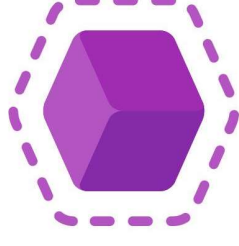


# Extending Existing Blockchains with Virtualchain

Jude Nelson\*, Muneeb Ali<sup>\*†</sup>,  
Ryan Shea<sup>†</sup>, Michael J. Freedman\*



\*



<sup>†</sup> **blockstack**

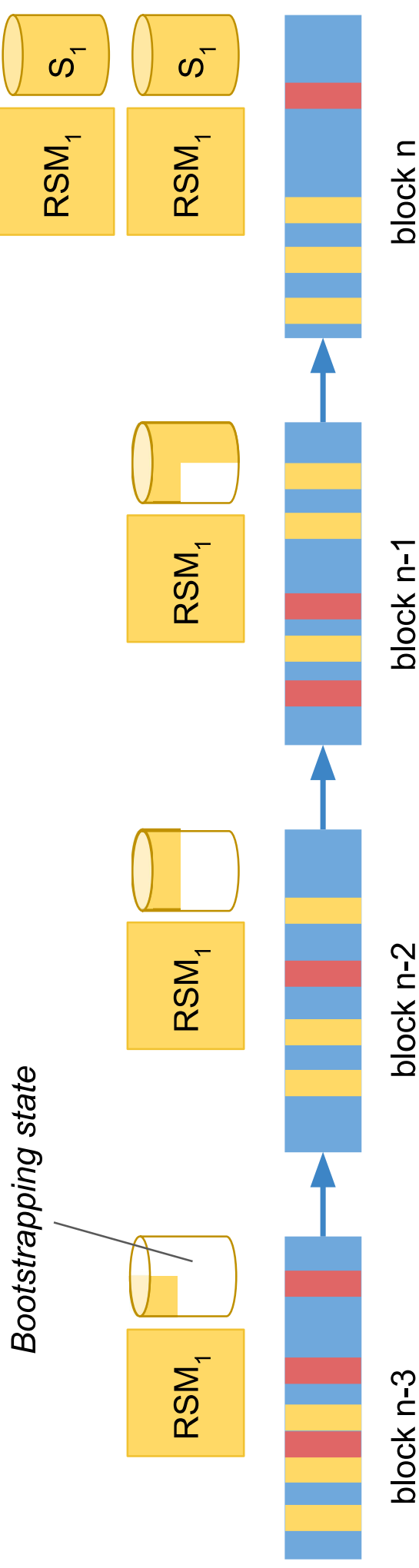
Pretend cryptocurrencies  
do not exist

# What's in a Proof-of-Work Blockchain?

- Total ordering of writes
- “Stable” view ordering (\*)
- Append-only
- 100% replicated
- Tamper-resistant
- Anyone can write
- Fixed growth rate (pay-to-play)
- **Hard to upgrade once deployed**

# Distributed Applications and Blockchains

- Replicated state machines (RSMs) on top?
- Strategy: store input history



## Advantages

- Open app membership
- Survive total app failure
- Blockchain-agnostic
- App-agnostic

