Working Draft: Views and View Addresses for Blockchain/DLT Interoperability

V. Ramakrishna, Vinayaka Pandit, Sandeep Nishad, Krishnasuri Narayanam, Dhinakaran Vinayagamurthy (IBM)

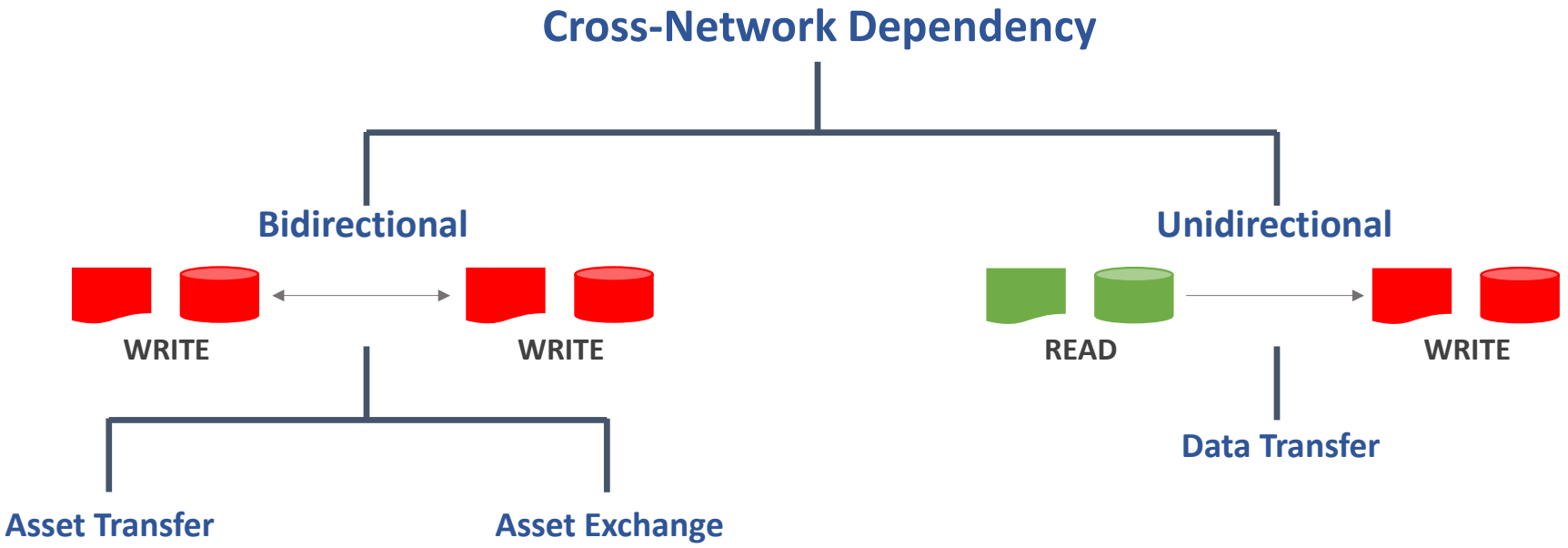
Ermyas Abebe (Consensys)

IETF 112 Side Meeting:
November 12, 2021

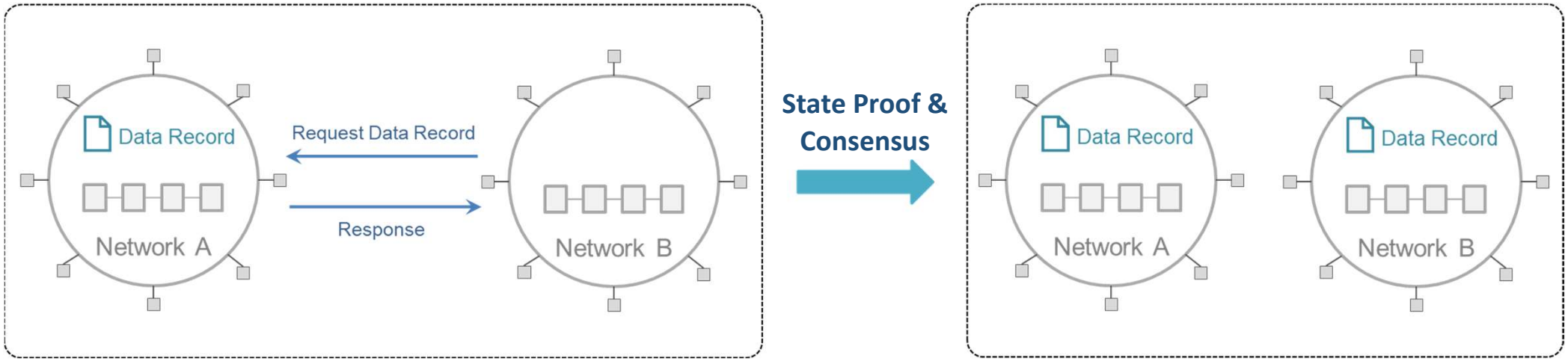
Keppel Terminal
Photo by @chutternsnap from Unsplash

