Consensus on Demand

Roger Wattenhofer

Asynchronous Proof-of-Stake

Jakub Sliwinski, Roger Wattenhofer

ETH Zurich
jsliwinski,wattenhofer@ethz.ch

Abstract. We introduce a new permissionless blockchain architecture called Cascade (Consensusless, Asynchronous, Scalable, Deterministic and Efficient). The protocol is completely asynchronous, and does rely on neither randomness nor proof-of-work. Transactions exhibit finality within one round trip of communication.
Cascade is consensusless and only satisfies a relaxed form of consensus by introducing a weaker termination property. Without full consensus, the protocol does not support certain applications, such as general smart contracts. However, many important applications do not require general
smart contracts, and Cascade is an advantageous solution for these appli-

Consensus on Demand

Jakub Sliwinski
Distributed Computing
ETH Zurich
Zurich, Switzerland
jsliwinski@ethz.ch

Yann Vonlanthen
Distributed Computing
ETH Zurich
Zurich, Switzerland
yvonlanthen@ethz.ch

Roger Wattenhofer
Distributed Computing
ETH Zurich
Zurich, Switzerland
wattenhofer@ethz.ch

Abstract—**Digital money can be implemented efficiently by avoiding consensus. However, no-consensus implementations have drawbacks, as they cannot support smart contracts, and (even more fundamentally) they cannot deal with conflicting transactions.**

We present a novel protocol that combines the benefits of an asynchronous, broadcast-based digital currency, with the capacity to perform consensus. This is achieved by selectively performing consensus a posteriori, i.e., only when absolutely necessary. Our on-demand consensus comes at the price of restricting the byzantine participants to be less than a one-fifth minority in the system, which we show to be the optimal threshold.

We formally prove the correctness of our system and present an open-source implementation, which inherits many features from the Ethereum ecosystem.

access to her account, and neither Alice nor Bob getting paid. Also, no-consensus systems cannot support smart contracts.

So now we have a choice: either we use a total ordering currency which cannot scale to a high transaction throughput, or we use a parallel no-consensus verification system that is functionally restricted, and cannot support conflicting transactions.

In this paper we propose a system which has the best of both worlds. Our system achieves the unlimited throughput of no-consensus solutions as it first tries to verify every transaction without performing consensus. Only if a transaction cannot be verified on this "fast path", we invoke a consensus routine to resolve potential conflicts.

“I think there is a worldwide market for maybe five computers.”

Thomas Watson, 1943

"I think there is a worldwide market for maybe five computers."

Thomas Watson, 1943

(Smart Contract Enabled Blockchain)

GameStop

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

"The problem of course is the payee can't verify that one of the owners did not double-spend the coin."

"We need a system for participants to agree on a single history of the order in which [transactions] were received."