# Challenges

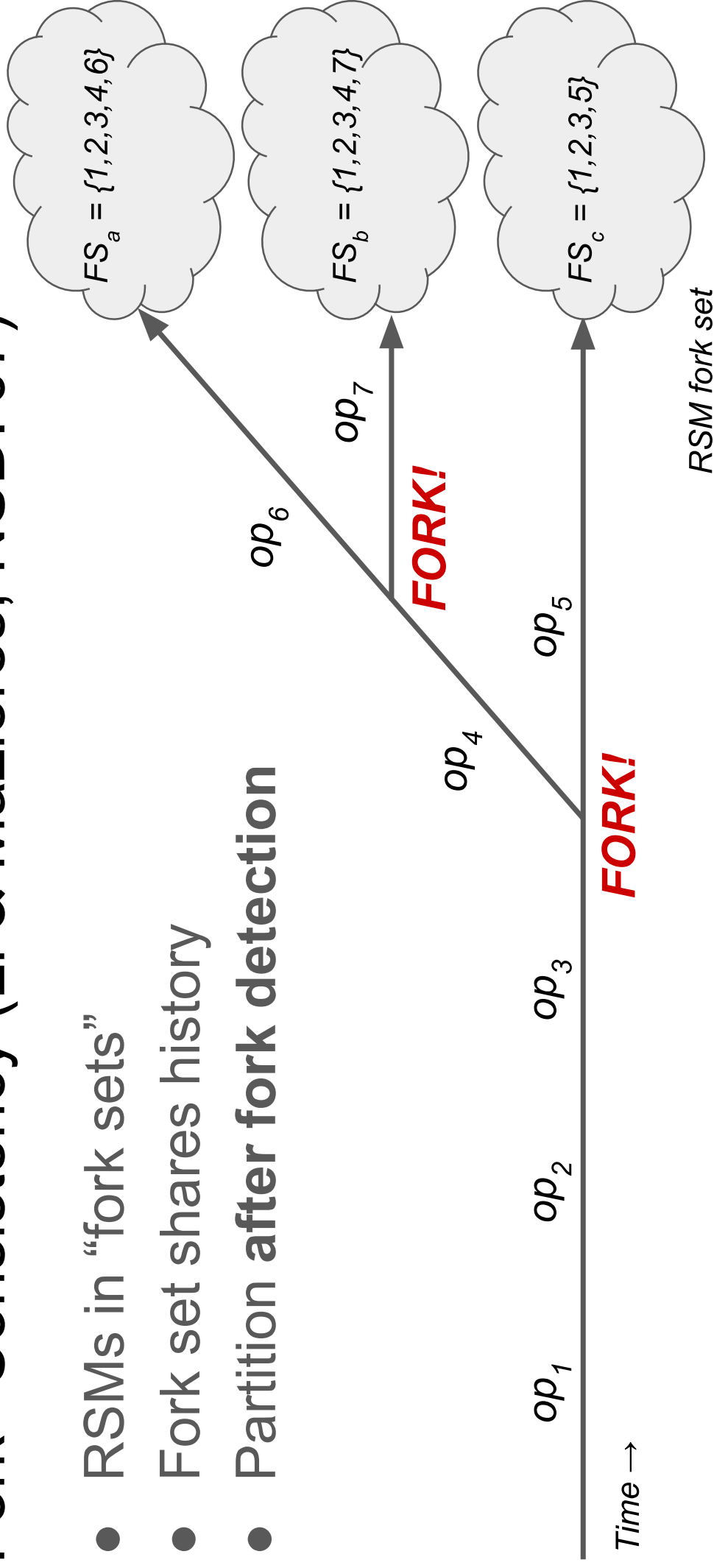
- Blockchain failure
  - Goes offline
  - “Centralization” attacks
- Blockchain forks
  - Data loss
  - Chain reorganization

# Virtualchain

- Fork\*-consistent RSMs on existing blockchains
- Fork detection & recovery
- Cross-chain migration