Modes of Interoperation Across DLT Networks

- *Claim*: any cross-network process interdependency can be realized as a combination of data transfers, asset transfers, and asset exchanges

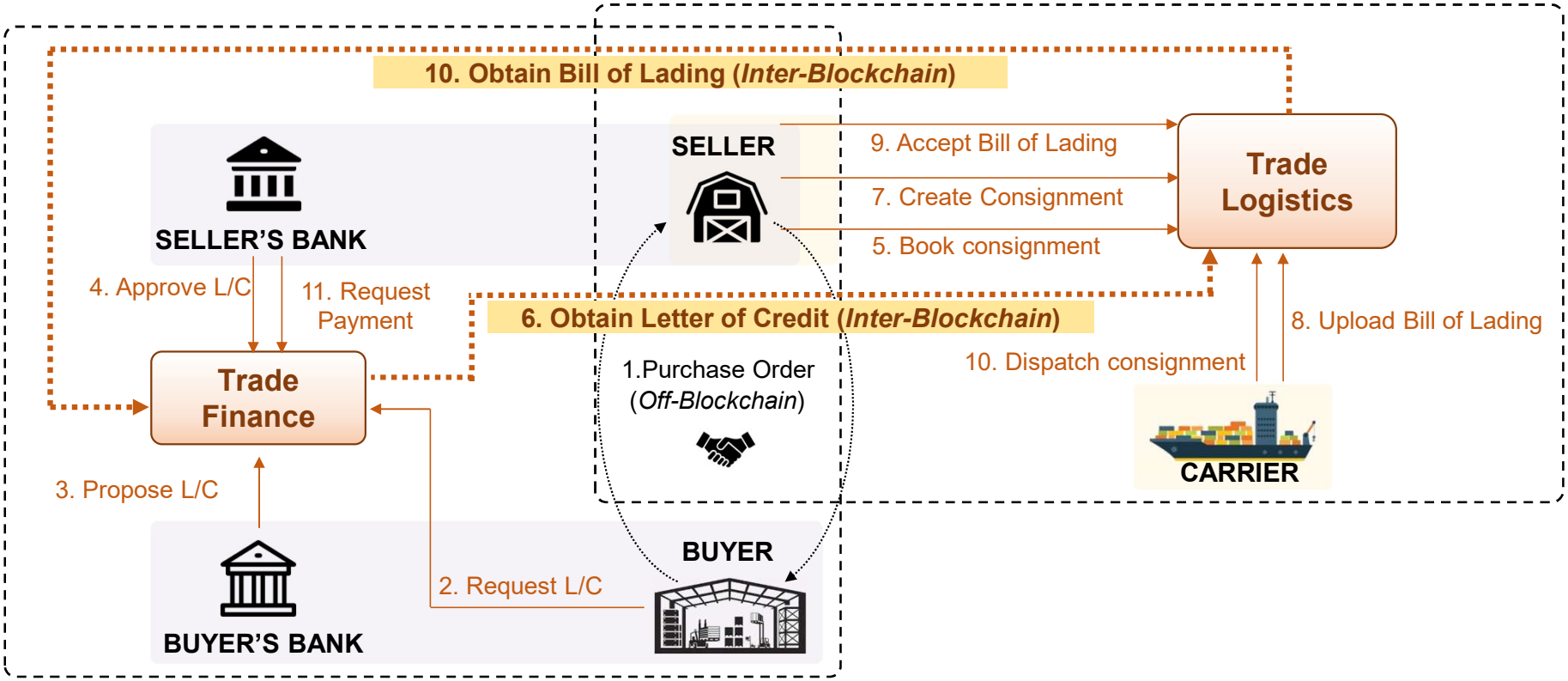


Cross-Network Data Transfer: Model

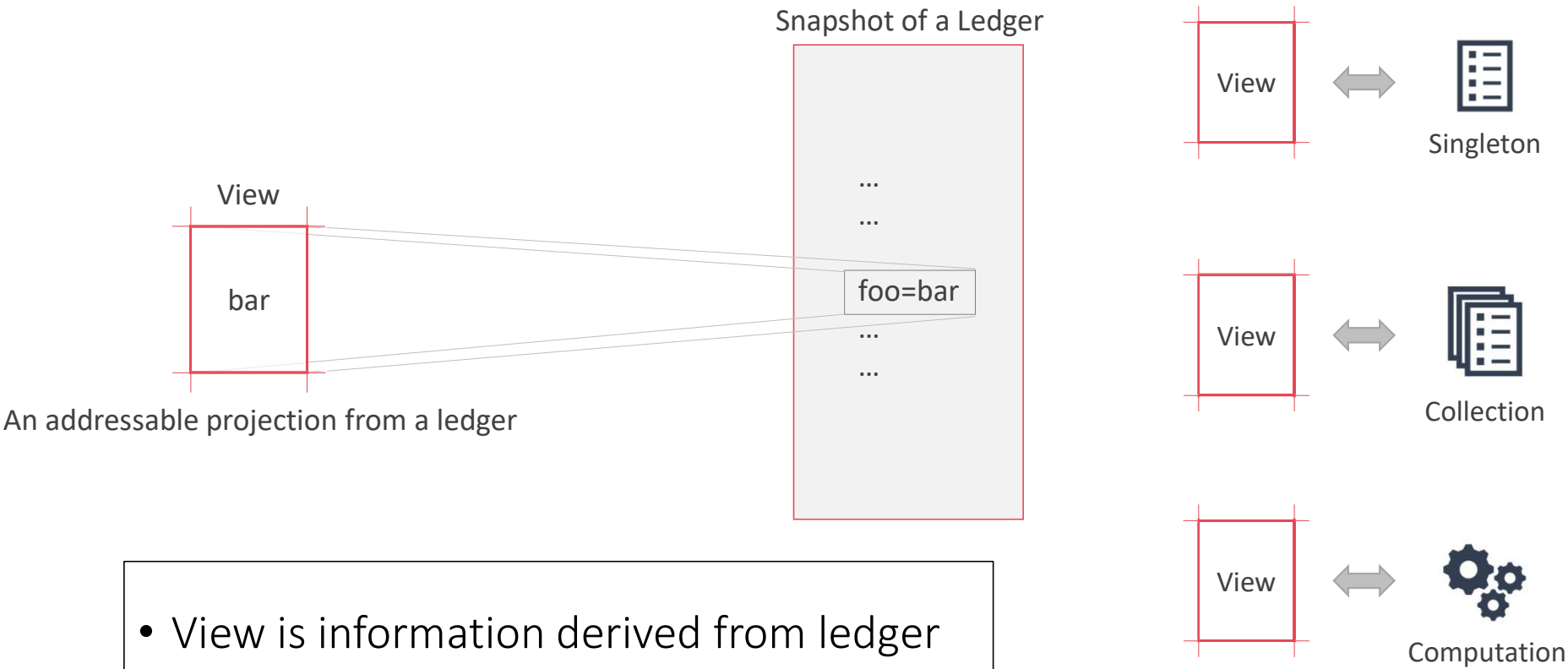


