

On the way to Omniledger: adding transaction batching and ByzcoinX to skipchains

Raphaël Dunant

DEDIS lab

Supervisor: Linus Gasser

Responsible: Prof. Bryan Ford

Omniledger improvements

- Block state storing



- Transaction Batching



- Improve decentralised Signing

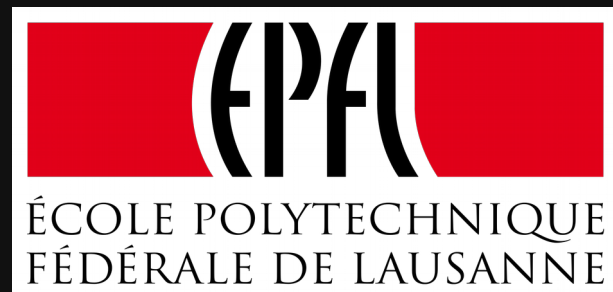


Summary

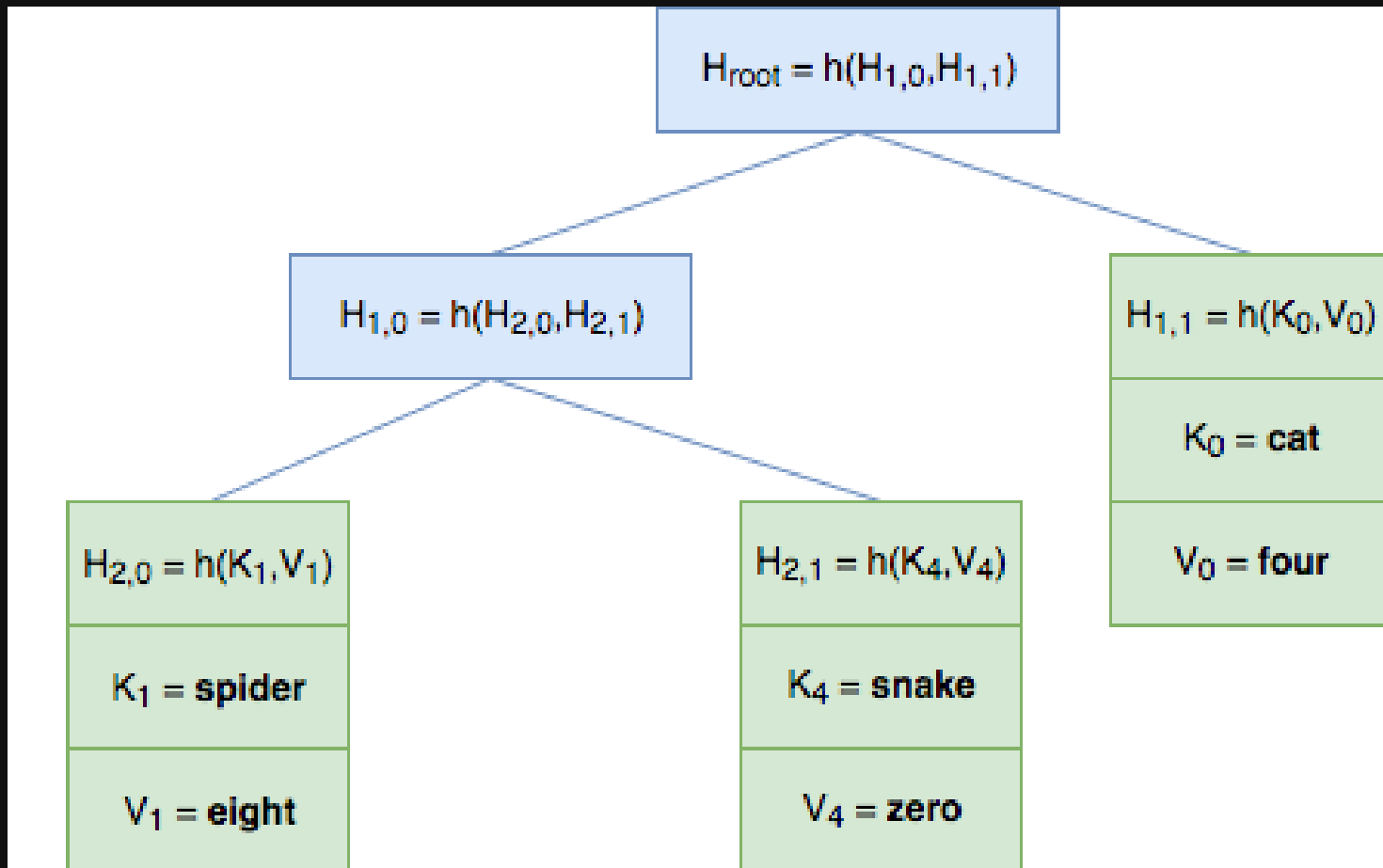
- Introduction (done)
- Collections
- ByzcoinX Quick Answers
- Simulation results
- Conclusion (results, lessons learned, etc.)