# Fork\*-Consistency (Li & Mazières, NSDI'07)

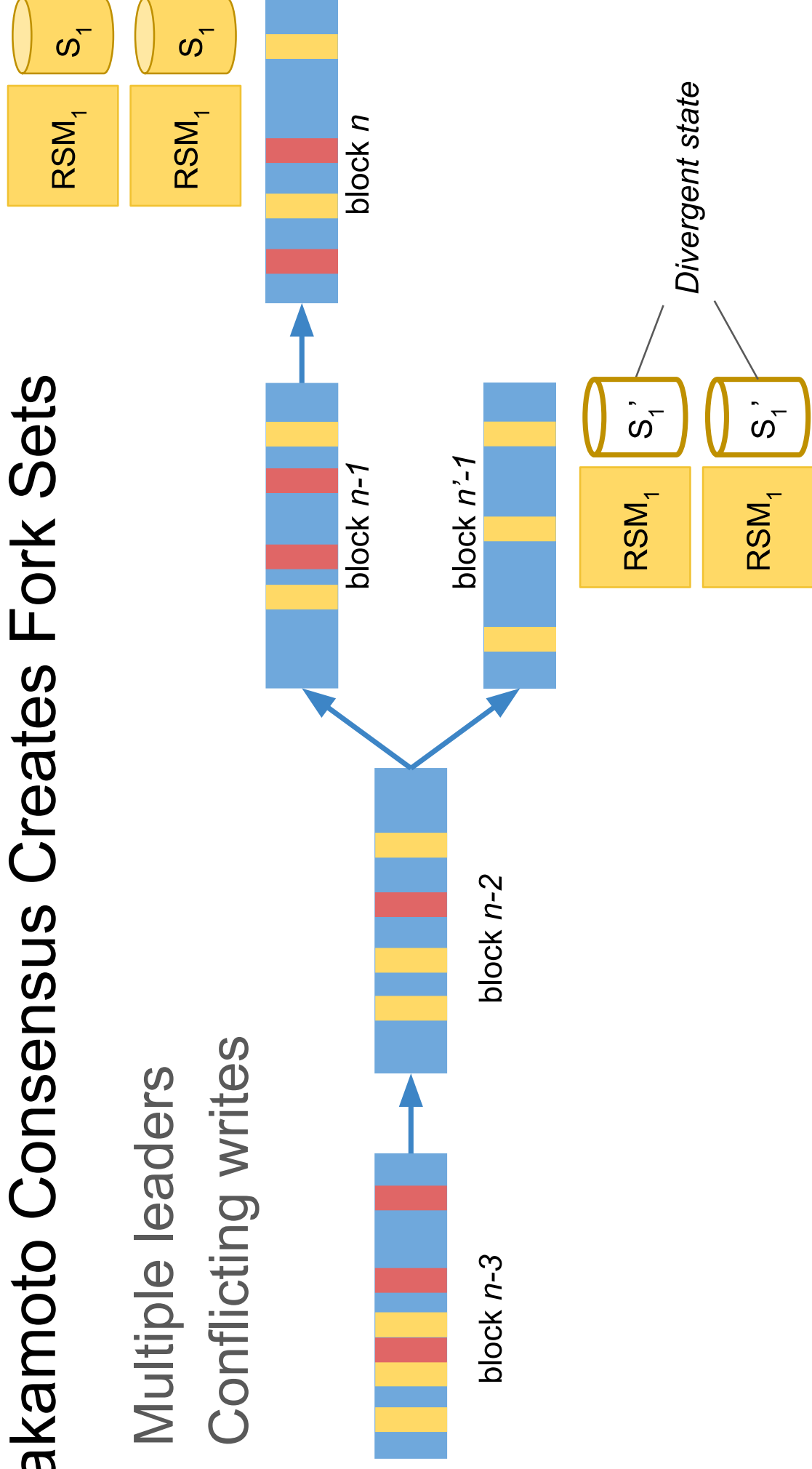
- RSMs in “fork sets”
- Fork set shares history
- Partition **after** fork detection





