TU Delft

SERG — Software Engineering Research Group

ByzzBench

A Benchmark Framework for BFT Testing Algorithms

João Miguel Louro Neto & Burcu Kulahcioglu Ozkan

FMBC'25 — May 4th, 2025
6th International Workshop on Formal Methods for Blockchains

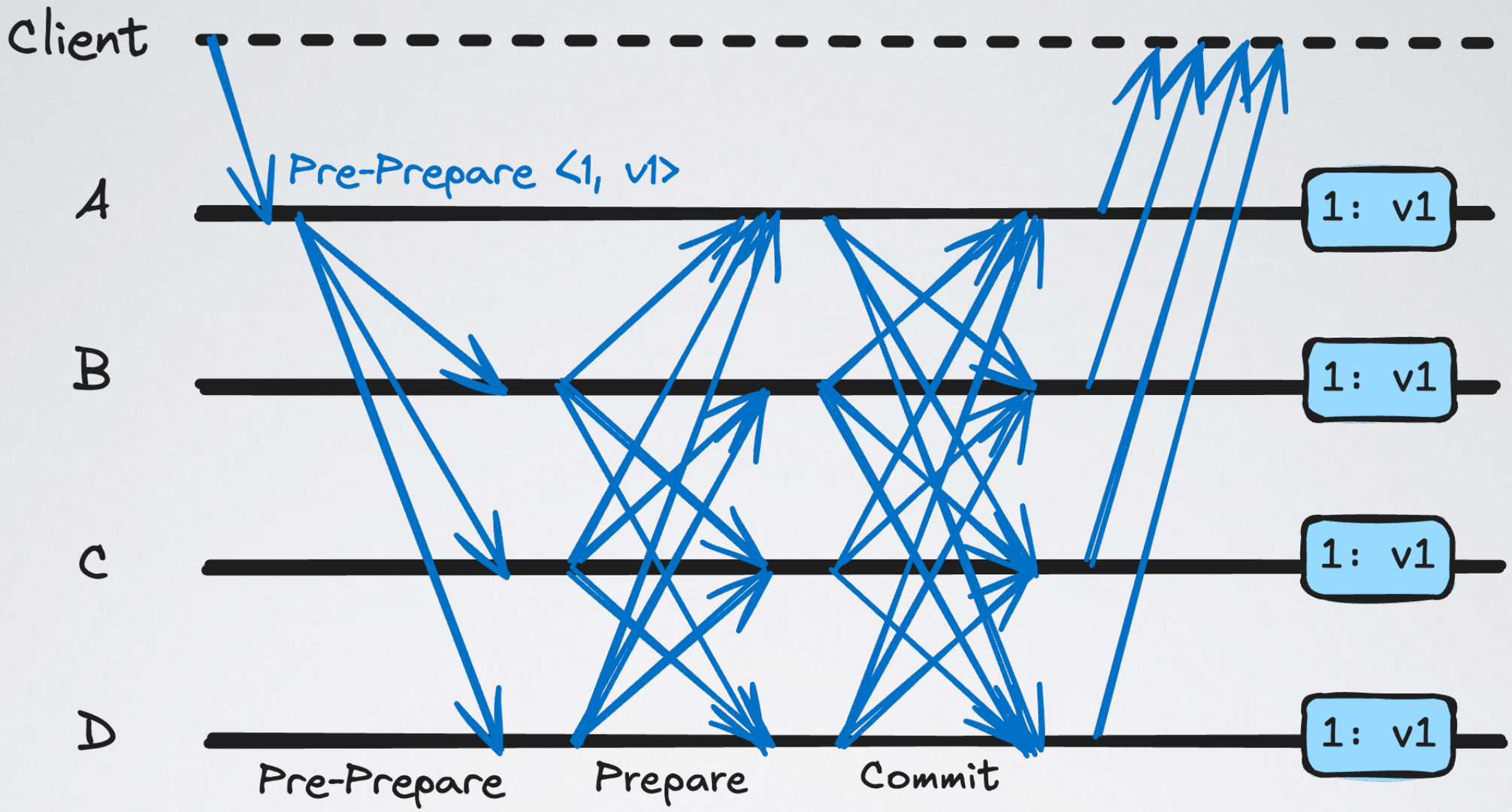
BFT protocols are central to all sorts of critical applications

including Blockchains

BFT PROTOCOLS ARE PRONE TO BUGS

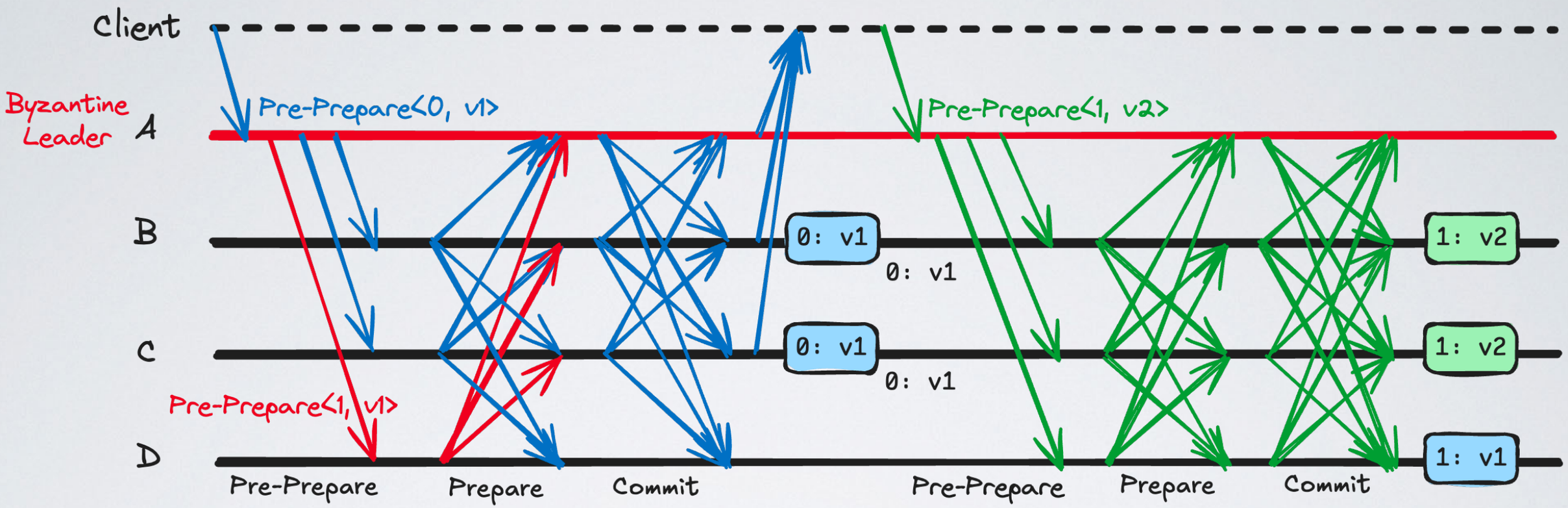
- Bugs in these protocols can lead to **severe security and reliability risks**.

- Despite 40+ years of research and formal proofs, new bugs are still being discovered in real-world BFT systems



PBFT

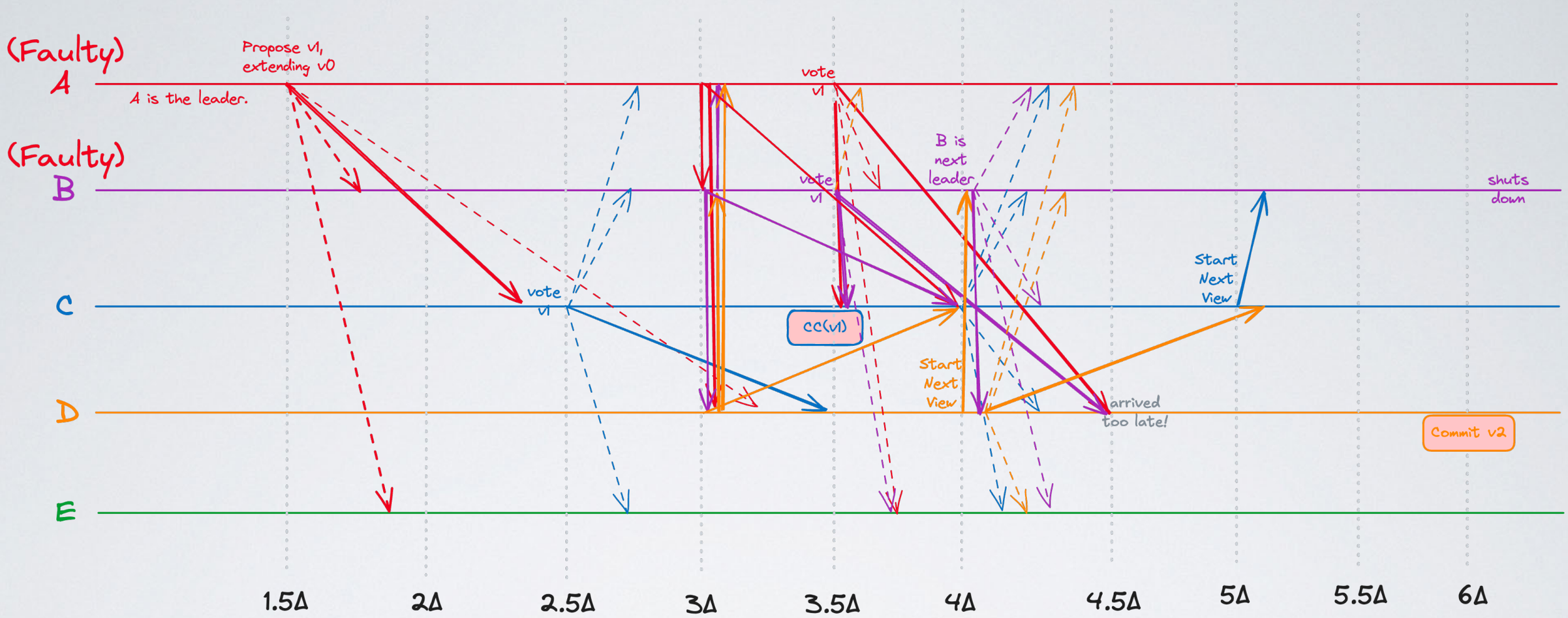
[Miguel Castro, Barbara Liskov, 1999]