The logo for DEDIS, consisting of the word "DEDIS" in a bold, black, sans-serif font centered within a white square.

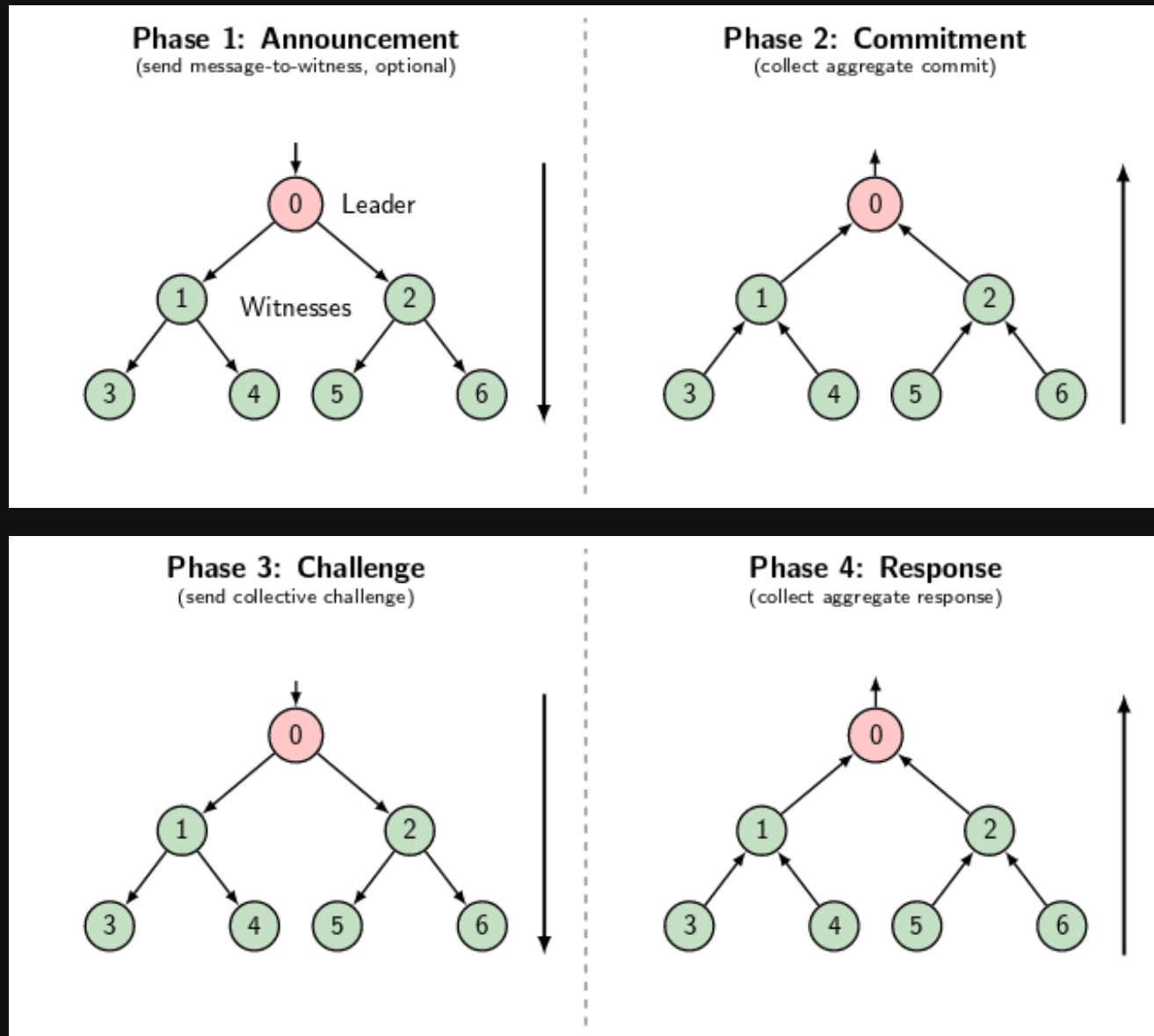
DEDIS



Patricia Merkle Tree



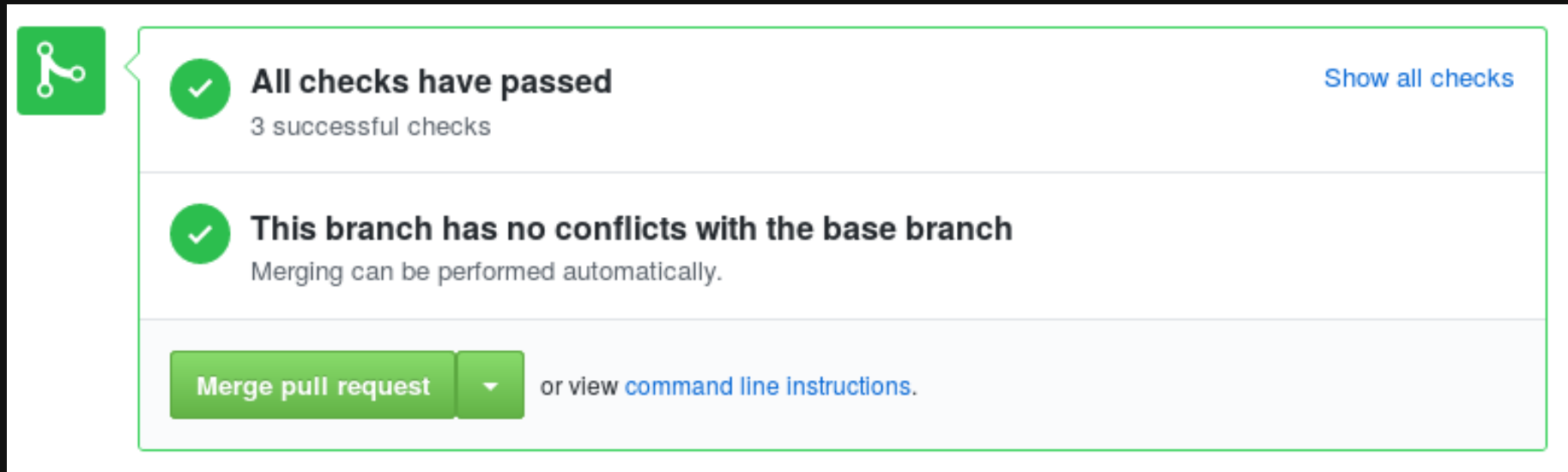
ByzcoinX: Decentralized Witness Cosigning




Objectives

- Understand and Document complete collection library
- Improve drastically running time of ByzcoinX
- Have nice, documented, tested code

Collections Code Cleaning



A screenshot of a GitHub pull request status card. On the left is a green icon with a white branching diagram. The main area contains two green checkmark icons. The first checkmark is followed by the text "All checks have passed" and "3 successful checks". To the right of this is a blue link "Show all checks". The second checkmark is followed by the text "This branch has no conflicts with the base branch" and "Merging can be performed automatically.". At the bottom, there is a green button labeled "Merge pull request" with a dropdown arrow, followed by the text "or view [command line instructions](#)."

