



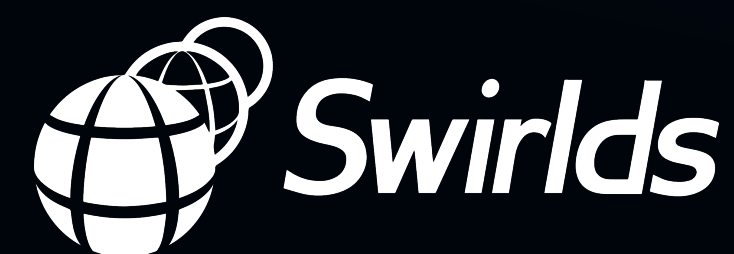
Hedera[™] Hashgraph

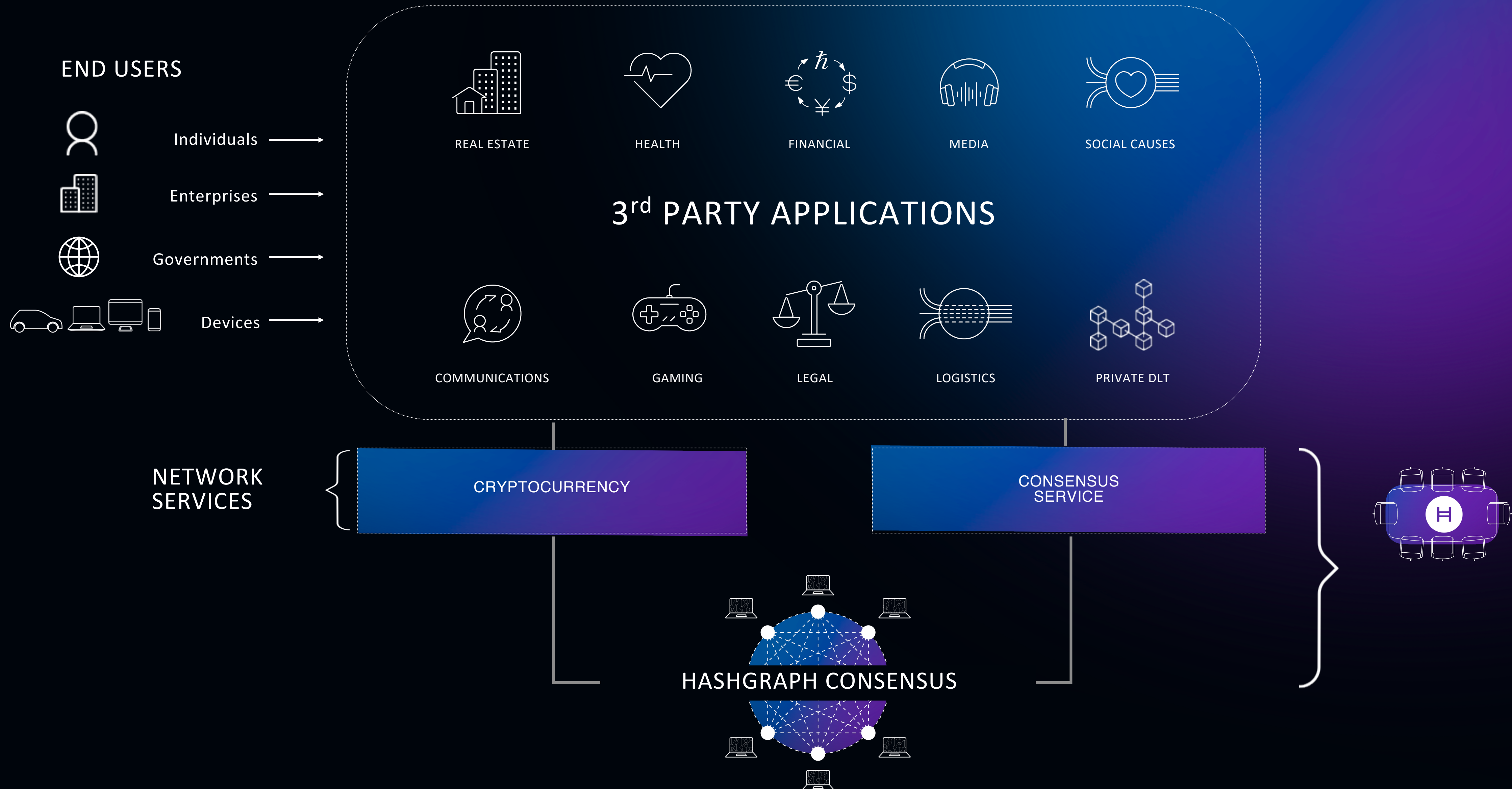


The Hedera Governing Council



Building the future together.







LOOK UP ACCOUNT

1D ▼

Node 1
140,710 *txns* * | 19.41739





Adoption to Date

Use Case	Description	Value Proposition	Example
<u>CBDC</u>	Build compliant and performant central bank digital currencies on a distributed architecture.	Hedera provides a global and scalable messaging layer for nodes operated by permissioned entities.	Project New Dawn InterWork Alliance Start-up (Confidential)
<u>Stable Coin</u>	Create crypto asset with stable market value for peer to peer payments.	Hedera provides a global and scalable messaging layer for nodes operated by permissioned entities.	Enterprise (Confidential) Carbon Council Member
<u>Digital Identity</u>	Establish verifiable credentials for any person or entity.	Hedera provides highly performant method for establishing and verifying credentials through the Hedera DID SDK.	MyEarthID
<u>Proof of Compliance</u>	Verifiably prove user information capture and deletion.	Hedera proof of action microservice generates cryptographically verifiable proofs of data capture and deletion.	Manetu
<u>Auditable Logs</u>	Applications prove events to third parties to improve trust and auditability of data.	Hedera proof of action microservice provides a simple API to generate cryptographic proofs for any app event.	AdsDax Coupon Bureau Global Top 10 University (Confidential)
<u>Enterprise Integration</u>	Extend core enterprise applications & middleware with verifiable proof of transactions between corporates silos and business partners	Hedera proof of action microservice together with an auditable log provides a decentralized means to validate transactions between business partners	Global Auto-Manufacturer (Confidential)
<u>Network Interoperability</u>	Easily enable atomic swaps between distinct permissioned networks with Corda or Hyperledger Fabric.	Hedera integrates with leading permissioned DLTs like Fabric and Corda to provide a universal ordering service.	Enterprise (Confidential) Insurance Consortium



