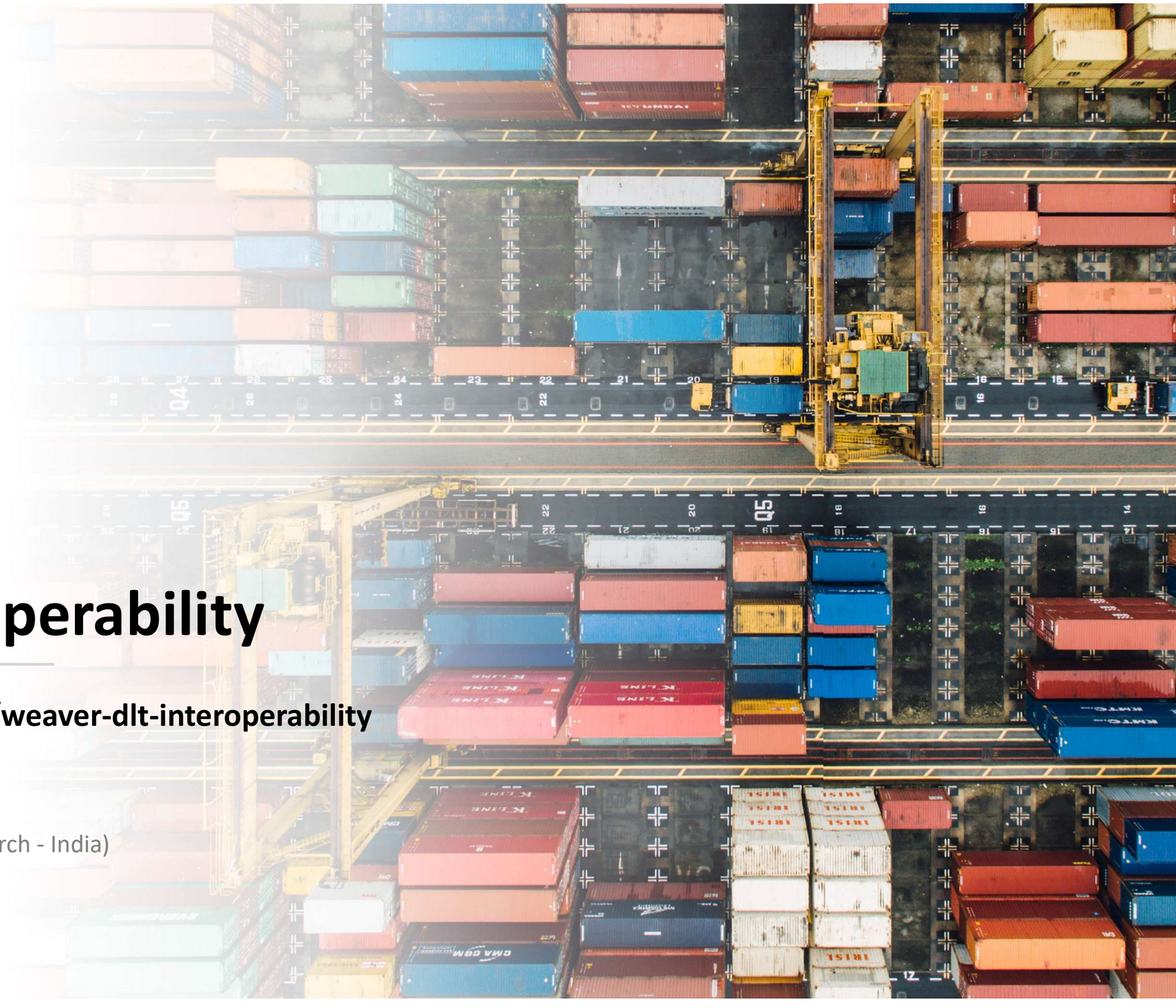


Weaver: DLT Interoperability

https://github.com/hyperledger-labs/weaver-dlt-interoperability

Presenter: Venkatraman Ramakrishna (IBM Research - India)

Keppel Terminal
Photo by @chutternsnap from Unsplash

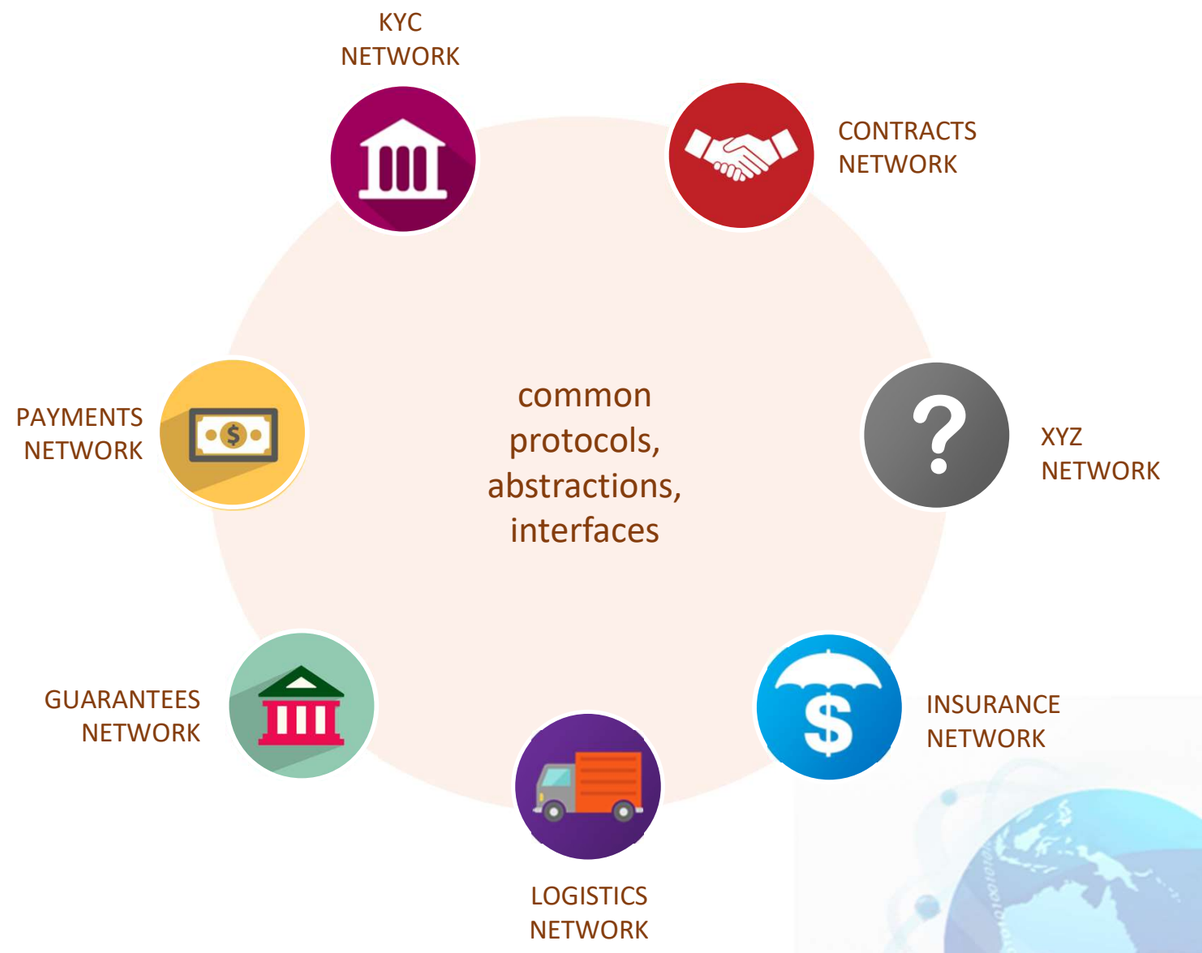
Outline

- Vision and Scope of the Weaver Project
- Description: Architecture and Protocols
- Standardization Opportunities and Discussion

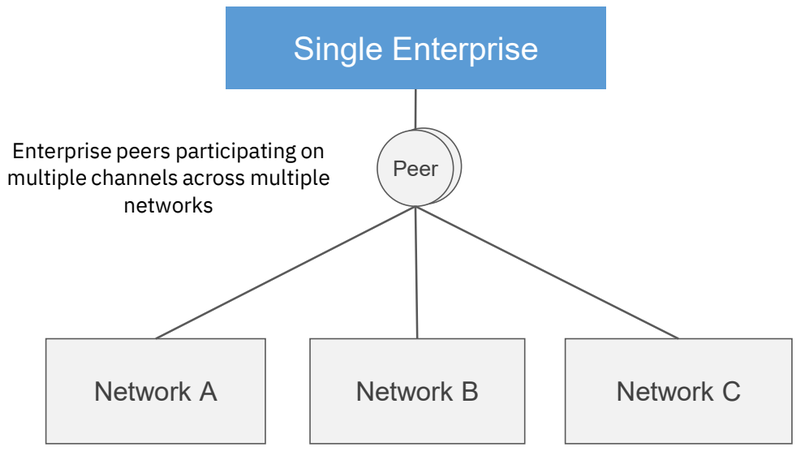
Vision and Scope

Objectives

- Enabling the seamless flow of data and value across disparate blockchain networks in a manner that preserves their trust and security tenets.
- Remove network data and value silos
- Increase market sizes, liquidity and overall efficiency
- Improve network effects
- Enable orchestration of complex business functionality across networks
- Enable scale and growth of networks
- Encourage further adoption of the technology