Cross-Network Data Transfer: Use Case

Trade finance and logistics

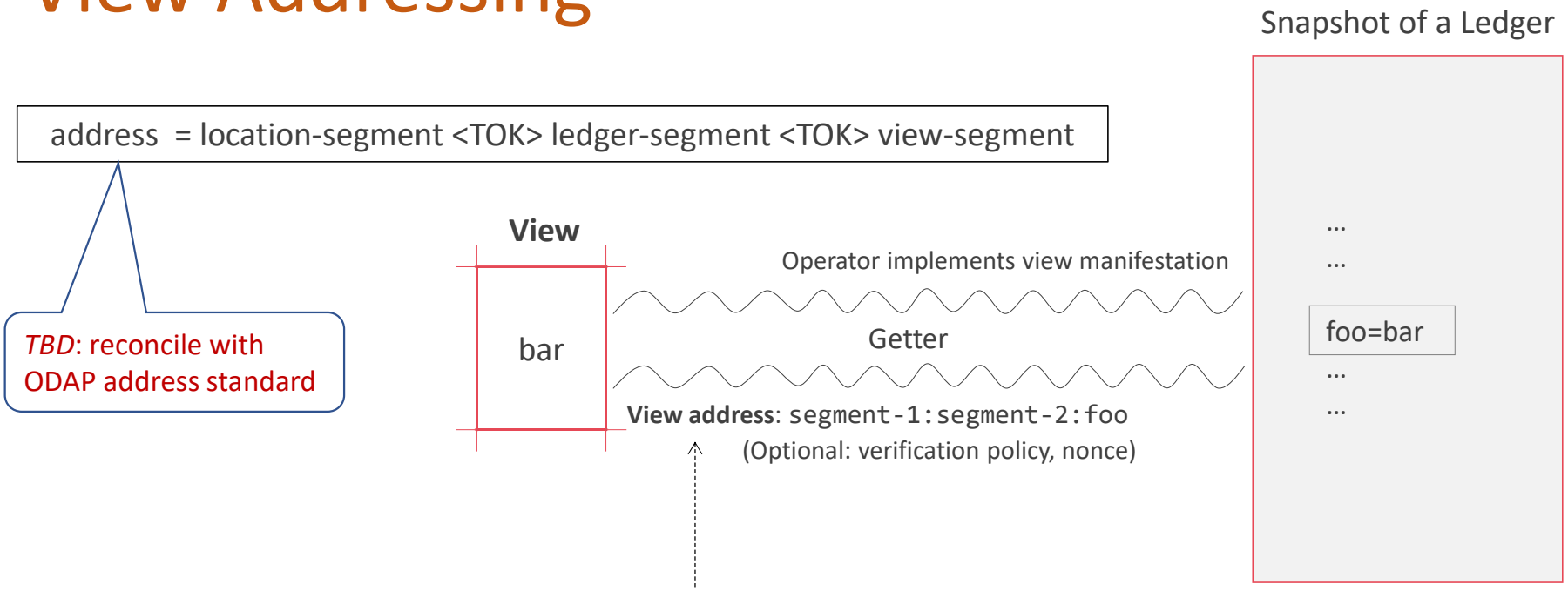


Remote State Views and Addressing



- View is information derived from ledger
- View is also proof of state of a blockchain

View Addressing



Hyperledger Fabric `<relay-address>/<network-id>/<channel-id>/<chaincode-id>:<function-id>:<arg-1>:<arg-2>:.....`

