Theory Meets Practice in the Algorand Blockchain

Victor Luchangco

Algorand, Inc.

OPODIS, 13 Dec 2022

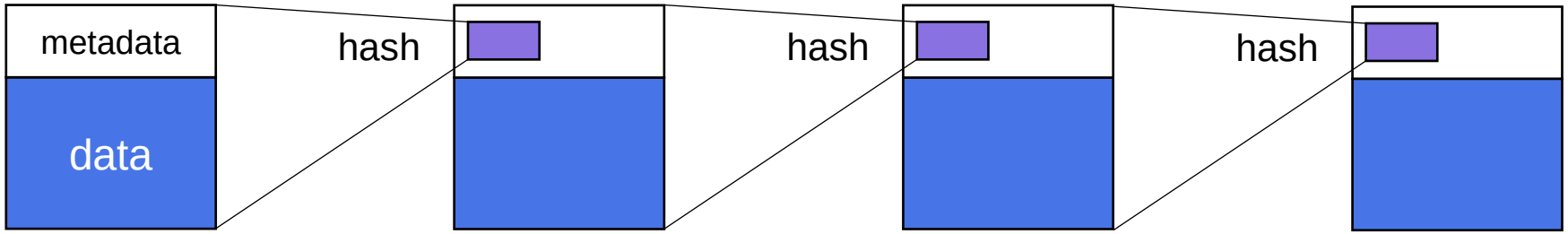
Algorand™

Theory and Practice in Distributed Systems

- Good theory requires good models
- Good models capture essence of problem/protocol
- Good system models are modular
  - Makes analysis tractable
  - Allows systems to be composed from modules
- Good models abstract away irrelevant details
- In practice, implementations may resist good modeling
  - They may transgress module boundaries
  - “Implementation details” may induce algorithmic changes
  - Examples: GHS minimum-spanning-tree algorithm, operating systems, Algorand blockchain

What is a blockchain?

- Tamper-resistant distributed ledger (append-only)
- Consists of sequence of blocks shared by all participants
  - Each participant keeps its own copy
- Each (non-genesis) block contains cryptographic commitment to previous block



- Blockchain data is sequence of signed transactions
  - Transactions issued by clients
  - Transactions in blocks must be valid
  - Most transactions commute with each other

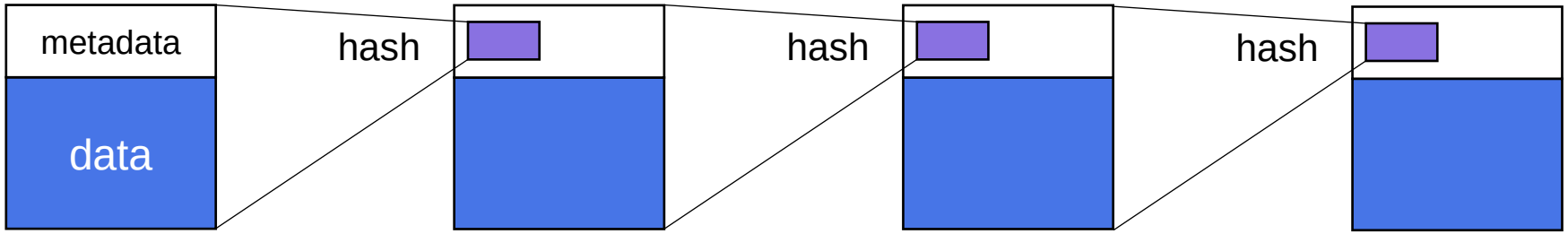
Valid signature
Valid state transition

Desired properties
- Secure
- Fast
- Efficient
- Decentralized

Chief difficulty: how to add new blocks

What is a blockchain?

- Tamper-resistant distributed ledger (append-only)
- Consists of sequence of blocks shared by all participants
  - Each participant keeps its own copy
- Each (non-genesis) block contains cryptographic commitment to previous block