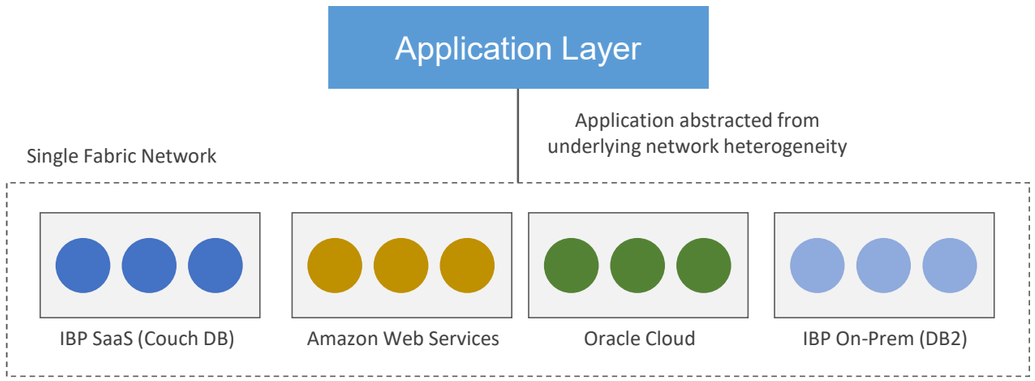


Network Integration Patterns



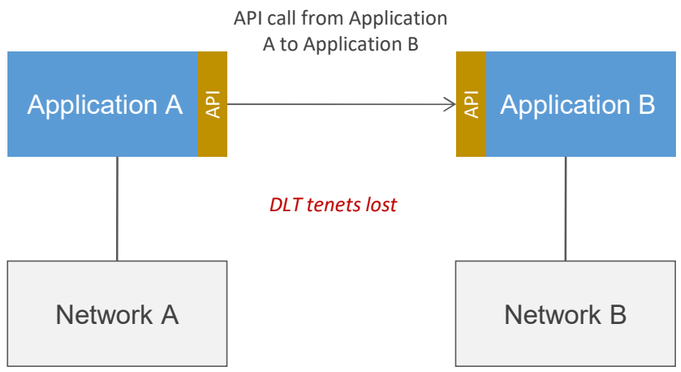
Pattern 1: Single enterprise participating in multiple networks

(Possible today)



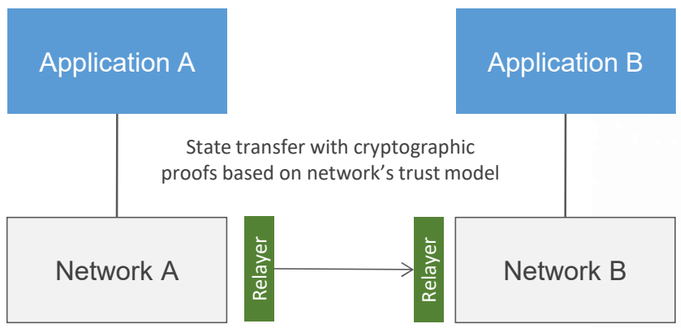
Pattern 2: Single network deploy on multiple heterogenous infrastructure

(Possible today)



Pattern 3: Standard API integration between applications

(Possible today, however loses blockchain tenets and not suitable for value exchange)

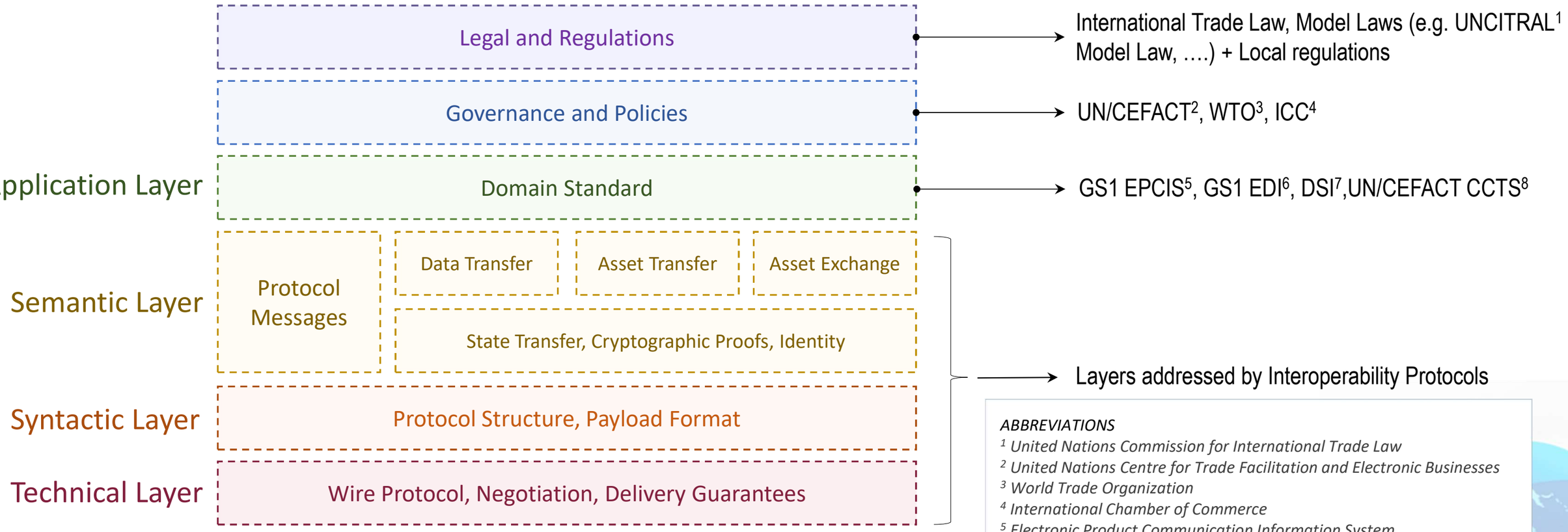


Pattern 4: Data and value exchange with DLT tenets preserved

(Work in progress, provides strongest assurance on validity of state across DLTs)

Layers of Interoperability

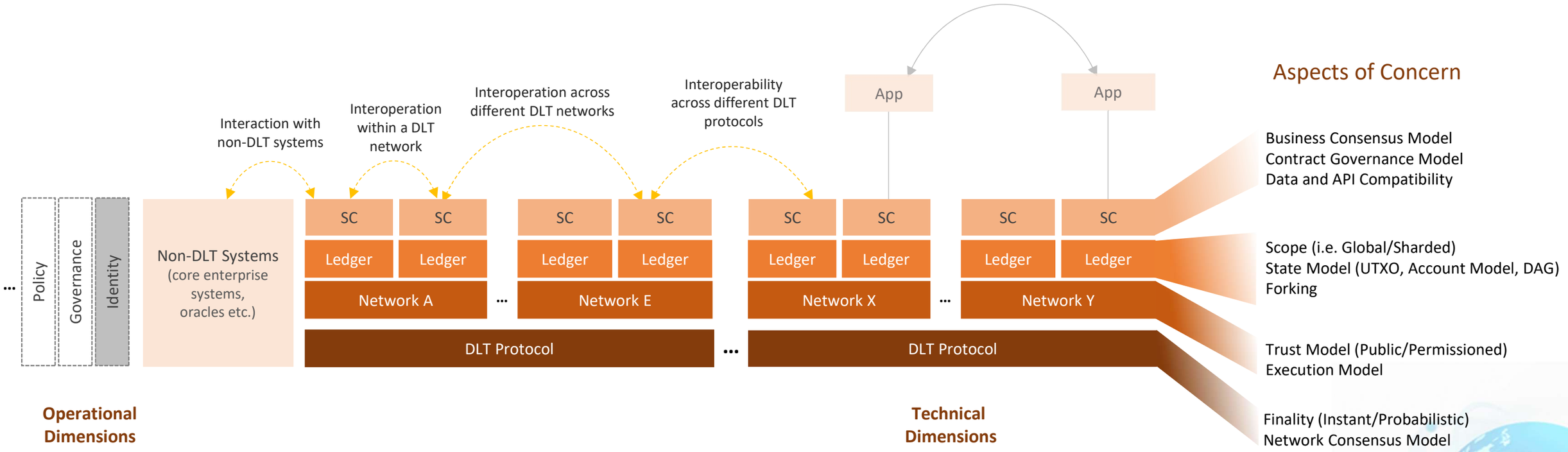
not only a technical matter



ABBREVIATIONS

[1] United Nations Commission for International Trade Law
[2] United Nations Centre for Trade Facilitation and Electronic Businesses
[3] World Trade Organization
[4] International Chamber of Commerce
[5] Electronic Product Communication Information System
[6] Electronic Document Interchange
[7] Digital Standard Initiative
[6] Core Components Technical Specification

Spectrum of Blockchain Interoperability

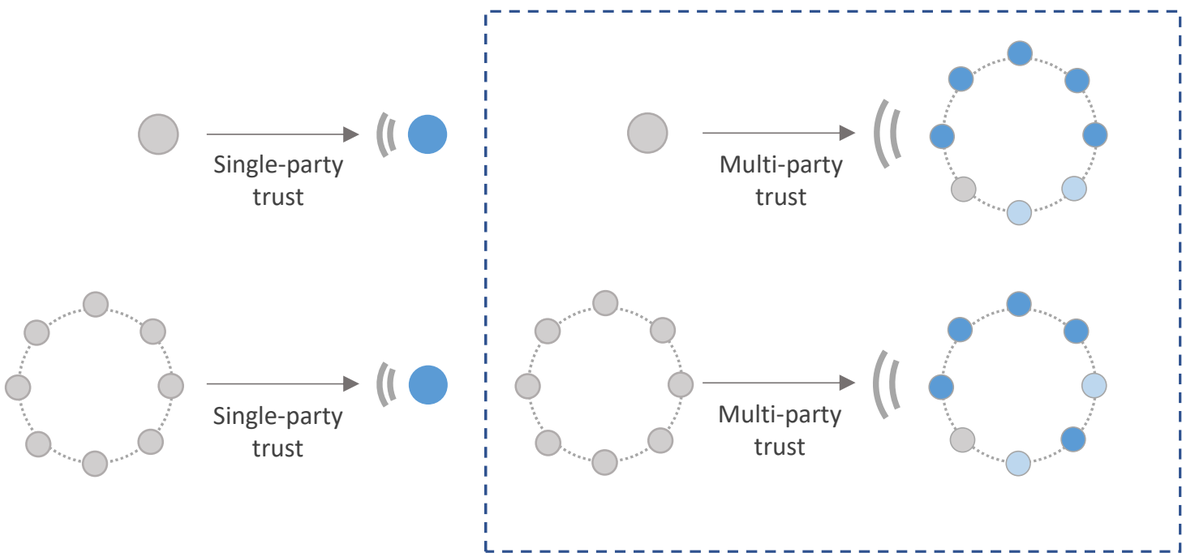


SC -> Smart Contract

What is DLT Interoperability? *(the unique challenges)*

Single-party vs Multi-party Trust

- In distributed ledger architectures, the authority over state lies in a collective and the protocol they employ to ensure its integrity.
- When one network or an entity consumes state from another, it would need to establish the veracity of the state according to the shared consensus view of parties in the network



Proofs and Verifications

- Establishing the veracity of state in a decentralized network is not trivial. In most cases, a consumer of state might not be able to observe the full ledger of the network itself.
- Hence, a consumer needs to obtain an independently verifiable cryptographic proof on the validity of state according to the source network.

