latest gossip on BFT consensus"
arXiv preprint arXiv:1807.04938 (2018).

Tendermint

Algorithm

impl.

Tendermint Core

RIP

Impl. in Go
Consensus Engine

fork & continuation

Comet

Impl. in Go
Consensus Engine

many lessons from

Malachite

Impl. in Rust
Consensus Engine

* Most widely used BFT consensus engine

* Originally called "Tendermint Core"

* Implements Tendermint in Go

* A lot of "batteries included"

* discovery
* RPC server
* mempool
* indexer
* write-ahead log

→ Simplified adoption by users

Comet

Both implement Tendermint

OSS & long-term maintenance

~ O(1000) nodes

Malachite

Go	Rust
Monolithic	Modular
Many "batteries" included	Lean & extensible
Build blockchains (i.e., Layer 1s)	Build whatever: sequencers, provers, .. ?
Building happens on top of it	Building happens with it
"Enough" performance ~ 1-2 years to iterate	Progressively higher perf ~ 3-6 months iterations

Open areas of R&D

Multi-proposer

Two-phase Tendermint

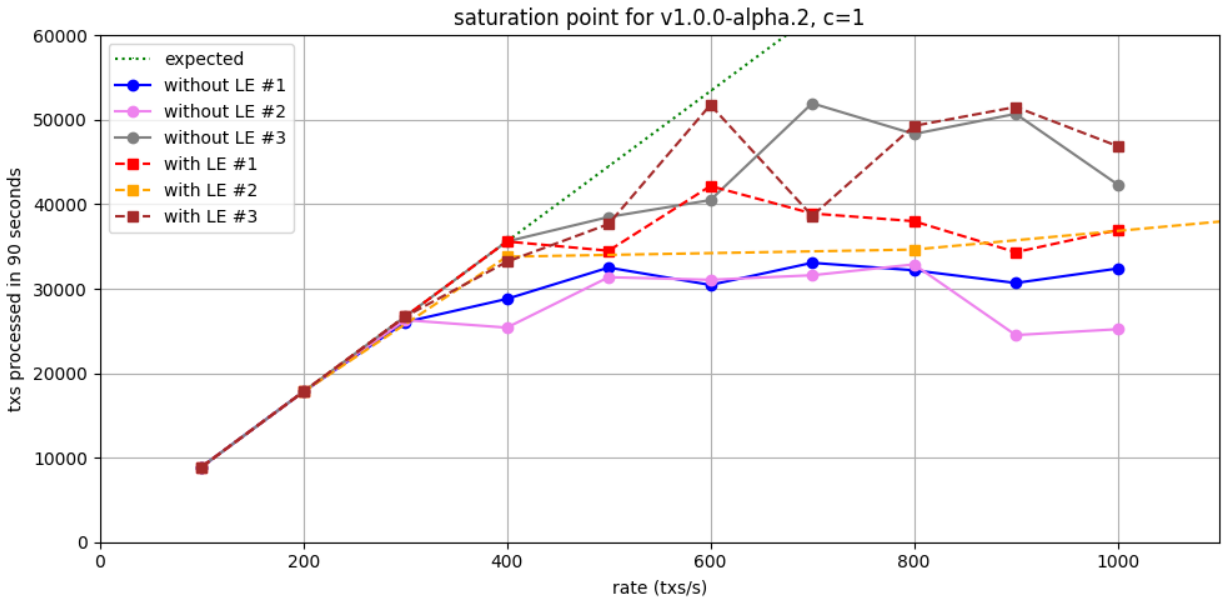
EVM (Reth) integration

native oracle	• problem: coordination with other L1s takes herculean effort • outcome: reusable primitives for processing proofs from other base layers
automatic upgrades	• problem: dev. velocity is low due to slow L1 upgrades • outcome: versioning & upgrade support for block protocol
provable consensus	• problem: prove anti-censorship to external observer • outcome: zk-proof on block construction properties

