Secure Asset Transfer Protocol (SATP)

Future Extensions: Asset And Process State Queries

V. Ramakrishna (IBM)

IETF 117: Secure Asset Transfer Working Group

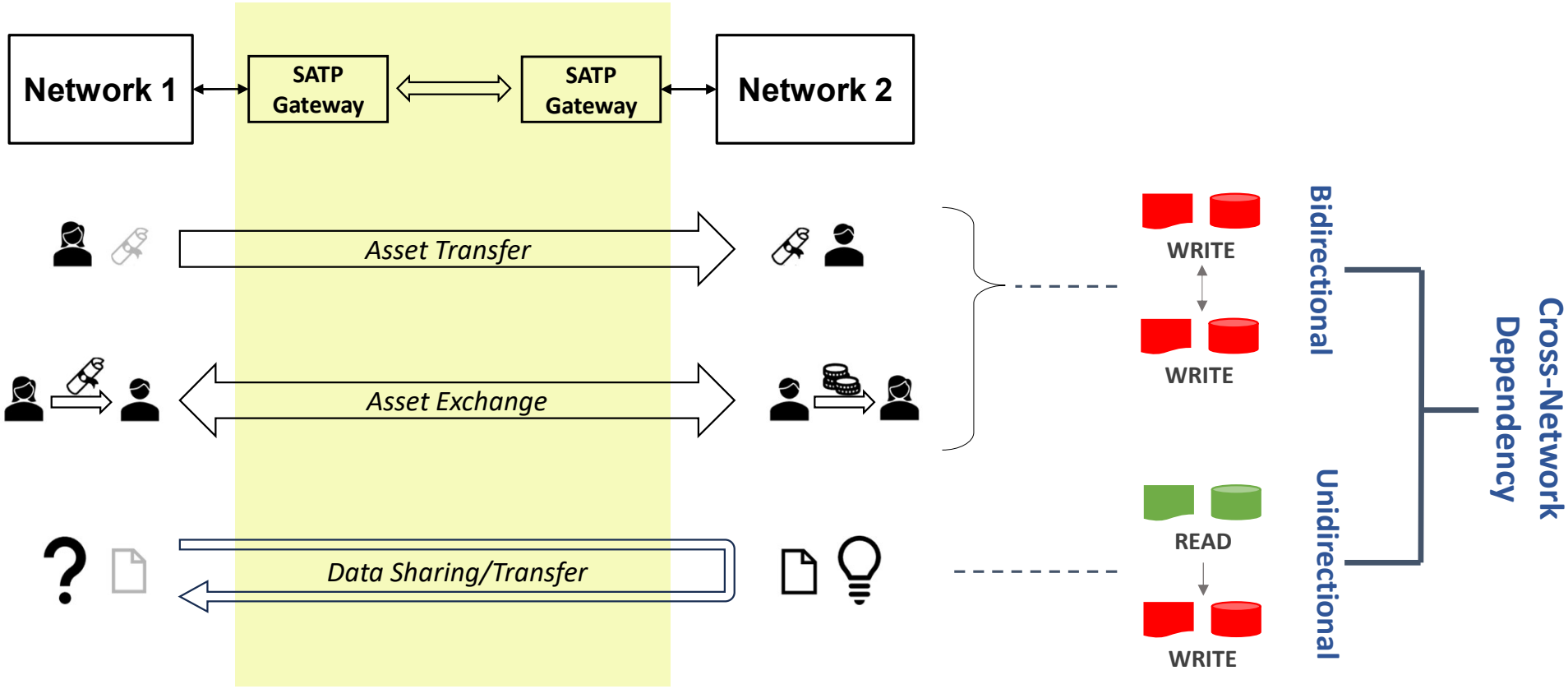
San Francisco, CA

July 25, 2023

Views and Addresses: https://datatracker.ietf.org/doc/draft-ramakrishna-satp-views-addresses/

Data Sharing: https://datatracker.ietf.org/doc/draft-ramakrishna-satp-data-sharing/

Interoperation Modes Using SATP Architecture



Requirements for Cross-Network Workflows

- Each network runs processes (business workflows) managing digital assets
- Processes in two different networks may be interdependent, and therefore require interlinking
  - Cross-network of multi-network workflow is the desired outcome, as long as security, privacy, and network independence is maintained
  - See https://datatracker.ietf.org/doc/draft-ietf-satp-usecases/ for examples
- Asset transfer is one of these interdependencies, but may typically be the culmination of a longer workflow
  - Before an asset is transferred, it may need to be discovered by another network….
  - And/or its properties and current state may need to be queried from another network
  - Hence the need for a data sharing protocol to complement an asset transfer protocol whereby the state of an asset or the process managing it, can be queried by another network
  - Proof or evidence of authenticity is a key artifact, to enable network-to-network trust
  - Example: Requesting and finding evidence of liquidity in the form of tokenized CBDC
  - Just like with asset transfer, the network and the query details can be opaque to the inter-gateway protocol

