Tendermint

Build, Deploy, Manage

Speed, security, and scalability for the world's blockchains

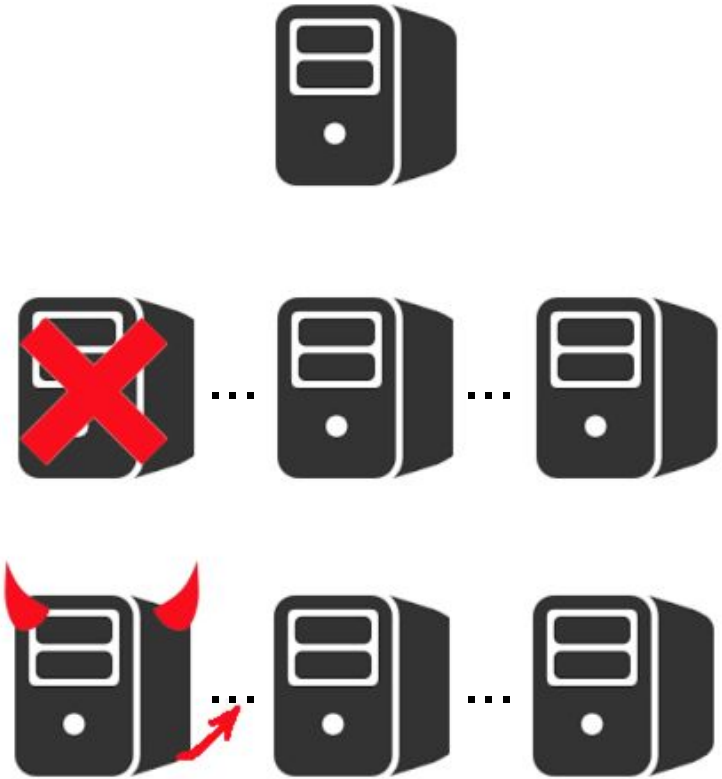
What are Blockchains?

Distributed transactional systems that use:

- Byzantine fault-tolerance (BFT)
- Applied cryptography
- Economic (dis)incentives

a.k.a. *“crypto-economics”*

Evolution of Fault-Tolerance



No fault-tolerance

Crash fault-tolerance
(e.g. Raft, Paxos)

Byzantine fault-tolerance
(e.g. DLS, PBFT, Tangaroa)

Partial History of Byzantine Fault-Tolerance

“Impossibility of Distributed Consensus with One Faulty Process”
- Fischer, Lynch, Paterson (1985)

“Consensus in the Presence of Partial Synchrony”
- Dwork, Lynch, Stockmeyer (1988)

“Practical Byzantine Fault Tolerance”
- Castro, Liskov (1999)

“Bitcoin: A Peer-to-Peer Electronic Cash System”
- Satoshi Nakamoto (2008)

Overcoming FLP Impossibility

1. **Assume Partial-Synchrony instead of Asynchrony**
   - Max latency of msgs are unknown but finite
   - Network disruption duration is unknown but finite

2. **Assume Non-determinism instead of Determinism**
   - “Common coin” protocol (e.g. HoneyBadger)
   - Randomized timeouts?

PoW vs “classical” BFT

- "Nakamoto" BFT consensus (PoW)
  - Invented by Satoshi Nakamoto for Bitcoin
  - Security is proportional to ongoing energy burn costs
  - Slow, expensive, and environmentally costly?

- "Classical" BFT consensus
  - DLS (1988), PBFT (1999), Tendermint (2014), Tangaora (2015)
  - Based on public/private key cryptography and *quorums*
  - Very fast & no energy waste
  - Efficient light client proofs
  - Does not “fork”, based on rounds
  - The basis of BFT Proof-of-Stake (BFT PoS)

Question:

How many confirmations would you wait for to confirm a $100M Bitcoin transaction?

12.5BTC * $600 = $7500/block

6*24blocks/day = $1M/day

“Classical” BFT + Proof-of-Stake

1. The system only fails if 1/3+ are Byzantine
2. Those Byzantine actors can be identified
3. On-chain collateral can be slashed

“Classical” BFT + Proof-of-Stake

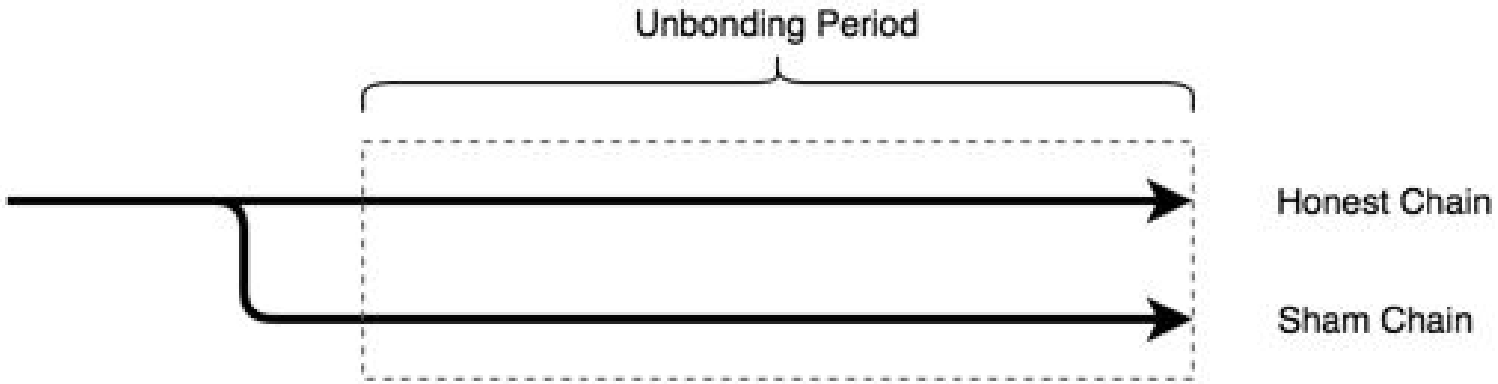
- Separate the staking token from other tokens
- Staking token != store of value
- Staking token != medium of exchange
- Merely “hodling” staking token incurs inflation tax
- To earn back inflation, tokens must be “at stake”
- Those who don’t want to validate can delegate