no double-spending

$\neq$

single order

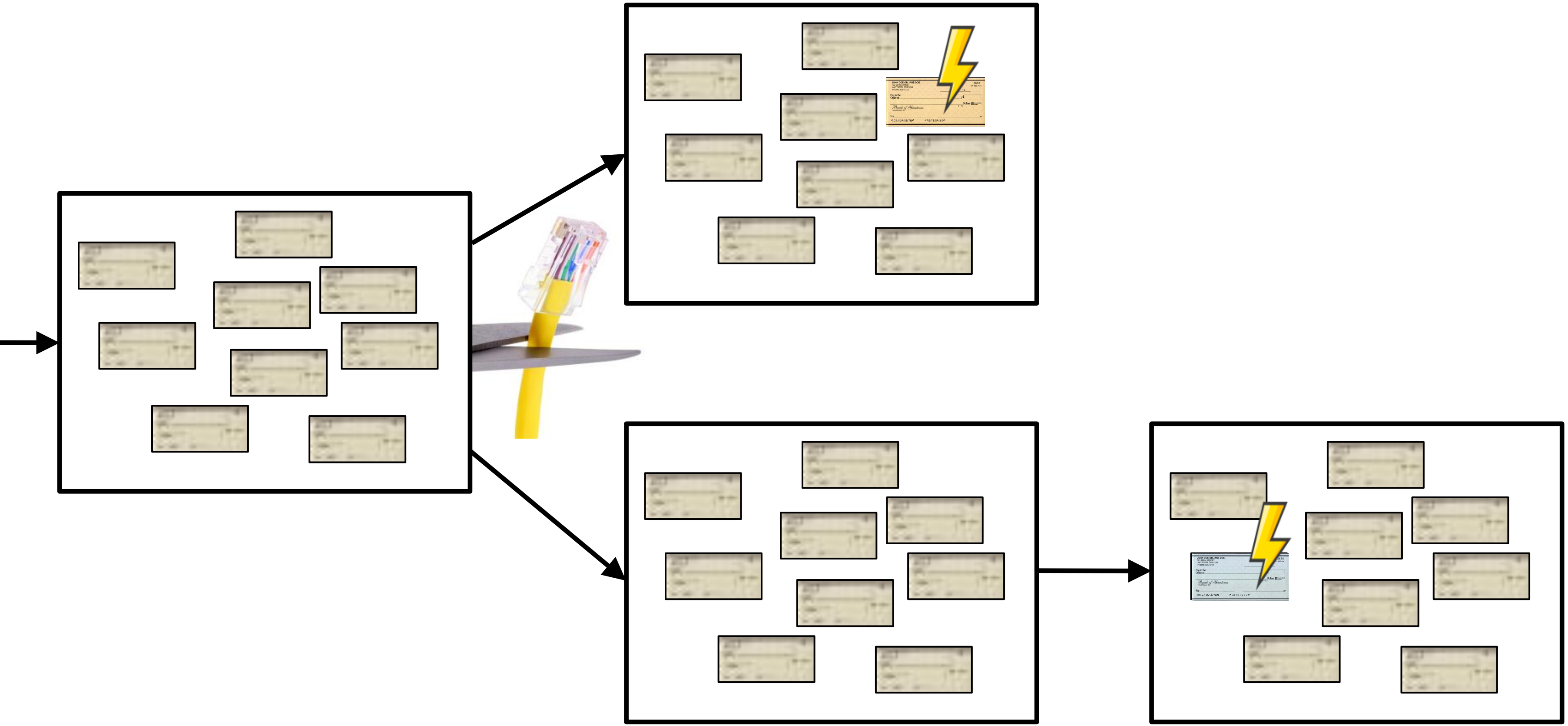
$=$

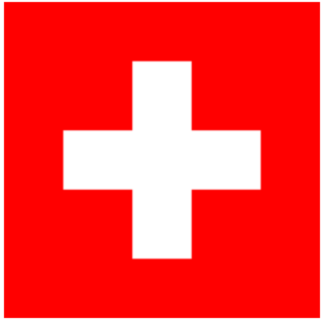
consensus

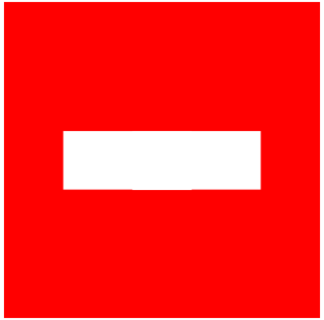
Double-Spending

Blockchains Solve Double-Spending Problem

What About Network Outages?





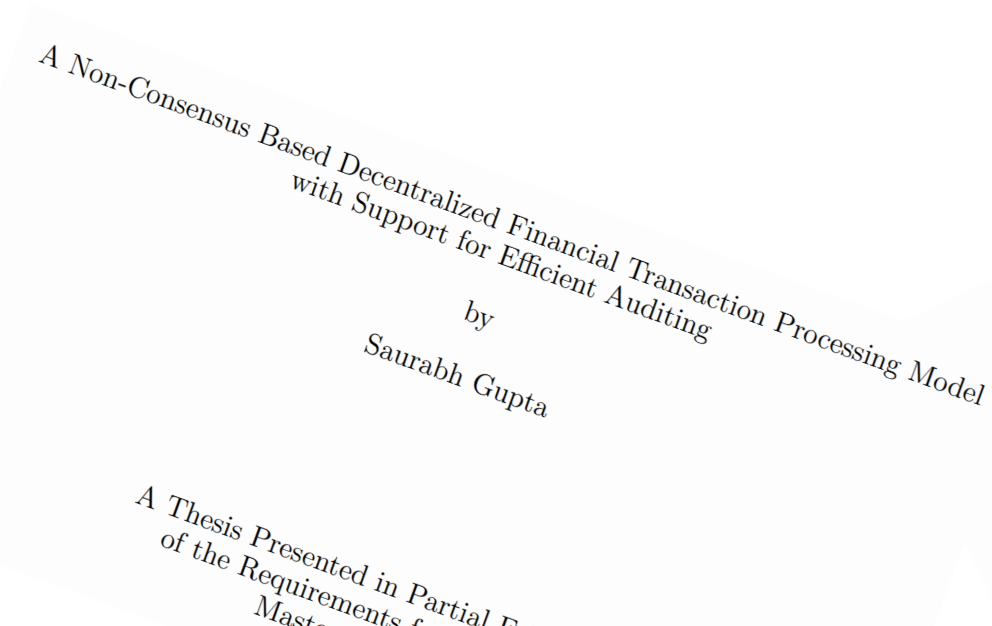


Unchangeable
Market Cap

Anonymous?
Permissionless?
Scalable = Secure?

Asynchrony
Finality
Throughput
Energy (PoW)
Smart Contracts
Unchangeable

Without Consensus

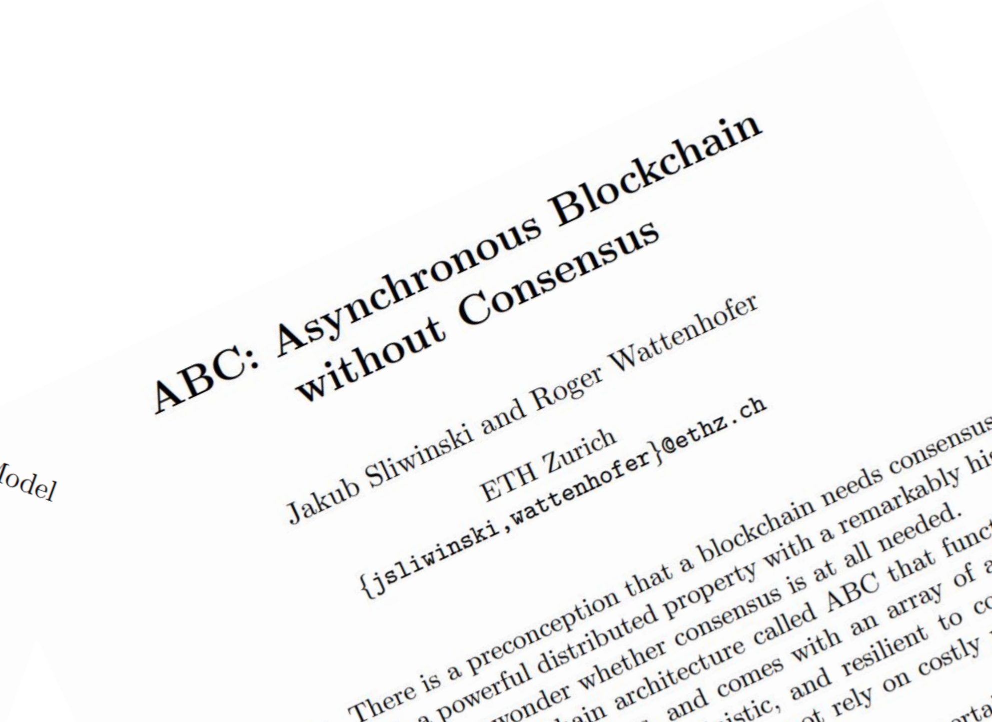


A Non-Consensus Based Decentralized Financial Transaction Processing Model
with Support for Efficient Auditing