Data vs Asset

- Interoperation should not compromise the invariants of networks and protections against double spends on assets

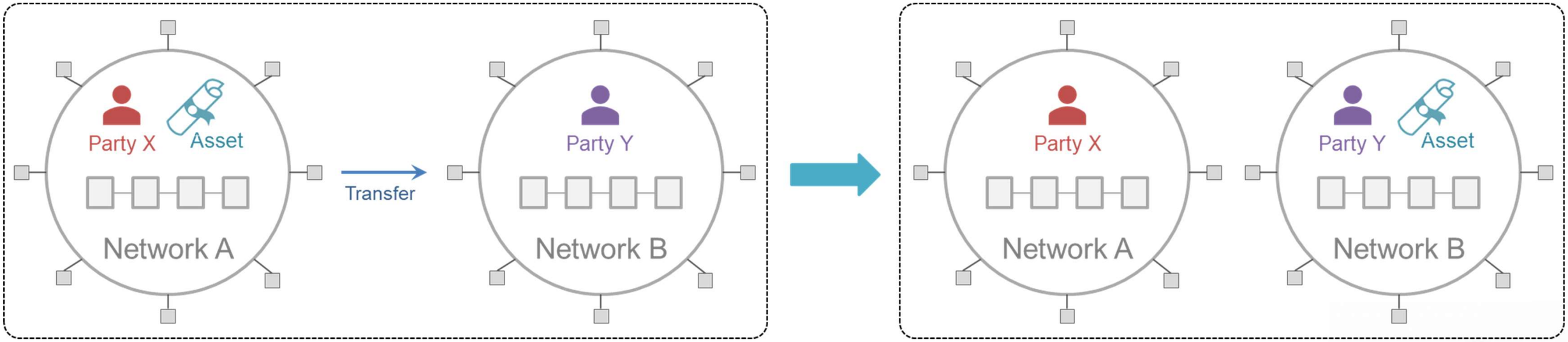
Modes of Interoperation

- Data Transfer: The transfer of data from a source ledger to a consuming ledger. The data transfer can either be a result of a transaction in the source network, or an explicit request from a consuming network.

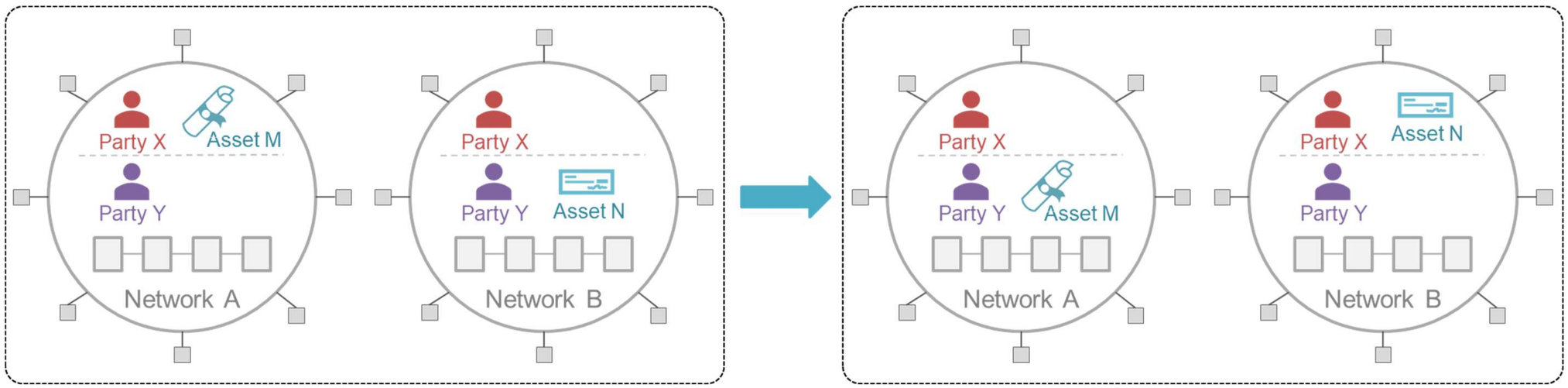
- Asset Exchange: The change of ownership of an asset in a source network and a corresponding change of ownership in another network. No actual value leaves the networks boundaries. Example: atomic cross-chain swap.

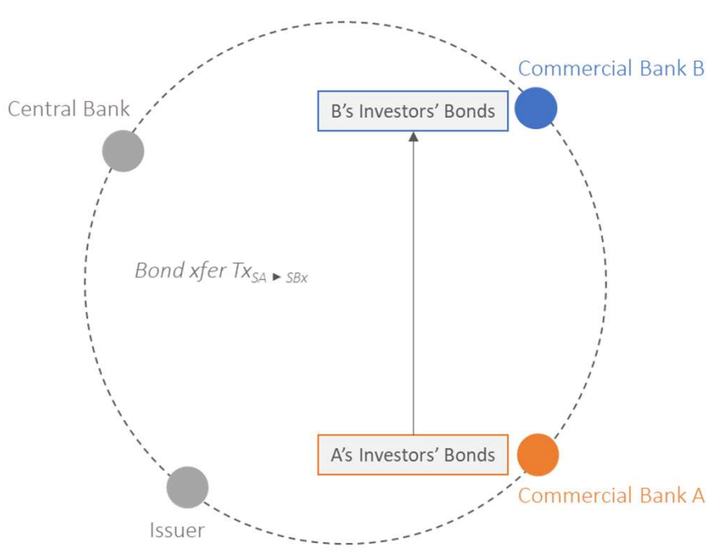
- Asset Transfer: The movement of an asset from the source ledger to a consuming ledger. As an asset cannot be double spent, transfer of an asset should result the termination/locking of its use in the source ledger, and its creation into the target ledger. Mechanisms that support such capabilities are one-way and two-way pegs