BUGS?

PBFT-Java

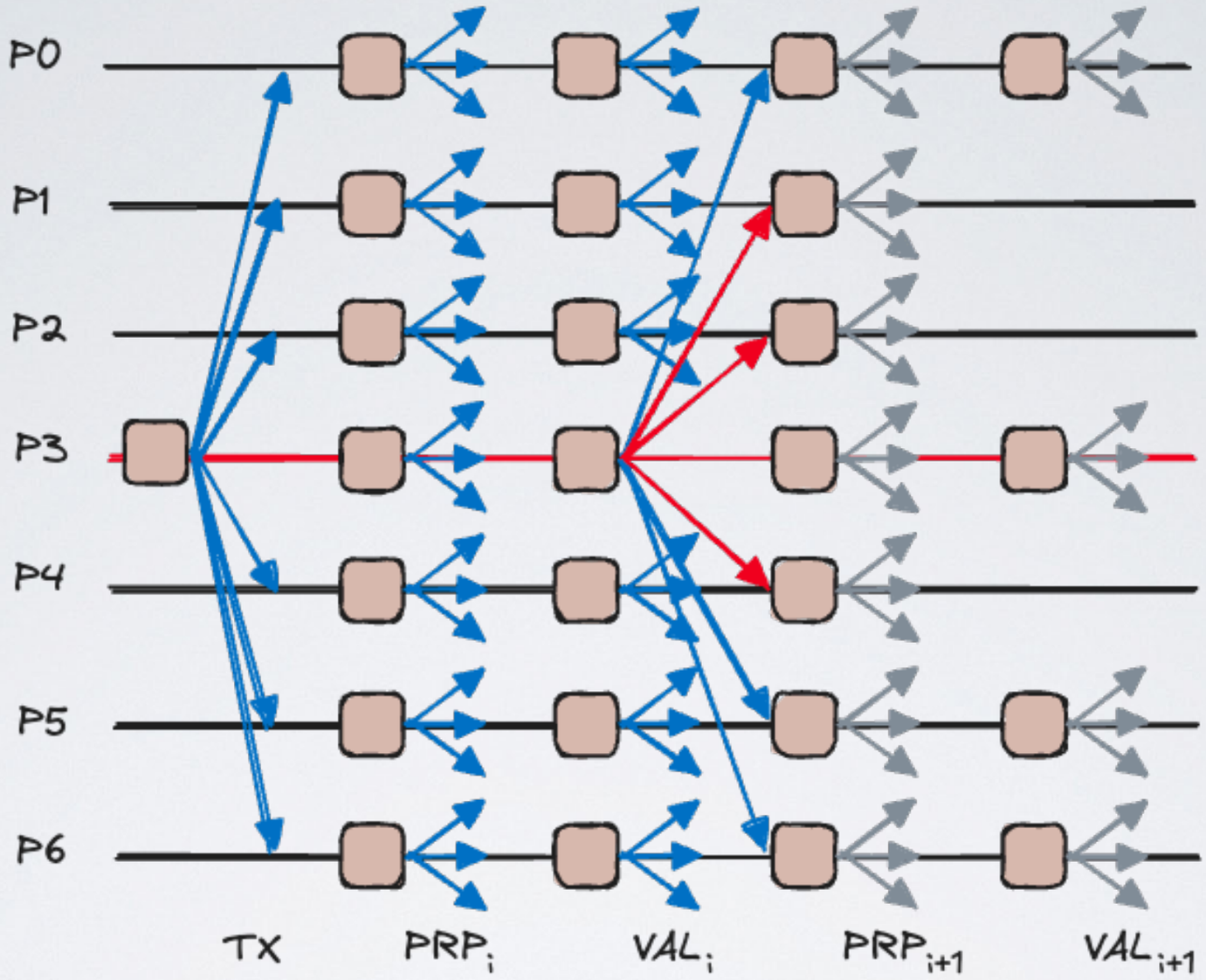
[L. Winter et al., 2023]



BUGS?

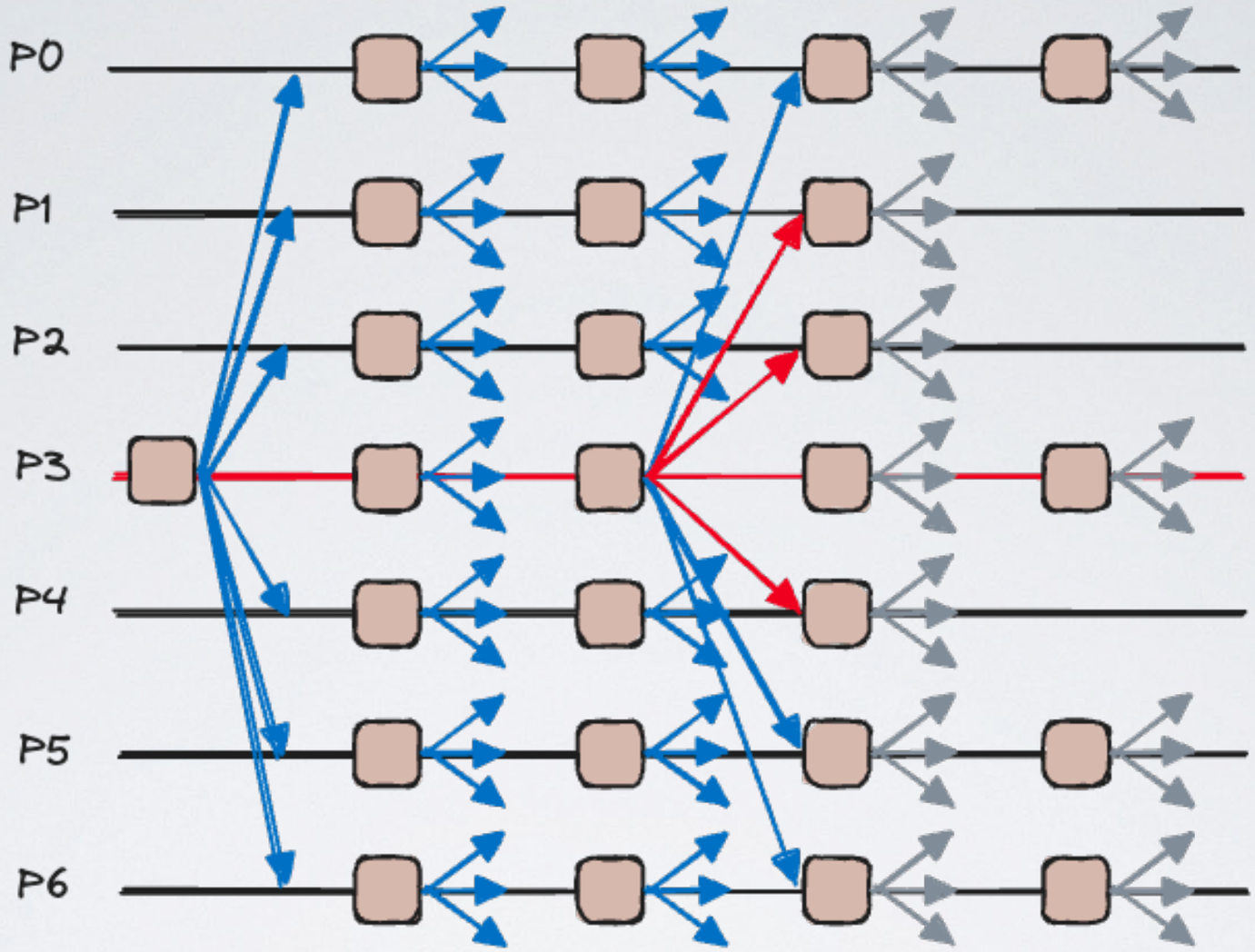
Sync HotStuff

[Atsuki Momose and Jason Paul Cruz 2020]