by

Saurabh Gupta

A Thesis Presented in Partial F
of the Requirements f
Maste



ABC: Asynchronous Blockchain without Consensus

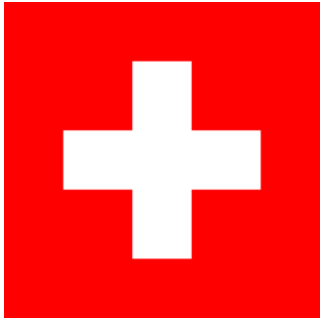
Jakub Sliwinski and Roger Wattenhofer

ETH Zurich

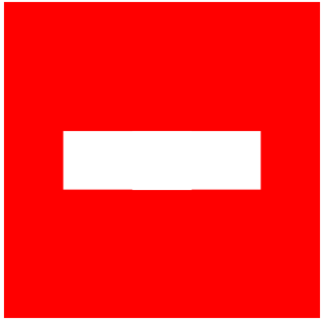
{jsliwinski,wattenhofer}@ethz.ch

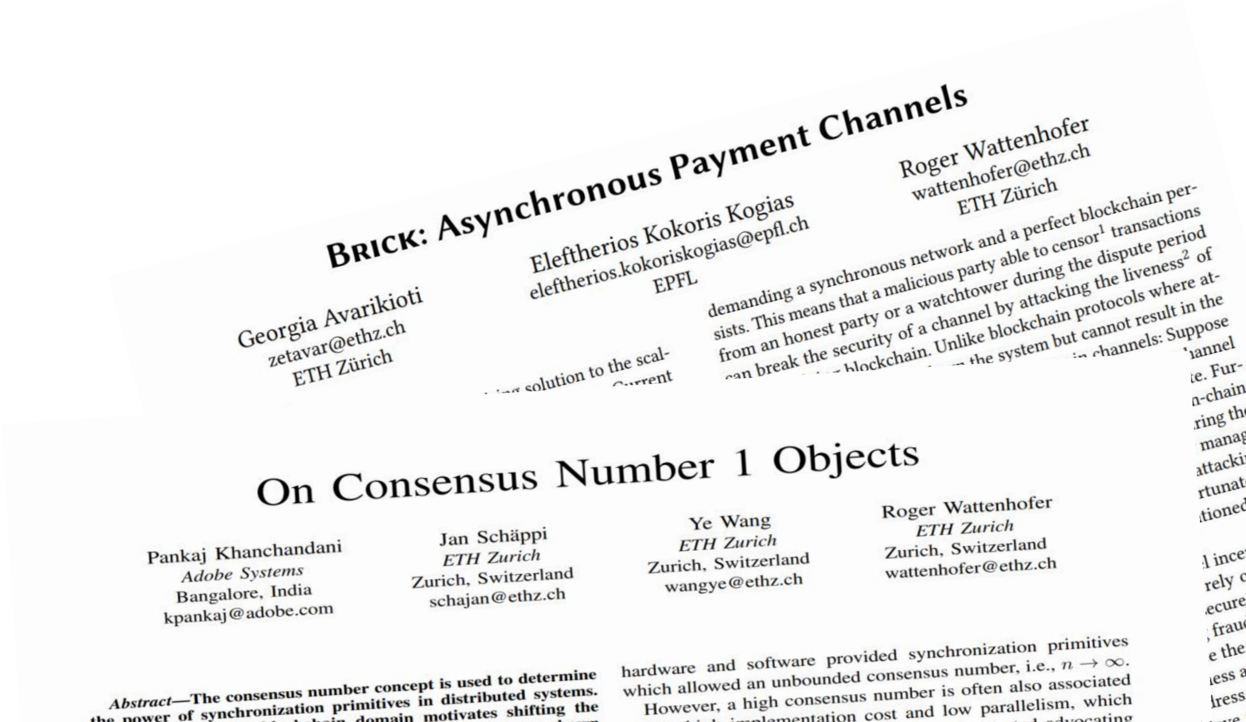
There is a preconception that a blockchain needs consensus
a powerful distributed property with a remarkably hi
wonder whether consensus is at all needed.
ain architecture called ABC that func
and comes with an array of a
istic, and resilient to co
t rely on costly

	PBFT[3]	HoneyBadger BFT[12]	Broadcast-based[7]	Bitcoin and Ethereum[17]	Ouroboros[9]	Algorand[4]	Cascade
Permissionless				✓	✓	✓	✓
Proof-of-work free	✓	✓	✓		✓	✓	✓
Finality	✓	✓	✓			✓	✓
Asynchronous		✓	✓				✓
Deterministic	✓		✓				✓
Parallelizable			✓				✓
General smart contracts	✓	✓		✓	✓	✓	



Asynchronous*
Throughput
Finality
Energy (PoS)
Permissionless
Scalable





BRICK: Asynchronous Payment Channels

Georgia Avarikioti
zetavar@ethz.ch
ETH Zürich

Eleftherios Kokoris Kogias
eleftherios.kokoriskogias@epfl.ch
EPFL

Roger Wattenhofer
wattenhofer@ethz.ch
ETH Zürich

demanding a synchronous network and a perfect blockchain persists. This means that a malicious party able to censor[1] transactions from an honest party or a watchtower during the dispute period can break the security of a channel by attacking the liveness[2] of the blockchain. Unlike blockchain protocols where at-

On Consensus Number 1 Objects

Pankaj Khanchandani
Adobe Systems
Bangalore, India
kpankaj@adobe.com

Jan Schäppi
ETH Zurich
Zurich, Switzerland
schajan@ethz.ch

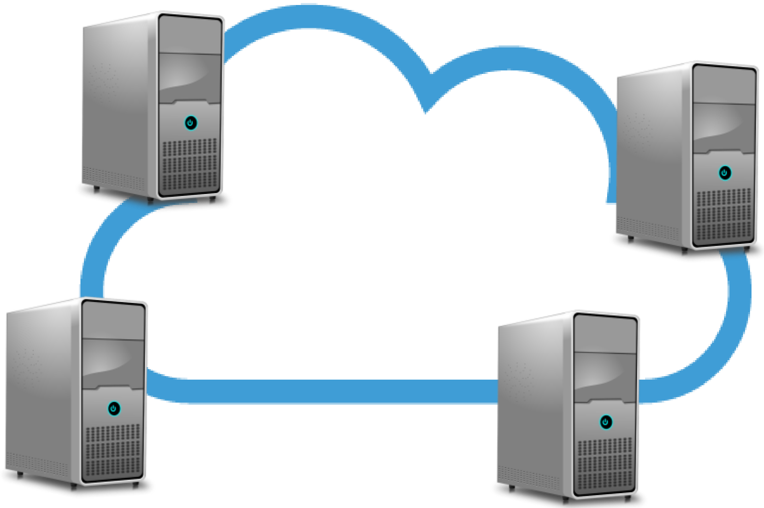
Ye Wang
ETH Zurich
Zurich, Switzerland
wangye@ethz.ch