Asset Transfers

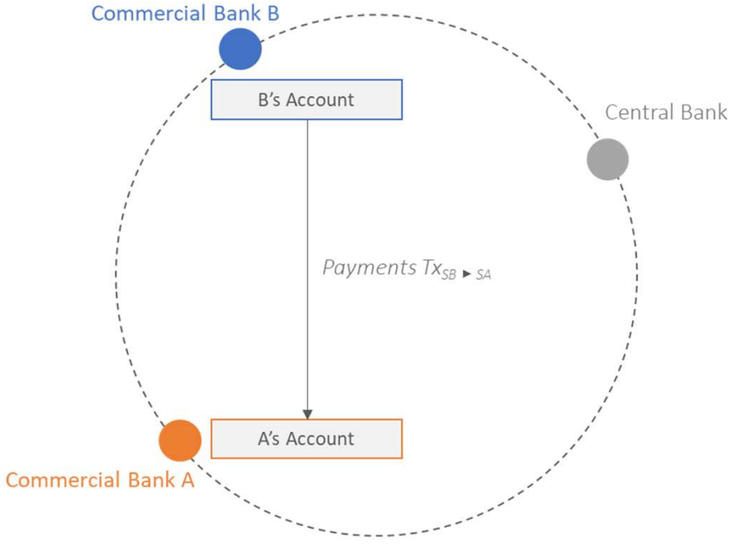


Asset Exchanges



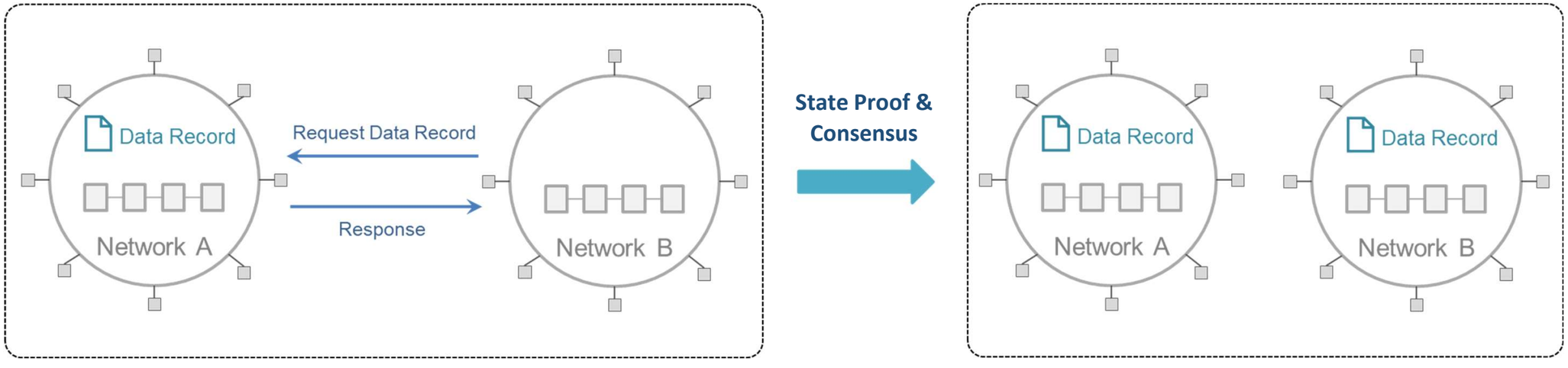


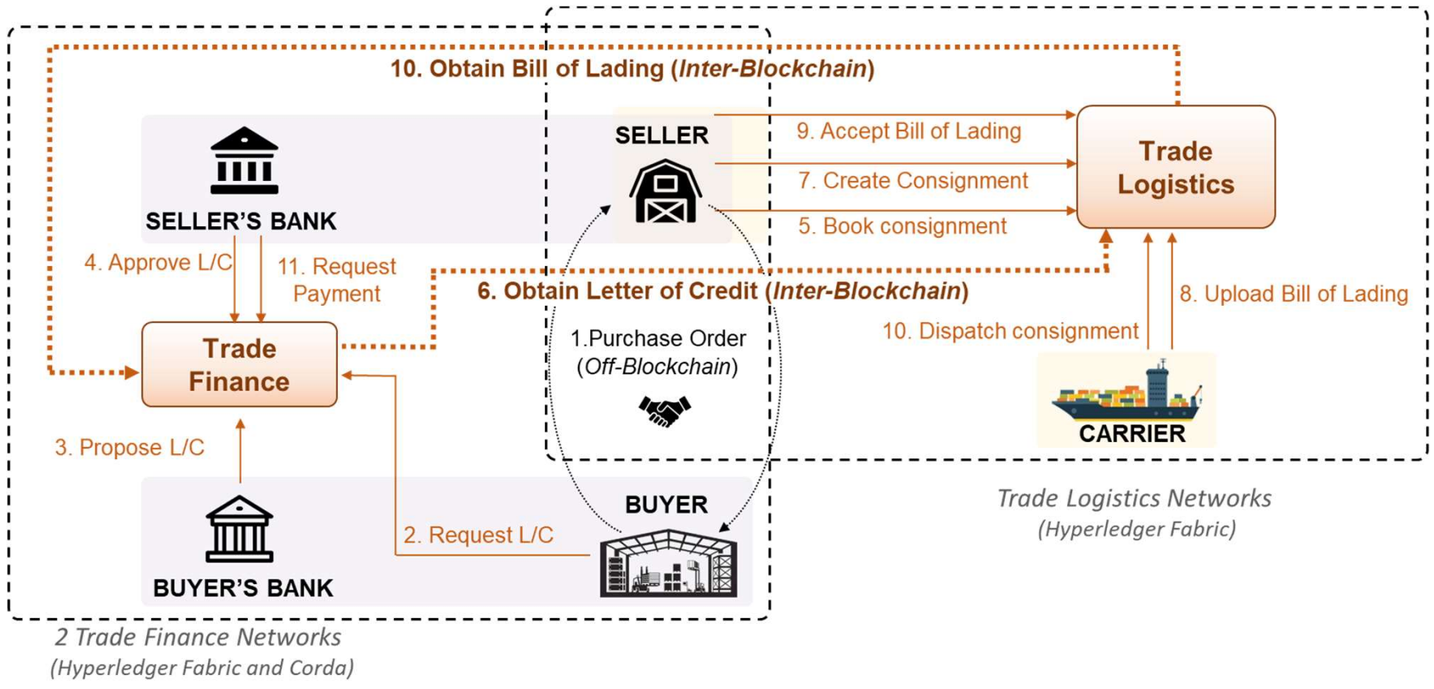
Asset Network for Bonds (ANB)



Central Bank Digital Currency Network (CBDCN)

Data Transfers





Weaver Architecture and Protocols

Weaver

fully decentralized based on trustless intermediaries

Inclusiveness
Avoid approaches that are specific to a particular DLT implementation and design.

Independence
Interoperable networks retain sovereignty on their own processes and access control rules.

Minimum Trust
Reduce trust to only what is essentials (i.e. identity providers in the network).

Privacy by Design
Interaction between parties across networks should be kept private and confidential and revealed only to the interested parties.

No Intermediaries
No third-party intermediary should be relied upon for the purpose of cross network data verification or settlement.

We believe that these constitute a minimal set of requirements that will facilitate adoption and applicability of interoperability in various contexts.

Design Principles

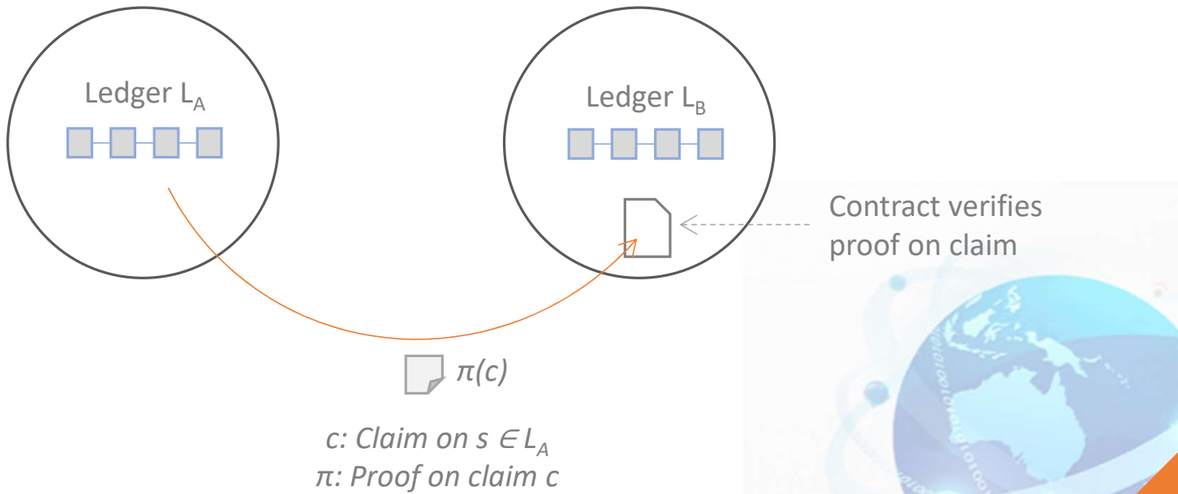
Weaver

interoperability protocol components

Proofs of State Validity

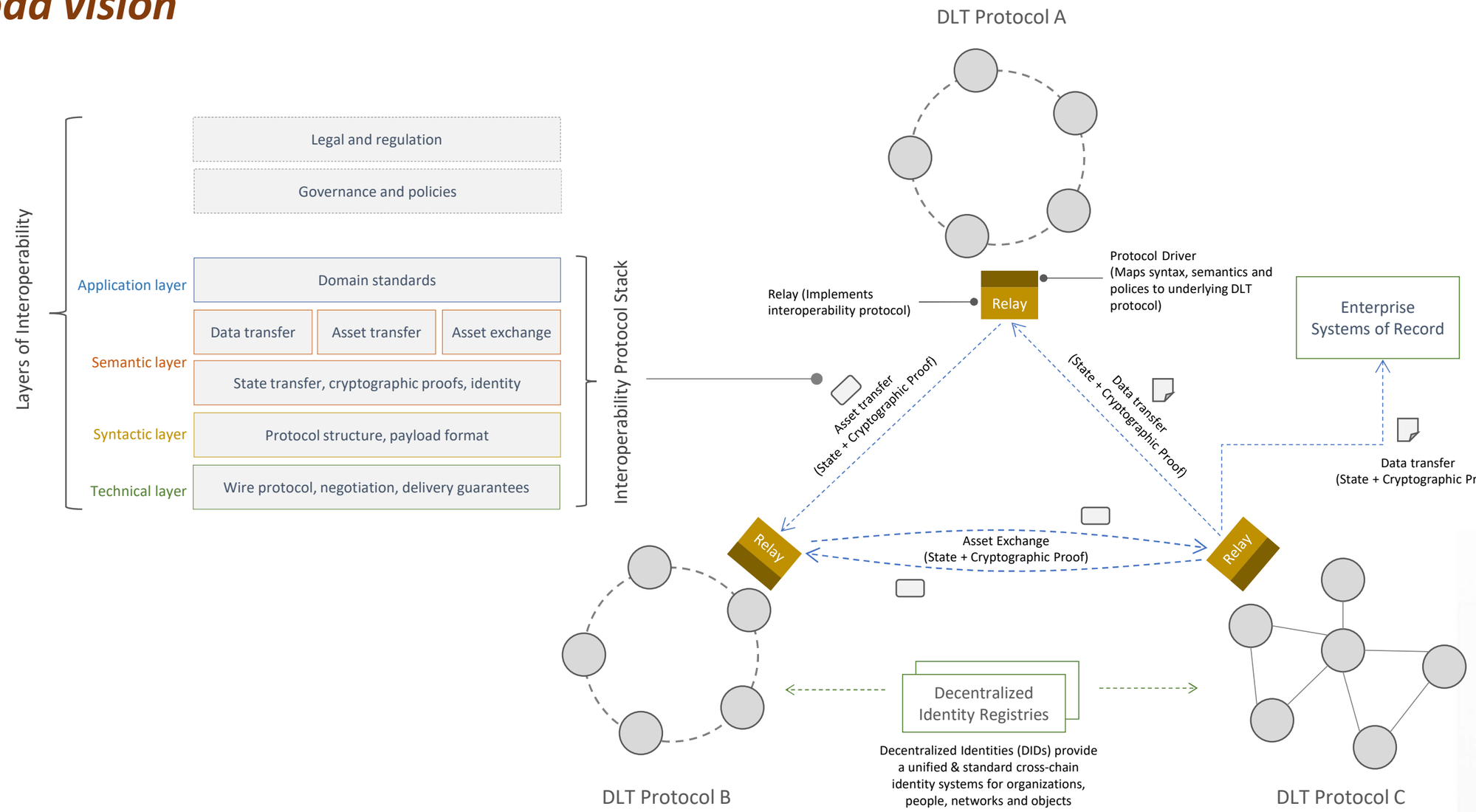
- The communication of state between networks can be generalized to take the form of a claim c and an associated proof π(c).
- The construction of the proof (and verification) is derived from the protocol of the network being interacted with assumptions made on the validity of the ledger.
- In permissioned networks, finality is achieved when all authoritative signatures are received for a transaction and/or block recorded on the ledger.
- For networks based on Nakamoto consensus, finality is subjective.
    - Low value transactions require few blocks of confirmation (e.g. 6 blocks)
    - High value transactions require larger number of of confirmation (e.g. 60 blocks)

- Standardized relay
    - Component and standardized protocol for querying networks, obtaining proofs, registering for events
    - Minimize trust assumptions on relay*



Weaver

broad vision



Weaver

components

Communication

Mode and Structure

Definition of the technology for exchanging messages and the operation modality of the protocol.