BUGS IN PRODUCTION BLOCKCHAINS

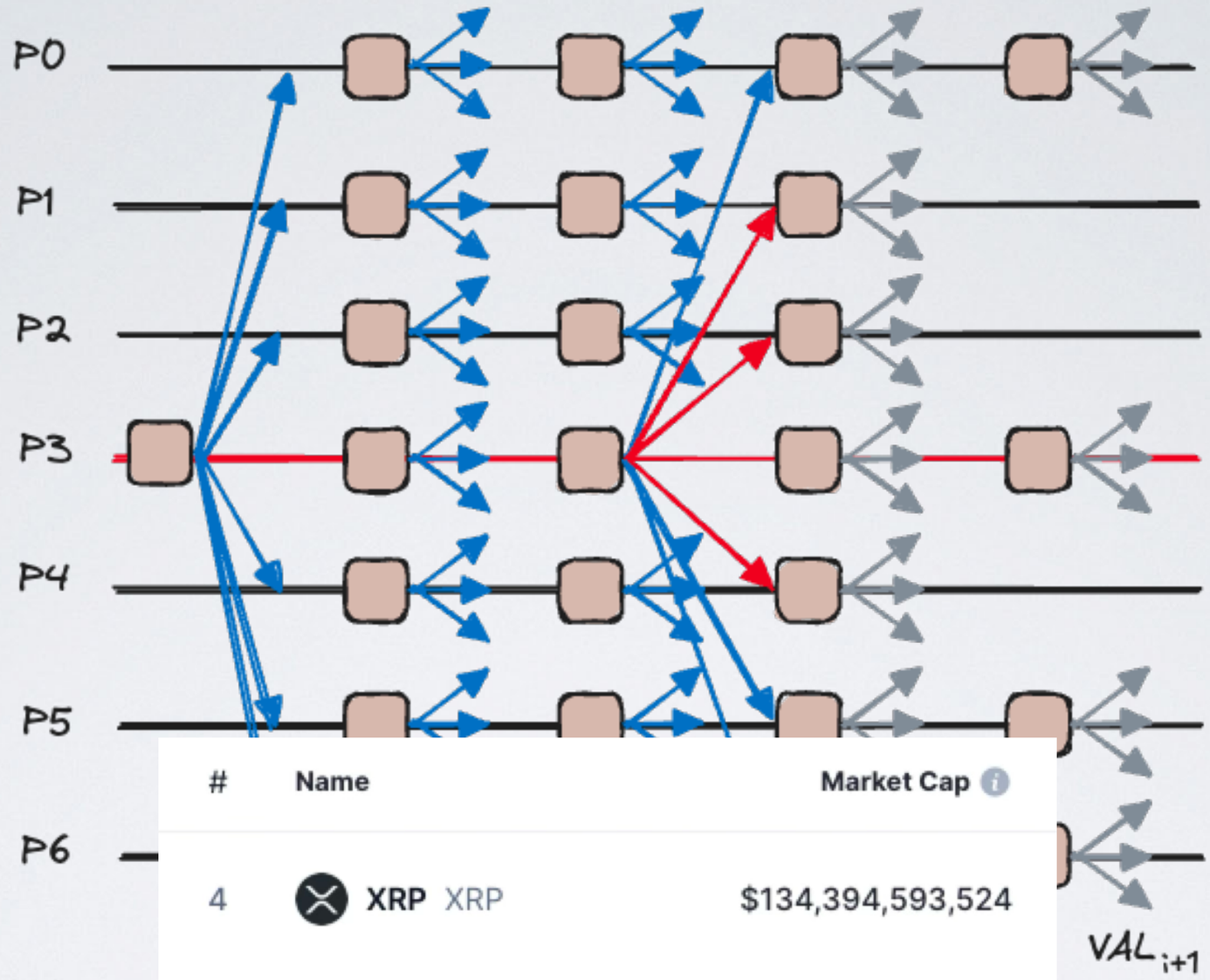
XRPL CP Liveness Violation
[L. Winter et al., 2023]



How was this bug found?

BUGS IN PRODUCTION BLOCKCHAINS

XRPL CP Liveness Violation
[L. Winter et al., 2023]



Levin Winter and Florena Buse win Ripple Bug Bounty Award

NEWS - 25 APRIL 2023 - COMMUNICATION EWI

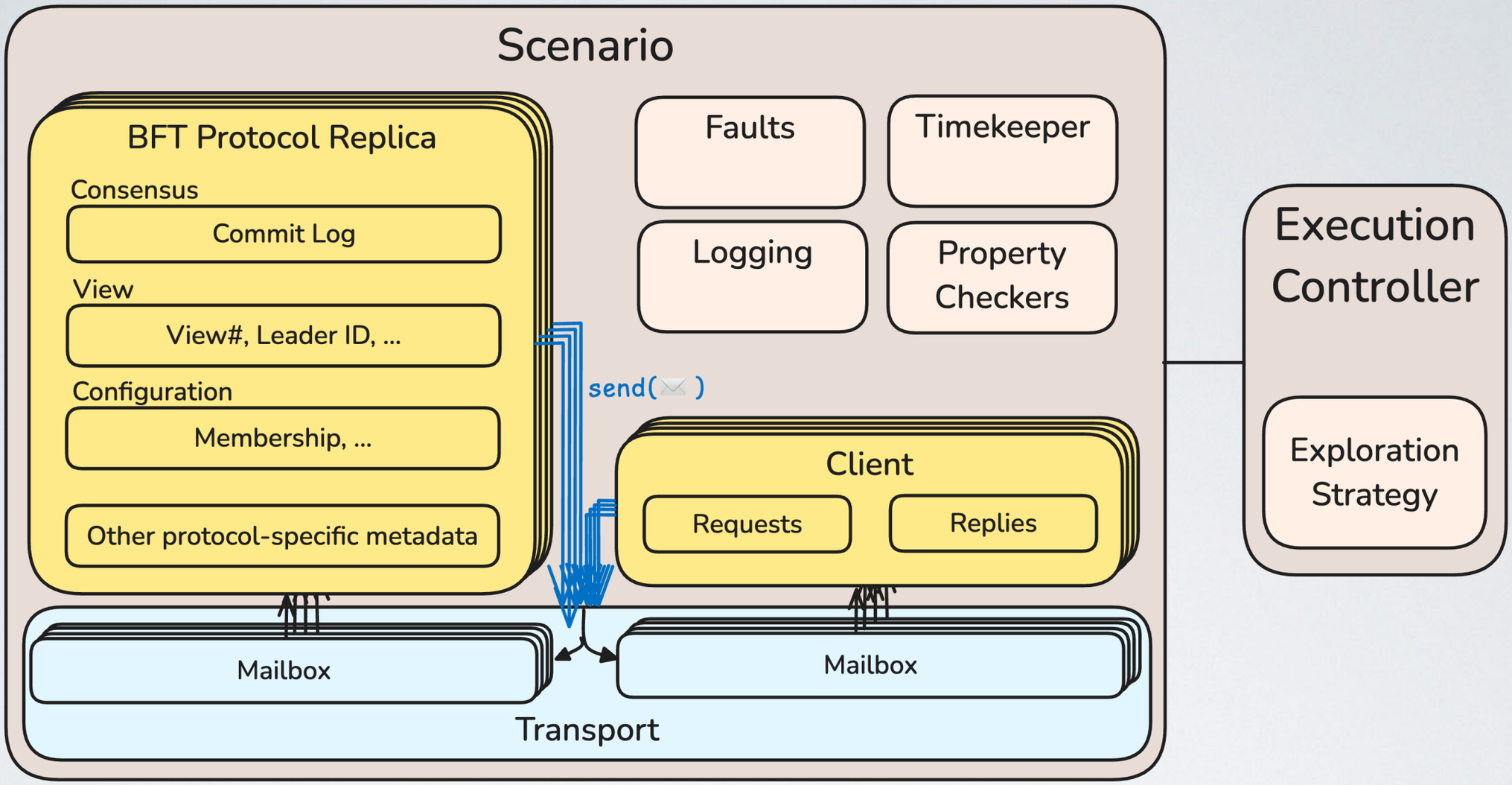
Burcu Özkan's Honours Programme students, Levin Winter and Florena Buse, have been awarded by Ripple's Bug Bounty Program for the new bug they discovered in the XRP Ledger using the programme's new testing method. Read more about their work in "Randomized Testing of Byzantine Fault Tolerant Algorithms" in OOPSLA 2023.