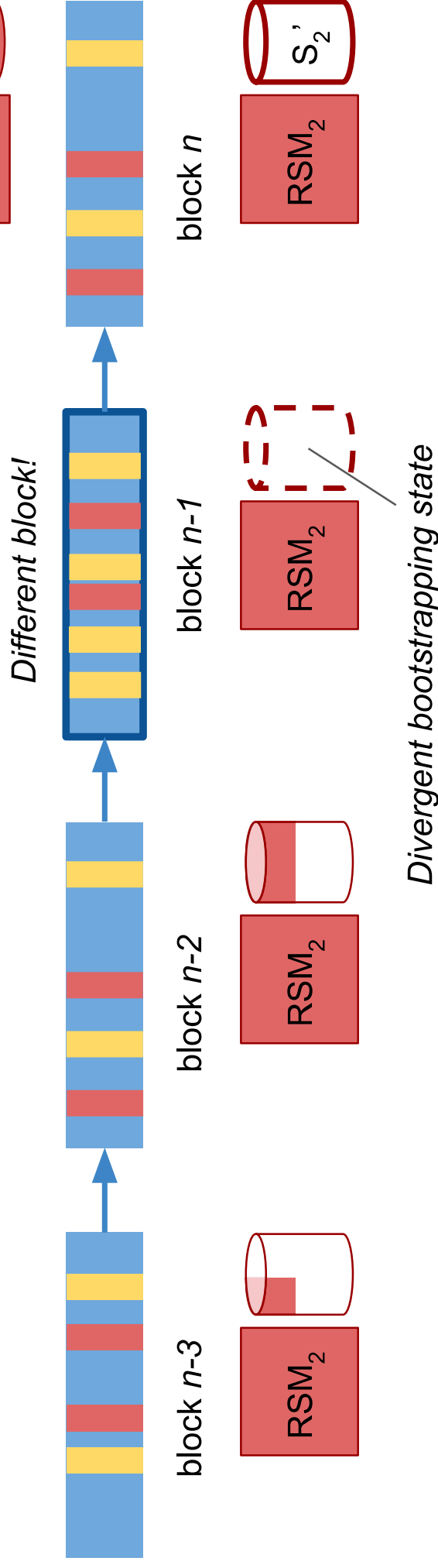
# Nakamoto Consensus Creates Fork Sets

- Multiple leaders
- Conflicting writes



# Reorganizations Create Fork Sets

- Conflicting TXs discarded
- Accepted TXs re-ordered

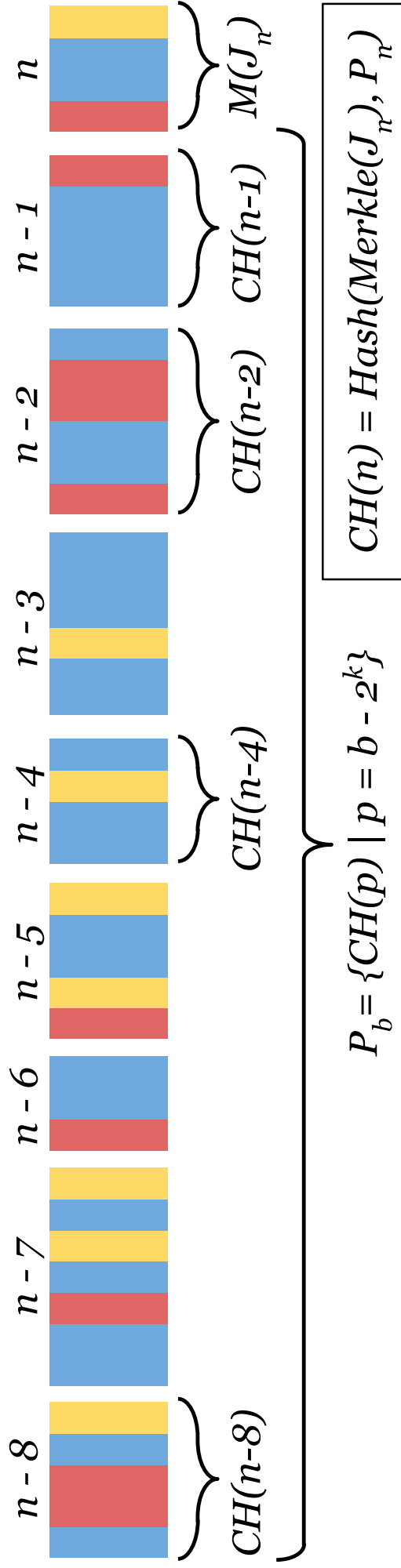


## Solution: Consensus Hashes

- In-band app-level consensus
- Used for:
  - Identifying fork sets (multiplexing)
  - Fork detection and recovery
  - Blockchain migration
  - Lightweight fork set selection

# Consensus Hash Construction

- $CH(n)$ : cryptographic hash
- Covers **state transition history** (“journal”)