Current design: *message oriented, asynchronous, Protobuf.*

Messages

Definition of syntax and semantic of messages, addressability of entities across networks.

Current capabiities: *ability identify entity via ledger views, return queries results accompanied by verifiable proofs.*

Services

Definition of services exposing the interoperability protocol functions, and mechanisms for their discoverability.

Current capabiities: *implementation of network agnostic gRPC services called relays, currently statically configured to talk to each other.*

Platform(s) Integration

Smart Contracts

Definition and implementation of smart contracts (SCs) that provide services to the interoperability protocol. SC(s) are independent on the business purpose of the networks.

Current capabilities: *support for on-chain validation of foreign-network state, registration of crypto material, and production of signed and verifiable views for network consumption (Hyperledger Fabric and Corda).*

SDK Adaptations

Modification (if needed) of existing SDK with interoperability features.

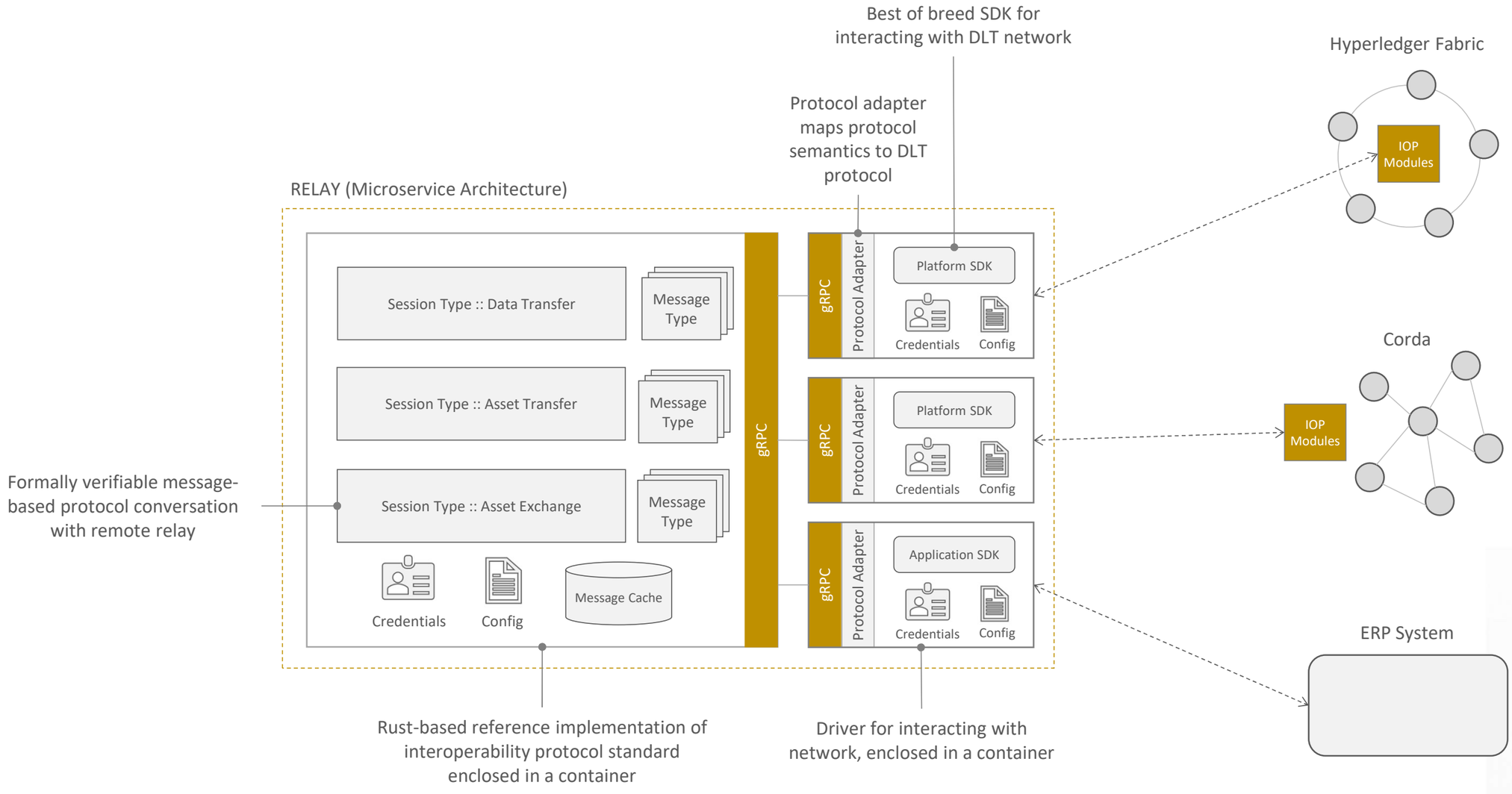
Current design: *minor to no changes on the SDKs of the supported platforms.*

Network Drivers

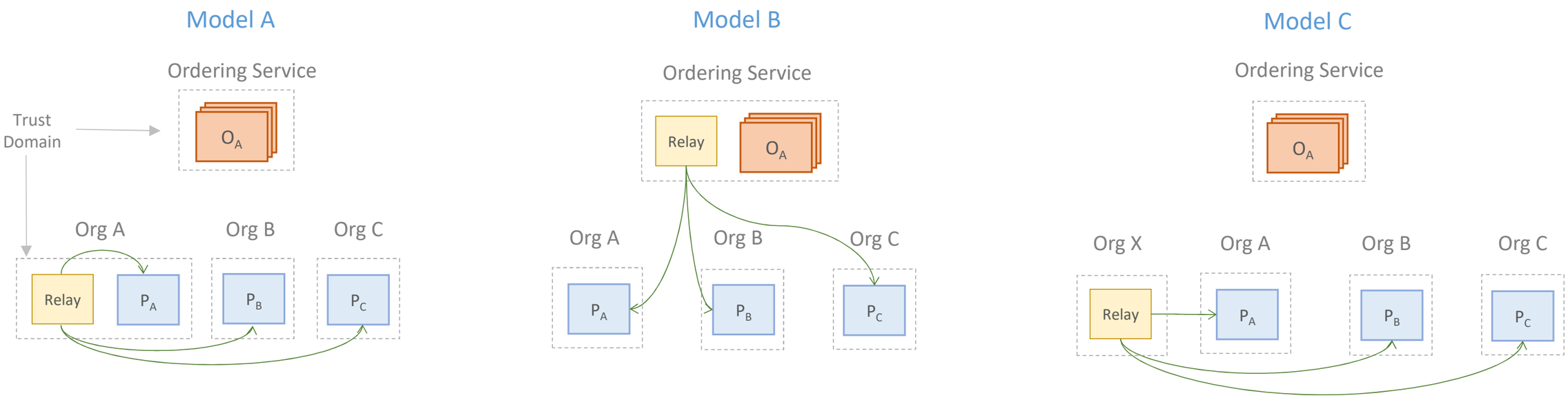
Definition of network-specific pluggable components in the relay service to customize its behavior for a specific blockchain platform.

Current status: *support for Fabric and Corda.*

Relay Architecture



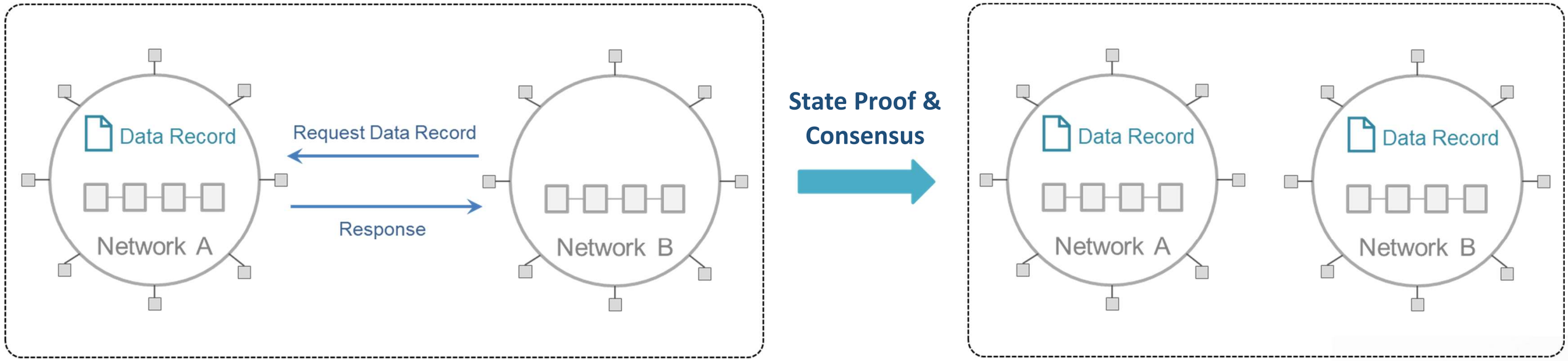
Relay Deployment Models (Fabric)