Roger Wattenhofer
ETH Zurich
Zurich, Switzerland
wattenhofer@ethz.ch

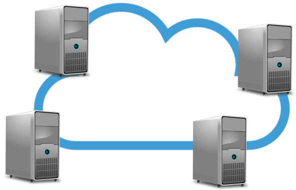
***Abstract*—The consensus number concept is used to determine the power of synchronization primitives in distributed systems.**

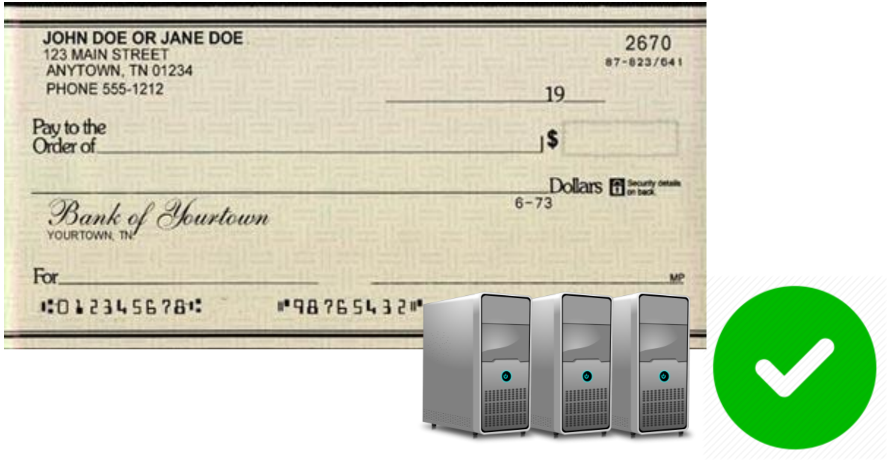
hardware and software provided synchronization primitives which allowed an unbounded consensus number, i.e., $n \to \infty$. However, a high consensus number is often also associated