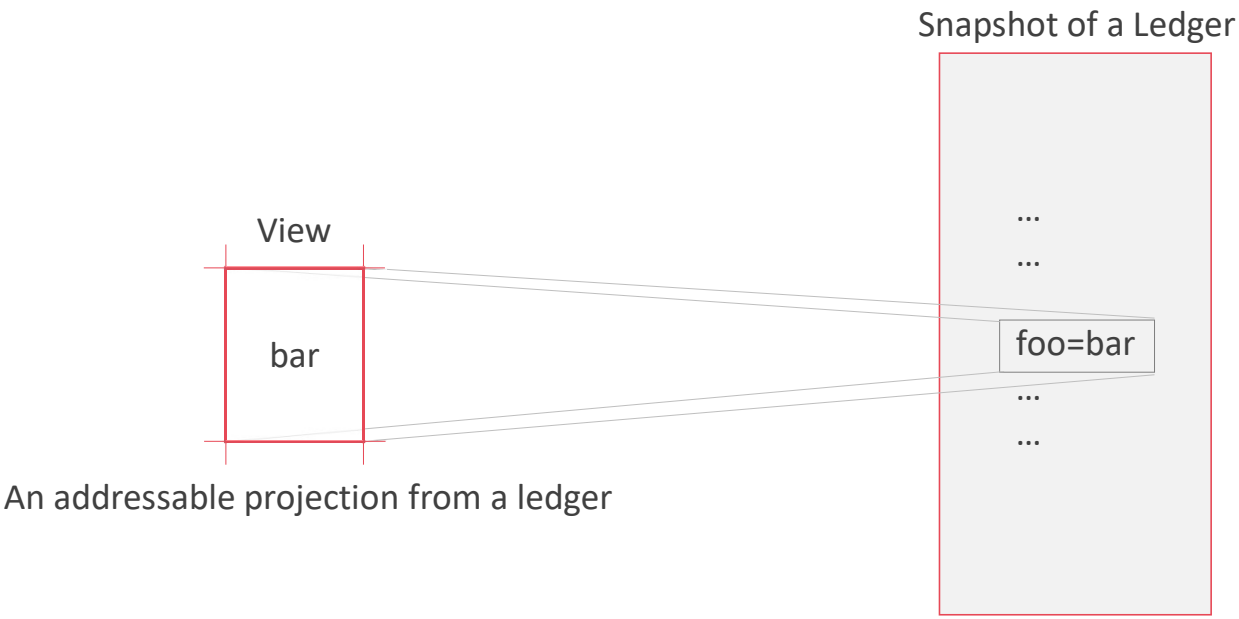
Ledger State Views and Addressing

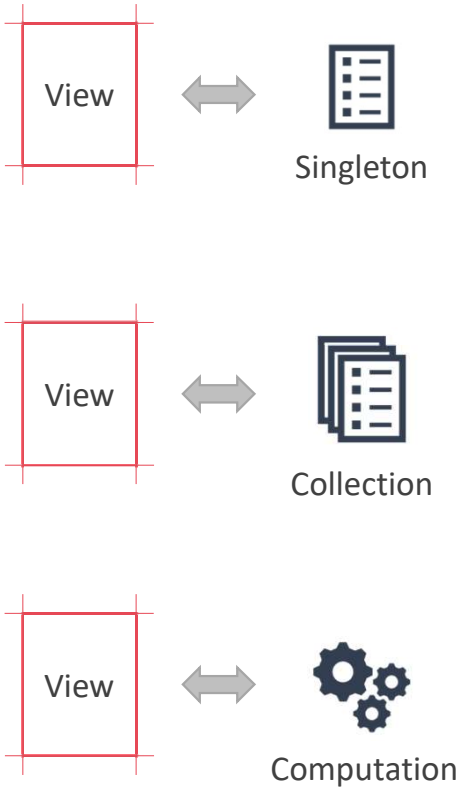
- Views and view addresses: https://datatracker.ietf.org/doc/draft-ramakrishna-satp-views-addresses/
- Data sharing protocol for querying view addresses and receiving view responses: https://datatracker.ietf.org/doc/draft-ramakrishna-satp-data-sharing/

- Peer-reviewed publications
  - Verifiable Observation of Permissioned Ledgers, ICBC 2021: https://arxiv.org/abs/2012.07339
  - Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer, Middleware 2019 - Industry Track: https://arxiv.org/abs/1911.01064
- Open-source implementation within Hyperledger Cacti: https://github.com/hyperledger/cacti/tree/main/weaver