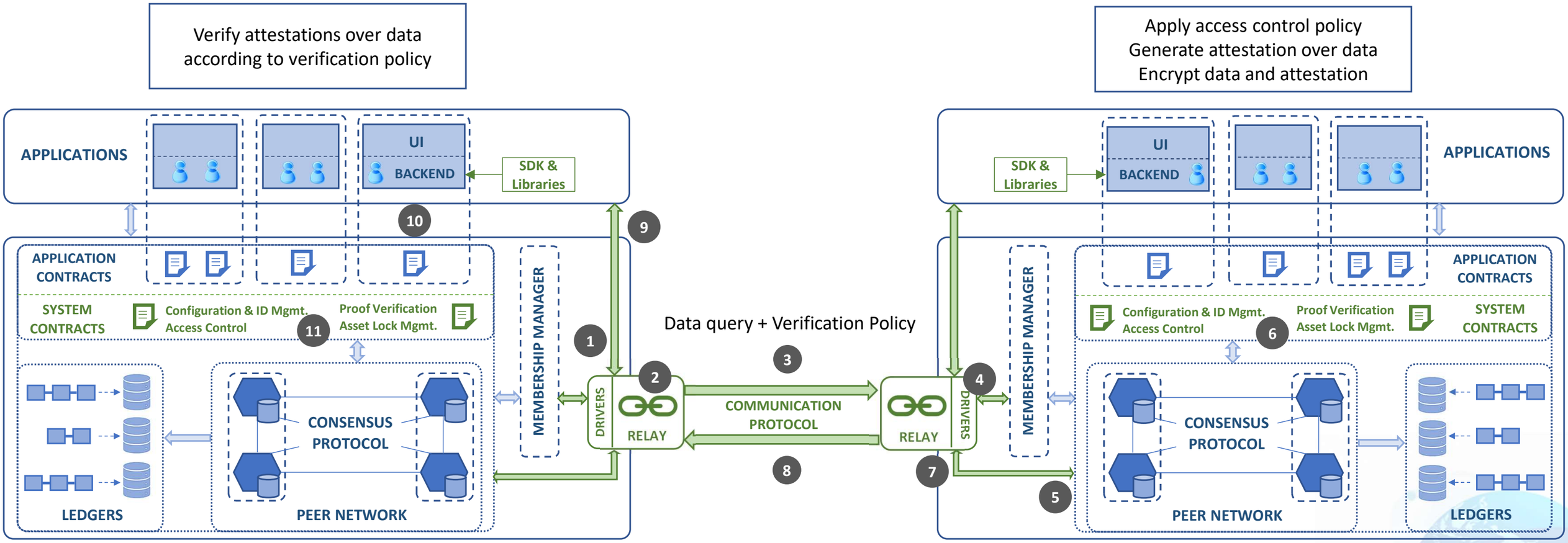
Deployment Models

- The relay is designed to support multiple deployment models based on network requirements and trust assumptions.
- The relay uses a valid network identity in order to service requests on behalf of remote networks or enterprise systems.
- Multiple relays can be deployed for high-availability and confidentiality (e.g. relay per network, per channel, subset of orgs in a channel, per org).
- The scope of data visibility can be controlled by access control contracts (applies to Model B and C).

Data Transfer: General Use Case

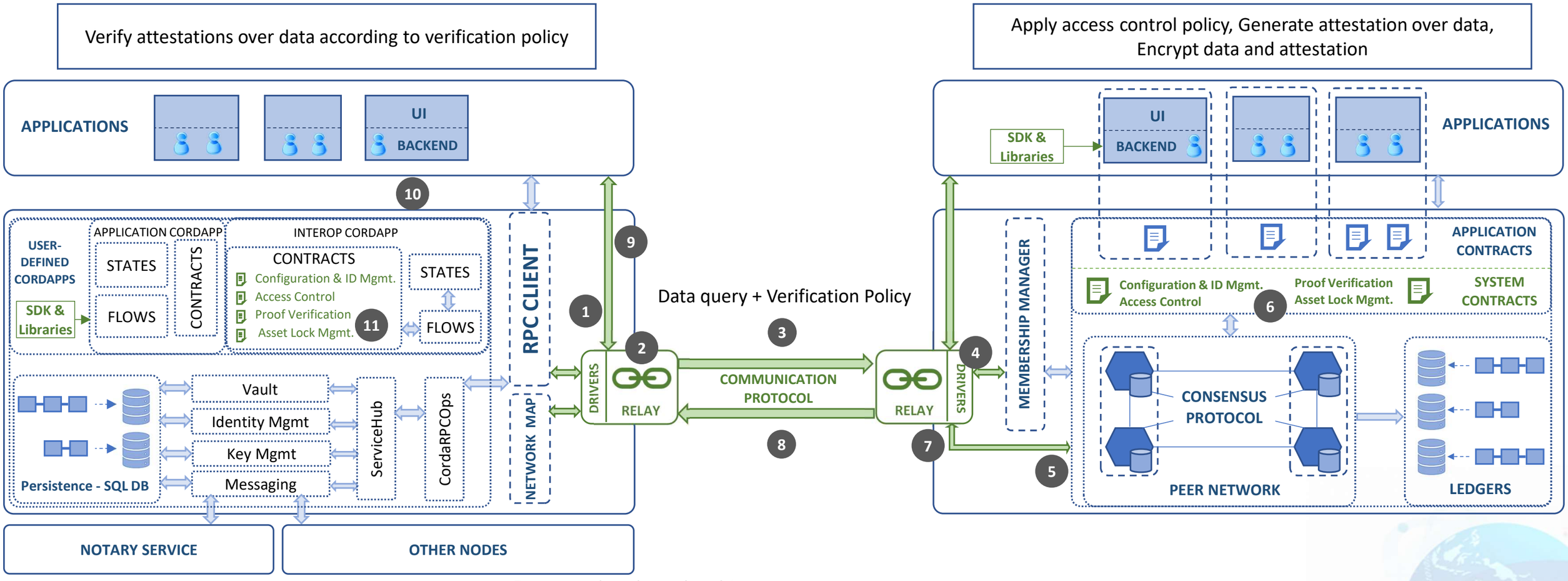


Cross-Network Data Transfer: Fabric-Fabric



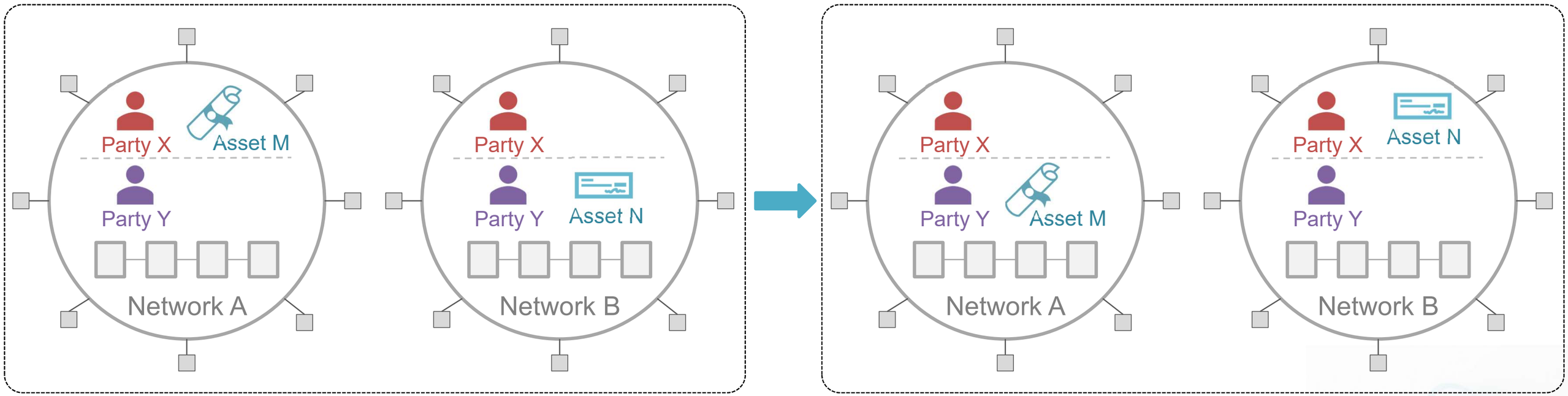
1. Application prepares a relay query to fetch remote data to supply as argument to a local chaincode invocation
2. Local relay determines how to route the request
3. Local relay sends the request to the remote network's relay
4. Remote relay, through a driver, parses query and verification policy and determines how to collect data and proof
5. Driver sends query to selected peers
6. Remote network peers perform access control checks before executing query and signing the result
7. Driver collects attested results and packages a response
8. Remote relay sends the response to the local relay
9. Local relay queues response to the application
10. Application submits data and proof within a chaincode invocation
11. Peers verify proof before processing the data