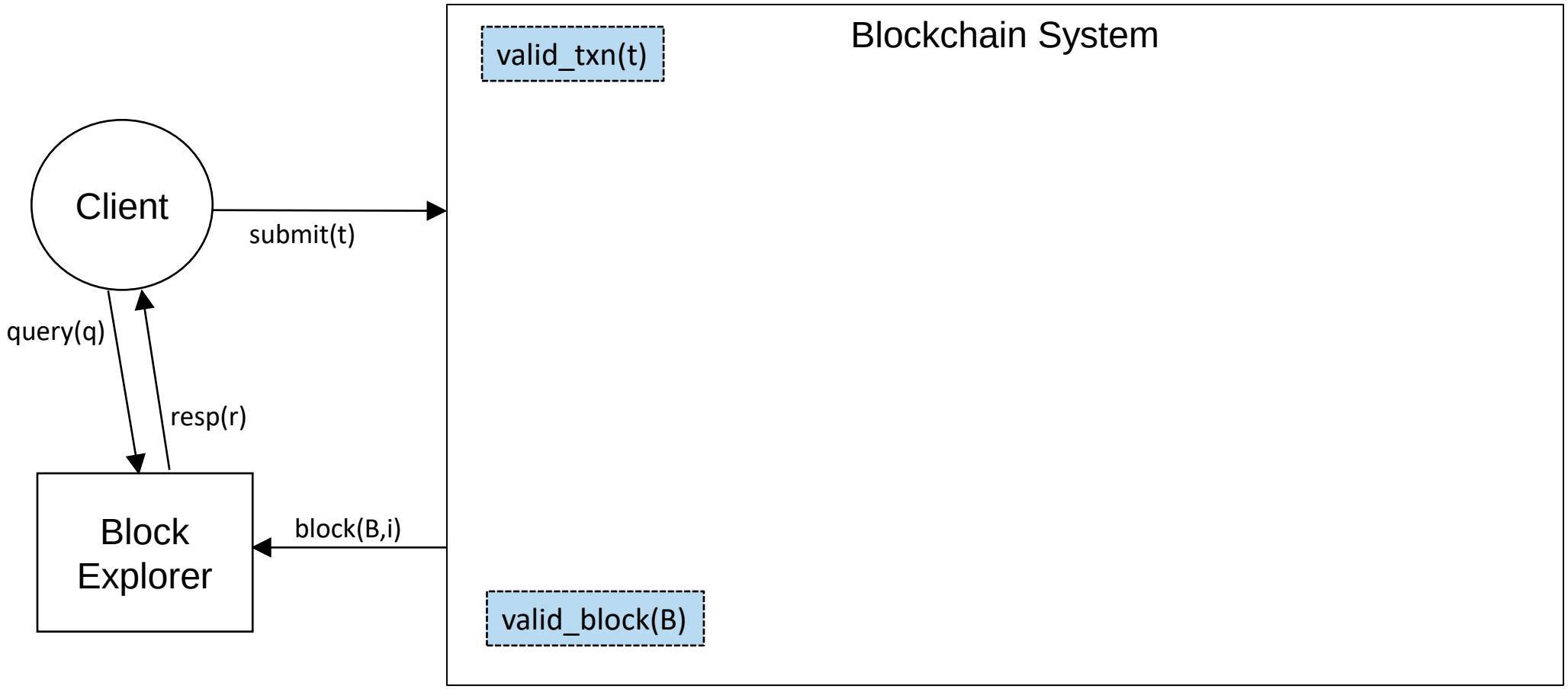
- Blockchain data is sequence of signed transactions
  - Transactions issued by clients
  - Transactions in blocks must be valid
  - Most transactions commute with each other

Valid signature
Valid state transition

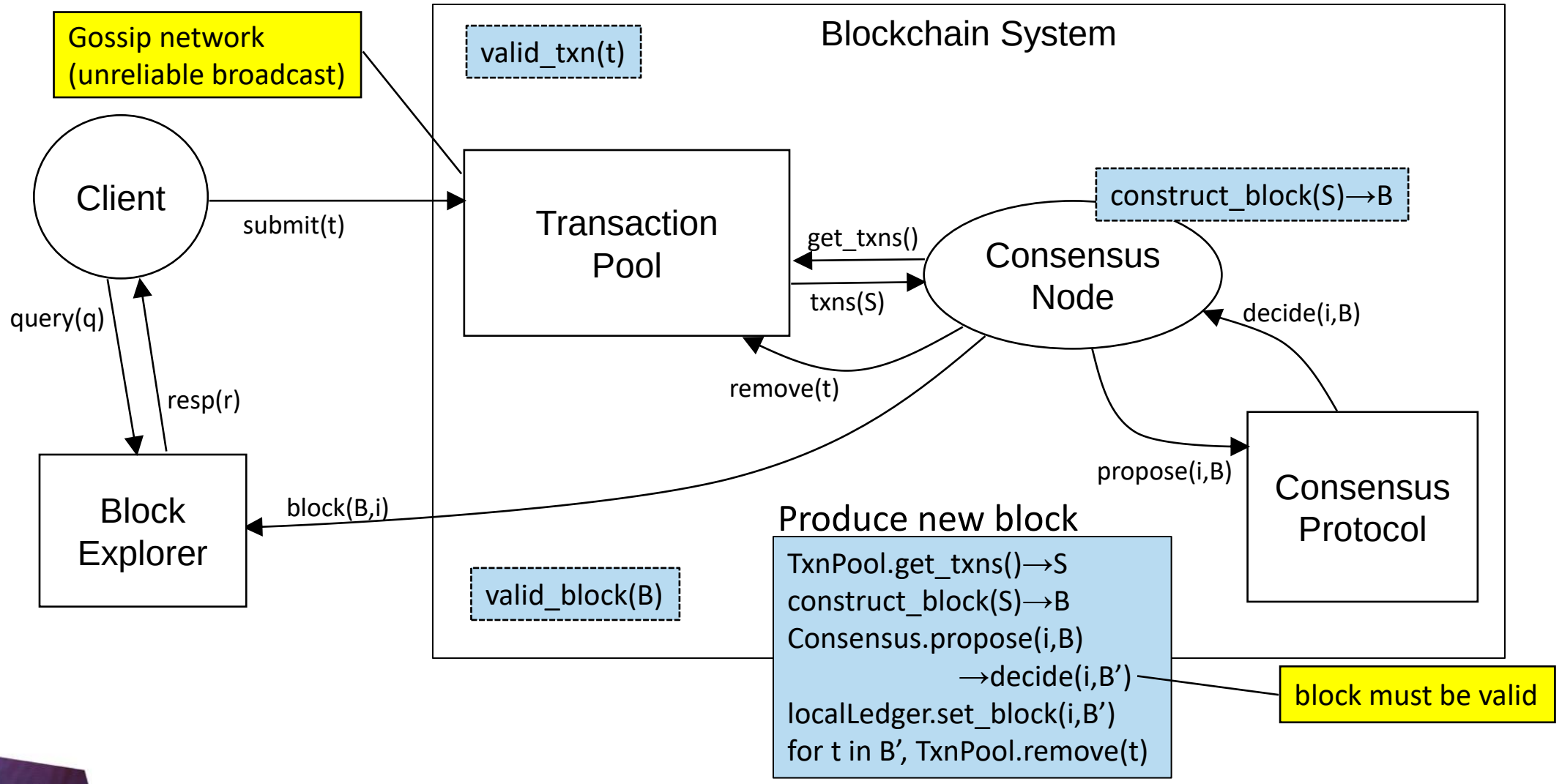
Desired properties
- Secure
- Fast
- Efficient
- Decentralized

