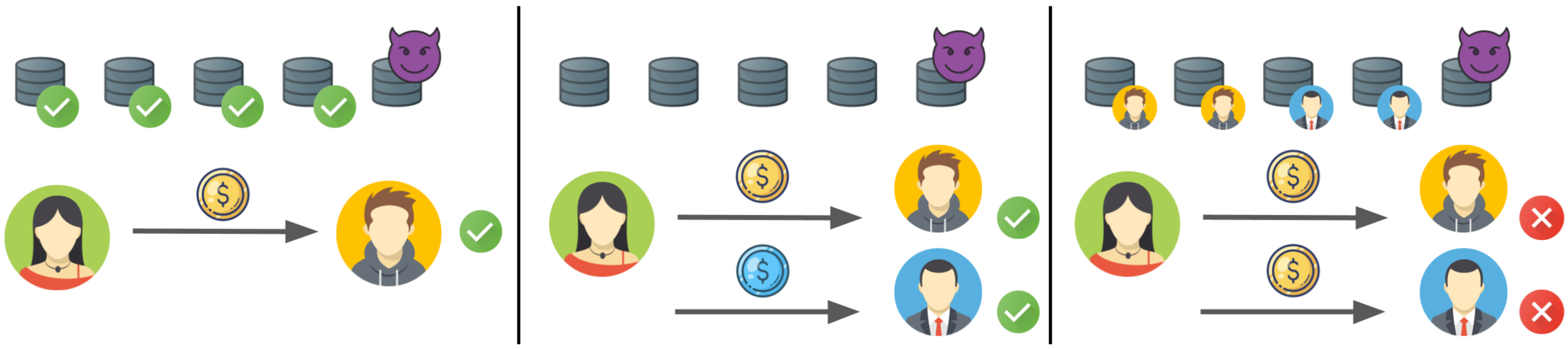
Consensus on Demand

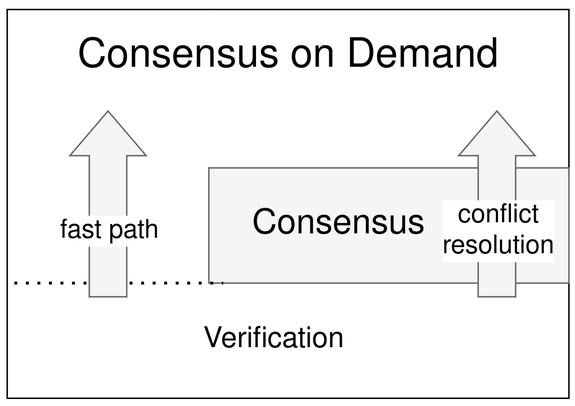
Jakub Sliwinski, Yann Vonlanthen, Roger Wattenhofer
ETH Zurich, Switzerland

A Broadcast-Based Currency



Combining the Best of Both Worlds

Whenever possible we use fast verification. Only if there is a (non-resolvable) conflict, we use consenus to decide which transaction should be accepted.



The main challenge is to make sure that all correct servers accept the same transaction, whether they use the fast path or not.

Comparison to Related Work

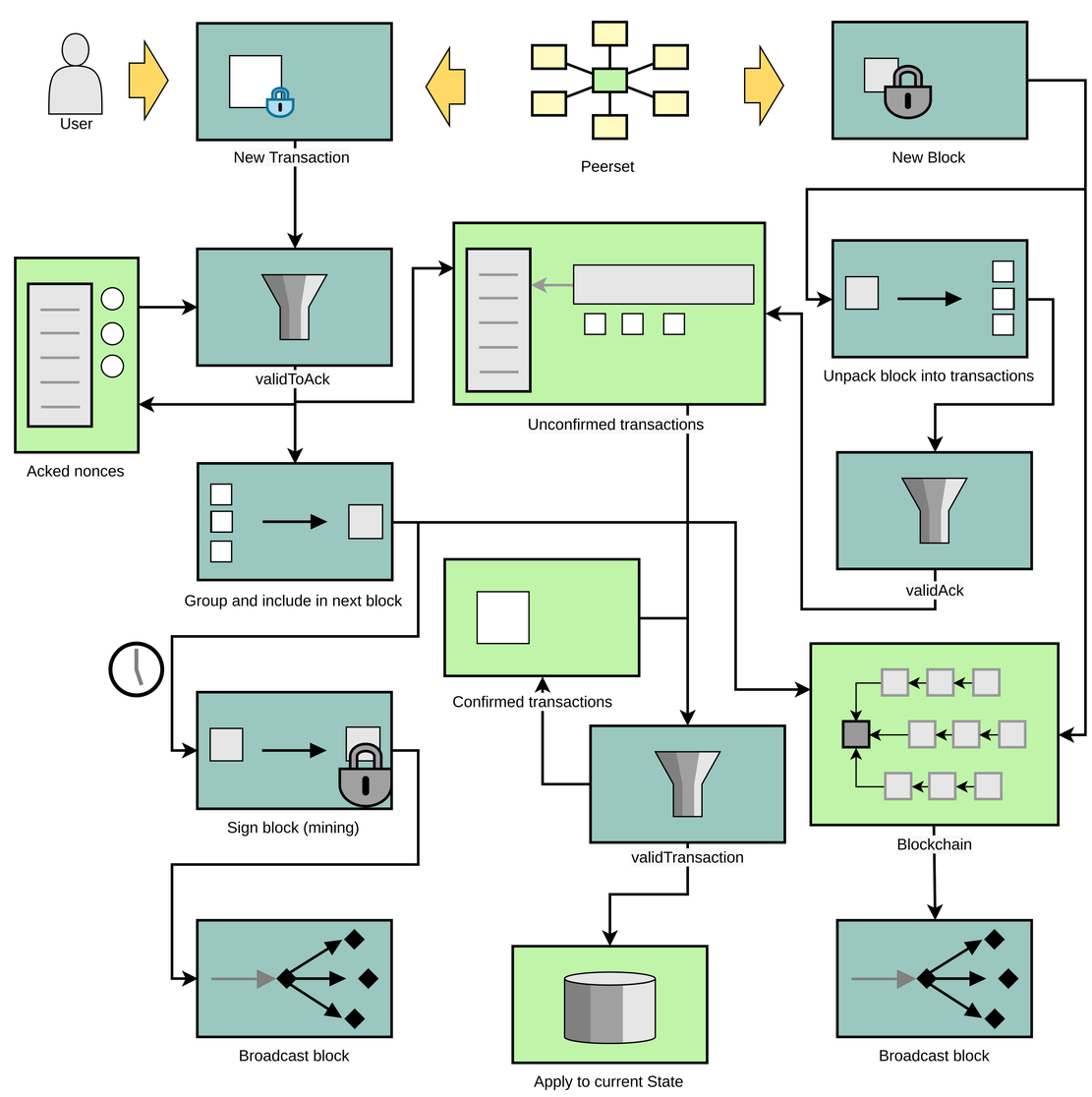
Our Consensus on Demand (CoD) algorithm is modular, works without synchronicity assumptions and can be implemented on top of any consensus algorithm, as long as less than one fifth of the n servers are Byzantine. $(n = 5f + 1)$