Pluggable Consensus with HCS

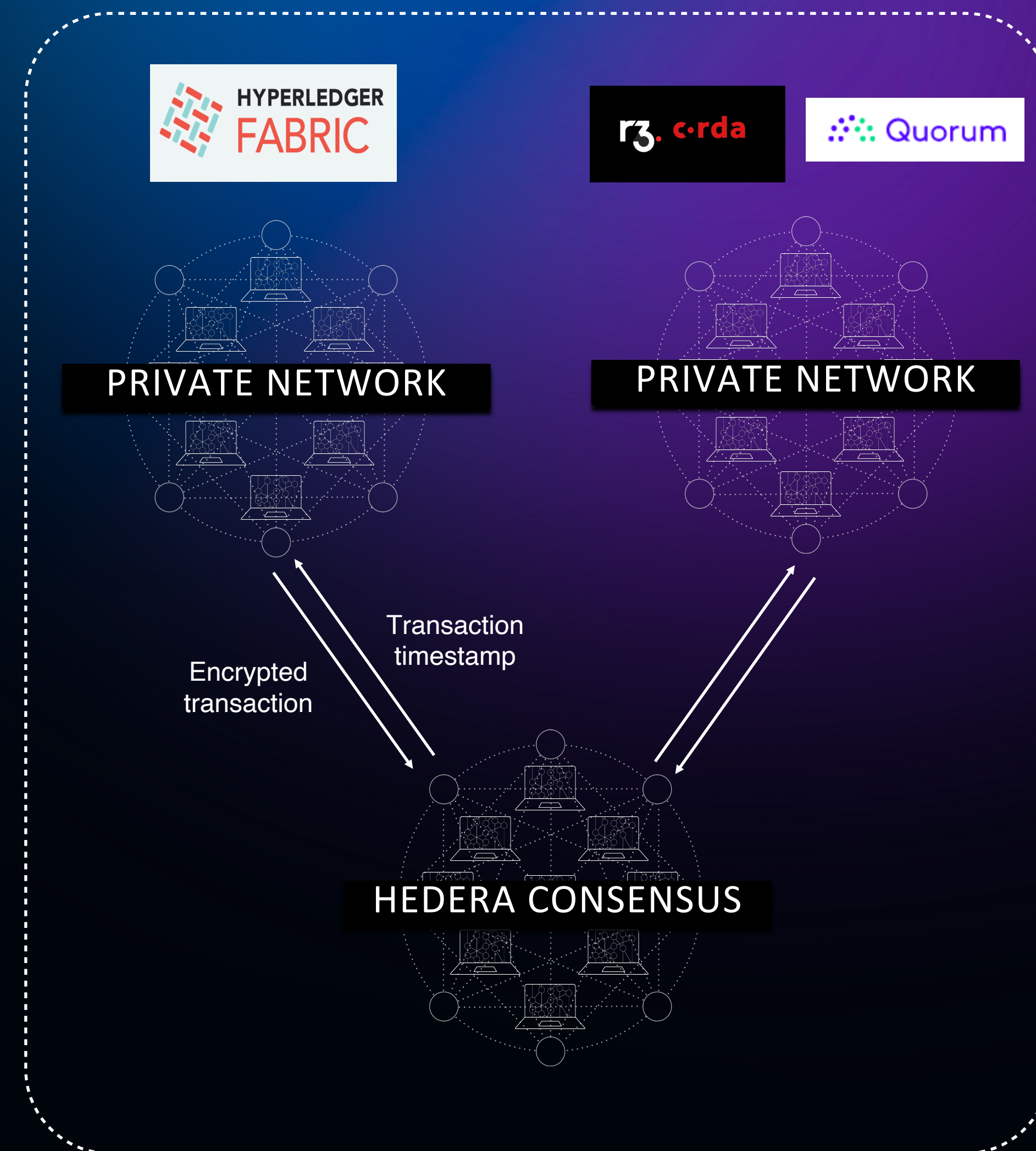
Hedera Consensus Service provides auditable log of transaction for distinct networks and applications.

Permissioned networks now share a single consensus engine, enabling:

- Custom data access and encryption
- Public trust
- Verifiable timestamps
- Decentralized, shared service
- Finality of transaction order
- High performance

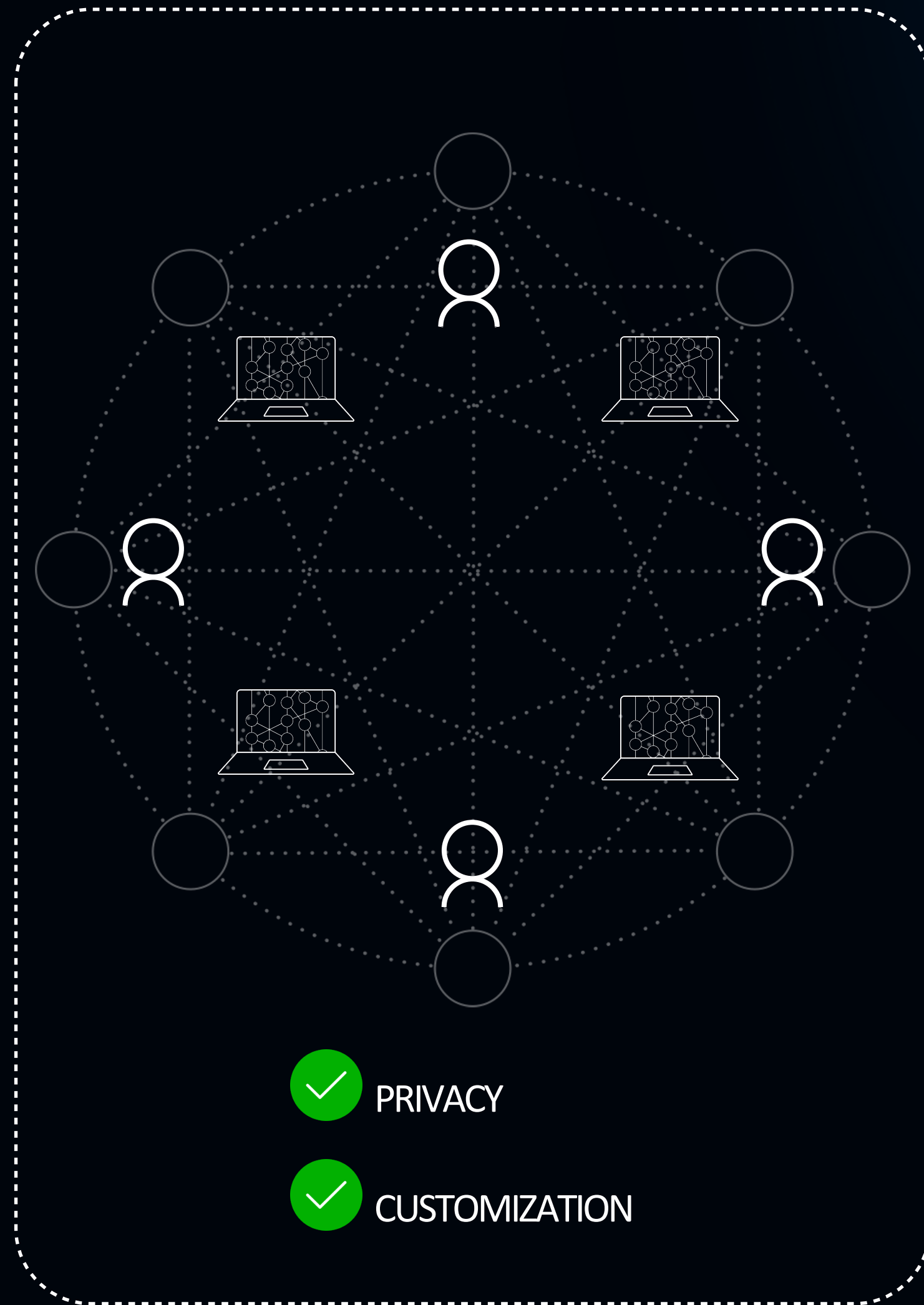
Hedera supports integrations between the Hedera Consensus Service, Hyperledger Fabric, and R3's Corda

- ✓ PRIVACY
- ✓ CUSTOMIZATION
- ✓ DATA ENCRYPTION
- ✓ IDENTITY MANAGEMENT
- ✓ AUDITABILITY
- ✓ AVAILABILITY
- ✓ DECENTRALIZATION
- ✓ PERFORMANCE





