BLOCKCHAINS

ARCHITECTURE, DESIGN AND USE CASES

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

PRAVEEN JAYACHANDRAN
IBM RESEARCH,
INDIA

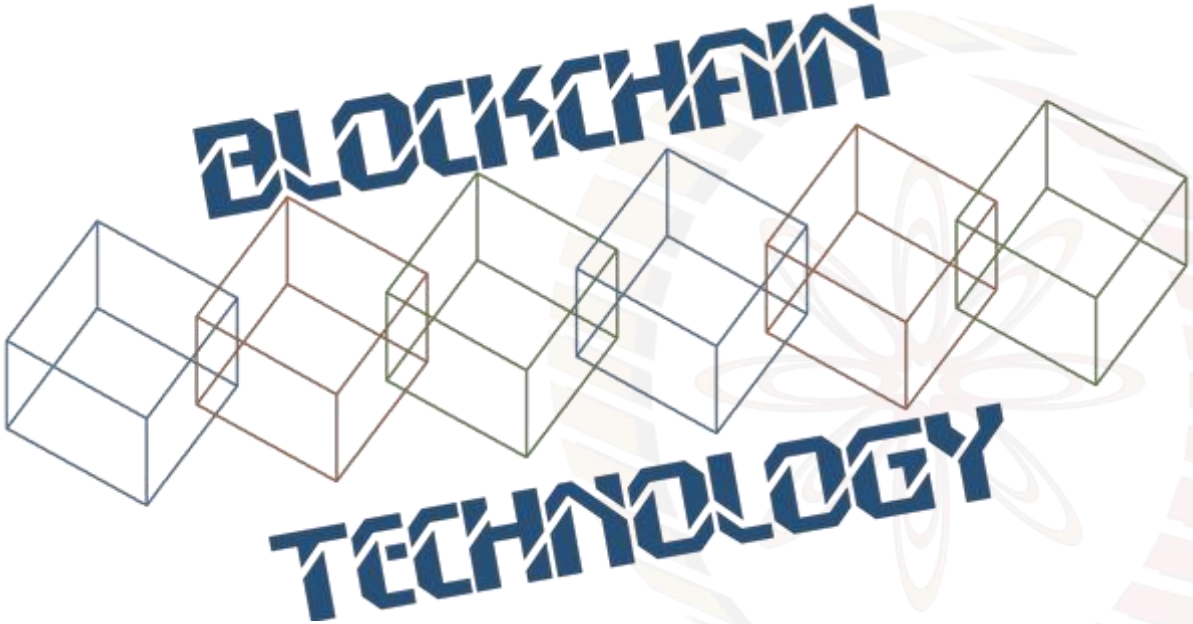
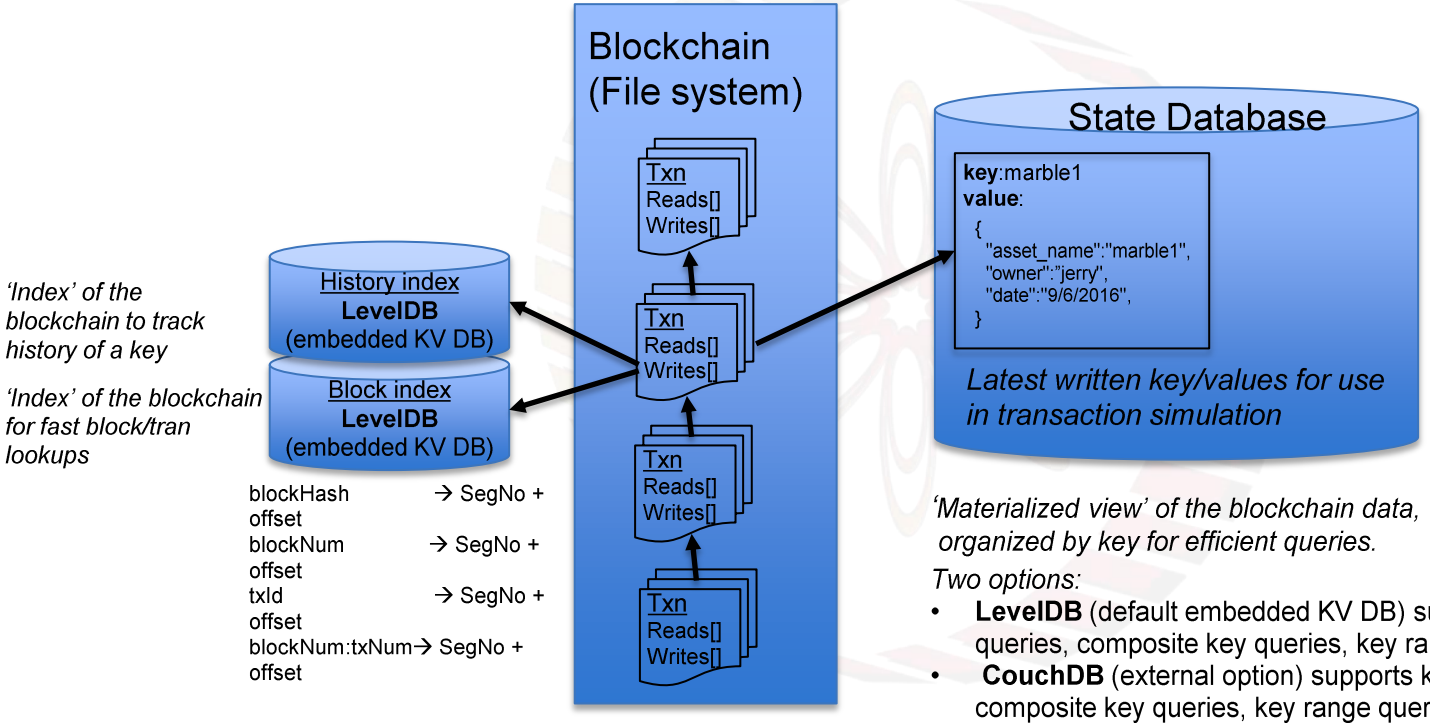