Overcoming “Nothing at Stake”

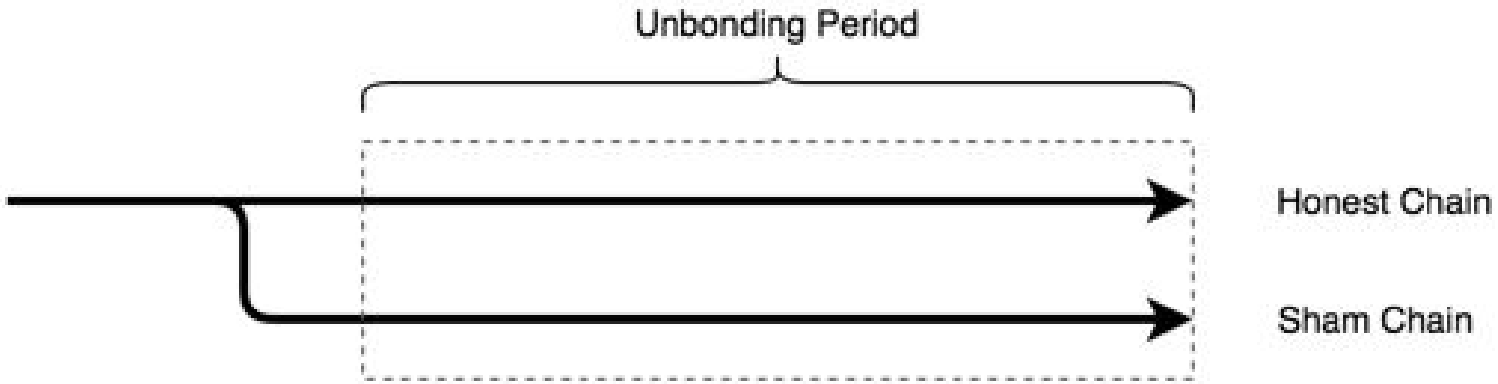
1. **Solve the “Short Range Attack” Problem**
   - Post stake tokens as collateral w/ BondTx to participate in consensus
   - To get stake tokens back, first post an UnbondTx to leave validator-set
   - Must wait “unbonding period” to get stake tokens back

2. **Solve the “Long Range Attack” Problem**
   - Full nodes and validators sync within the “unbonding period”
     - New nodes and really-out-of-sync must receive trusted block-hash
   - For full nodes and light-clients, query long-bonded oracles
     - Lying oracles get slashed

Overcoming “Nothing at Stake”



Overcoming “Nothing at Stake”



Etc

Governance

- Compare to ETH/ETC fork
- Constitution

Validator Alliance

- Slash one slash all on many blockchains

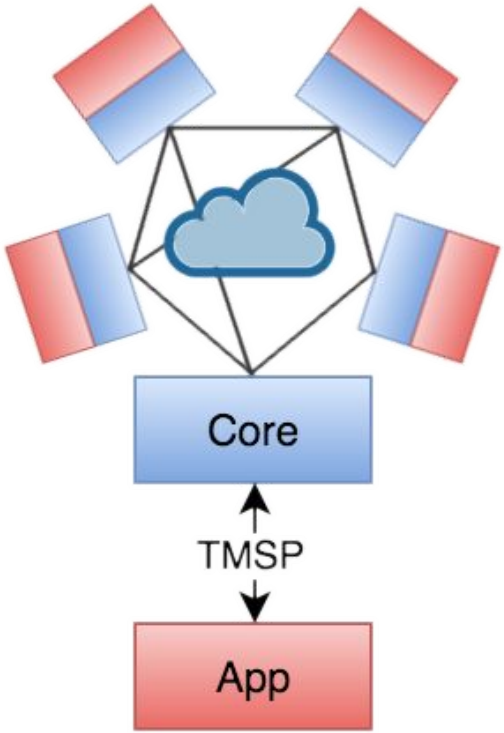
Vs Mining Pools

Nielson’s Law

Tendermint

- In development since early 2014
- Free, open-source, now Apache2.0
- Production version ready Q1 2017
- Round-robin leader w/ deterministic non-choking algo
- Based on gossip routing (of blocks & votes)
- Tendermint handles:
  - P2P networking
  - Consensus (tx ordering)
  - Tx broadcasting (mempool)
- Otherwise is “application agnostic”
  - Connects to a black-box state-machine via unix socket (TMSP)
  - Makes it easy to connect to existing blockchain stacks

Tendermint



Tendermint + Ethereum = Ethermint

We will plug in Tendermint into GoEthereum!

ETA: 2 months or less

Thank you!

Jae Kwon
Email: jae@tendermint
Twitter: @jaekwon

Tendermint (blockchain engine)
Website: http://tendermint.com
Github: http://github.com/tendermint/tendermint