`localhost:9080/network1/mychannel:simplestate:Read:a`

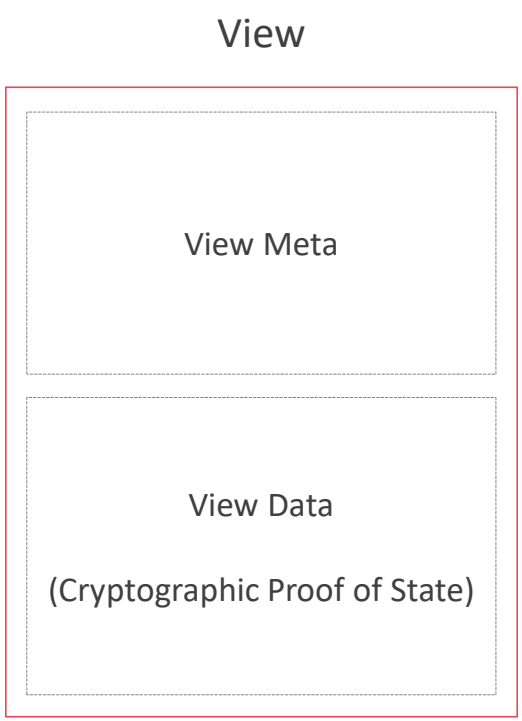
Corda `<relay-address>/<network-id>/<cordapp-node-addresses>#<flow-id>:<arg-1>:<arg-2>:.....`

`relay-corda:9081/Corda_Network/corda_partya:10003#com.cordaSimpleApplication.flow.GetStateByKey:H`