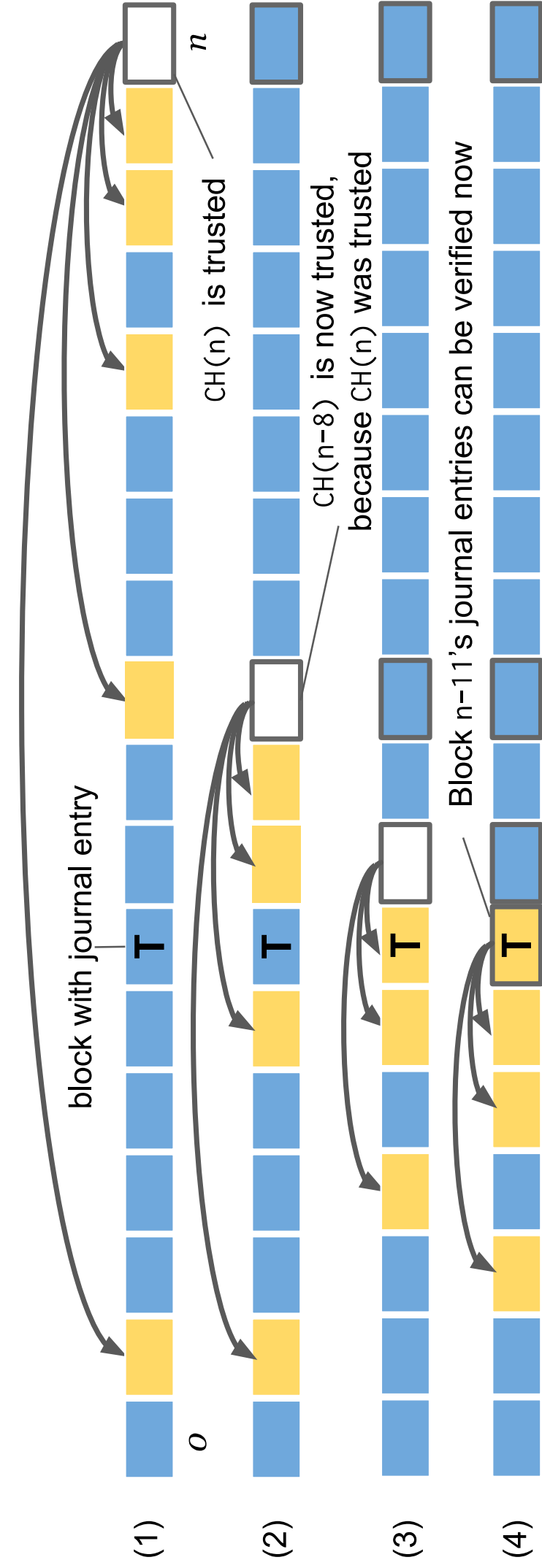


## In-band Consensus

- Fork sets: agree on  $CH(n)$  for all  $n$
- Client: embed latest  $CH$  in input TX
  - Obtained from preferred fork set
- Server: consider TX only if  $CH$  is “recent”
  - “Send/ACK” with  $K$ -block timeout

# Lightweight Fork Set Selection

- Given  $CH(n)$ , search for *characteristic state transitions*



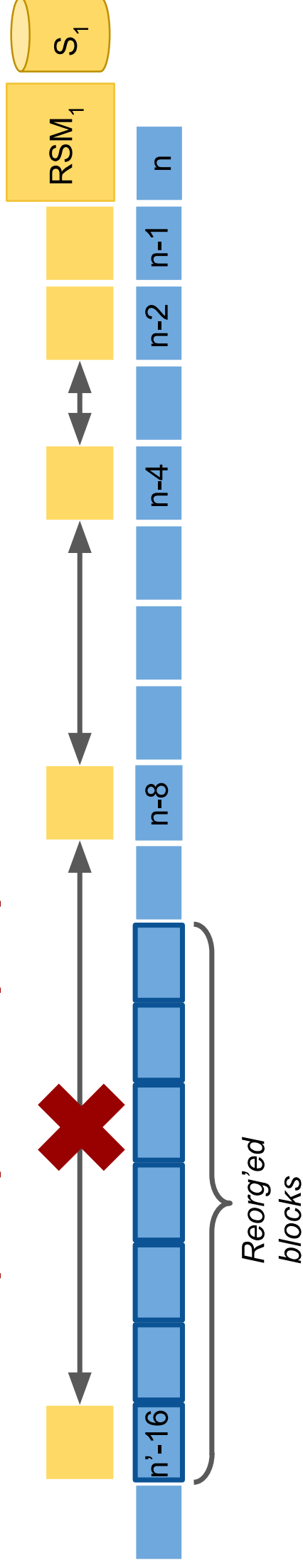
# Dealing with Blockchain Forks

- Most forks are short-lived
  - Avoid with “confirmations”
- Long-lasting forks are rare
  - But widely noticed!
  - Due to bugs or attacks