Cross-Network Data Transfer: Corda-Fabric

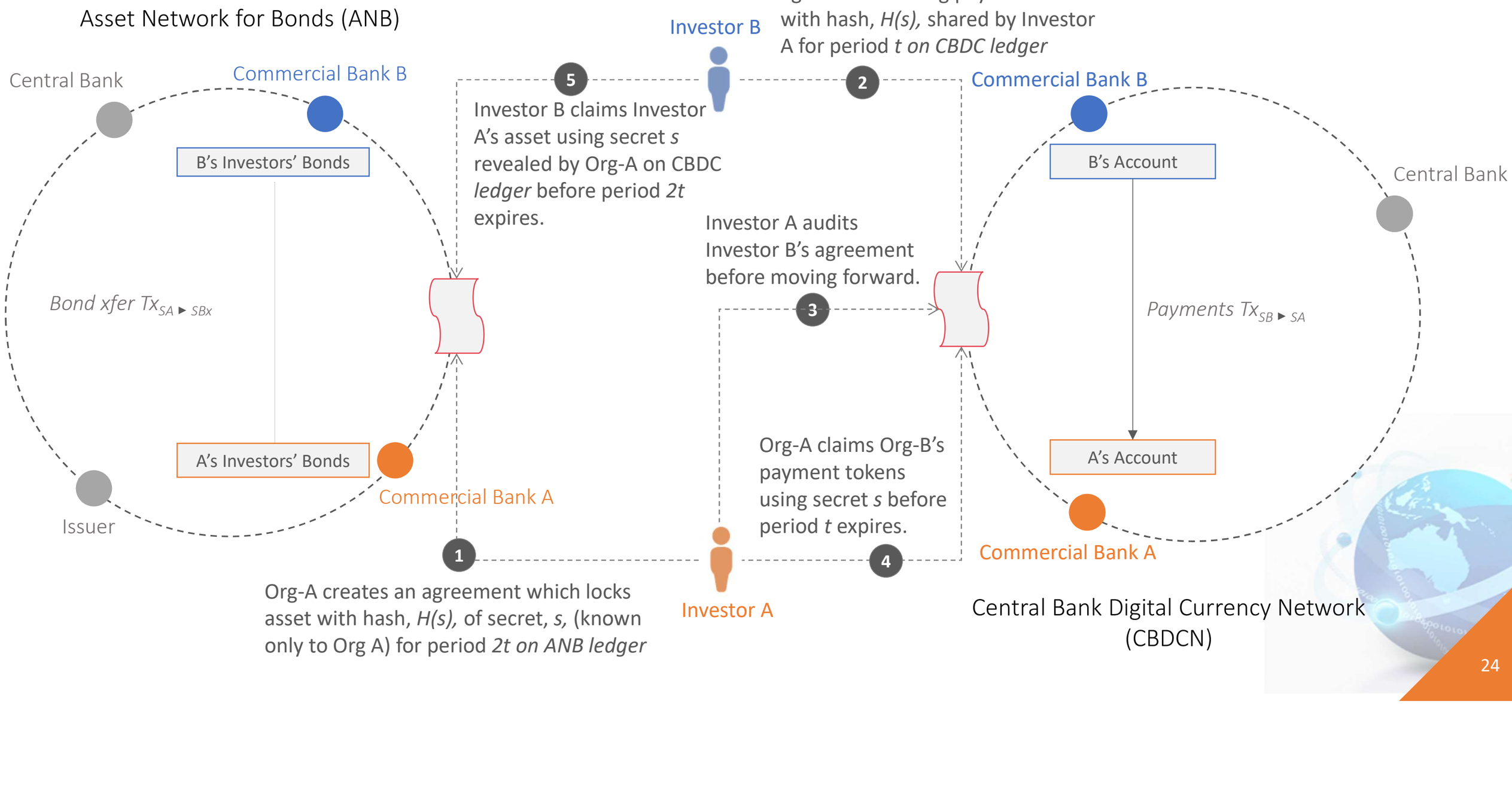


1. Application prepares a relay query to fetch remote data to supply as argument to a local chaincode invocation
2. Local relay determines how to route the request
3. Local relay sends the request to the remote network's relay
4. Remote relay, through a driver, parses query and verification policy and determines how to collect data and proof
5. Driver sends query to selected peers
6. Remote network peers perform access control checks before executing query and signing the result
7. Driver collects attested results and packages a response
8. Remote relay sends the response to the local relay
9. Local relay queues response to the application
10. Application submits data and proof within a chaincode invocation
11. Peers verify proof before processing the data

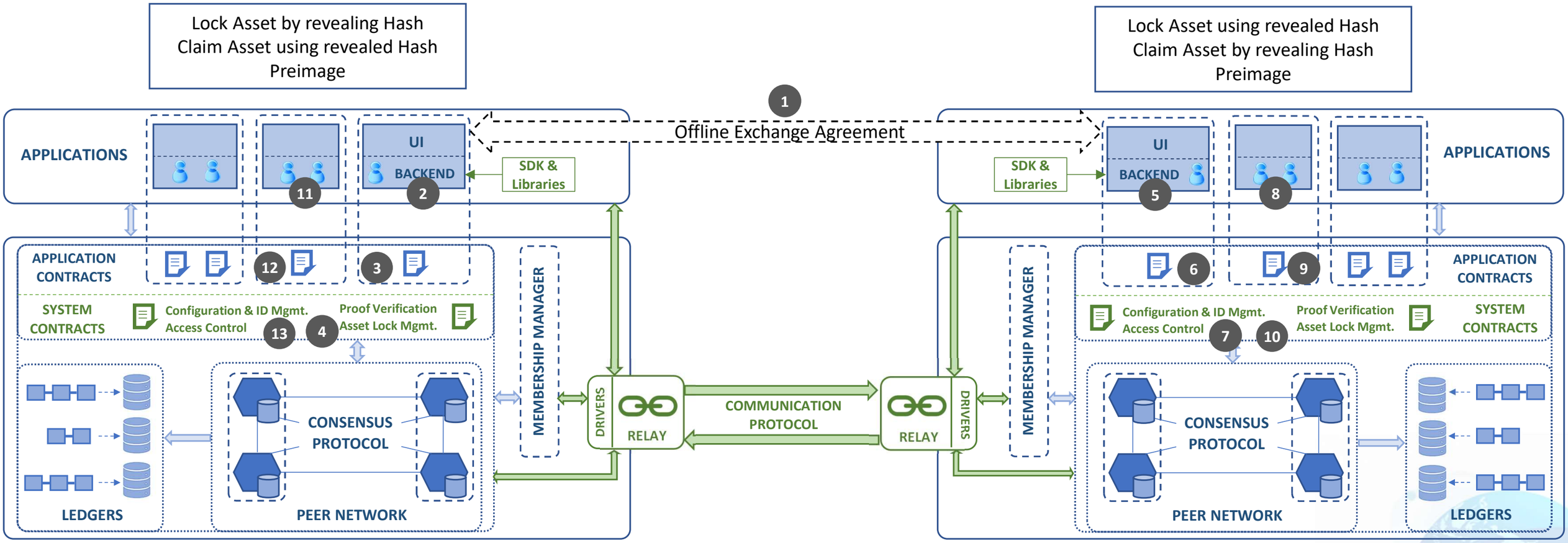
Asset Exchange: General Use Case



Example: Bond-CBDC Exchange Workflow



Cross-Network Asset Exchange: Fabric-Fabric



1. Alice and Bob, members of both networks agree on assets to be exchanged
2. Alice submits a request to lock an asset for Bob with hash value using SDK API
3. Adapted chaincode invokes asset lock mgmt contract via library
4. Interoperation contract locks Alice's asset for Bob
5. Bob submits a request to lock an asset for Alice with same hash value using SDK API
6. Adapted chaincode invokes interoperation lock contract via library
7. Interoperation contract locks Bob's asset for Alice
8. Alice submits claim for Bob's asset with hash preimage using SDK API
9. Adapted chaincode invokes asset lock mgmt contract via library
10. Interoperation contract unlocks Bob's asset for Alice, which is then claimed by adapted chaincode
11. Alice submits claim for Bob's asset with hash preimage using SDK API
12. Adapted chaincode invokes asset lock mgmt contract via library
13. Interoperation contract unlocks Bob's asset for Alice, which is then claimed by adapted chaincode

Standardization

Perspectives

- Layered approach: cross-network language for communication and negotiation
  - Asset specifications and semantics
  - Data formats for states and proofs
  - Framing policy and governance rules
  - Cross-network and DLT-agnostic requests and responses
  - Common mechanisms: security (encryption, signatures), hashing, locking, timing

- Modular approach: enabling components in the interoperation pipeline
  - Relay module API
  - DLT-specific drivers
  - Interoperation modules: proof generation and validation, asset locking and claiming
  - SDKs and libraries

Ongoing Efforts

- Active discussions in the Hyperledger community
  - Main Projects: Cactus, (potentially) Ursa, Quilt
  - Labs Projects: Weaver, Firefly, Yui (Cosmos IBC)
- Goals/targets identified:
  - Enabling compatibility among different interoperability frameworks (e.g., Cactus and Weaver and Yui)
  - Building a common relay module or at least making them interoperable (Cactus-Weaver effort ongoing and led by Accenture)
  - Creating a repository of common driver modules that can expose DLT-specific state and proofs in a network-neutral manner
  - (Tentatively) engage with ISO TC307

IETF Draft: An Interoperability Architecture for Blockchain/DLT Gateways

- Blockchain gateways in the Weaver framework are called *relays* (also in Cactus)
  - Presently, they only support data transfers: requests and responses for state views
  - Part of roadmap: support instructions and event pub/sub, enable asset exchanges (or any other atomic operation)
- IETF Draft: present version
  - *"Gateway nodes perform the transfer of virtual assets between blockchain systems while masking the complexity of the interior constructs of the blockchain that they represent."*
  - We have the same vision for the Weaver relay
  - Eventual goal: make our relay compliant with the final draft while using the lessons we have learned to help augment the draft and make it more comprehensive
- Other draft proposals: state views, proofs, policies, API, cross-network contracts?

Questions

Backup

Drivers for Interoperability *(enterprise perspective)*

Technology Evolution

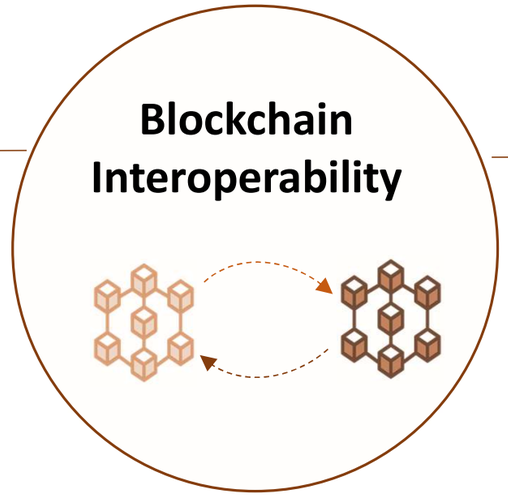
Still early days

Many available implementations

Different abstractions and approaches

Heterogeneity of technical stacks

Specialization vs generalization



Experimentation & Adoption

Gradual adoption via use cases

Consortia define network boundaries

Consortia are specific to use cases

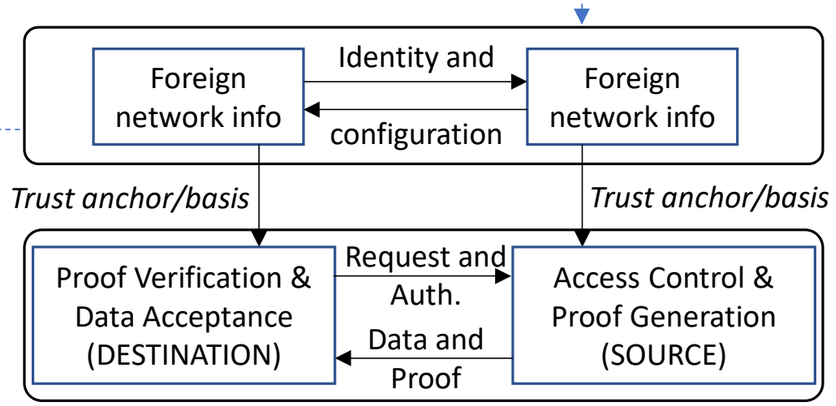
Use cases are often a slice of the business