Chief difficulty: h **Consensus!** s

Blockchain system model diagram

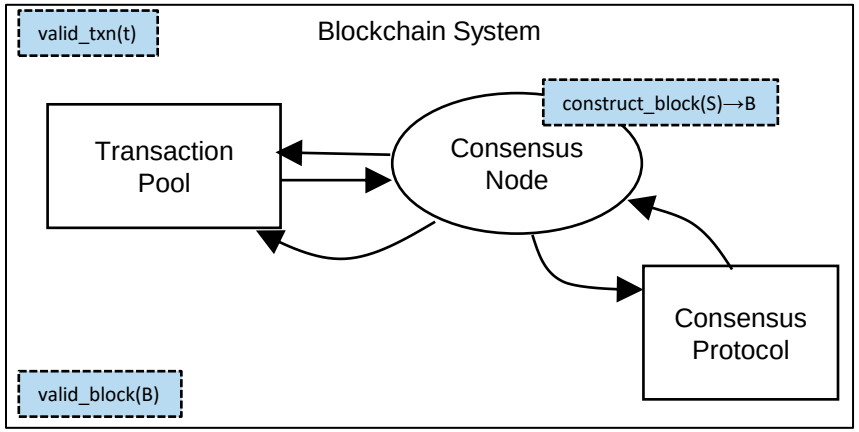


Blockchain system model diagram



Notes on blockchain system model

- Block decided must be valid
  - Incorporate check into validity condition of consensus
- Validity checking must be checked at (overlay) network level
  - Prevents malicious nodes from overloading network
- Proof of stake restricts number of participants in consensus protocol
  - Quadratic message complexity unsustainable for millions of nodes
- Signature verification dominates computational workload
- Consensus nodes have a lot of flexibility in how to construct blocks
  - “MEV” exploits
- Legacy code on the blockchain



What about proof of work?

Aside on proof-of-work consensus

- Unlike classic consensus, blocks are not final when adopted by a consensus node
  - Honest nodes adopt the longest chain they see, even if it contradicts previous chain
  - Blocks are expensive to create, so typically only one or a few are generated at a time
  - Blocks in the chain are “confirmed” by later blocks
  - Blocks with “enough” confirmations are considered final
- Alternatively: a block is not decided until it has sufficient confirmations
  - Later blocks must be proposed and adopted to decide on an earlier block
  - Requires additional assumption: nodes can detect if they are partitioned for too long

The Algorand Blockchain

- Permissionless
- Fast: new block every 4 seconds, 6000 transactions/sec
- Efficient (no proof-of-work puzzles), carbon negative
- *Pure* proof-of-stake
  - Consensus participants self-select using verifiable random function (VRF)
  - Requires an unpredictable seed for each round (written into block)

Self-selection-based attacks

- VRF produces deterministic output based on seed and address that looks random
  - Interpreted as number in [0,1), uniform distribution
  - Address is selected if number is sufficiently high (threshold depends on balance)
- If adversary can influence seed:
  - Make seed so its addresses are more likely to be selected
  - Prevent by making seed (almost) independent of block proposer's choices
- If adversary can predict seed:
  - Transfer tokens to an address more likely to be selected
  - Prevent by using balance from ~200 rounds ago for selection

Reducing computation on critical path

- Naïve signature verification for a single block could take over 1 sec
  - ~0.05ms to verify one signature
  - a block may contain over 20k transactions
- New block every 4 seconds or less

- Batch verification
- Transaction pre-validation on admission
- Layer-1 smart contract call cannot* verify signatures

Summary

- Good models help us understand and build systems better by abstraction and modularity.
- To achieve high performance, systems often transgress abstraction boundaries.
- Supposed implementation details may affect algorithmic choices.

The difference between theory and practice is bigger in practice than in theory.

All models are wrong; some models are useful.


Algorand™