# Fork/Reorganization Detection

- Continuously audit CH history
- Alert on disagreement

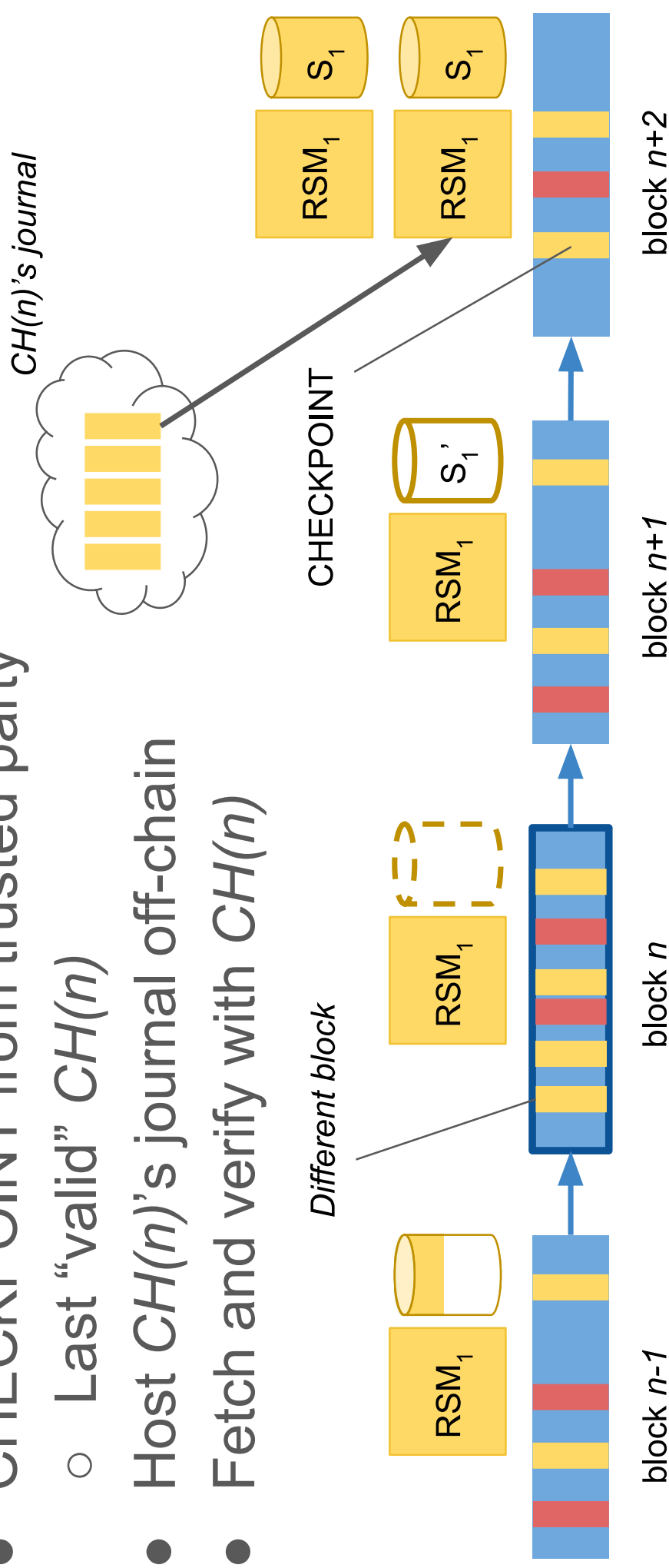
$$CH(n'-8) \neq CH(n-8)$$