- General-purpose addressing and query-response protocol

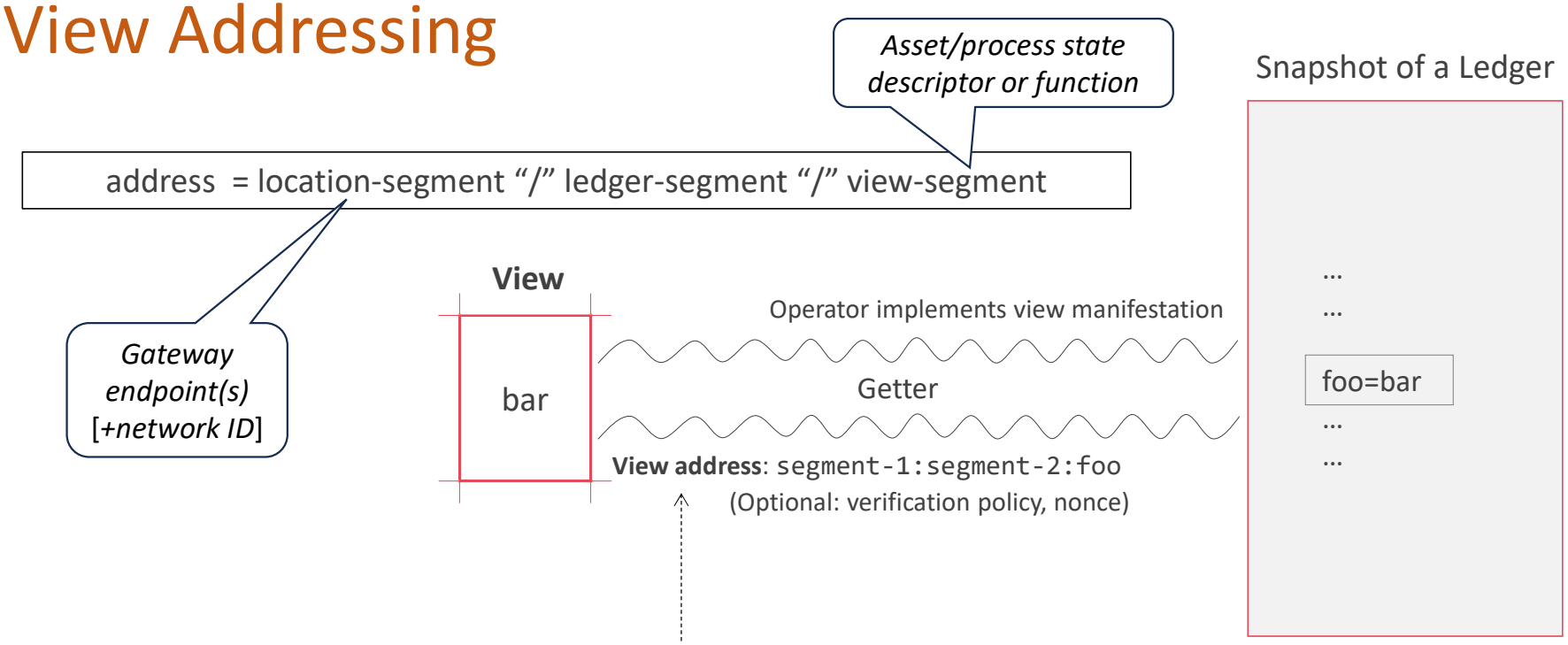
Remote State Views and Addressing





- View is information derived from ledger
- View is also proof of state of a blockchain
- Especially significant in permissioned ledgers

View Addressing



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

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

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

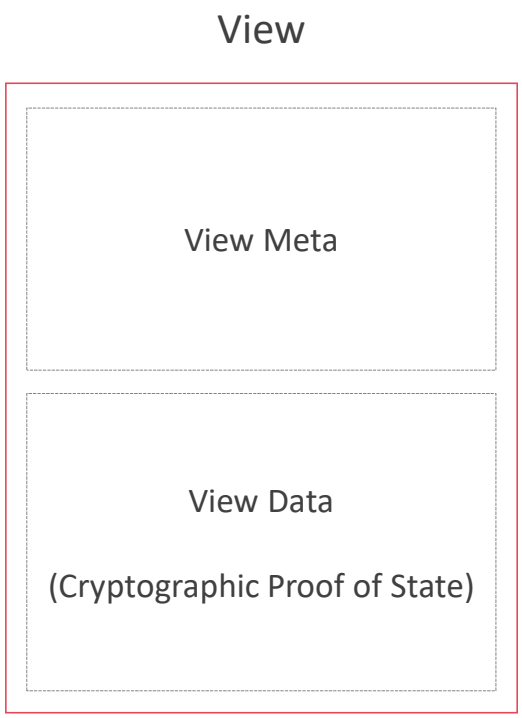
`relay-corda:9081/Corda_Network/corda_partya:10003#com.cordaSimpleApplication.flow.GetStateByKey:H`