View Structure

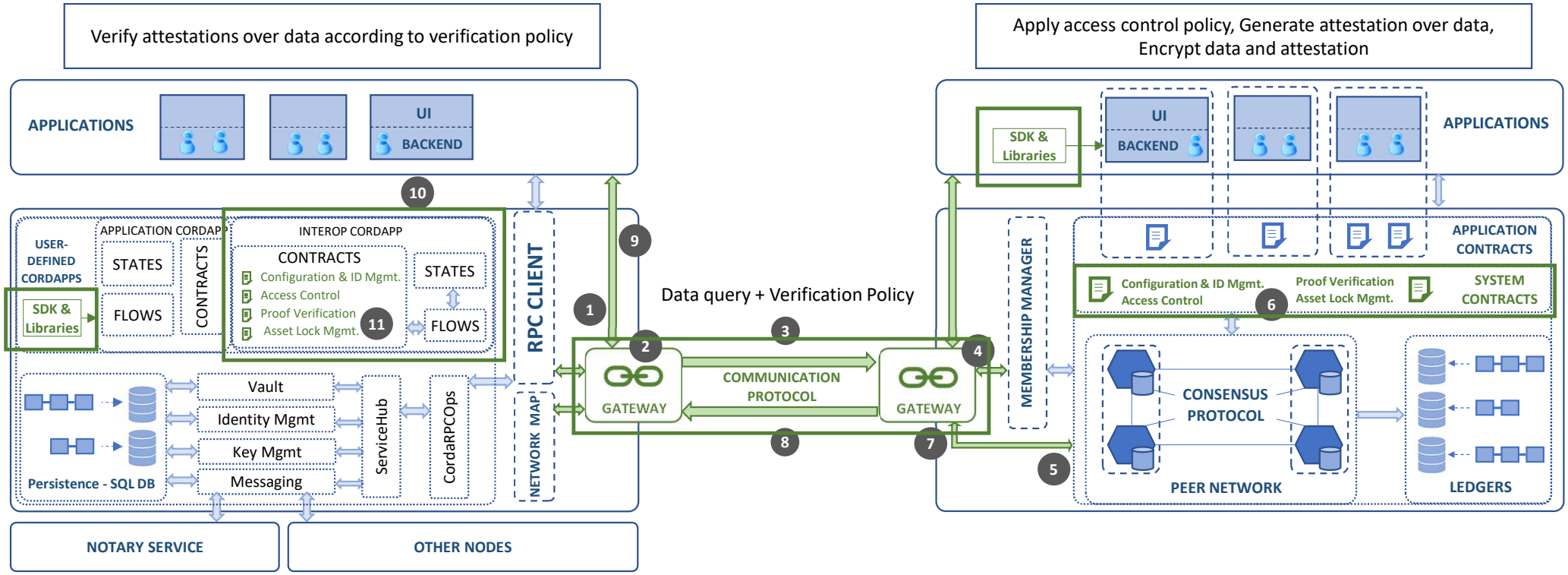
```
message Meta {
   enum Protocol {
      BITCOIN = 0;
      ETHEREUM = 1;
      FABRIC = 3;
      CORDA = 4;
   }
   Protocol protocol = 1;
   string timestamp = 2;
   string proof_type = 3;
   string serialization_format = 4;
}
```

```
message View {
  Meta meta = 1;
  bytes data = 2;
}

message ViewPayload {
  string request_id = 1;
  oneof state {
    View view = 2;
    string error = 3;
  };
}
```