THERE ARE TESTING METHODS BUT…

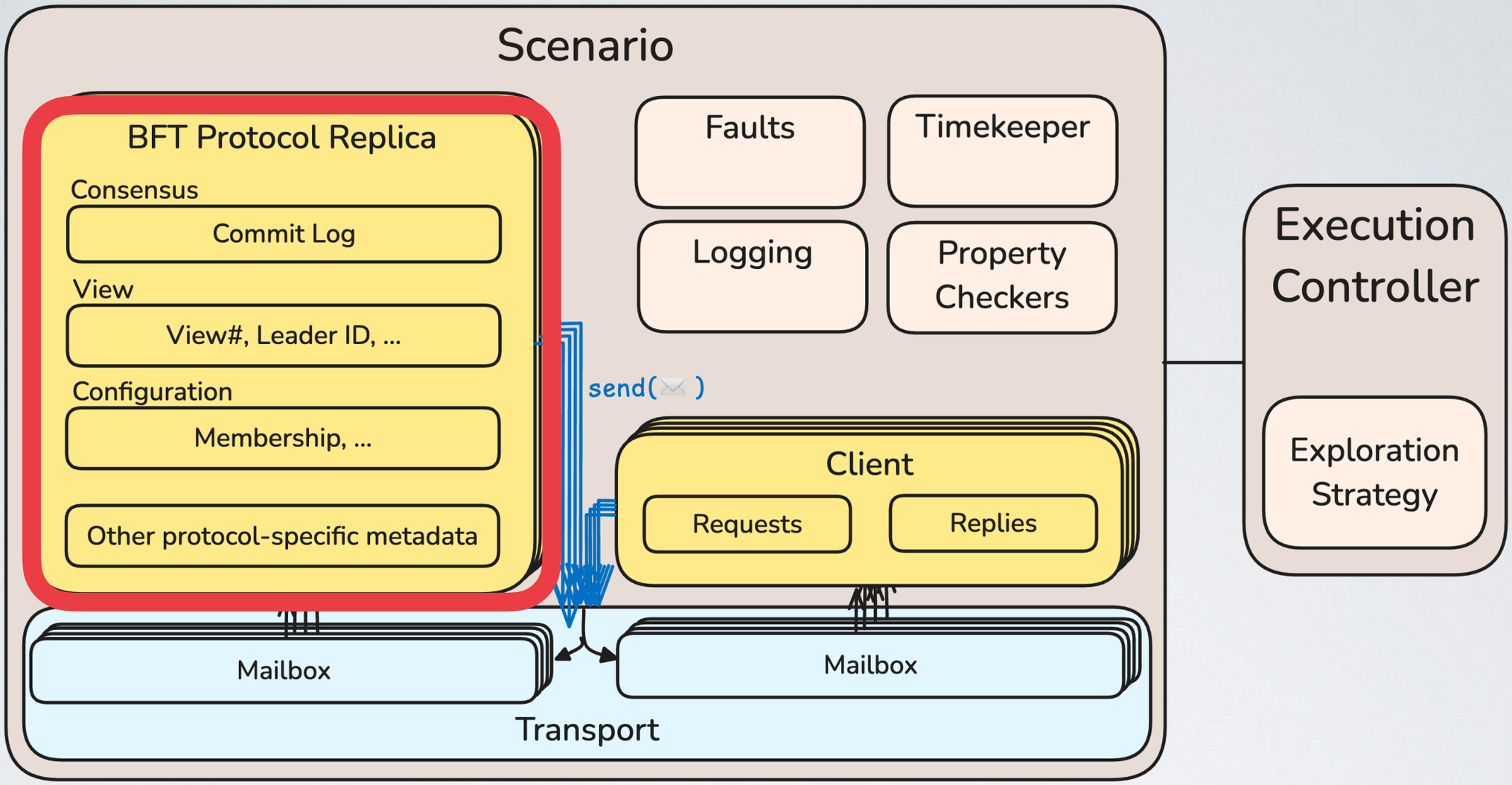
- Existing BFT testing methods are:
  - **System-specific** (e.g. ByzzFuzz for XRPL, Twins for DiemBFT)
  - **Not comparable** (how do they compare?!)
  - **Hard to use** (non-unified interfaces for SUTs; fault injection; non-determinism)

We need a Benchmark Suite for Evaluating BFT Testing Algorithms

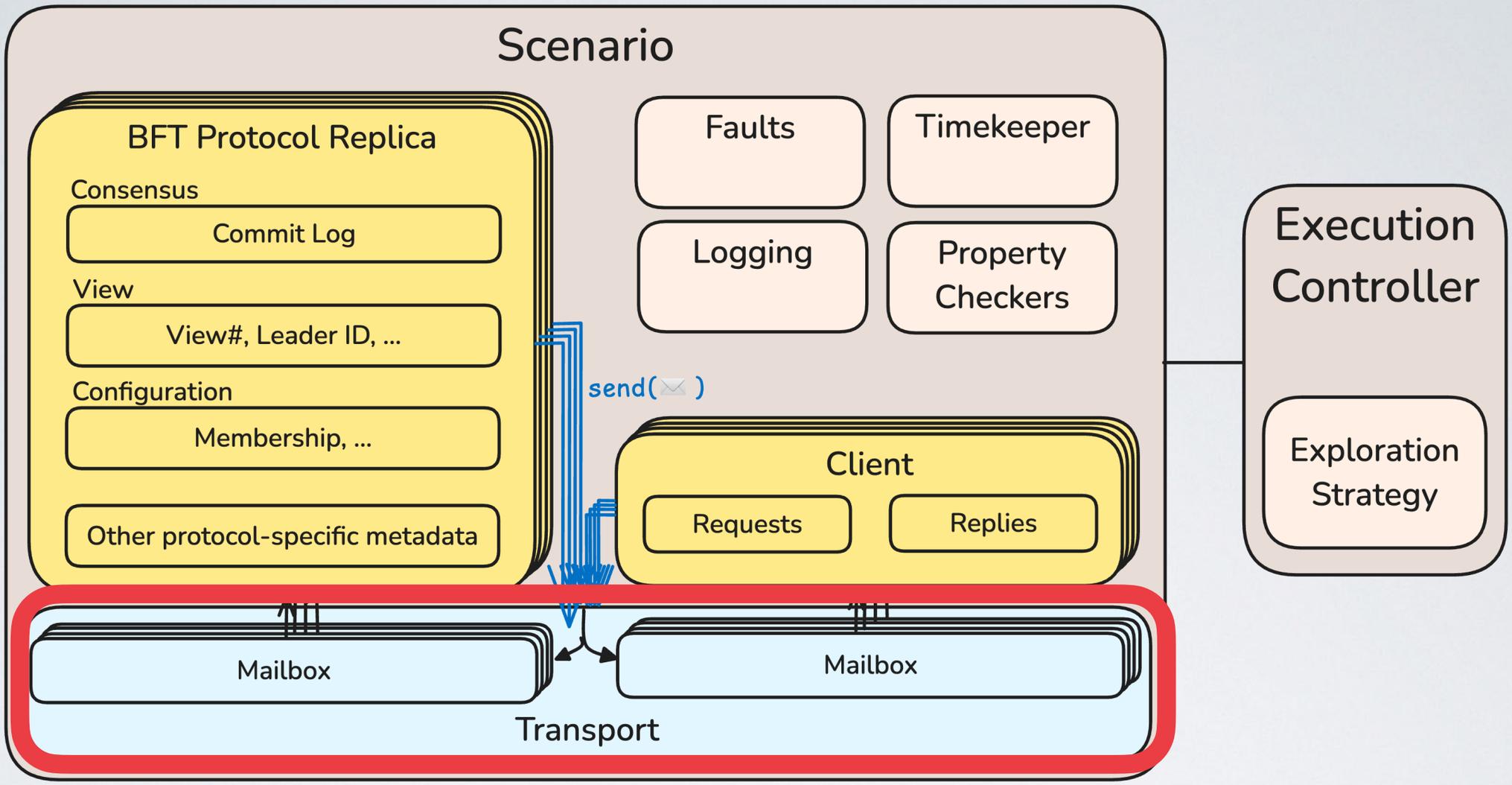
We need a Benchmarking Framework for BFT Testing Algorithms