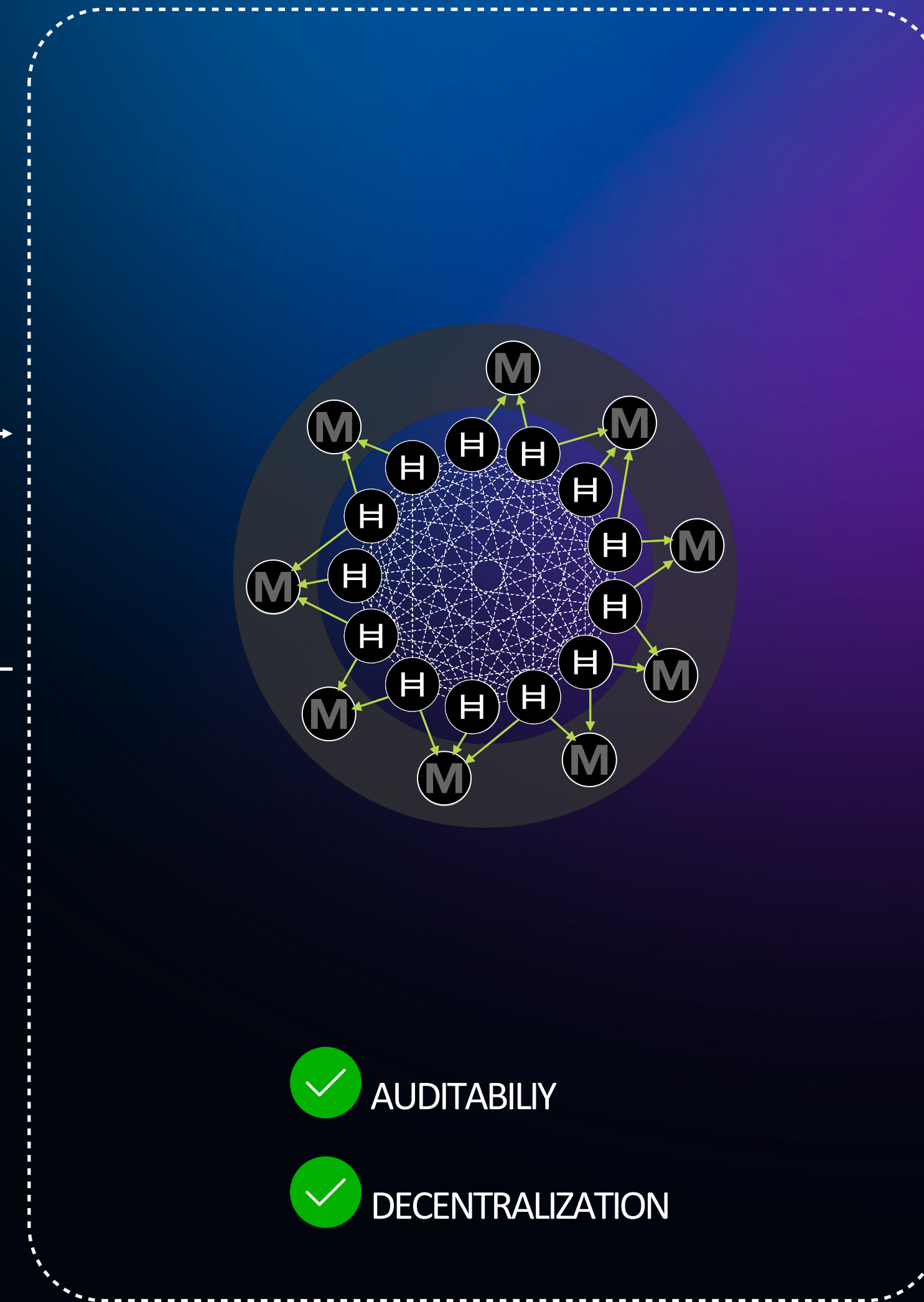
Fabric Network



Submit message

Consensus order

Consensus Service





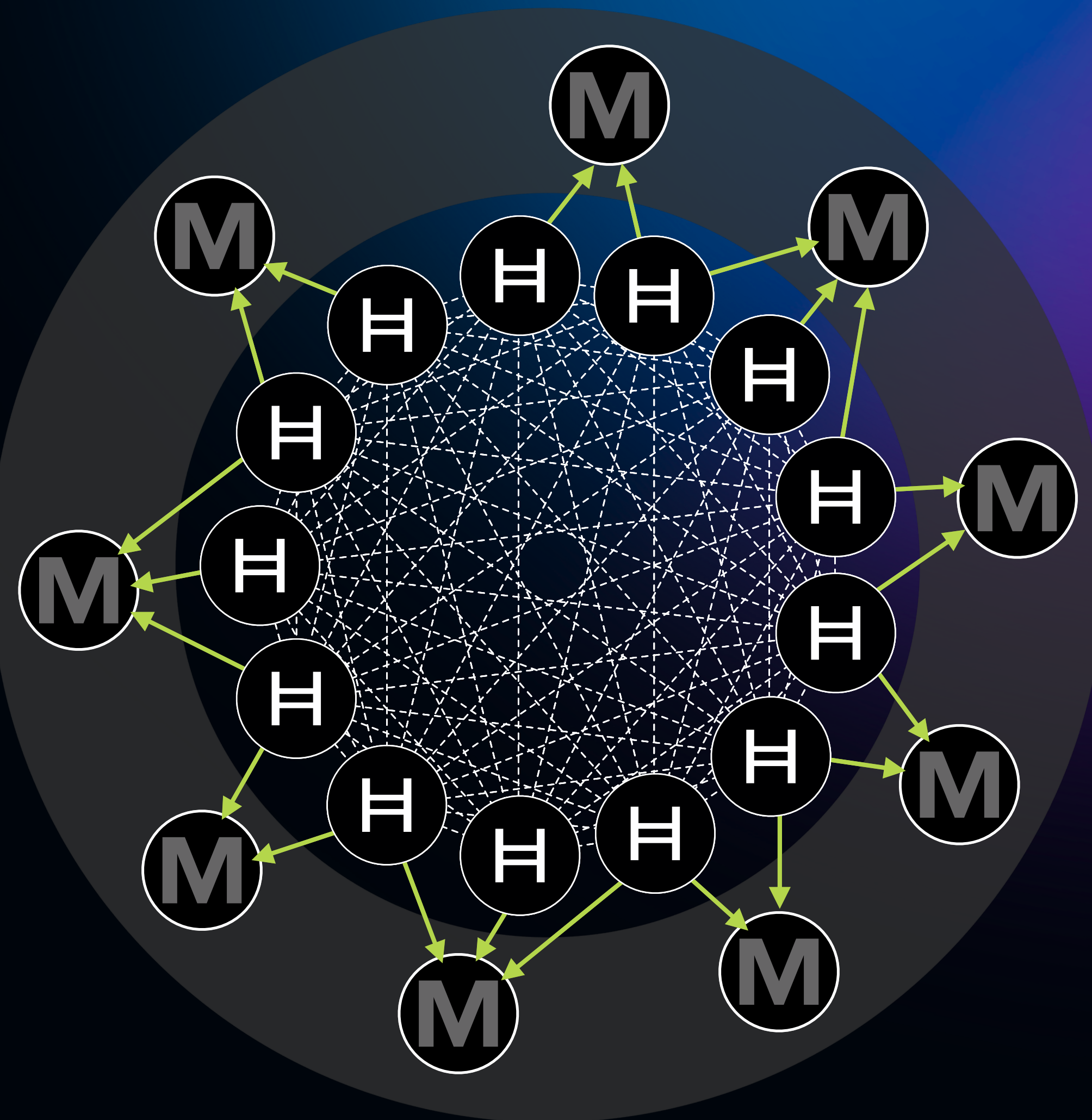
Hedera mainnet



Hedera mirrornet

HCS Message Submission

- Message
- TopicID