(Vanilla) Tendermint Throughput

Summary

- 300-400 tx/s
- 1 tx = 1024 bytes
- ~.4 MB/s



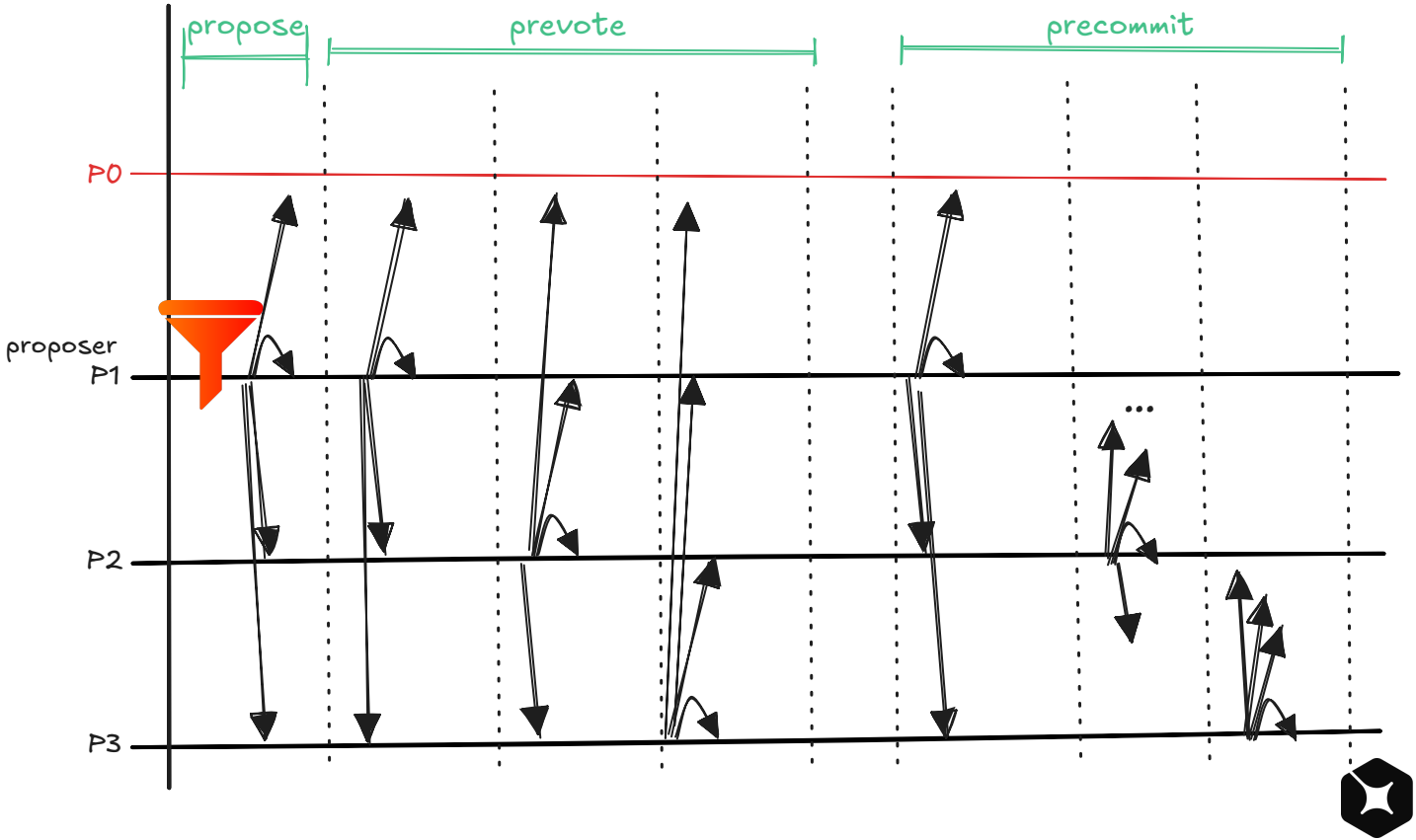
https://github.com/cometbft/cometbft/blob/main/docs/references/qa/CometBFT-QA-v1.mc

Bottleneck

leader's p2p for block dissemination @ Propose phase

Comet

(Vanilla) Tendermint Bottleneck



Comet