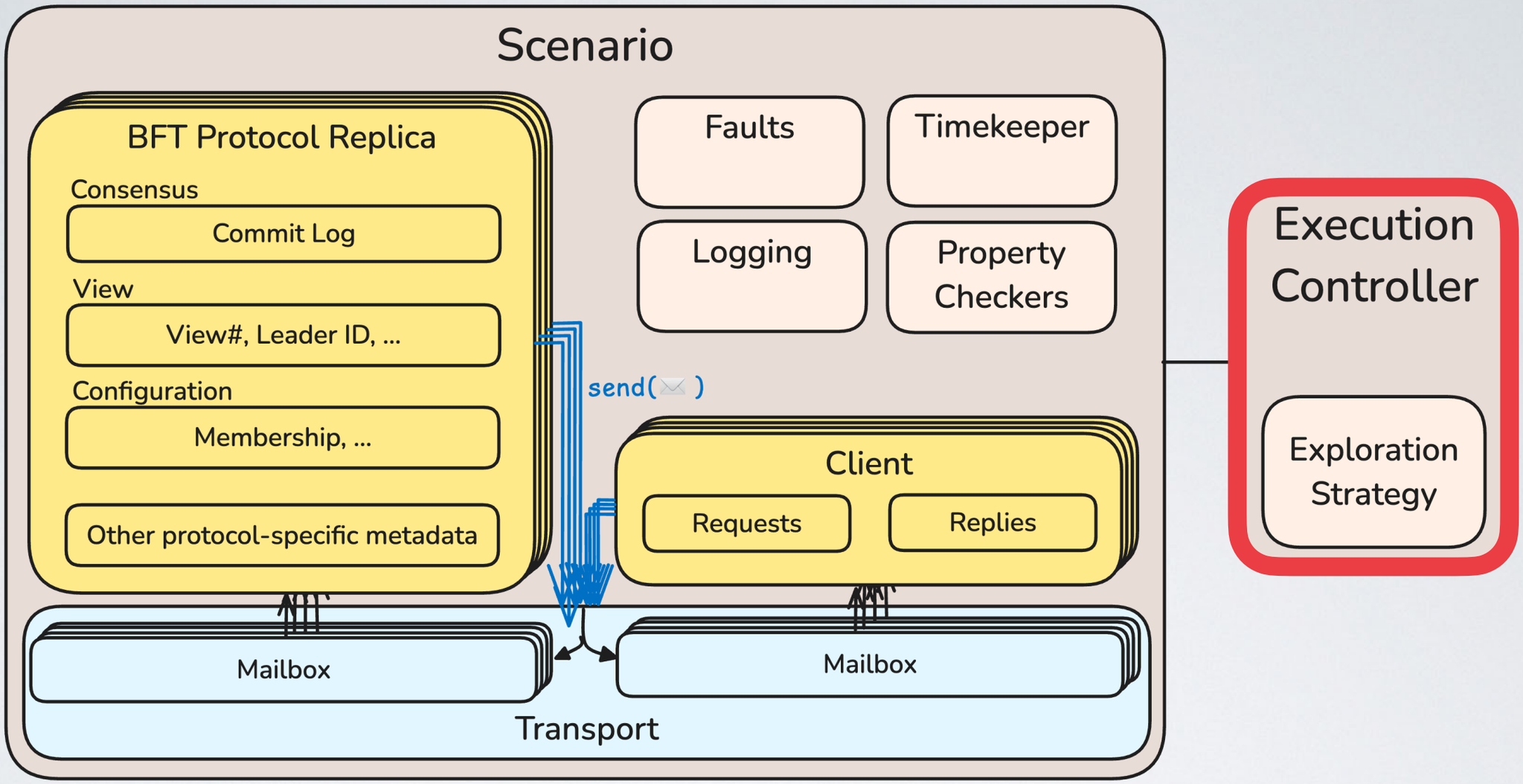
BYZZBENCH ARCHITECTURE



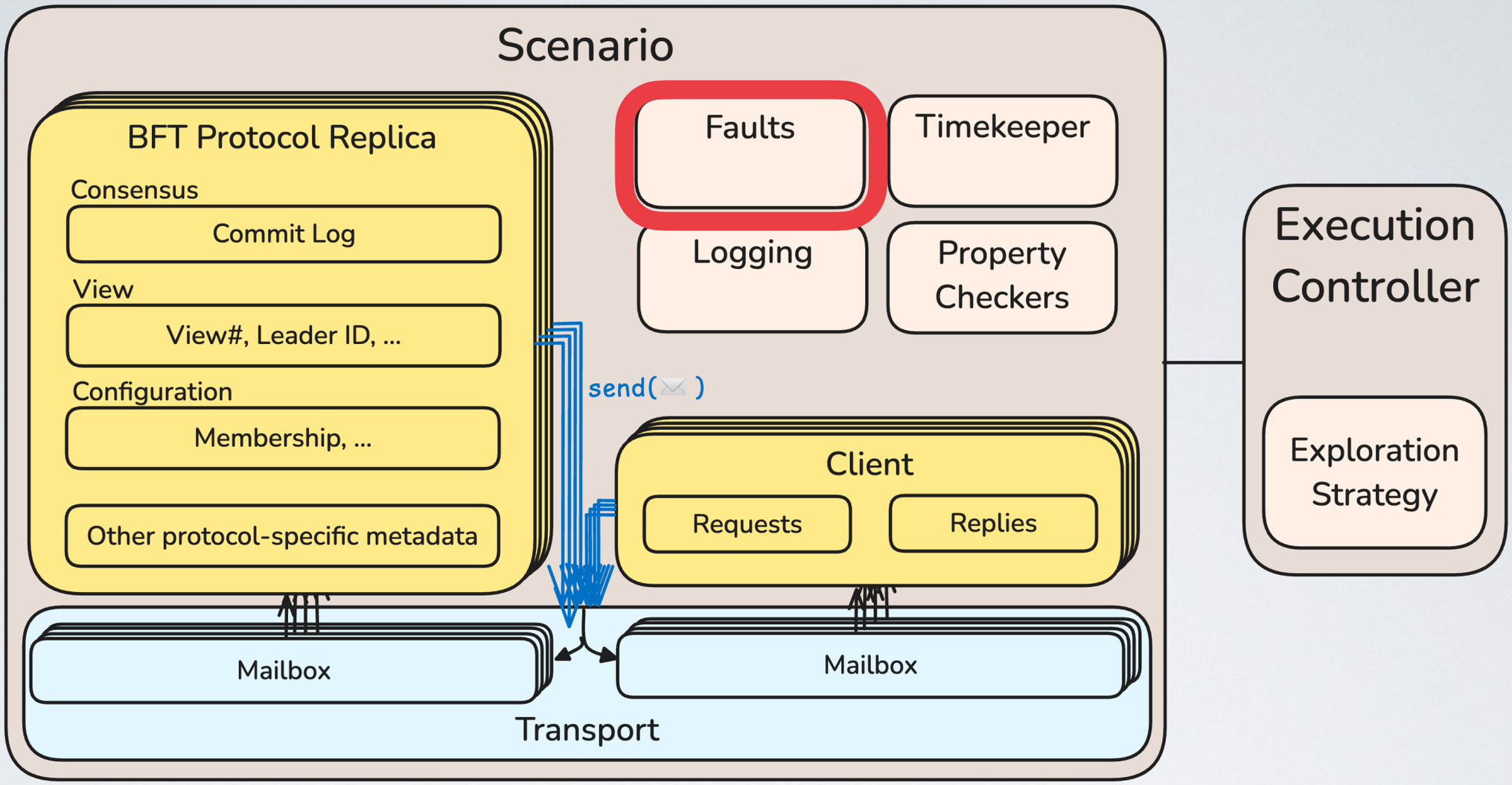
BYZZBENCH ARCHITECTURE



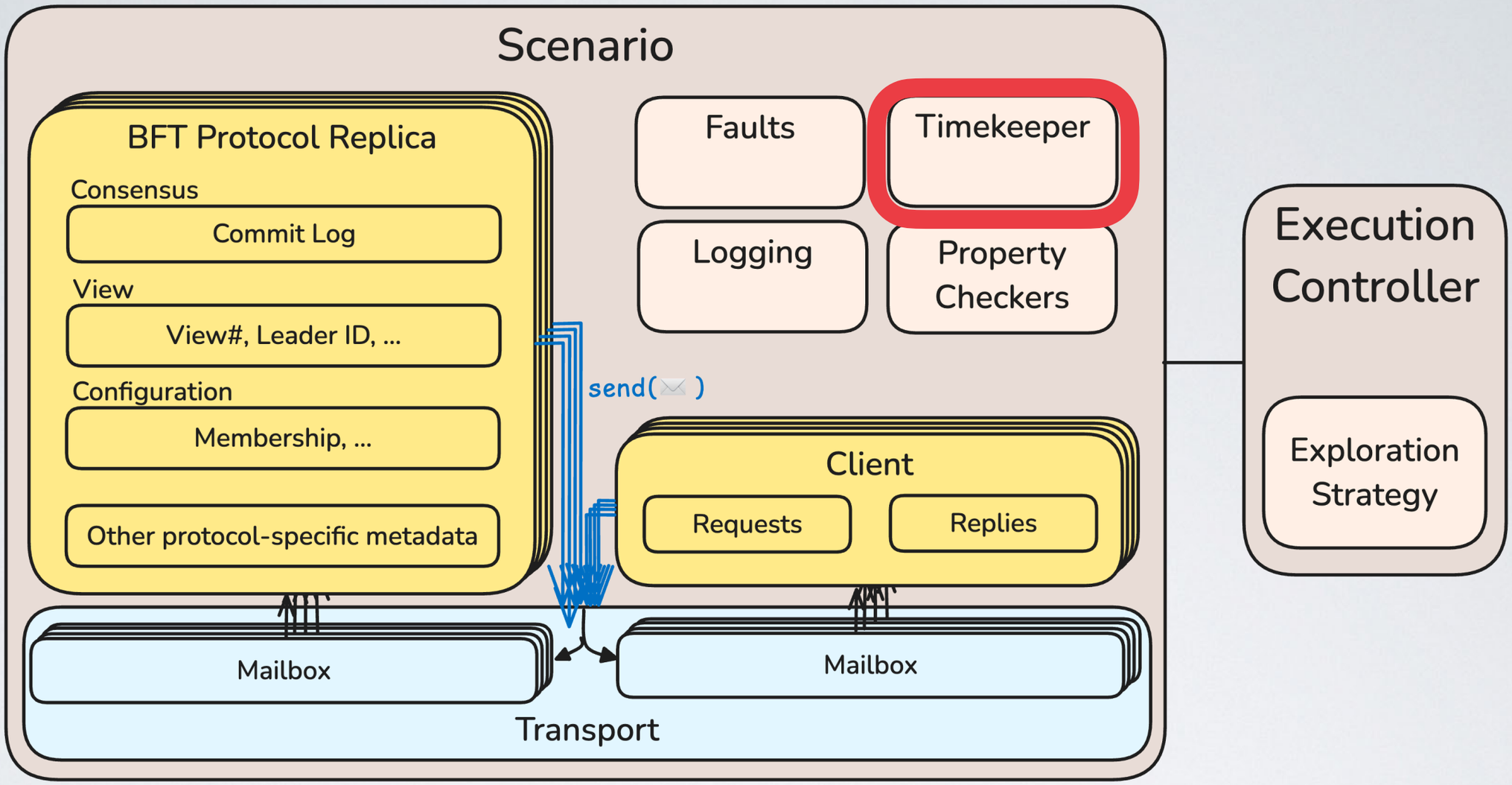
BYZZBENCH ARCHITECTURE



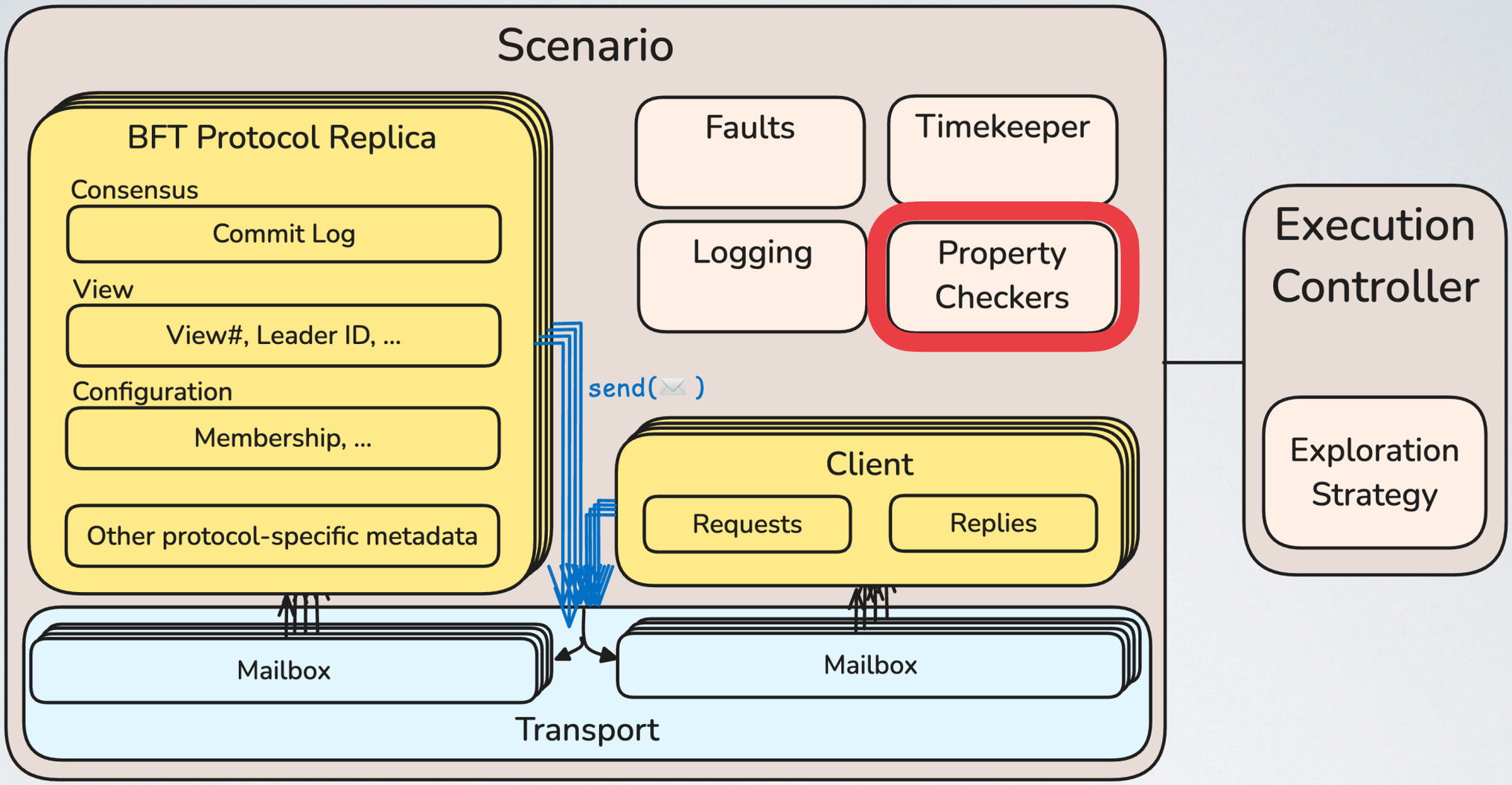
BYZZBENCH ARCHITECTURE



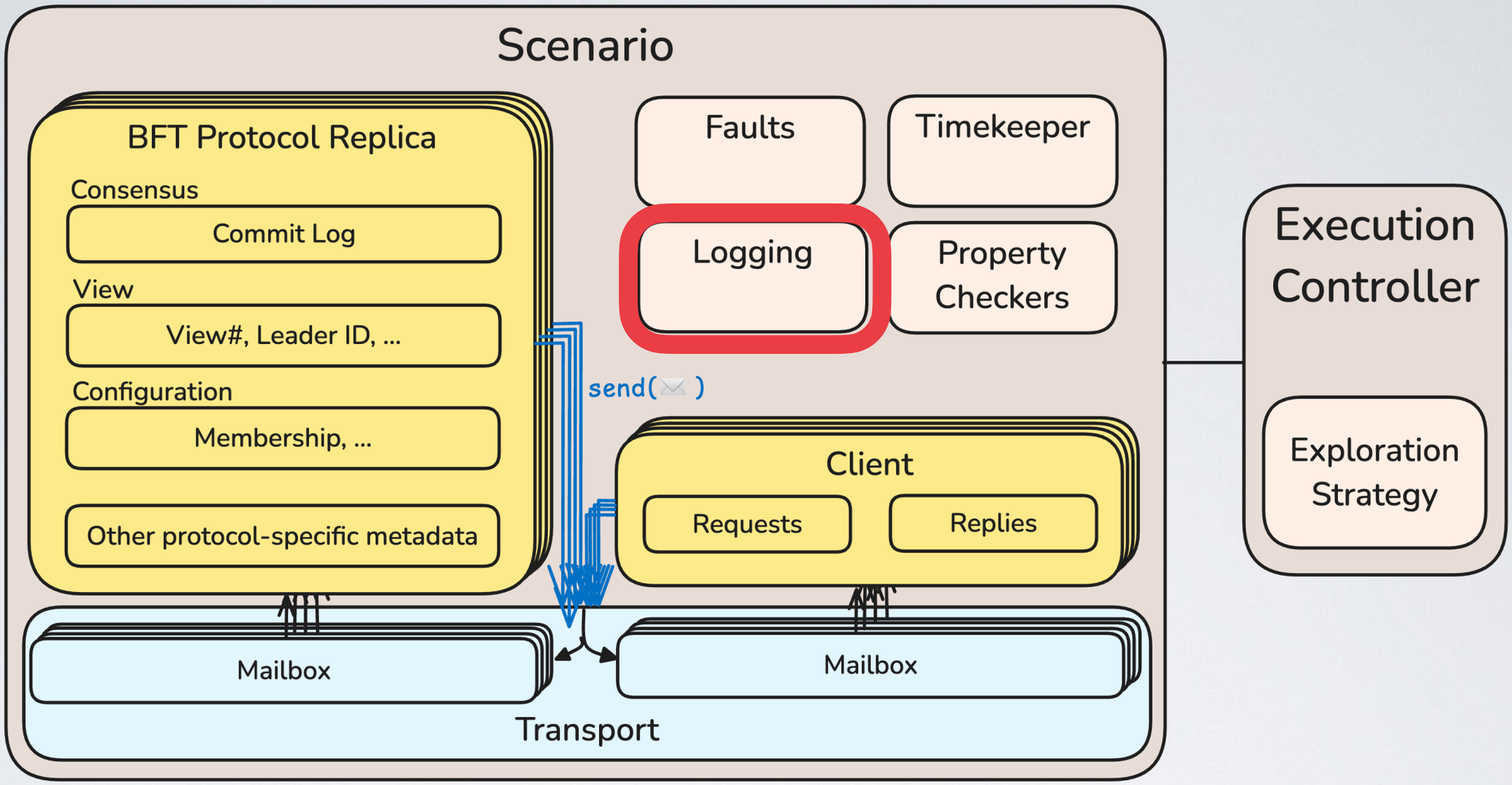
BYZZBENCH ARCHITECTURE



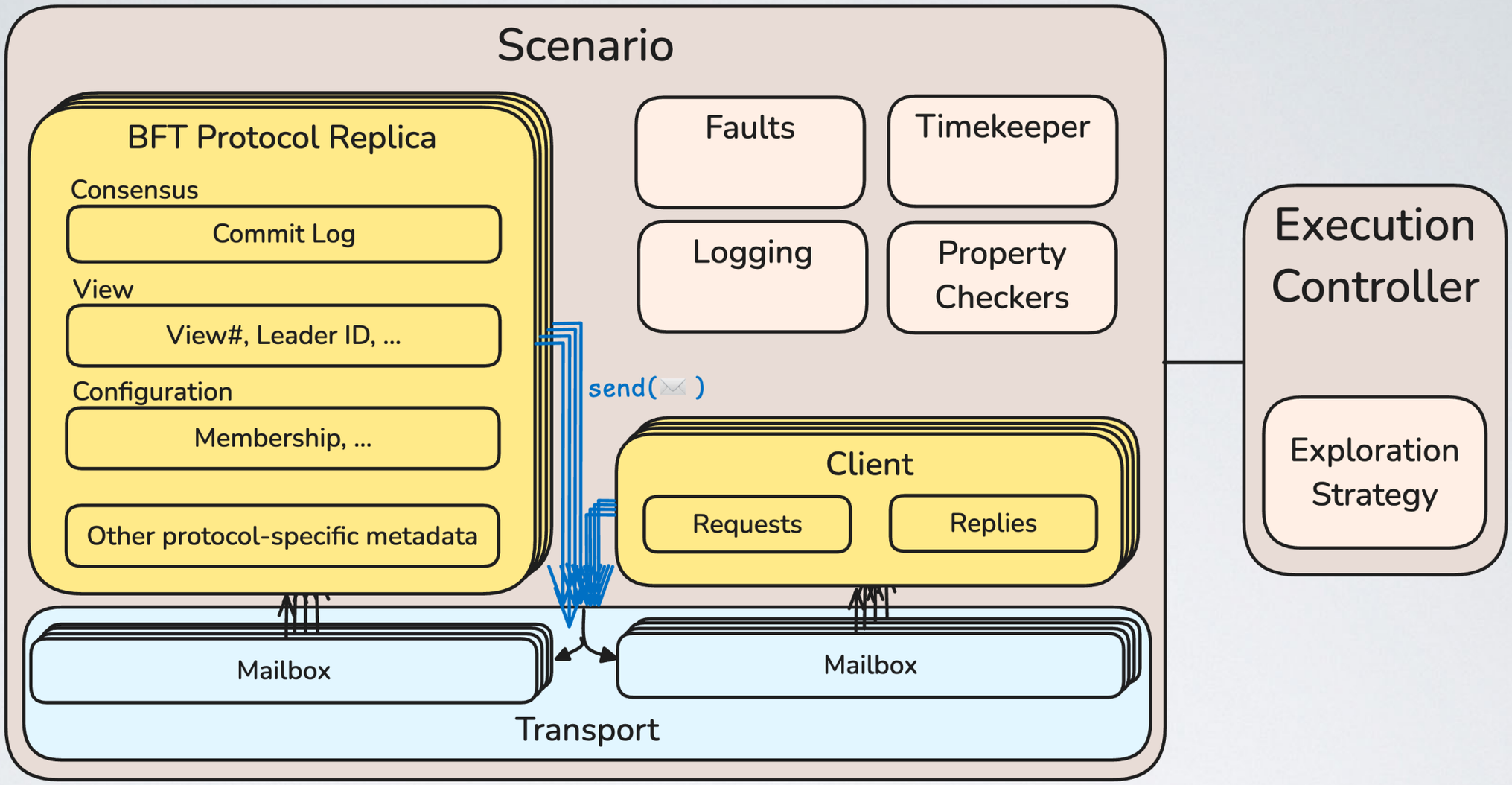
BYZZBENCH ARCHITECTURE



BYZZBENCH ARCHITECTURE

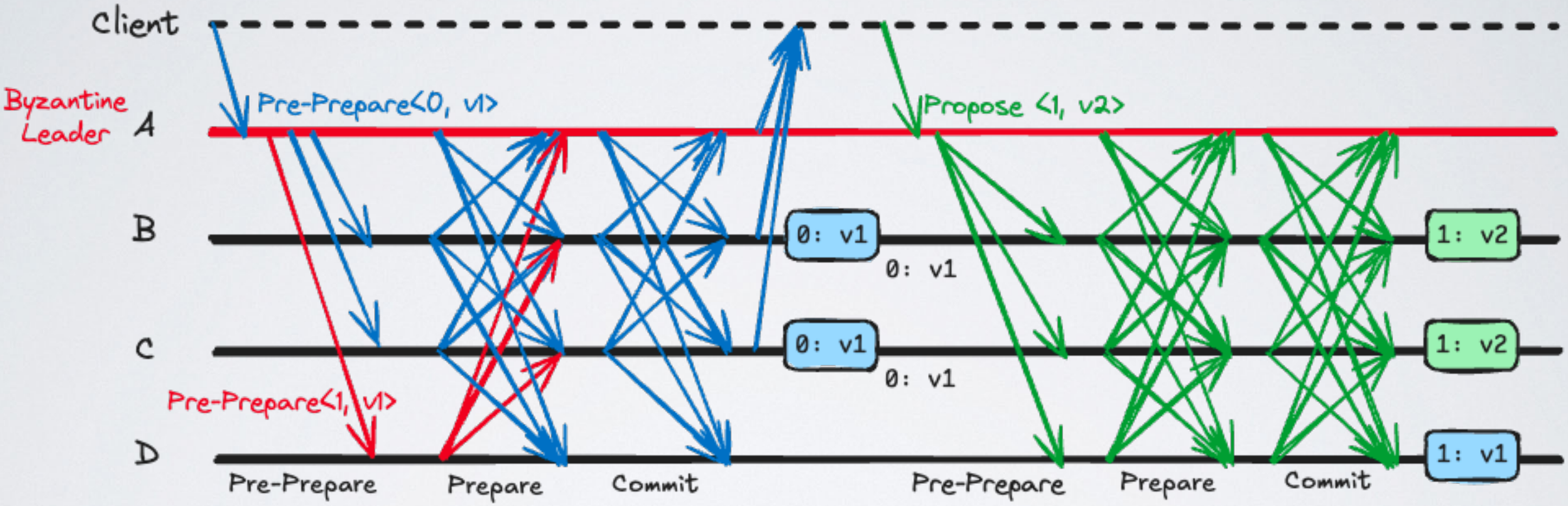