- Submitting Account



Hedera



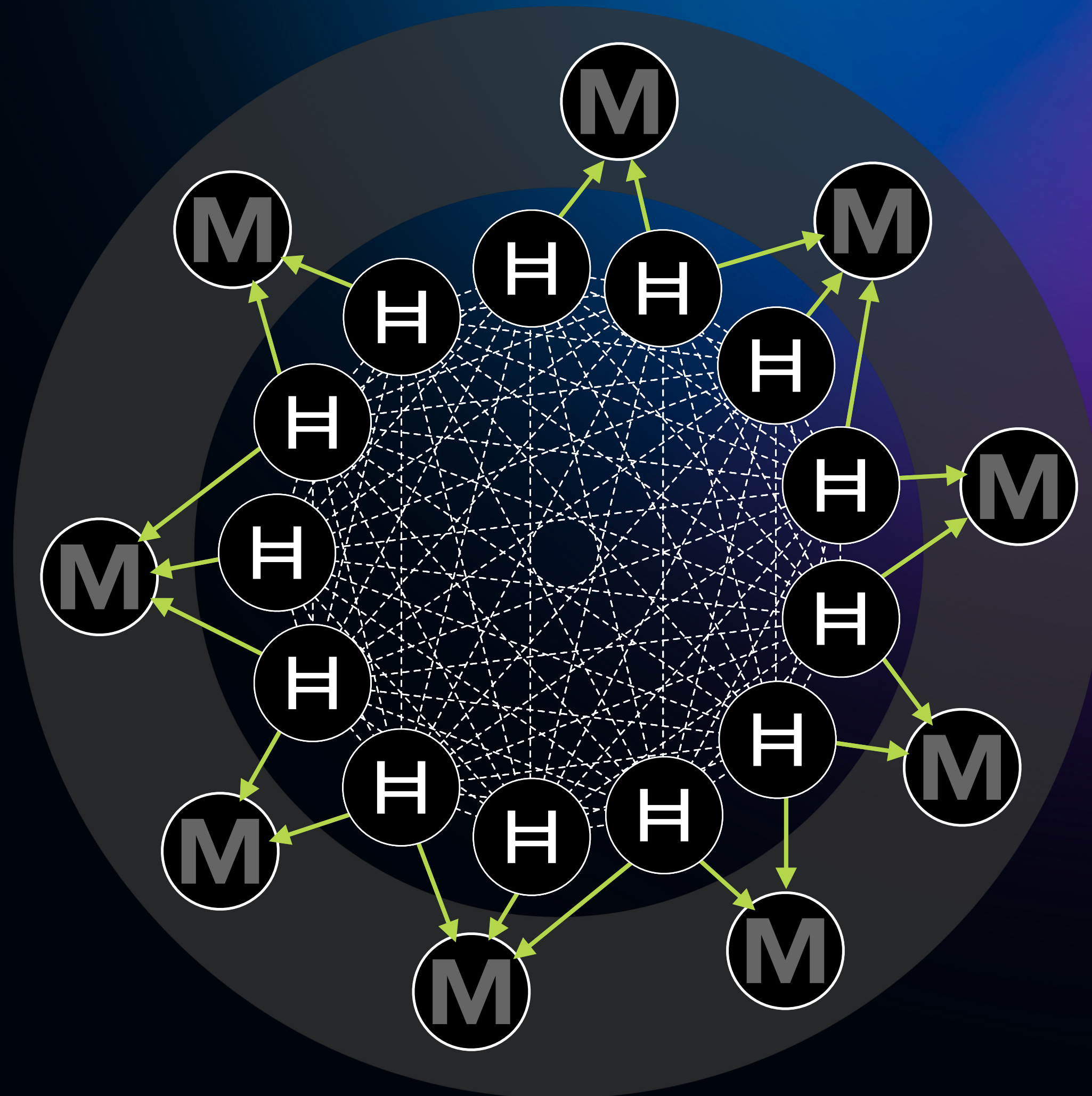
Hedera mainnet



Hedera mirrornet

HCS Message Response

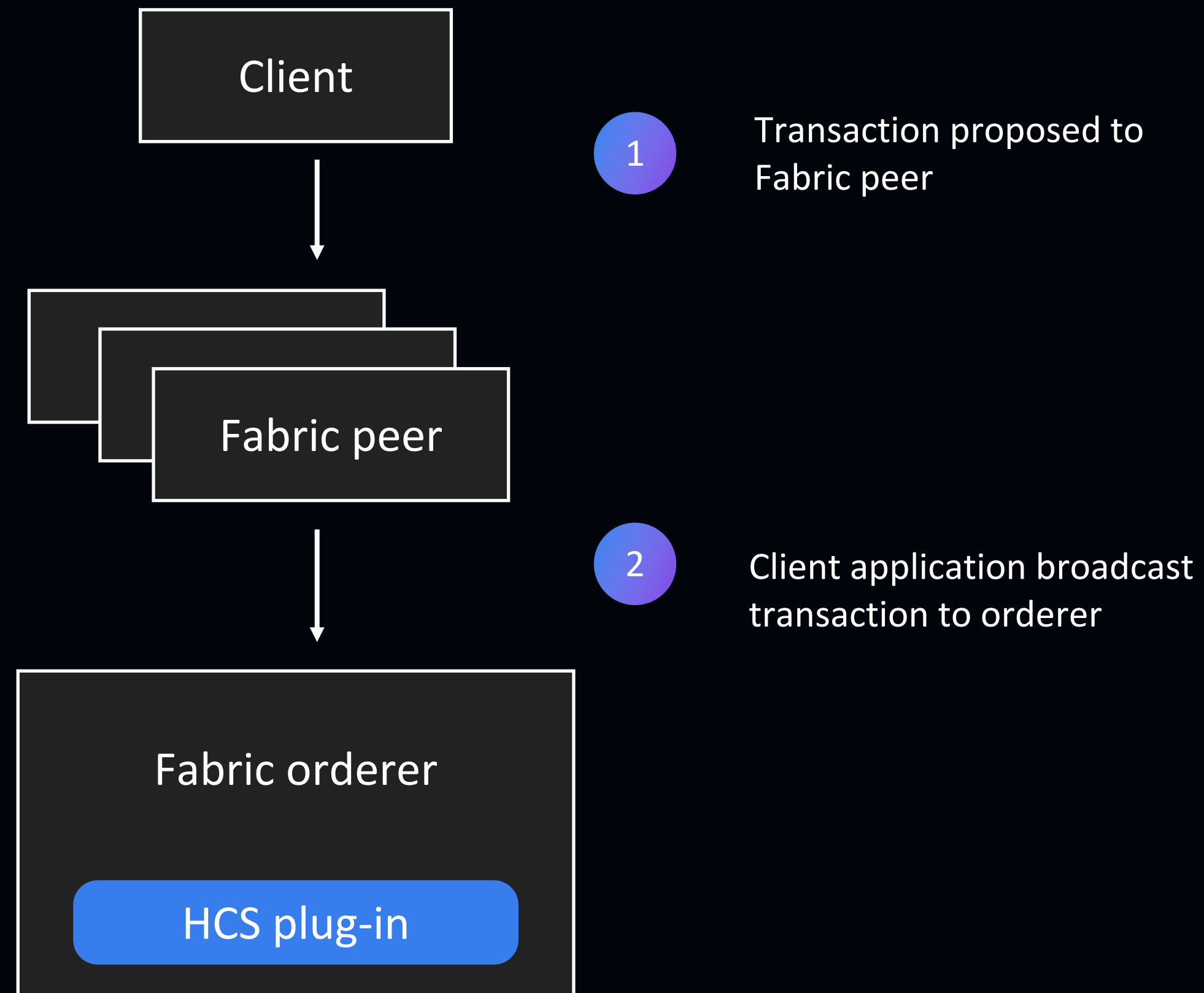
- Consensus Timestamp
- Sequence Number
- Running Hash