	Bitcoin and Ethereum	Algo-rand	Ouro-boros	PBFT	Honey Badger BFT	Broadcast-based	CoD with PBFT	CoD with Honey Badger BFT
Energy-efficient		✓	✓	✓	✓	✓	✓	✓
Asynchronous					✓	✓		✓
Parallelizable						✓	✓	✓
Finality		✓		✓	✓	✓	✓	✓
Permissionless	✓	✓	✓					
Consensus	✓	✓	✓	✓	✓		✓	✓
Leaderless					✓	✓		✓

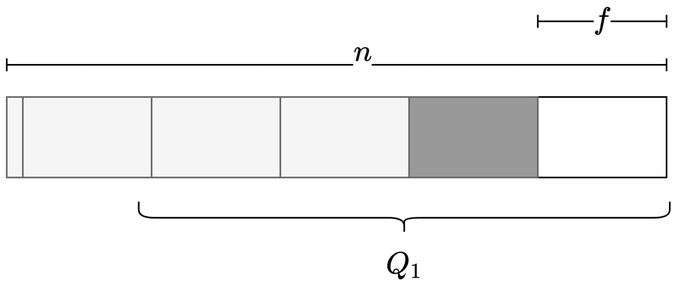
Algorithm

1. *Dissemination:* The transaction is broadcast to all servers.
2. *Verification:* Servers issue an acknowledgement for the first valid transaction they observe for a given $(sender, sn)$ pair. If at any point a server observes a quorum of more than $\frac{n+3f}{2}$ acknowledgements for a transaction t, the server accepts t.
3. *Consensus (opt.):* If after receiving $n - f$ acknowledgements servers observe conflicting acknowledgments, they propose the transaction for which they have observed the most acknowledgements to the consensus instance identified by the $(sender, sn)$ pair. The transaction delivered by the consensus routine is then accepted immediately.

Go Ethereum Based Implementation



Correctness Proof Intuition



The two shades of gray represent the share of honest servers acknowledging t (light gray) and t' (dark gray). The Byzantine adversary is depicted in white, and can acknowledge either transaction. While a server might see more than $4f$ acknowledgments for t, no server sees a majority of acknowledgements for t' in a quorum of $n - f$ servers.

Contact information:
Yann Vonlanthen
yvonlanthen@ethz.ch

References

[1] Daniel Collins, Rachid Guerraoui, Jovan Komatovic, Matteo Monti, Athanasios Xygkis, Matej Pavlovic, Petr Kuznetsov, Yvonne-Anne Pignolet, Dragos-Adrian Seredinschi, and Andrei Tonkikh. Online payments by merely broadcasting messages (extended version). *CoRR*, abs/2004.13184, 2020. URL `https://arxiv.org/abs/2004.13184`.

[2] J-P Martin and Lorenzo Alvisi. Fast byzantine consensus. *IEEE Transactions on Dependable and Secure Computing*, 3(3):202–215, 2006.

[3] Petr Kuznetsov, Andrei Tonkikh, and Yan X Zhang. Revisiting optimal resilience of fast byzantine consensus. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 343–353, 2021.

[4] Jakub Sliwinski, Yann Vonlanthen, and Roger Wattenhofer. Consensus on demand. *arXiv preprint arXiv:2202.03756*, 2022.

[5] Yann Vonlanthen. Cascadeth. `https://github.com/yannvon/cascadeth`, 2021.