BYZZBENCH ARCHITECTURE



BYZZBENCH ARCHITECTURE

PRELIMINARY EVALUATION



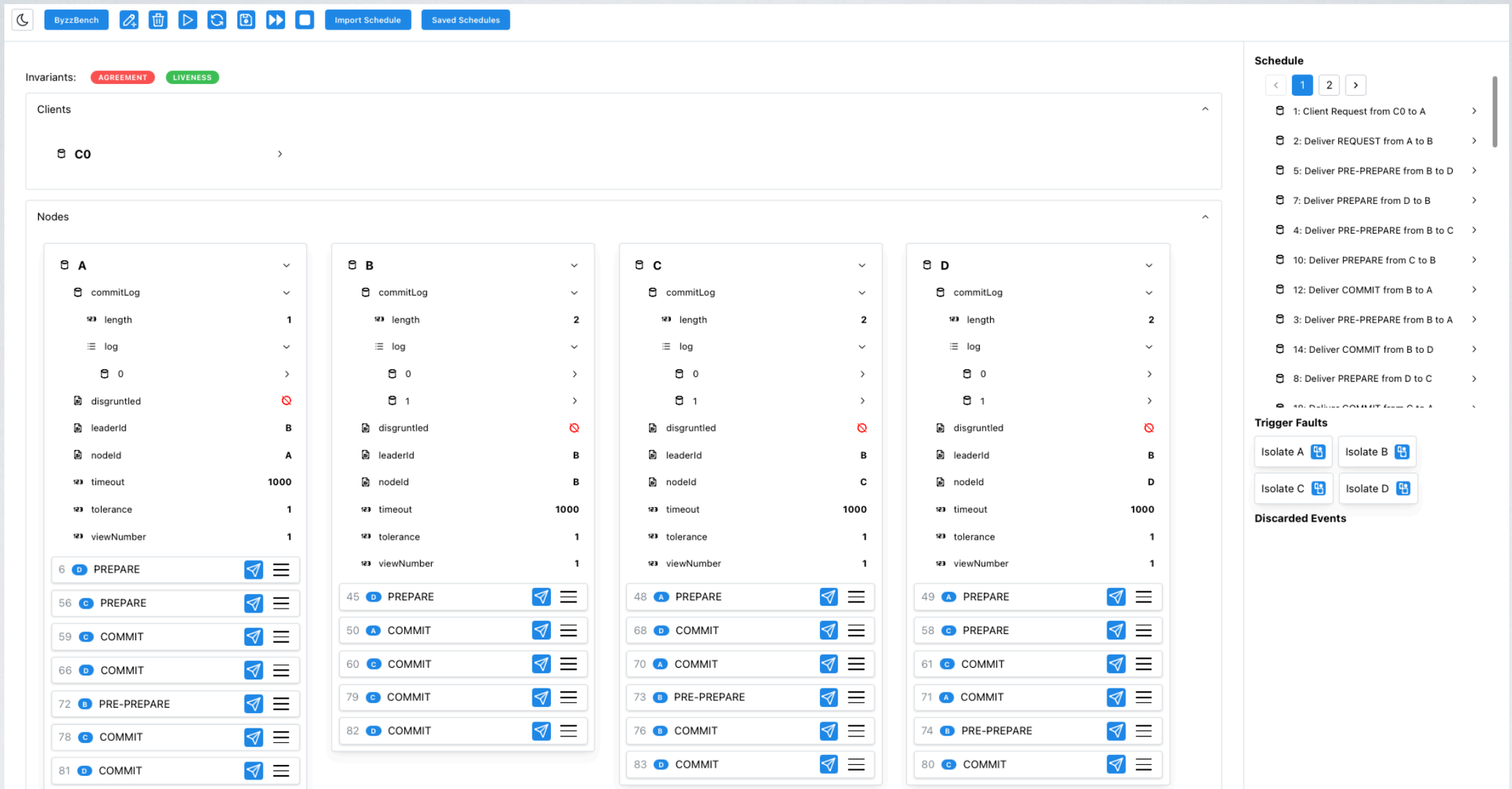
Random	BF(0,1)	BF(0,2)	BF(1,0)	BF(1,1)	BF(1,2)	BF(2,0)	BF(2,1)	BF(2,2)
0.0%	0.0%	0.0%	7.1%	5.4%	5.0%	11.6%	9.2%	9.5%

% scenarios with bugs

ONGOING WORK

- Extend the framework to a comprehensive benchmark suite of BFT protocols with known bugs

Year	Protocol	Violation	Bug Source	Execution Parameters				