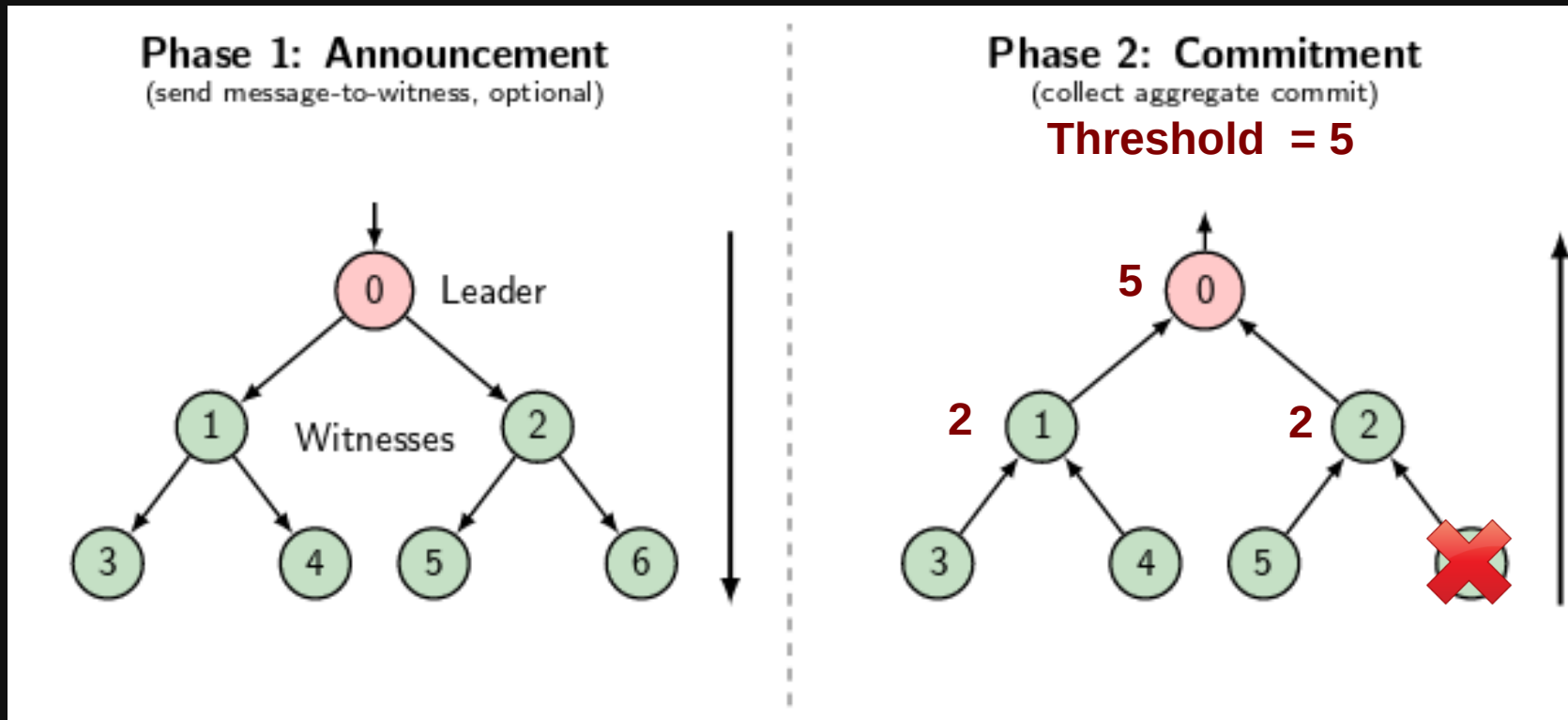
 **All checks have passed** [Show all checks](#)
3 successful checks

This branch has no conflicts with the base branch
Merging can be performed automatically.