*Image courtesy: http://beetfusion.com/

FABRIC - SIDEDB

Ledger in Hyperledger Fabric



State Database Options

- In a **key/value database** such as **LevelDB**, the content is a blob and only queryable by key
  - May not meet chaincode, auditing, reporting requirements for many use cases
- In a **document database** such as **CouchDB**, the content is JSON and fully queryable
  - Meets a large percentage of chaincode, auditing, and simple reporting requirements
  - For deeper reporting and analytics, replicate data to an analytics engine such as Spark (future)
  - Id/document data model compatible with existing chaincode key/value programming model, therefore no application changes are required when modeling chaincode data as JSON

SideDB Motivation

- Need for selectively sharing transaction data with certain entities, but blockchain by default replicates across all peers
  - All peers in a channel have access to state maintained by chaincode
  - Ordering service can view transaction data
  - Anyone querying for blocks can view all data in channel
- Data privacy needed in many applications including healthcare, KYC, financial services
- Only evidence (hash) needs to be sent to ordering service as well as stored in the chain of blocks
- Should be able to query/update private data using chaincode

SideDB Overview

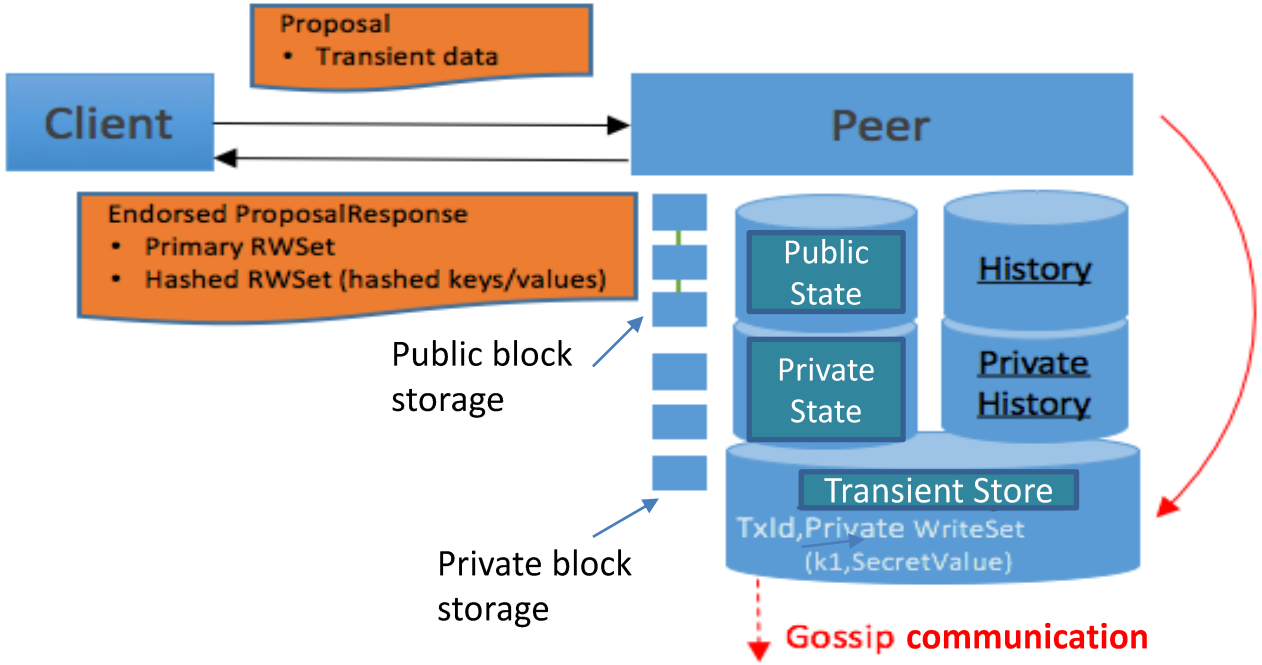
- Chaincode is tuned to store *state hashes* vs. state (private Data)
- Private data grouped in collections
- Collections associated to access policies
- Private data of a collection would be stored solely to peers who satisfy the collection's access policy

- Why not use channels? Not meant for data privacy, will mean that even transactions and the chain are distinct

SideDB: One Collection (1/2)

Endorsement Phase
<H(key), H(value>
written to public state,
while <key, value> held
in private state

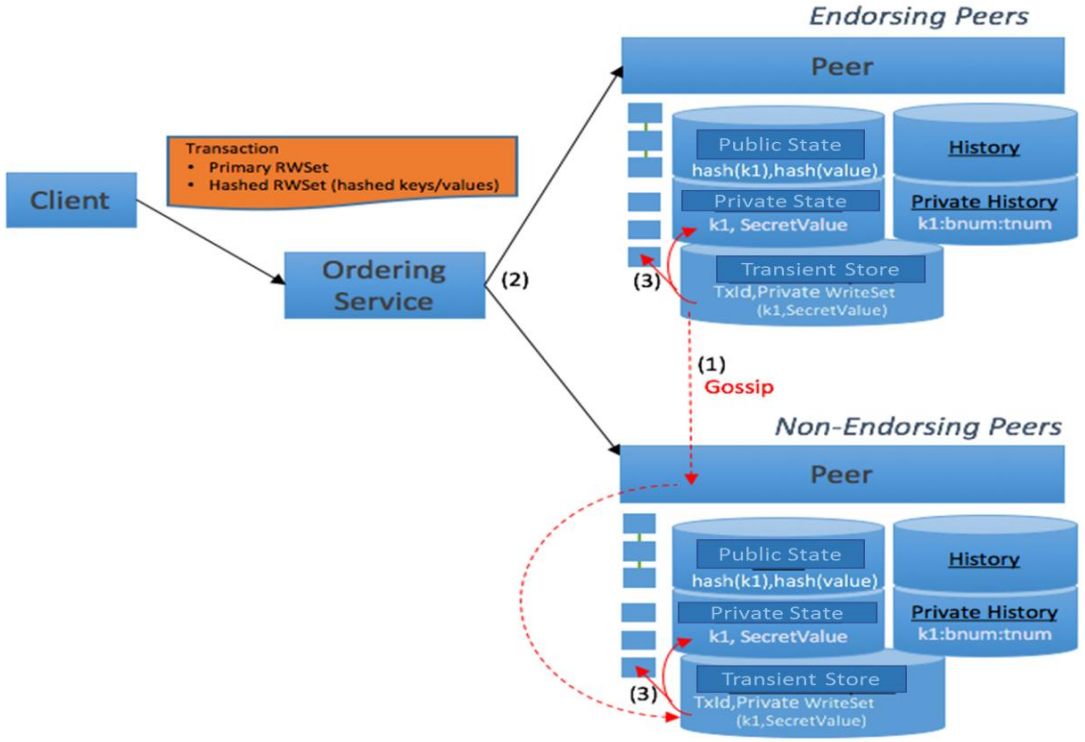
Private state
disseminated using
gossip communication
(also used for orderer-
peer communication)