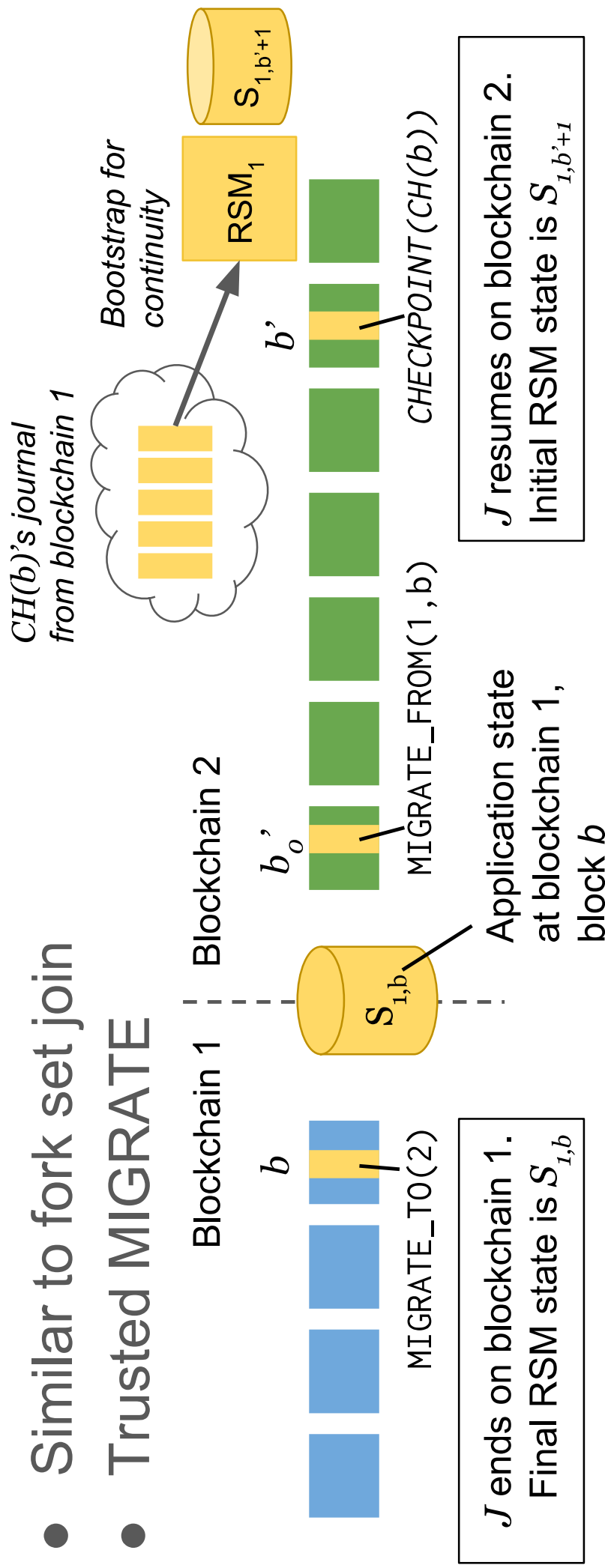
# Joining Fork Sets

- CHECKPOINT from trusted party
  - Last “valid”  $CH(n)$
- Host  $CH(n)$ ’s journal off-chain
- Fetch and verify with  $CH(n)$