Permissioned



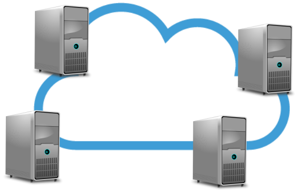
Permissioned

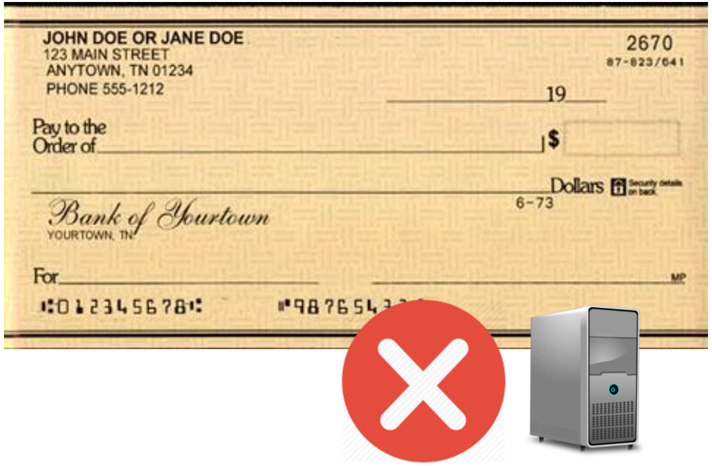


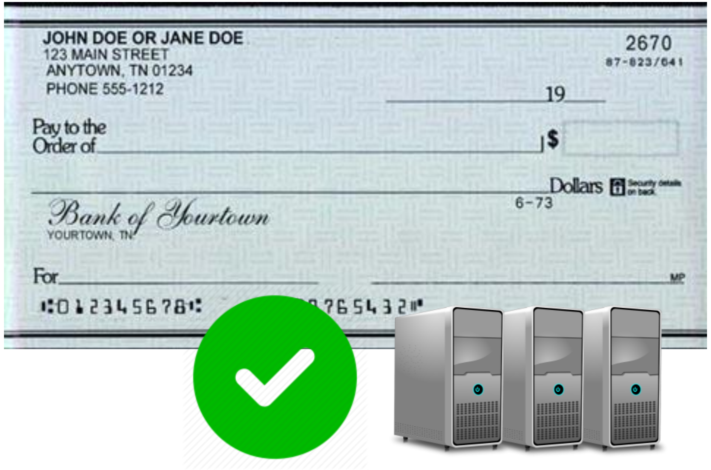


Needed: 3 out of 4 signatures

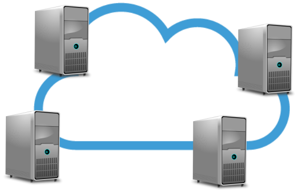
Double-Spending

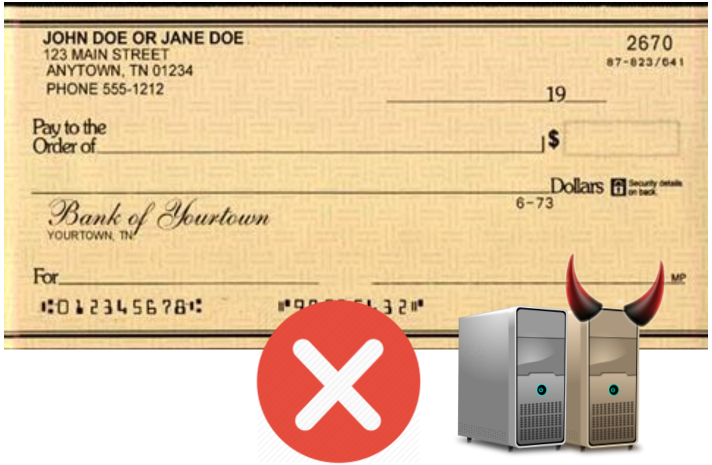


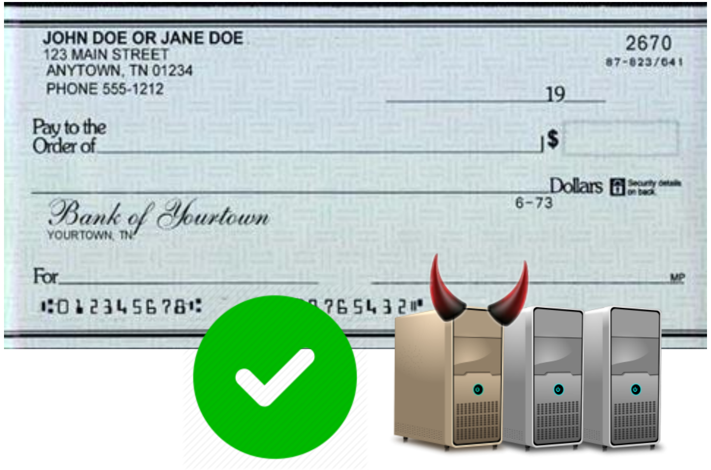




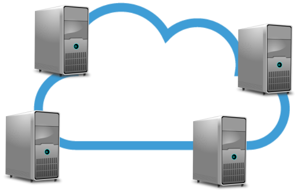
Double-Spending