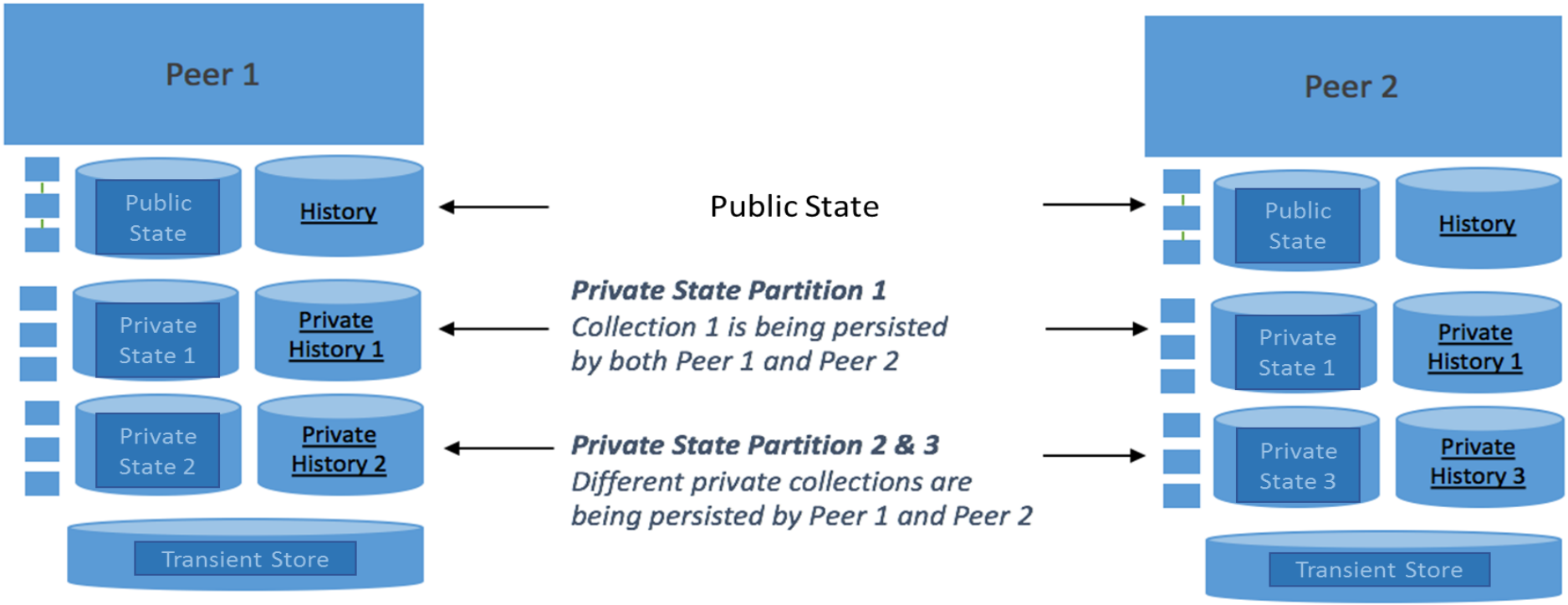
SideDB: One Collection (2/2)

Validation and Commit Phase

Validation of public RWset and hashed RWset on all committing peers



SideDB: Multiple Collections



Define Collections for each Chaincode and Channel

- Define collections during chaincode deployment
  - Need to have a special peer command to specify collections per chaincode per peer.
  - Can easily add/remove collections to existing chaincode

- Using channel configuration
  - Easy to configure collection
  - To add/remove collections on demand need channel reconfiguration message

Fun Reading

- Hyperledger Fabric documentation, Ledger: http://hyperledger-fabric.readthedocs.io/en/release-1.0/ledger.html
- Hyperledger Fabric SideDB: https://jira.hyperledger.org/secure/attachment/12720/PrivacyEnabledLedger20171022.pptx


thank you!