View Structure: Samples from Hyperledger Cacti

```
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;
  };
}
```

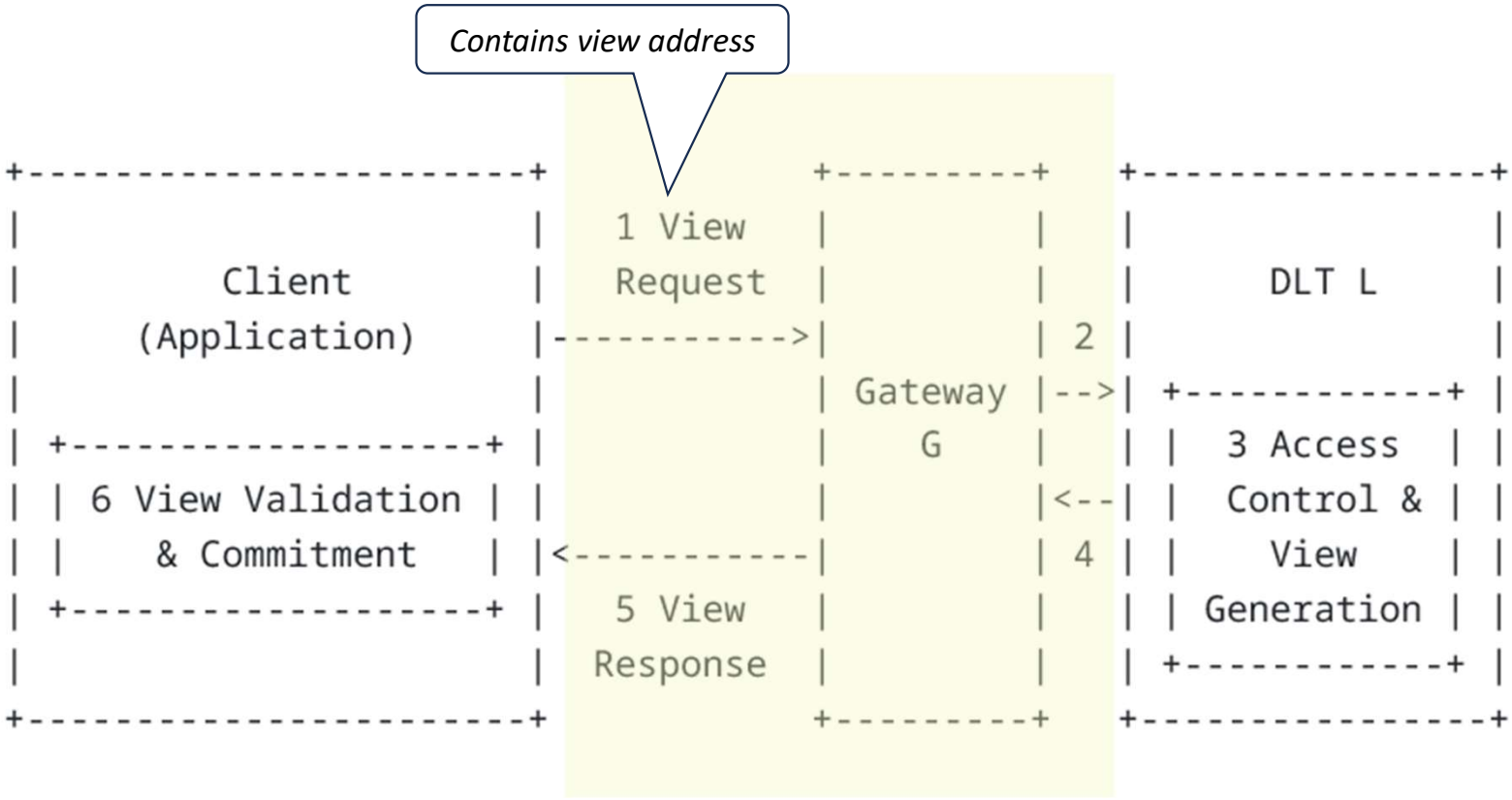



```
message FabricView {
  Response response = 1;
  ProposalResponsePayload proposal_response_payload = 3;
  repeated Endorsement endorsements = 4;
}

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;
}
```

Request-Response Protocol for Data Sharing: Single Network



Request-Response Protocol for Data Sharing: Two Networks