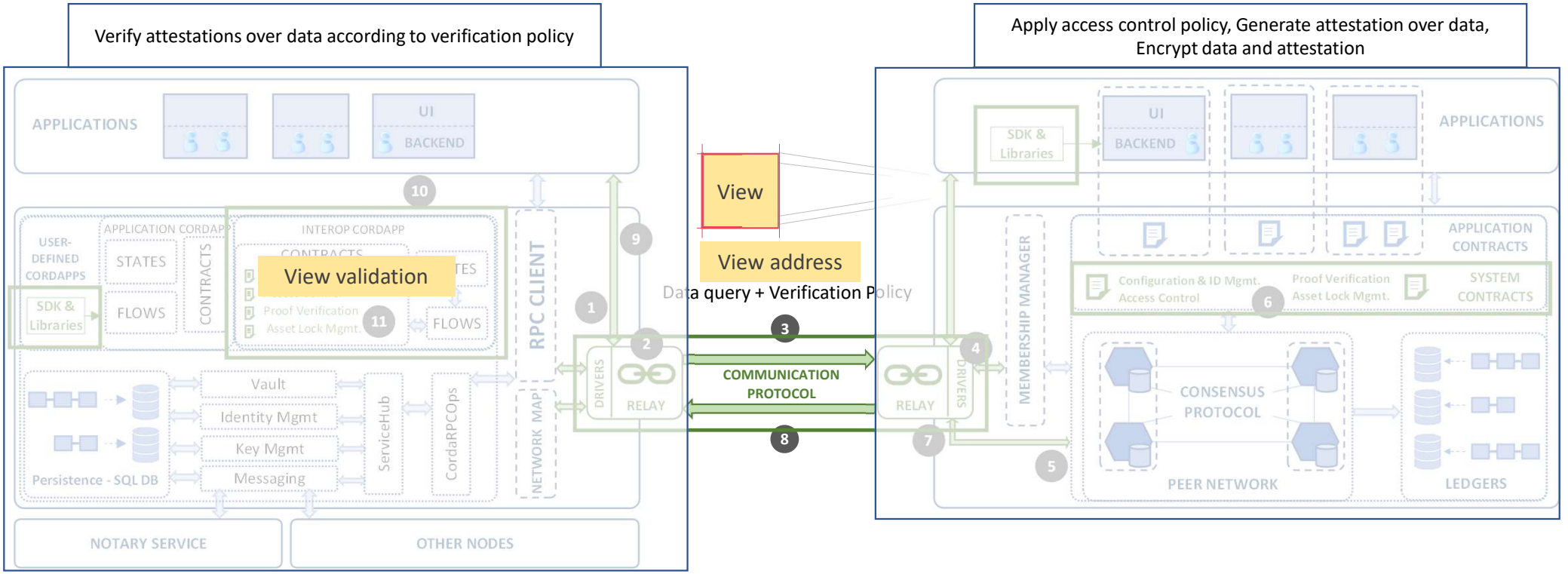

```
message FabricView {
  Response response = 1;
  ProposalResponsePayload proposal_response_payload = 3;
  repeated Endorsement endorsements = 4;
}

message CordaView {
  message Notarization {
    string signature = 1;
    string certificate = 2;
    string id = 3;
  }
  repeated Notarization notarizations = 1;
  bytes payload = 2;
}

message InteropPayload {
  bytes payload = 1;
  string address = 2;
}
```

Sample Cross-Network Data Transfer Protocol: Corda-Fabric



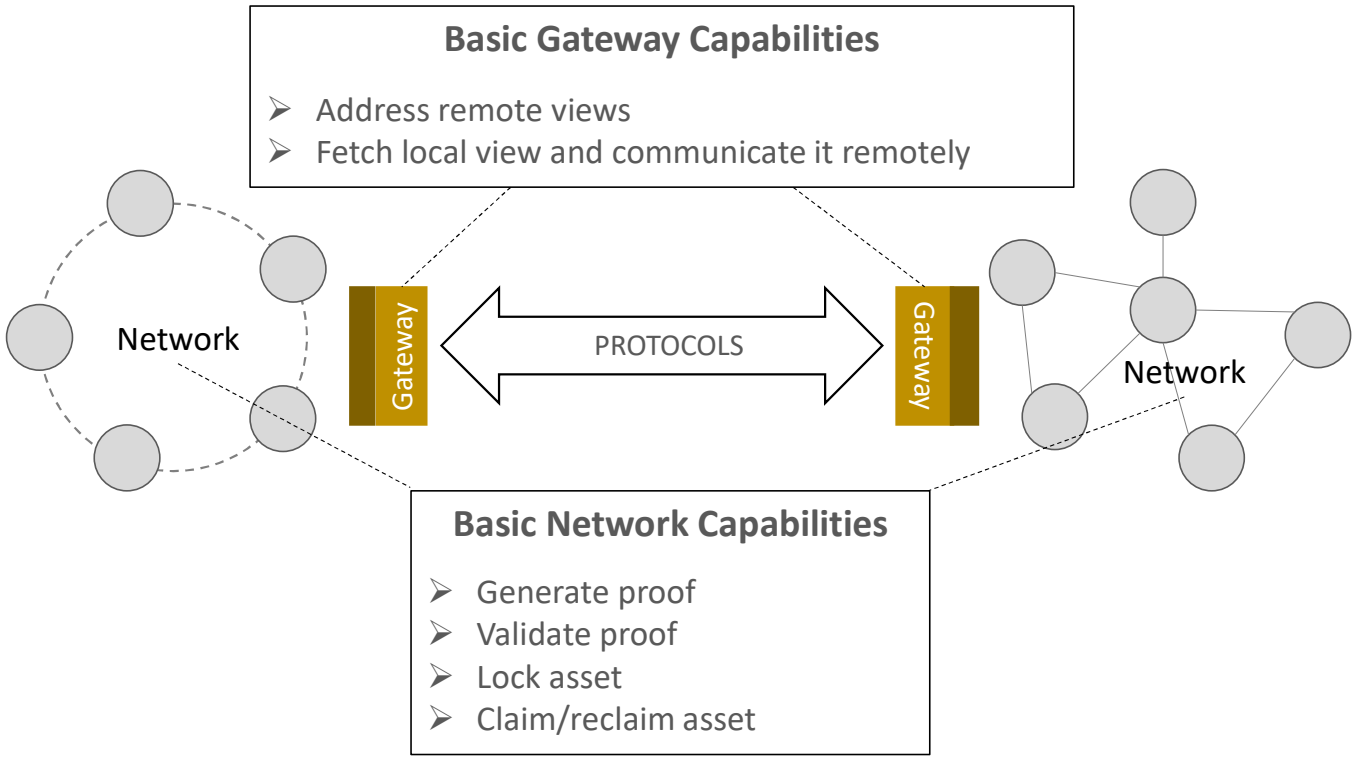
1. Application prepares a relay query to fetch remote data to supply as argument to a local chaincode invocation
2. Local relay determines how to route the request
3. Local relay sends the request to the remote network's relay
4. Remote relay, through a driver, parses query and verification policy and 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