Hedera

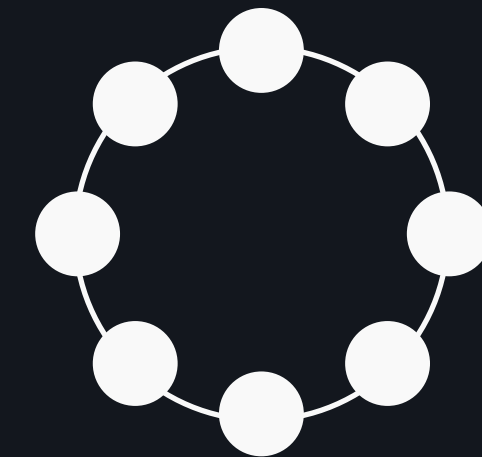
HYPERLEDGER FABRIC

Lifecycle of a transaction

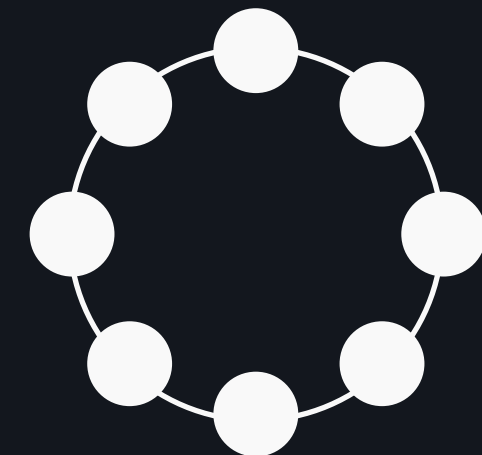


Hedera Hashgraph

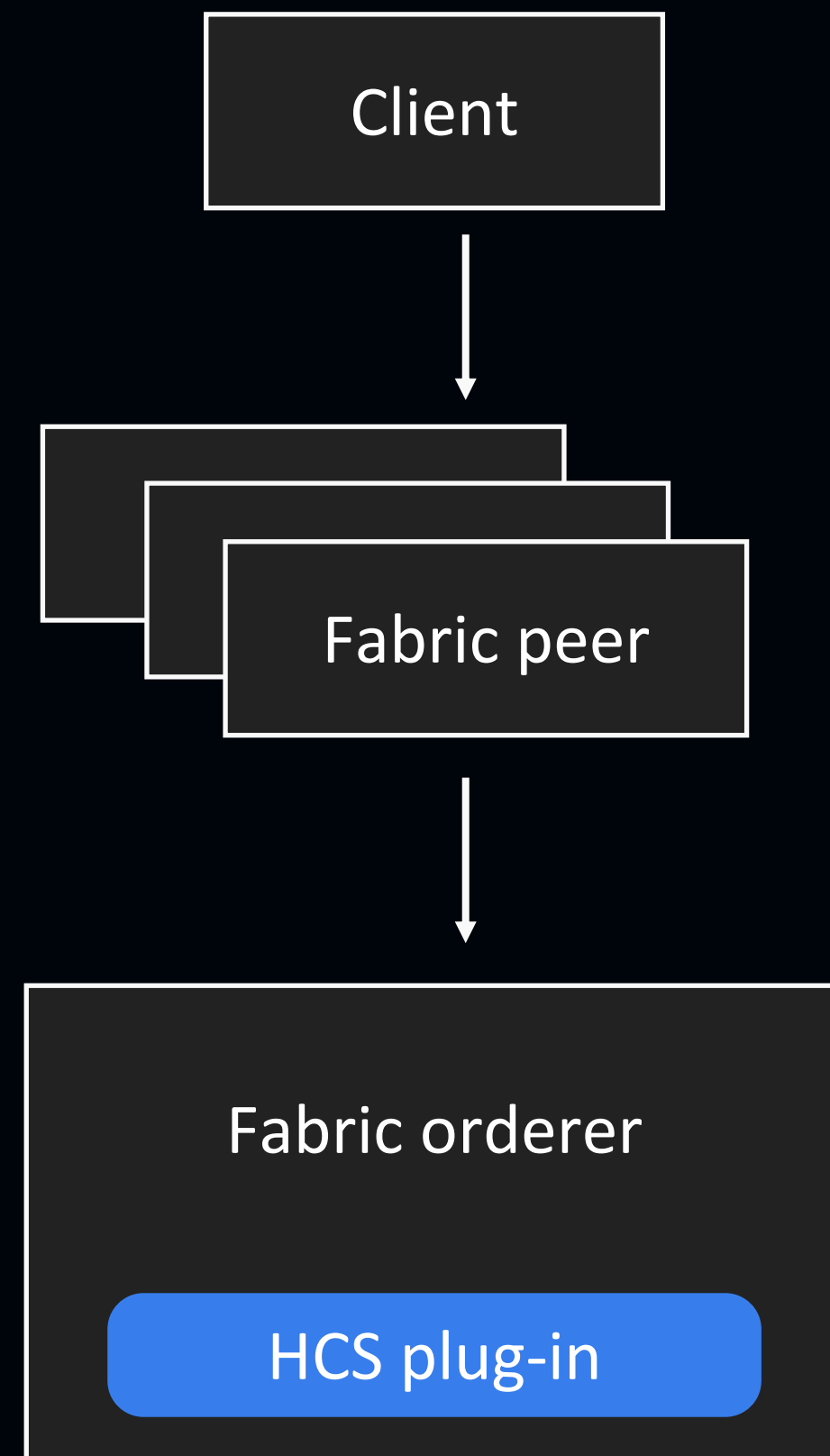
Consensus nodes



Mirror nodes



Lifecycle of a transaction



3

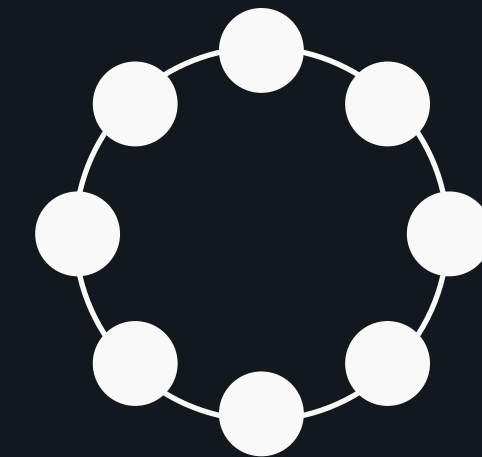
Transaction fragmented into messages, associated with topicID