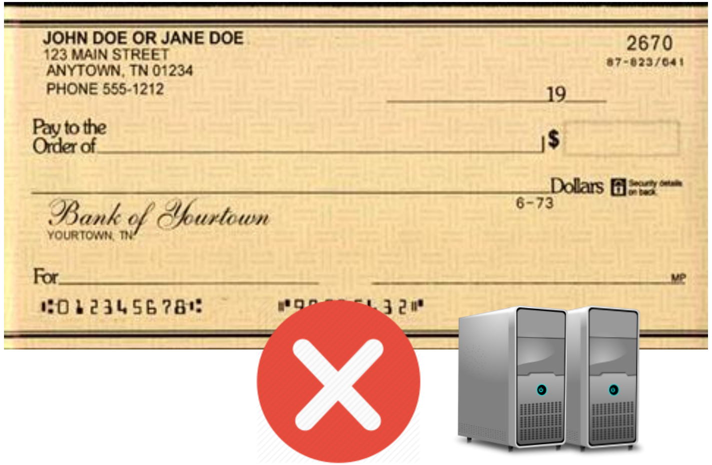


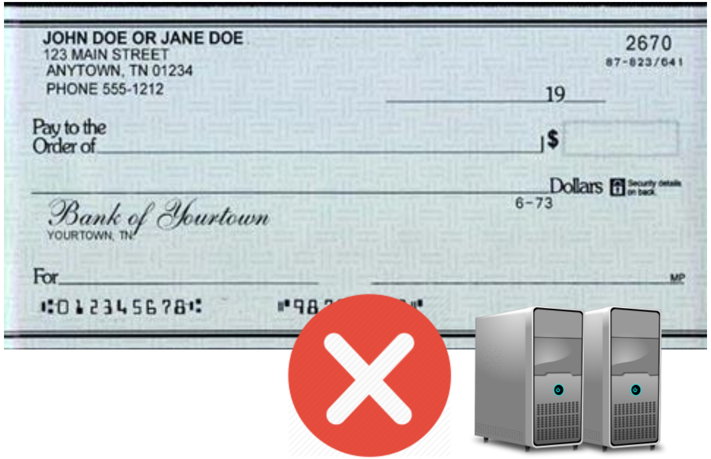




Double-Spending



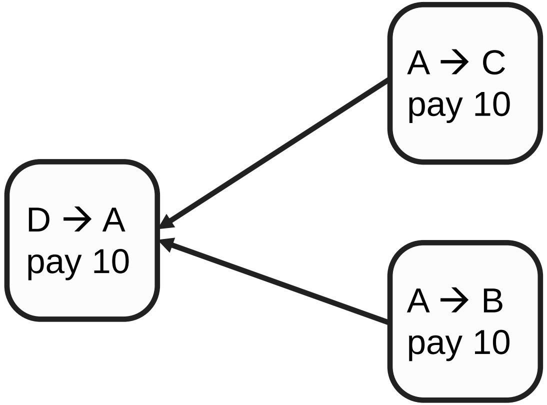




Usual Safety Condition

Less than 1/3 Byzantine

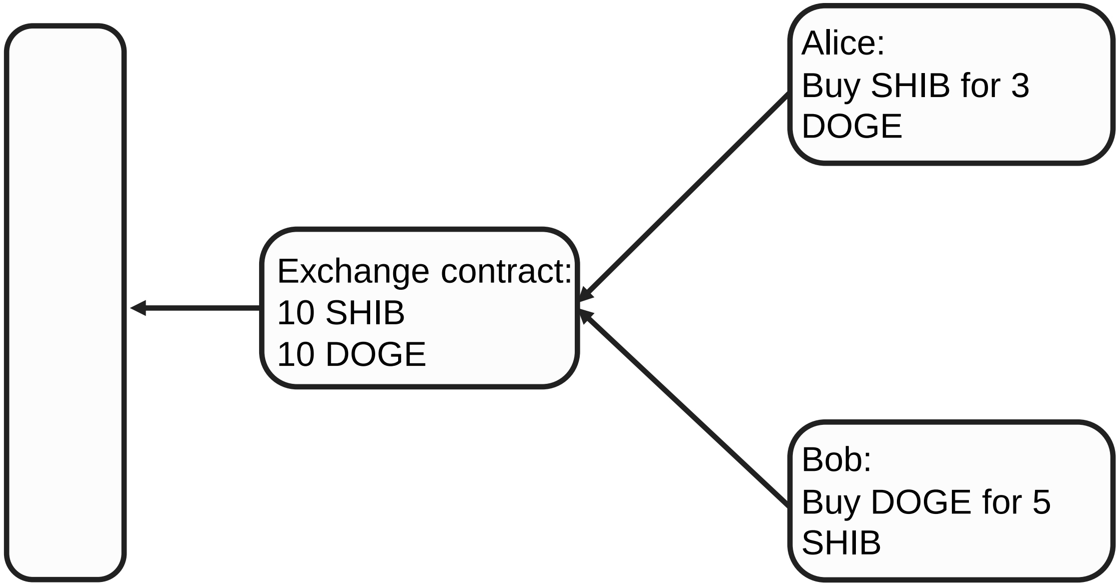
Without Consensus



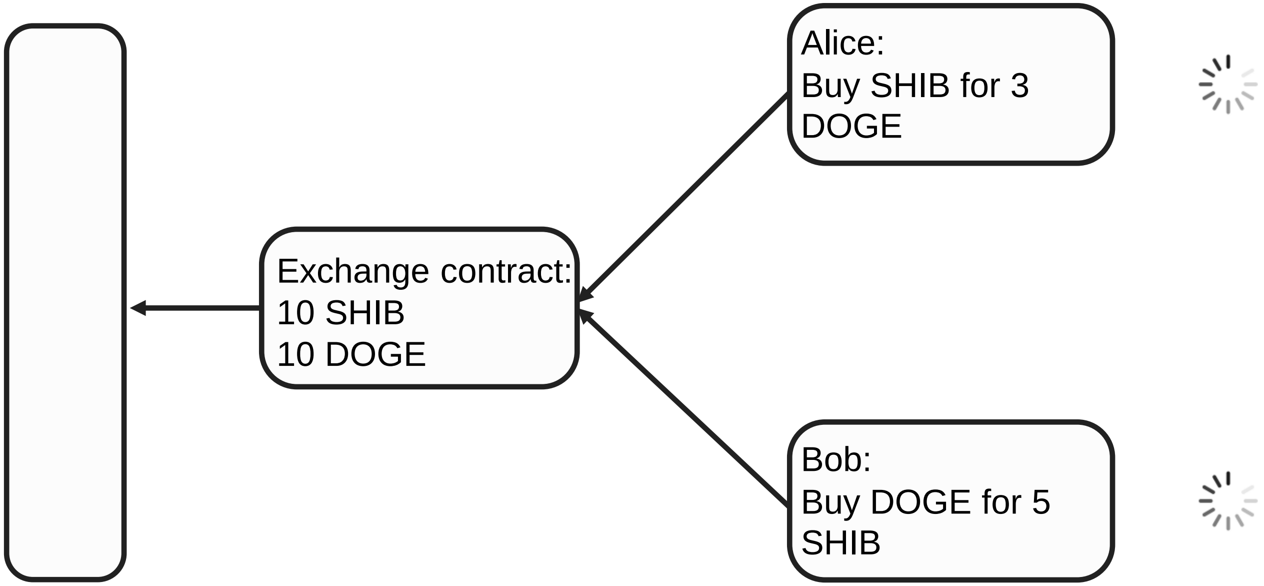
= confirm **exactly** one of Alice’s tx

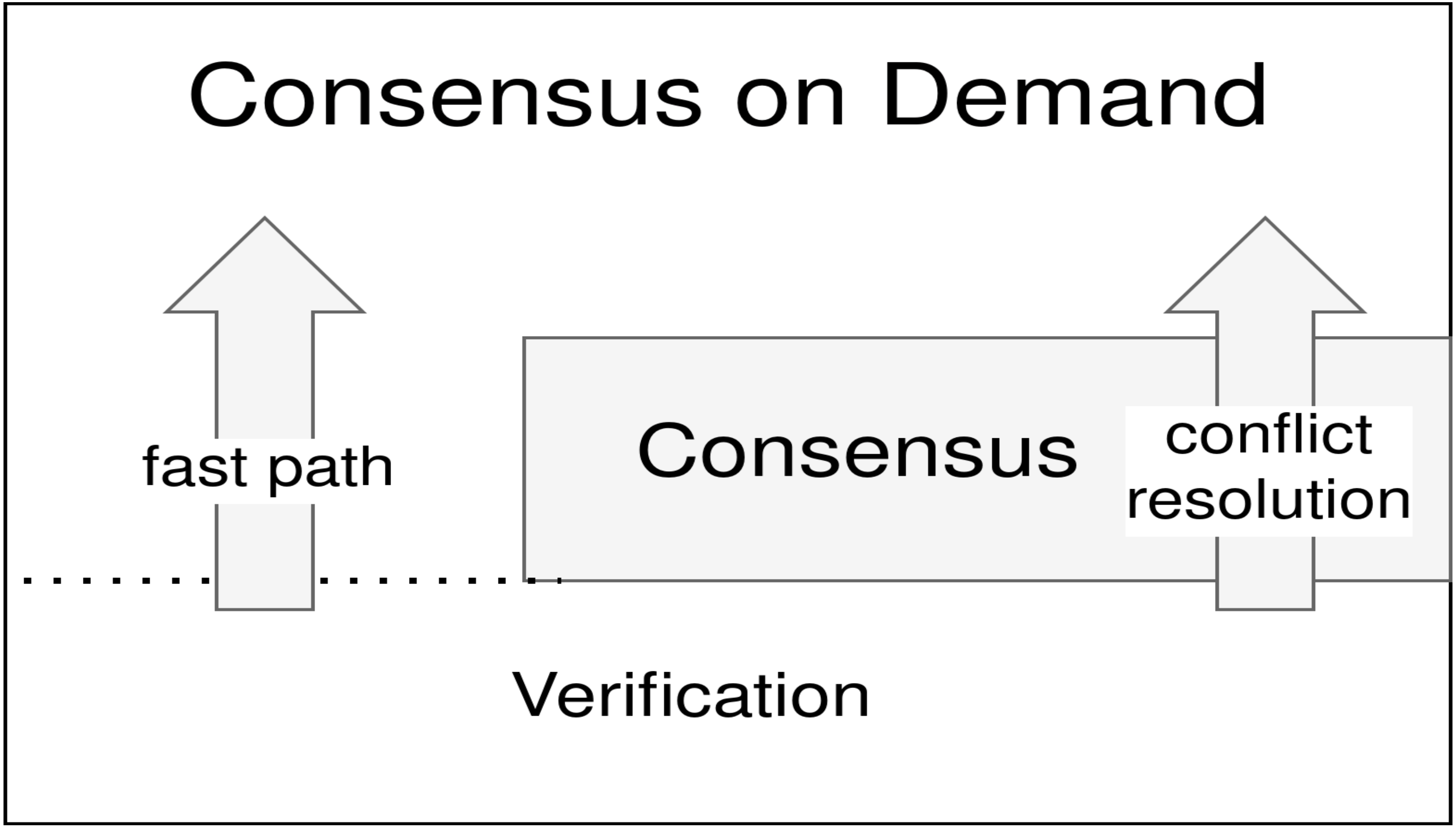
Cascade = confirm **at most** one of Alice’s tx