# Cross-chain Migration

- Similar to fork set join
- Trusted MIGRATE

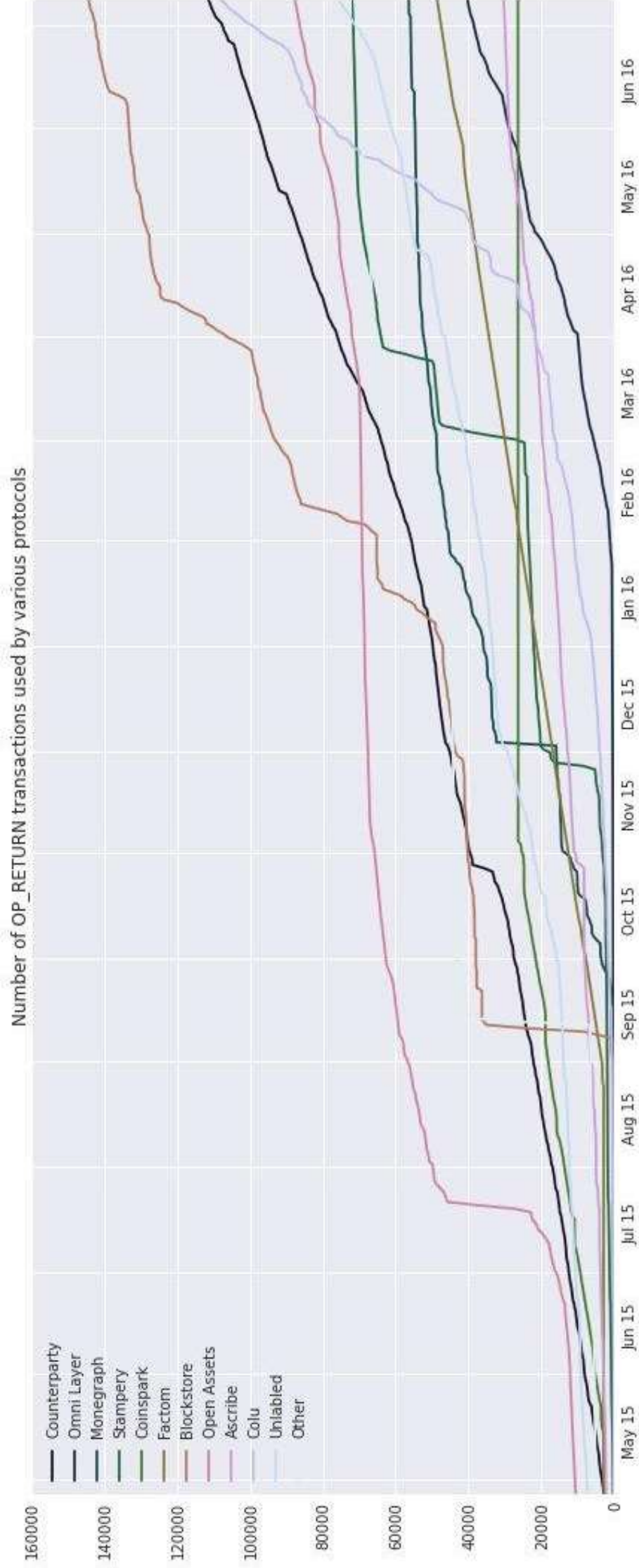


# On Centralization, Trust, and Cryptocurrencies

- Already trust RSM author
- Use CHECKPOINT, MIGRATE judiciously
  - Ignore with **no loss of security**
- Cryptocurrency: RSM input rate-limiter
  - RSMs becoming key use-case
  - Cloud market is >10x more valuable

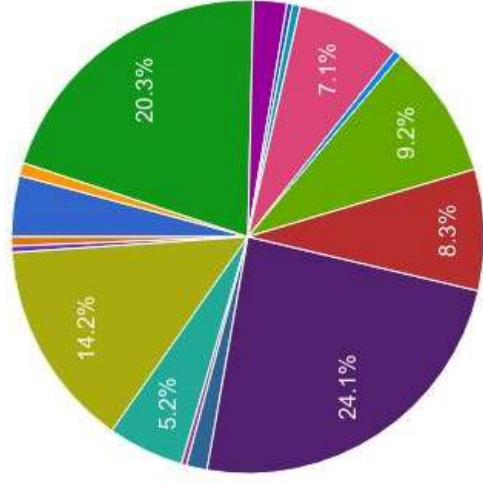
# Example: Bitcoin OP\_RETURN Usage

Source: Harry Kalodner

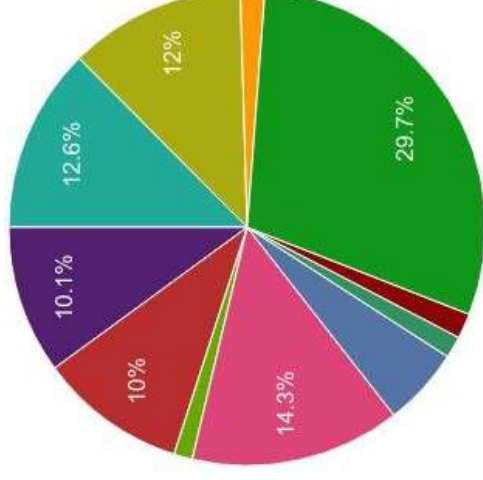


# Concluding Remarks

- In production for >1 year in Blockstack
- <https://github.com/blockstack/blockstack-virtualchain>
- Ali, Nelson, Shea, Freedman (ATC'16)
- Migrated from Namecoin to Bitcoin



Source:  
opreturn.org



Thank you!

Questions?