Sample Cross-Network Data Transfer Protocol: 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

Basic Capabilities for Protocols



Non-Fundamental (Extraneous) Capabilities

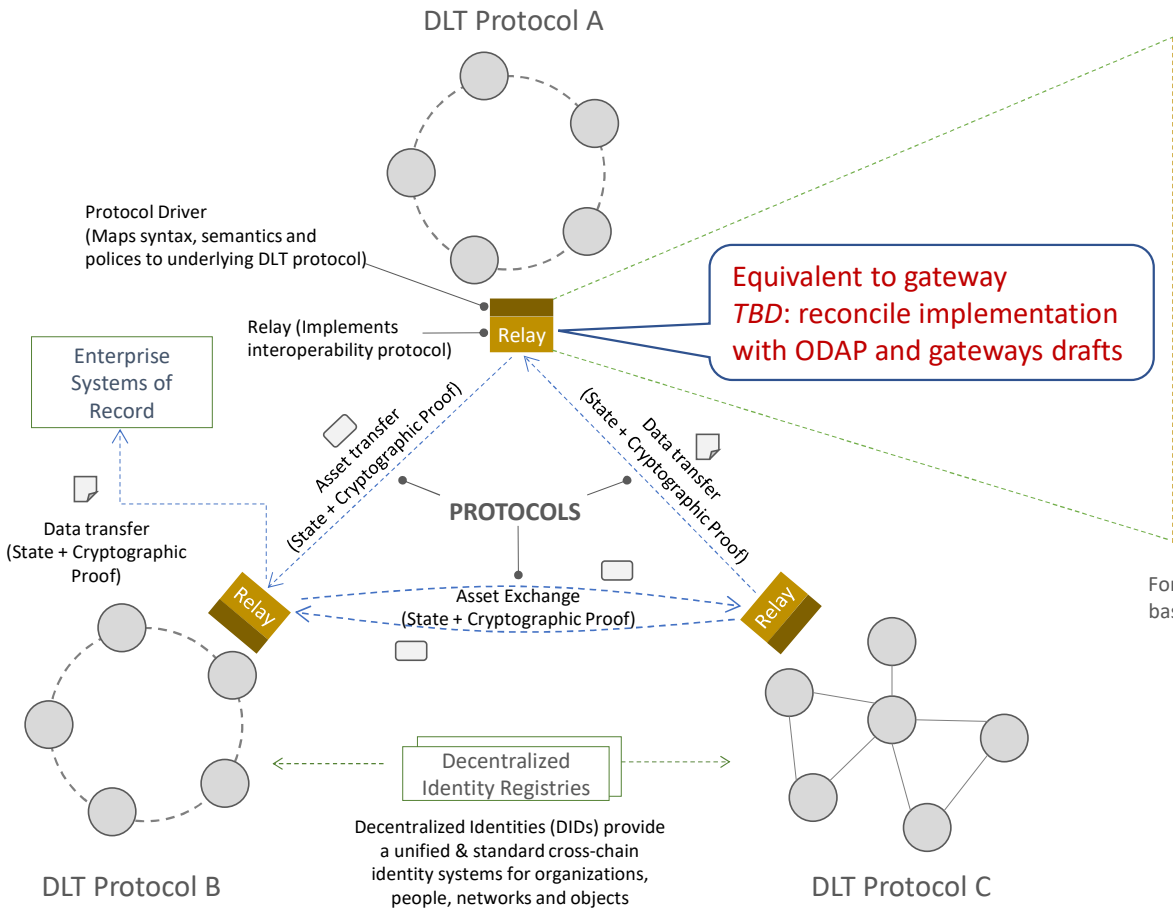
- Invoke instruction
- Event publish/subscribe