Complementary but isolated networks

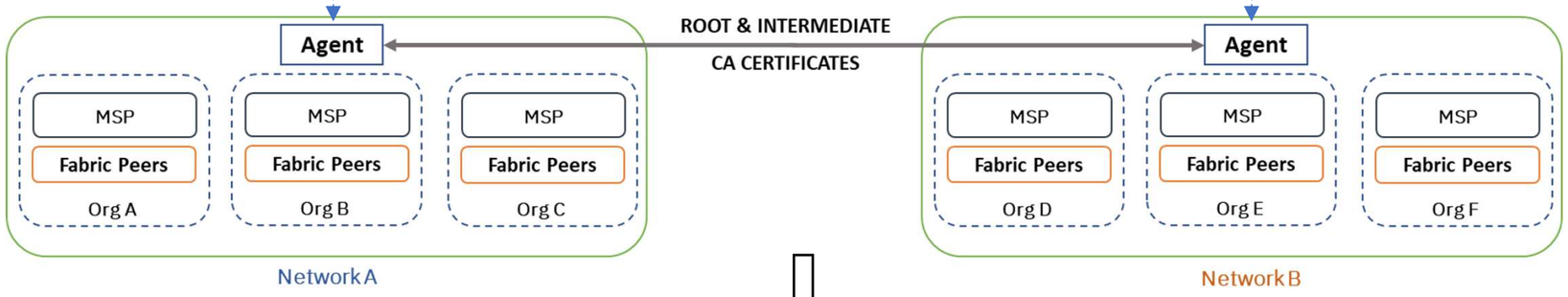
Decentralized Identity Management for Blockchain Interoperation

Goal: Bilateral on-demand transfers at scale

Requires 2-layer protocol with data transfer depending on identity exchange

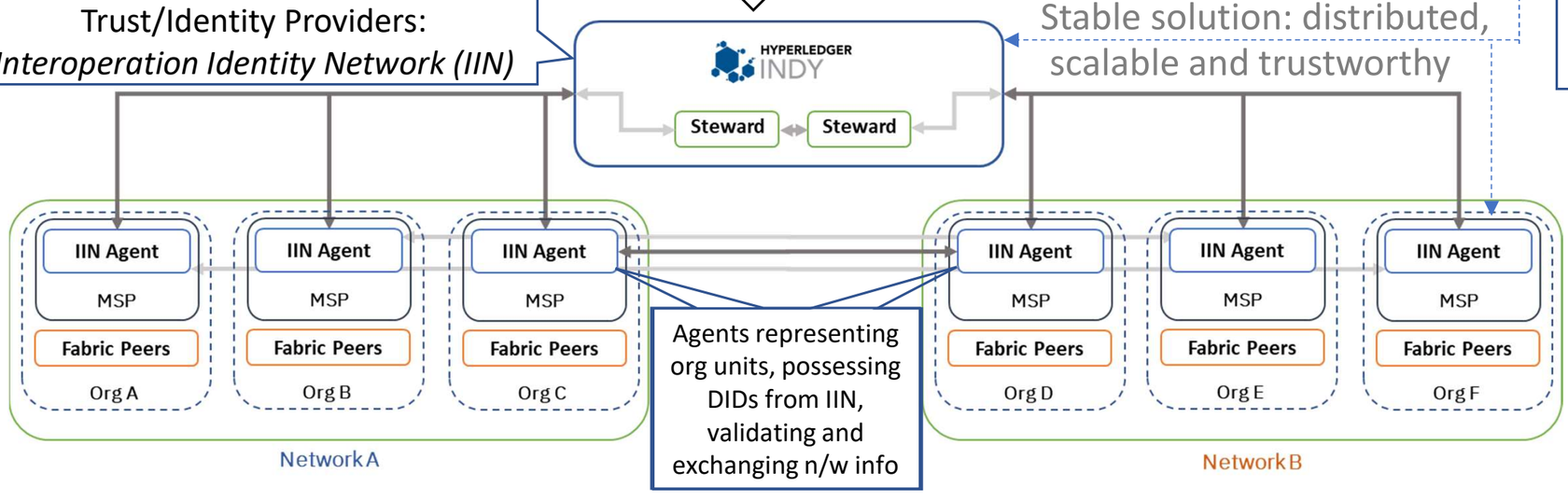


PoC: single agent exposing network certificates
Stopgap solution: insecure and not scalable



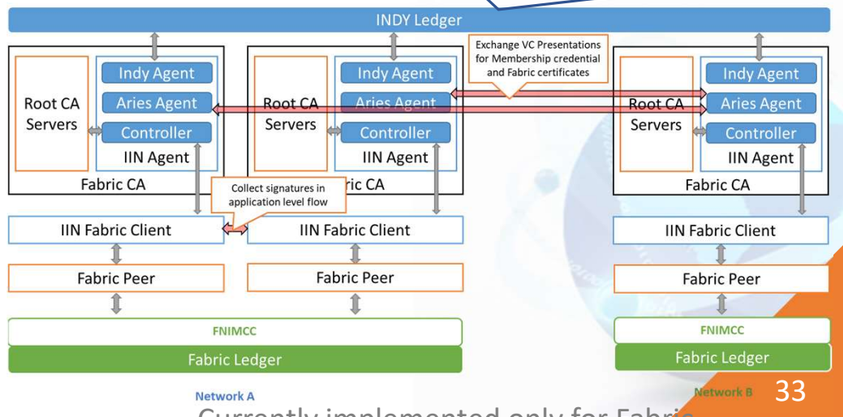
Common Platform with Trust/Identity Providers: *Interoperation Identity Network (IIN)*

Stable solution: distributed, scalable and trustworthy



Agents representing org units, possessing DIDs from IIN, validating and exchanging n/w info

Trust anchoring protocol: IIN Agents independently validate foreign n/w info, record with multisignature on local network ledger

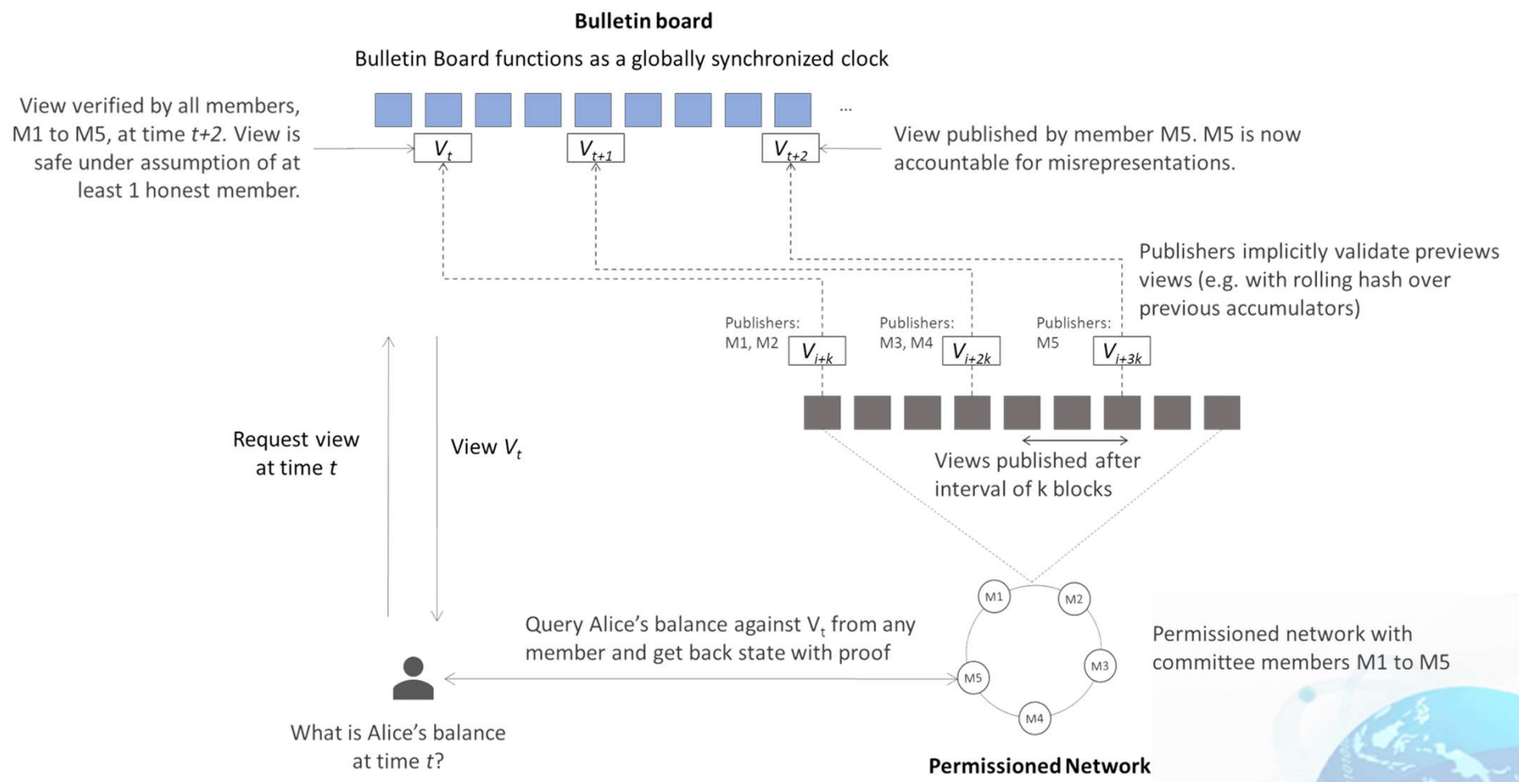


Currently implemented only for Fabric

Advanced Proof Mechanisms

Proof of Currency and Finality of State

- Permissioned chains are confidential by nature and no artifacts of the network are externally observable.
- We propose and demonstrate mechanisms for proving currency of state to remote agents in permissioned networks under stringent adversarial assumptions about the management committee of a network.
- The proofs of state must show evidence of finality on the ledger as well as proof that the state is linked to the published metadata (global snapshot) observable by the remote agent.
- The proof enables reasoning about the currency of a given state with respect to a global clock



Architectural Components and Protocol