But: No Consensus

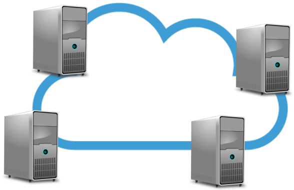


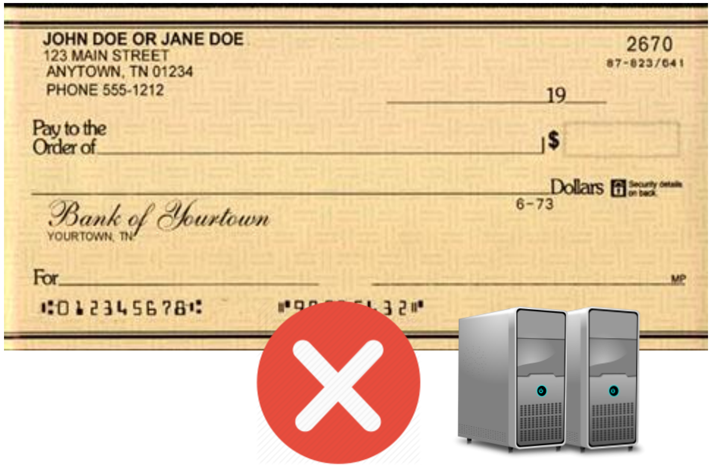
But: No Consensus

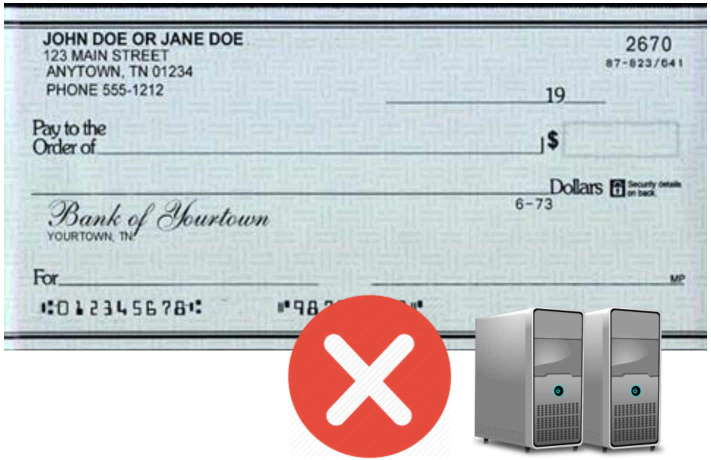



Consensus on Demand
fast path
Consensus
conflict resolution
Verification

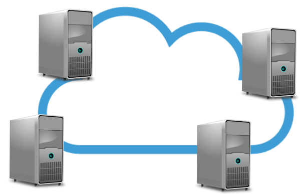
Consensus on Demand

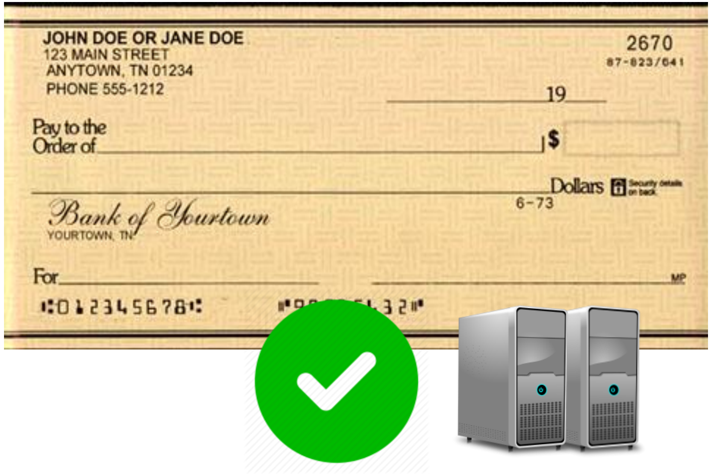


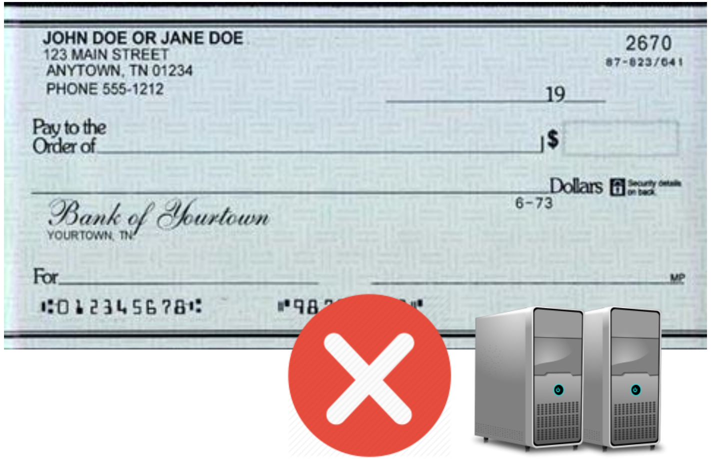




Consensus on Demand

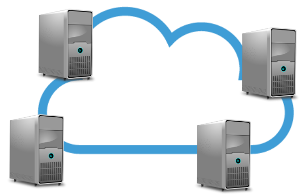


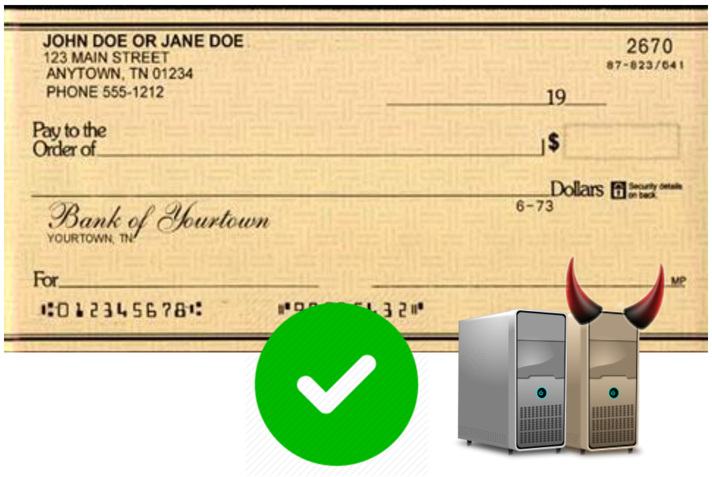


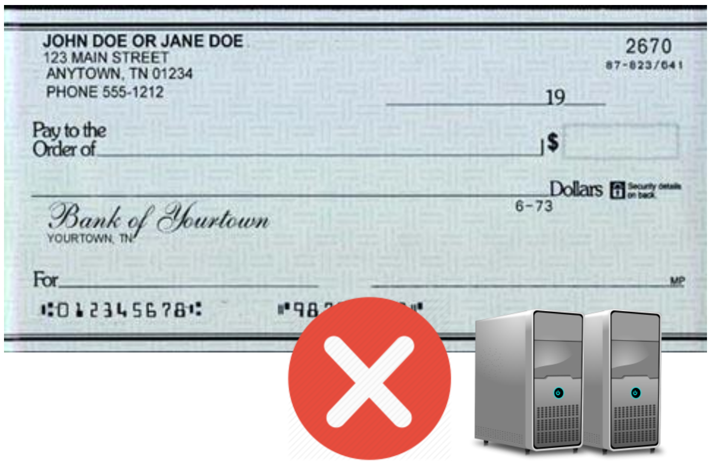


Consensus!

Consensus on Demand

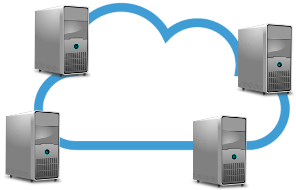


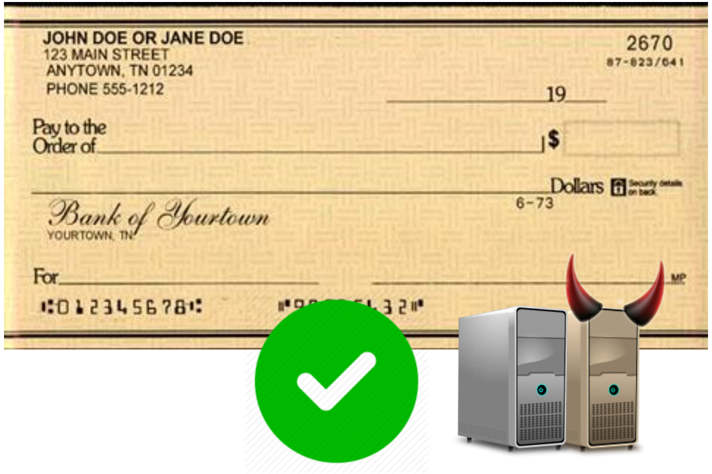


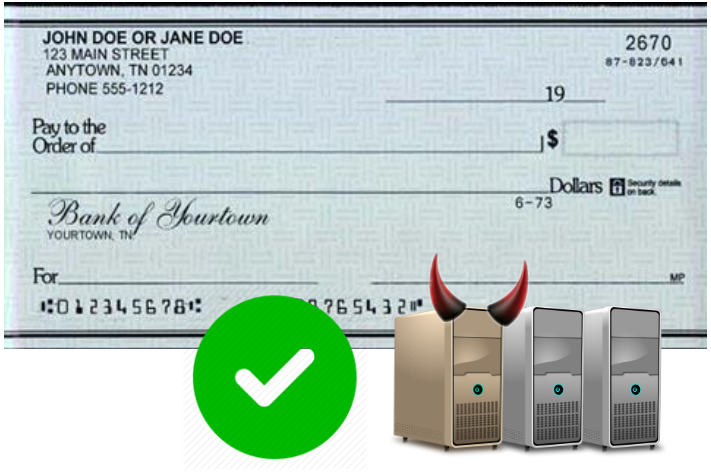


Consensus!

Double-Spending







Consensus!

Tight Impossibility Result

A system with n servers cannot reach consensus with a fast path (1 communication round) if

$f \geq n/5$ (asynchronous model)
$f \geq n/4$ (synchronous model)

The Best of Both Worlds

Fast Path

Speed-up through parallelization

Quick finality in the common path

Consensus

Account sharing

Updating transactions

Smart contracts

Summary and Comparison

	Bitcoin and Ethereum	Algorand	Ouroboros	PBFT	Honey Badger BFT	Broadcast-based	CoD with PBFT	CoD with Honey Badger BFT
Energy-efficient		✓	✓	✓	✓	✓	✓	✓
Asynchronous					✓	✓		✓
Parallelizable						✓	✓	✓
Finality		✓		✓	✓	✓	✓	✓
Permissionless	✓	✓	✓					
Consensus	✓	✓	✓	✓	✓		✓	✓

Questions? Comments?

Roger Wattenhofer