Hedera Hashgraph

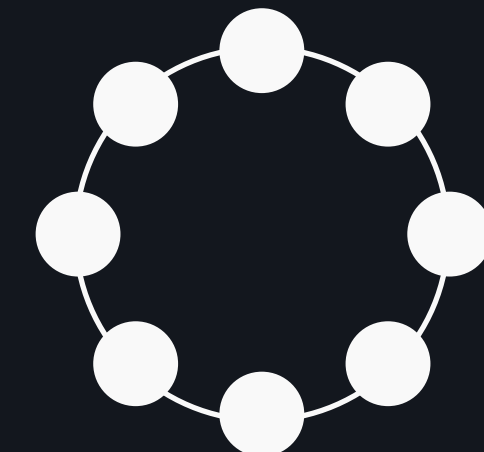
4

Messages submitted to topic using HCS

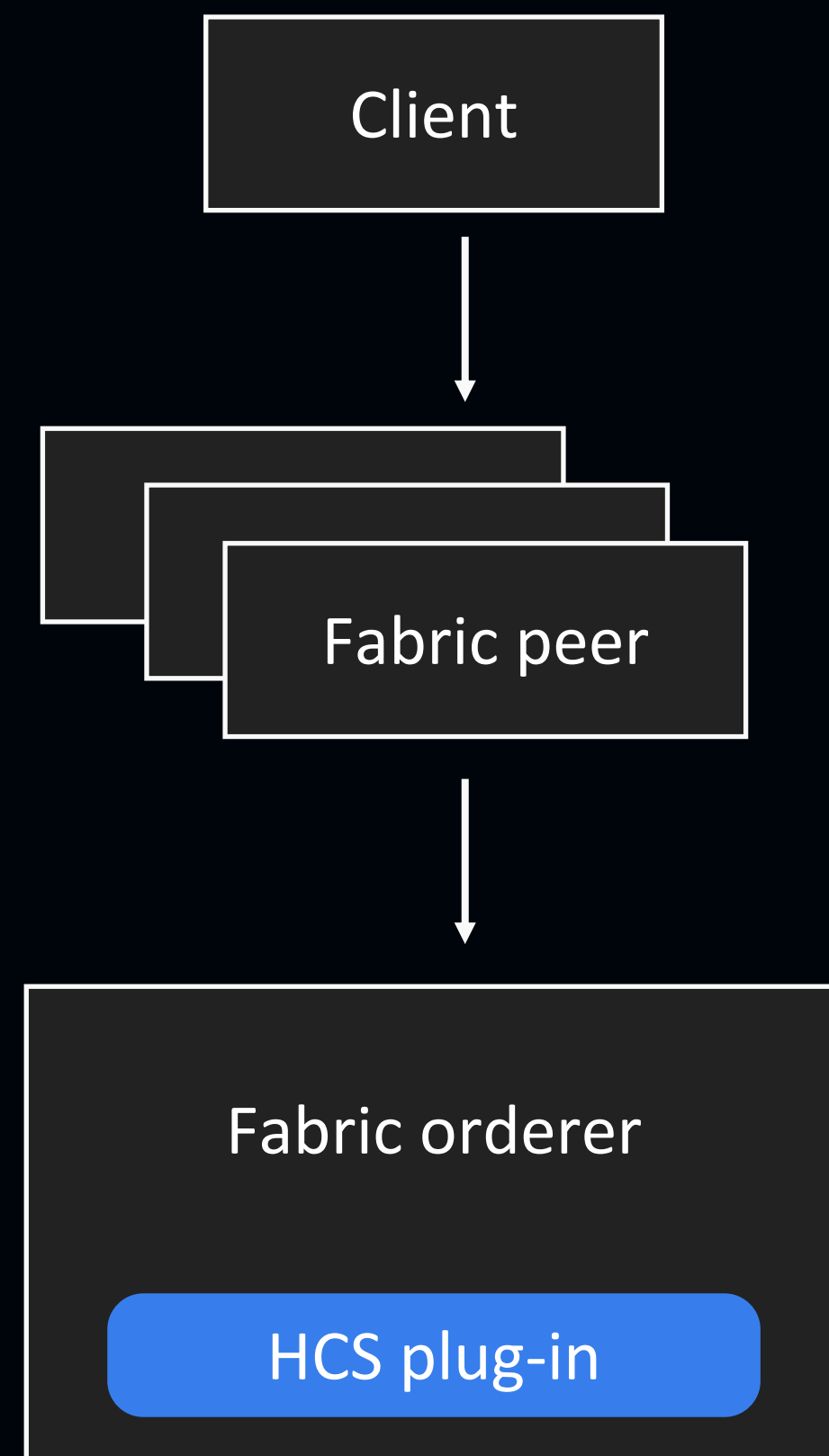
Consensus nodes



Mirror nodes

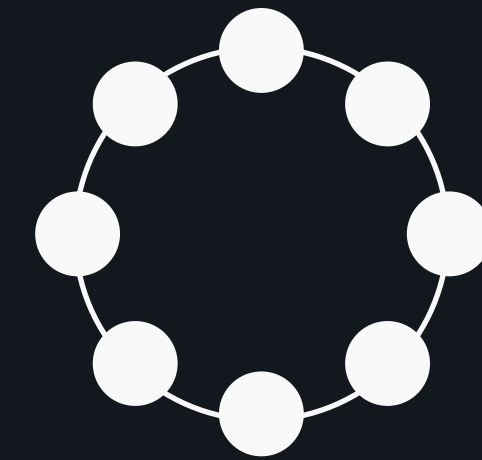


Lifecycle of a transaction



Hedera Hashgraph

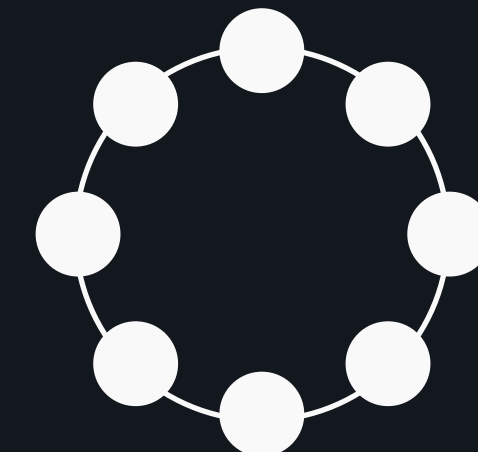
Consensus nodes



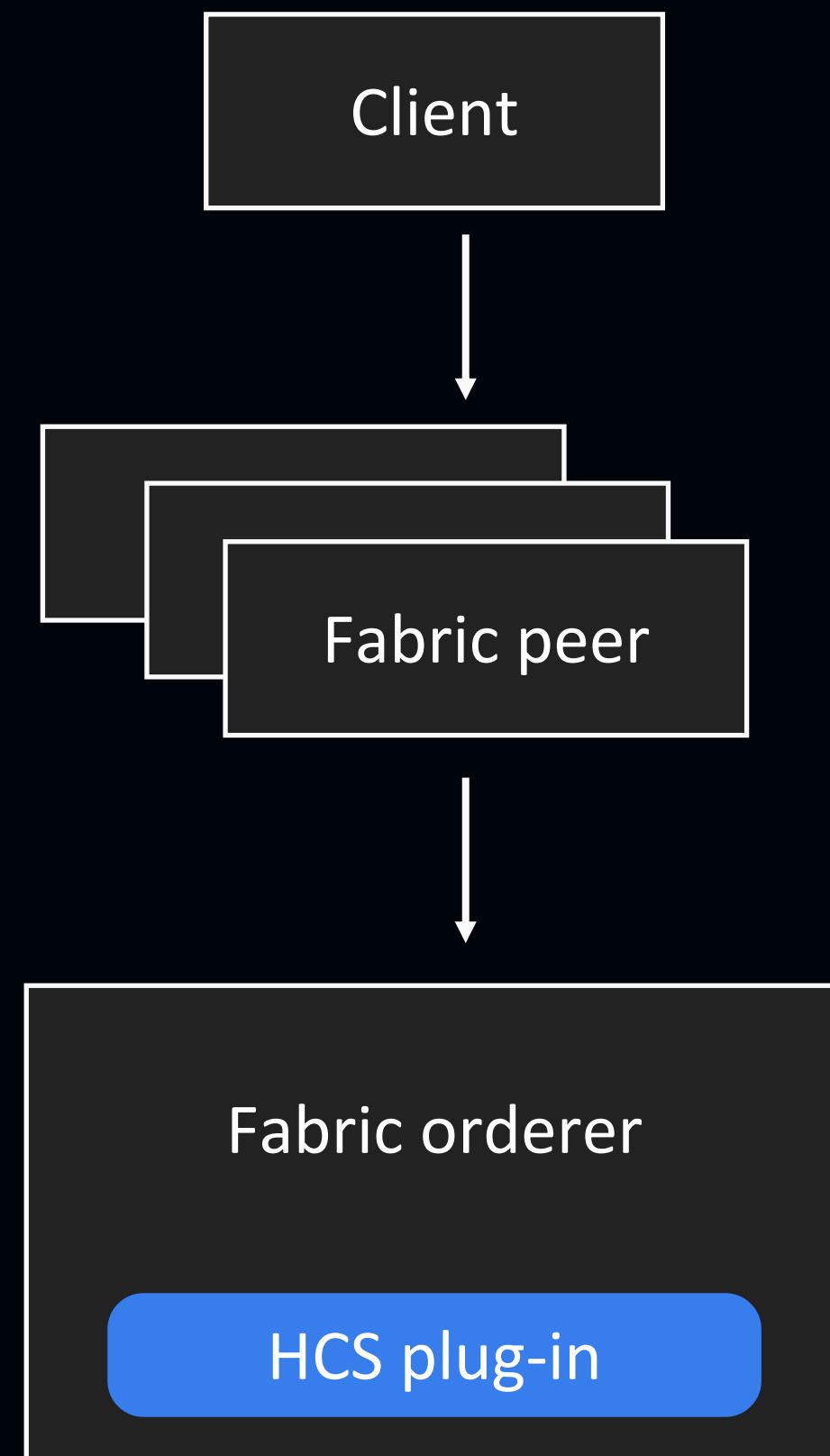
5

Event reaches consensus, receiving consensus timestamp and state proof

Mirror nodes

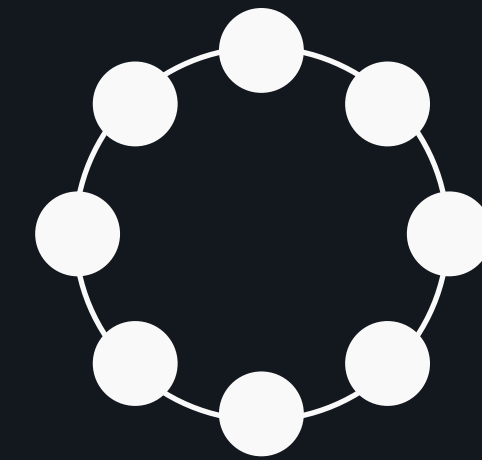


Lifecycle of a transaction

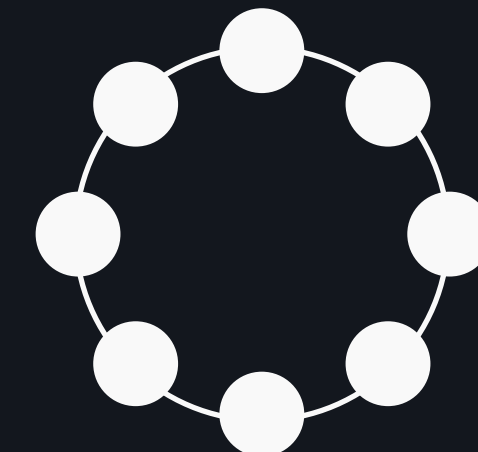


Hedera Hashgraph

Consensus nodes



Mirror nodes