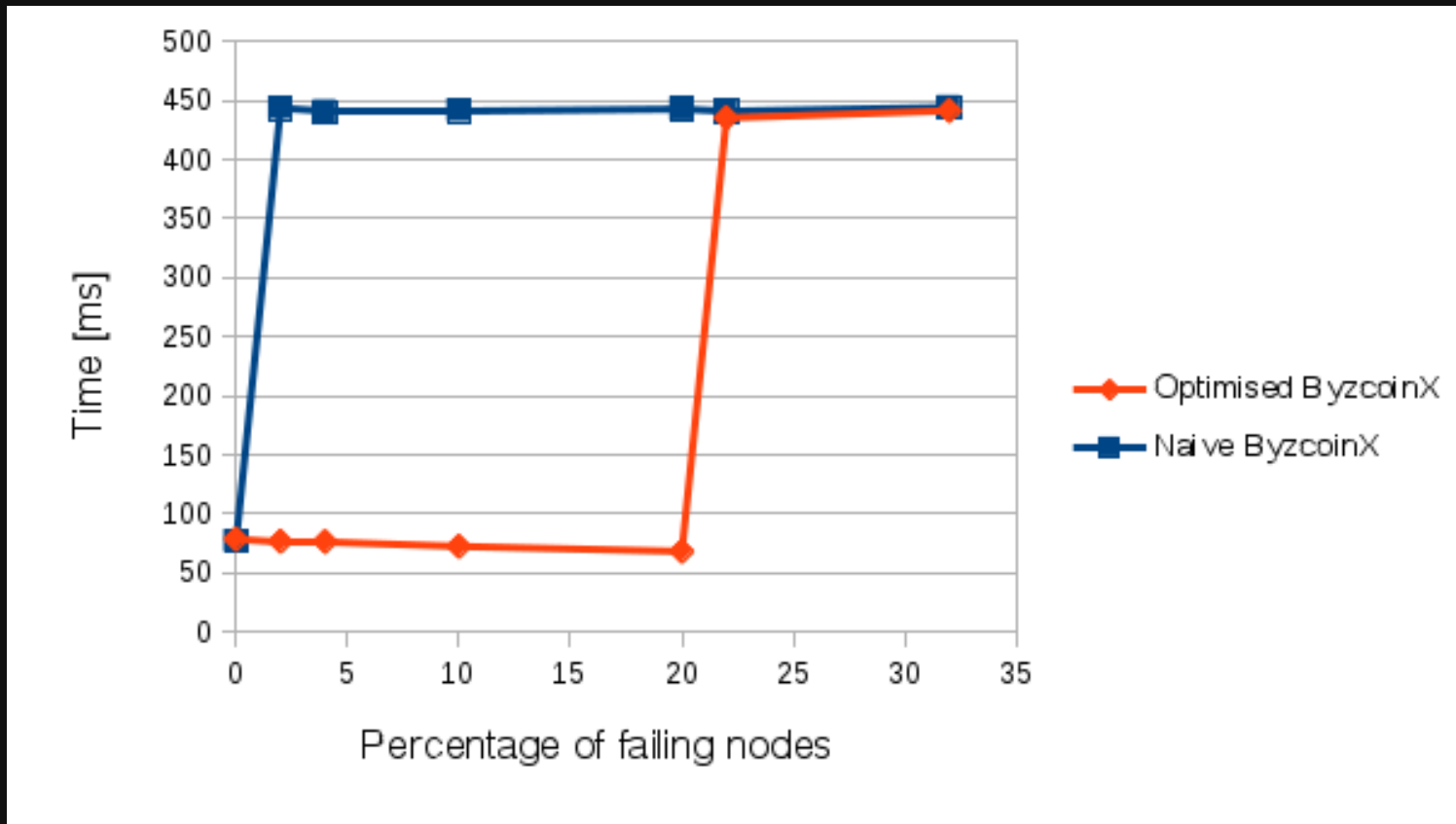
Merge pull request or view [command line instructions](#).

```
1 // Package collection is a Merkle-tree based data structure to securely and
2 // verifiably store key / value associations on untrusted nodes. The library
3 // in this package focuses on ease of use and flexibility, allowing to easily
4 // develop applications ranging from simple client-server storage to fully
5 // distributed and decentralized ledgers with minimal bootstrapping time.
6 package collection
7
8 // Collection represents the Merkle-tree based data structure.
9 // The data is defined by a pointer to its root.
10 type Collection struct {
11     root *node
12     fields []Field
```

ByzcoinX Quick Answers



Simulation results



- 50 Nodes, 6 Subleaders, Default Leafs Timeout: 417ms, average of 10 tries
- Threshold: $\lceil \frac{2}{3} \cdot 50 \rceil = 34$
- 2.9 GHz, 4 Core, 4MB cache, 8GB DDR3-1600 RAM

Future work

- Collections
 - Store on hard drive
 - Handle transactions conflicts more finely
- ByzCoinX
 - Add backward Compatibility
 - Rework Timeouts
 - Improve security
- Add more unit tests



Conclusion

- Complete, working and reusable collections code
- Quick ByzcoinX performances
- Will be used in production
- Scalable and tested
- Can still get better
- Personal improvement