				#processes	#views (or blocks)	#process faults	#network faults	
2017	FaB	liveness	protocol	4	2	1	2	✓
2017	Zyzzyva	safety	protocol	4	3	1	4	
2019	hBFT	safety	protocol	4	2	2	2	✓
2020	Sync HotStuff	safety	protocol	5	3	2	7	
2020	XRPL	liveness	trust config	7	2	1	0	
2020	XRPL	safety	trust config	7	2	2	0	
2021	PBFT	liveness	protocol	4	1	2	0	
2022	Fast-HotStuff	safety	protocol	4	11	0	3	
2023	PBFT	safety	pbft-java	4	2	1	0	✓
2023	XRPL	liveness	rippled v1.7.2	7	3	1	0	



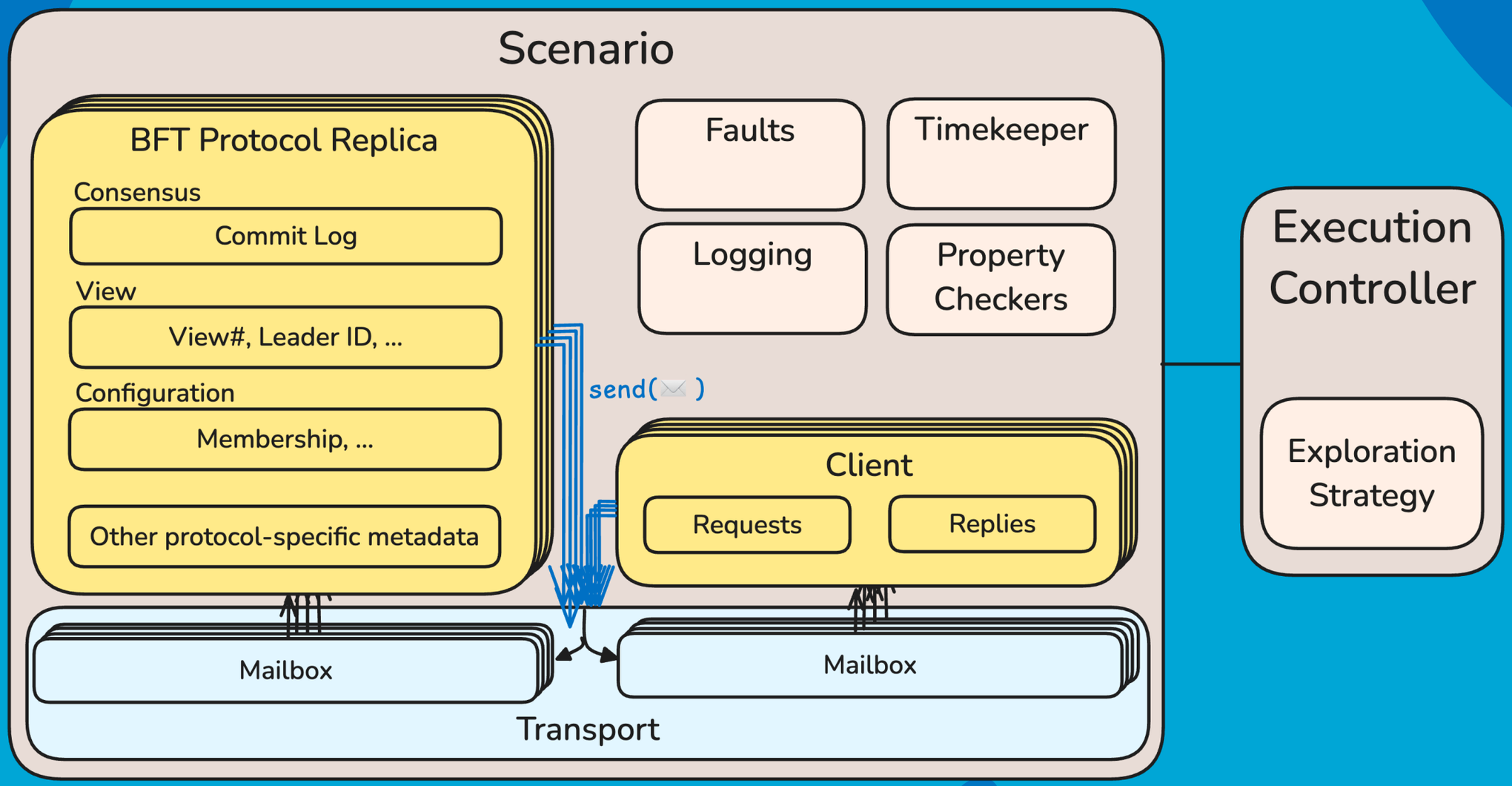
BYZZBENCH UI

BYZZBENCH SUMMARY

- A framework to implement BFT protocols and plug-and-play testing strategies.
- Built-in **controlled fault injection** and **property checkers**
- Allows for the **evaluation of different testing algorithms** for BFT protocols
- **First step** towards a **benchmark suite of BFT protocol bugs.**

ByzzBench

A Benchmark Framework for BFT Testing Algorithms



github.com/joaomlneto/byzzbench