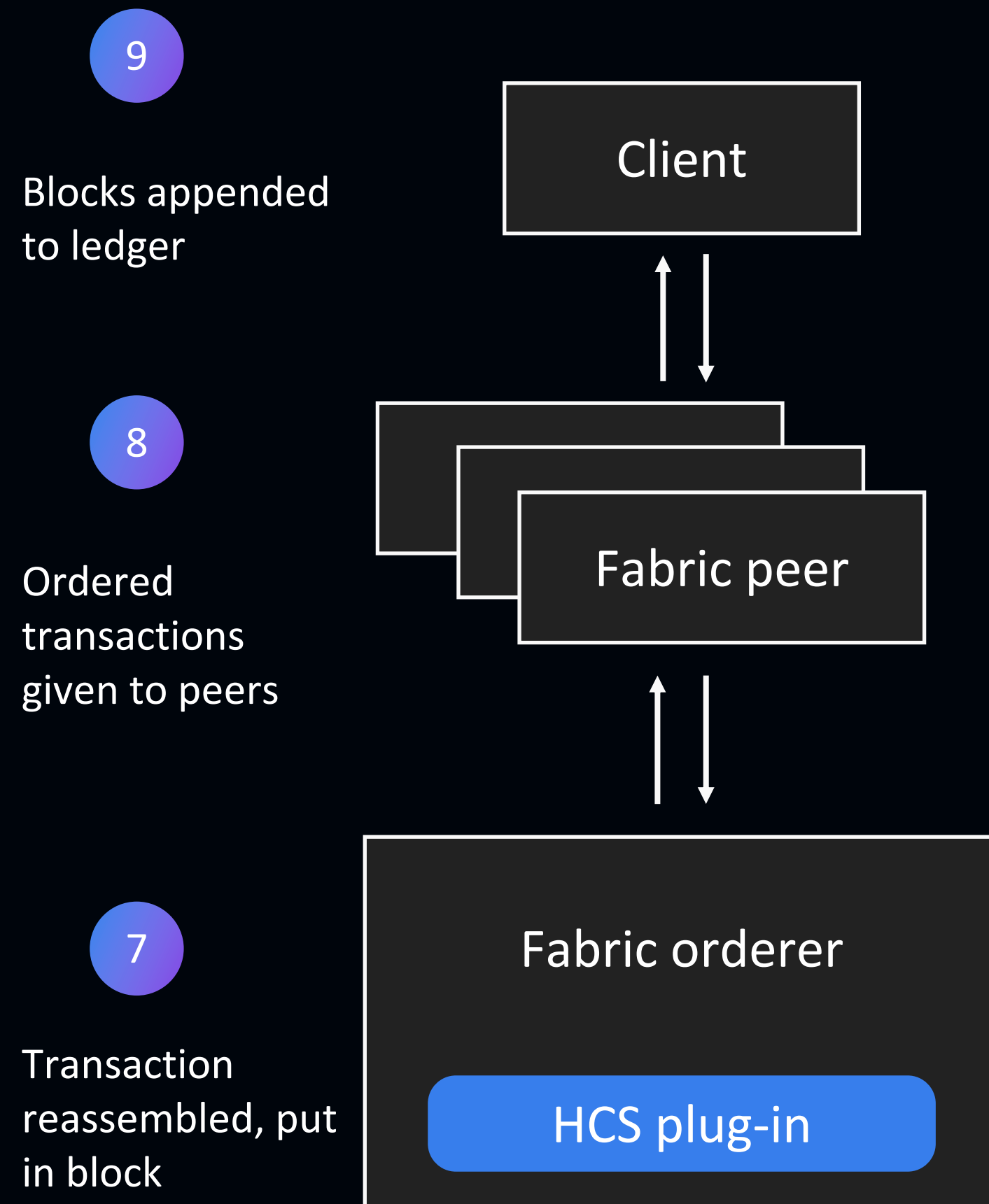


6

Notify of message order

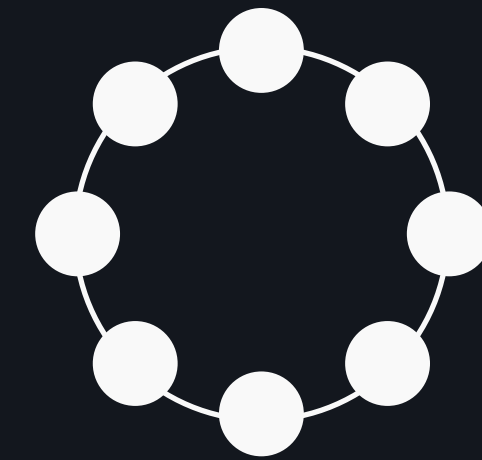
Message + TopicId + timestamp + sequenceNumber + runningHash
+ state proof (optional)

Lifecycle of a transaction

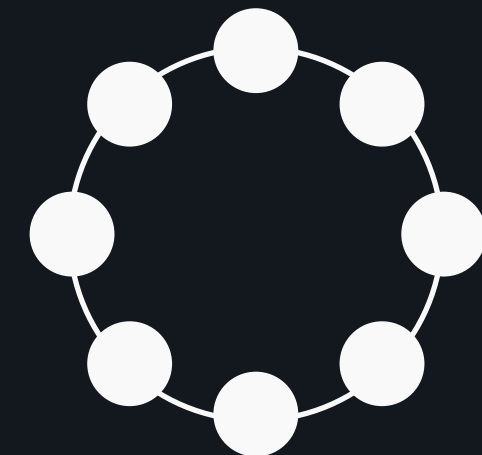


Hedera Hashgraph

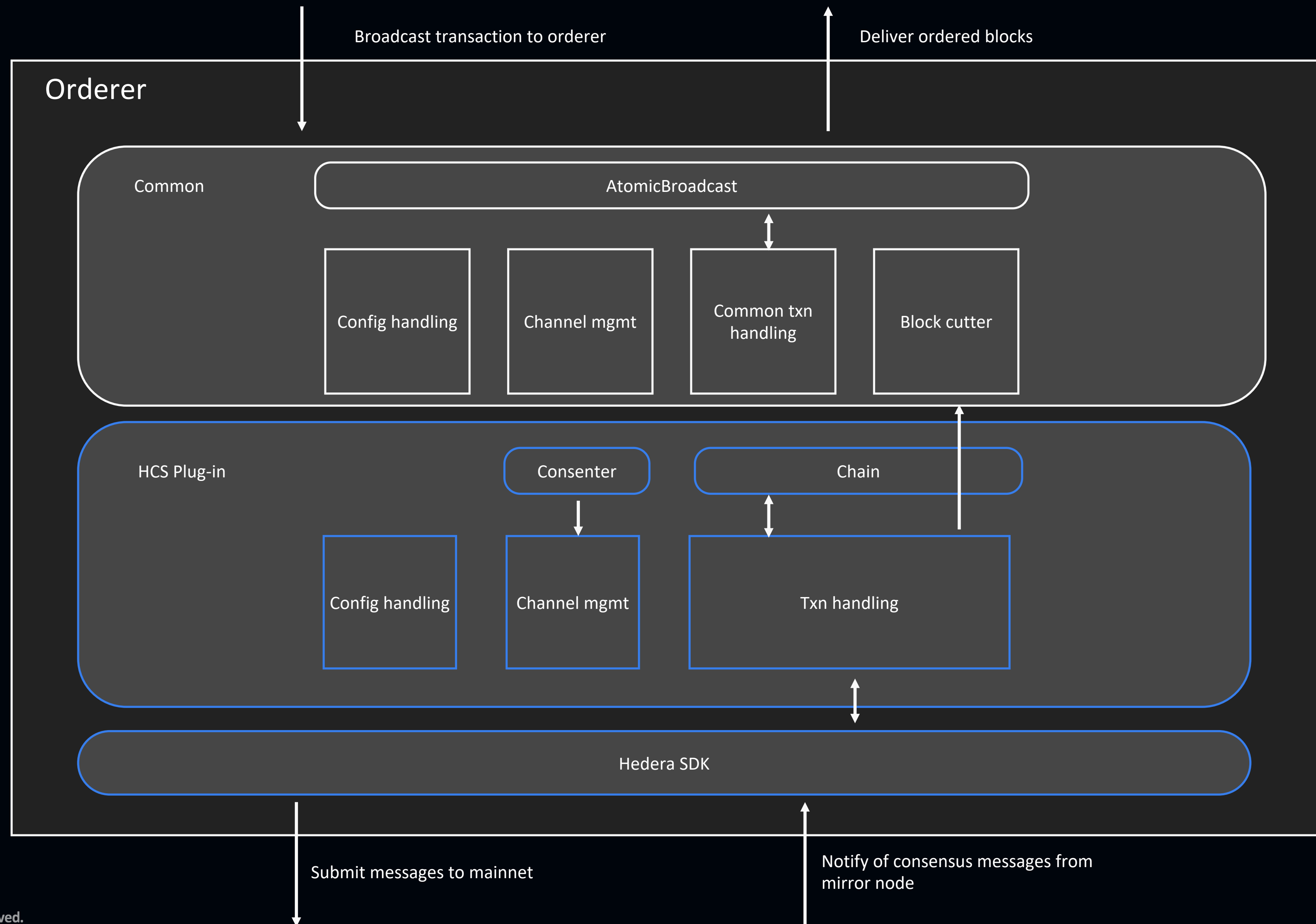
Consensus nodes



Mirror nodes



HCS-based Orderer



Configuration changes



Channel related common configuration

- configtx.yaml
- configtxgen reads in configtx.yaml and generates genesis block / channel config update request txn for orderers to consume
- example config items
 - ConsensusType – decides which plug-in to use for a channel
 - list of Kafka broker addresses for Kafka plug-in

Per orderer configuration

- orderer.yaml
- the orderer executable reads orderer.yaml during bootstrapping
- example config items:
 - tls cert / private key
 - retry config for Kafka plug-in

Configuration changes - configtx.yaml



- New ConsensusType - hcs
- Channel id <-> HCS topic id

Configuration changes - orderer.yaml



- Mainnet node id / node addresses
- Associated account / key for message submission to topics
- Mirror node address



To get started visit:
docs.hedera.com/hyperledger-fabric-hcs



[.com/hashgraph](https://github.com/hashgraph)