- *Claim*: this is a complete set of capabilities to realize any cross-network dependency

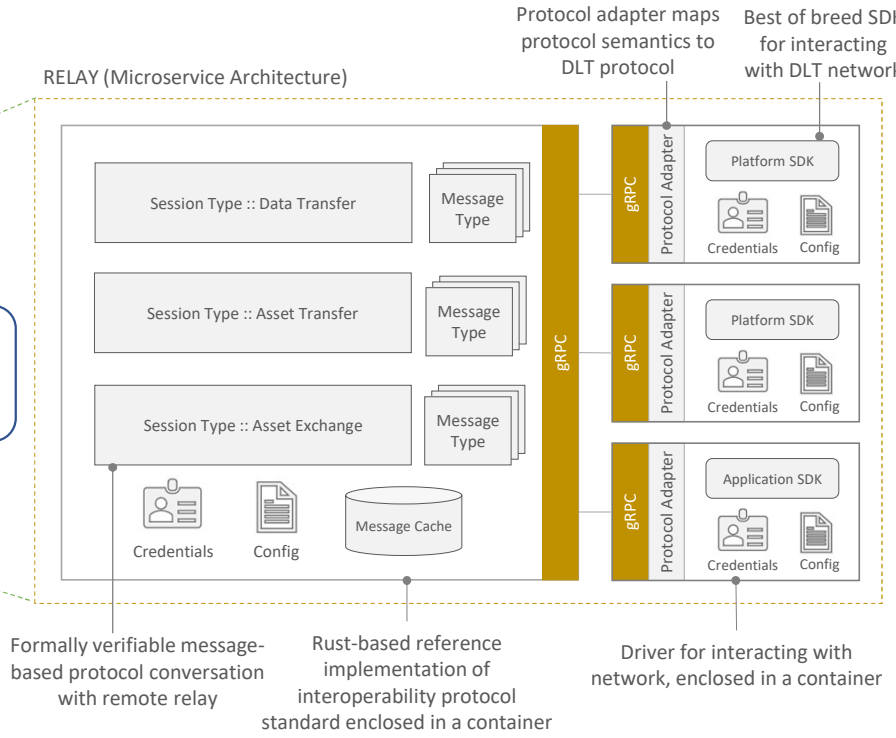
Interoperation Modes Using Basic Capabilities

- Data Transfer
  - Generate and verify proofs
- Asset Transfer
  - Multiple data transfers (and local ledger commitments)
- Asset Exchange
  - Multiple asset locks, claims, and unlocks

Reference: Weaver: DLT Interoperability (Hyperledger Labs)



Equivalent to gateway
TBD: reconcile implementation with ODAP and gateways drafts



- Research prototype (2019-)
- Open sourced in March 2021

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

References

- Dileban Karunamoorthy and Ermyas Abebe, **On the Interoperability of Distributed Ledgers**, *medium.com*
- Ermyas Abebe, Dushyant Behl, Chander Govindarajan, Yining Hu, Dileban Karunamoorthy, Petr Novotny, Vinayaka Pandit, Venkatraman Ramakrishna, Christian Vecchiola, **Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer**, *Middleware 2019 - Industry Track*
- Ermyas Abebe, Yining Hu, Allison Irvin, Dileban Karunamoorthy, Vinayaka Pandit, Venkatraman Ramakrishna, Jiangshan Yu, **Verifiable Observation of Permissioned Ledgers**, *ICBC 2021*
- Bishakh Chandra Ghosh, Venkatraman Ramakrishna, Chander Govindarajan, Dushyant Behl, Dileban Karunamoorthy, Ermyas Abebe, Sandip Chakraborty, **Decentralized Cross-Network Identity Management for Blockchain Interoperation**, *ICBC 2021*
- Venkatraman Ramakrishna, **Meet Weaver, one of the new Hyperledger Labs taking on cross-chain and off-chain operations**, *Hyperledger Global Forum 2021 Blog*
- Venkatraman Ramakrishna and Vinayaka Pandit, **Making permissioned blockchains interoperable with Weaver**, *Blockchain Pulse: IBM Blockchain Blog*