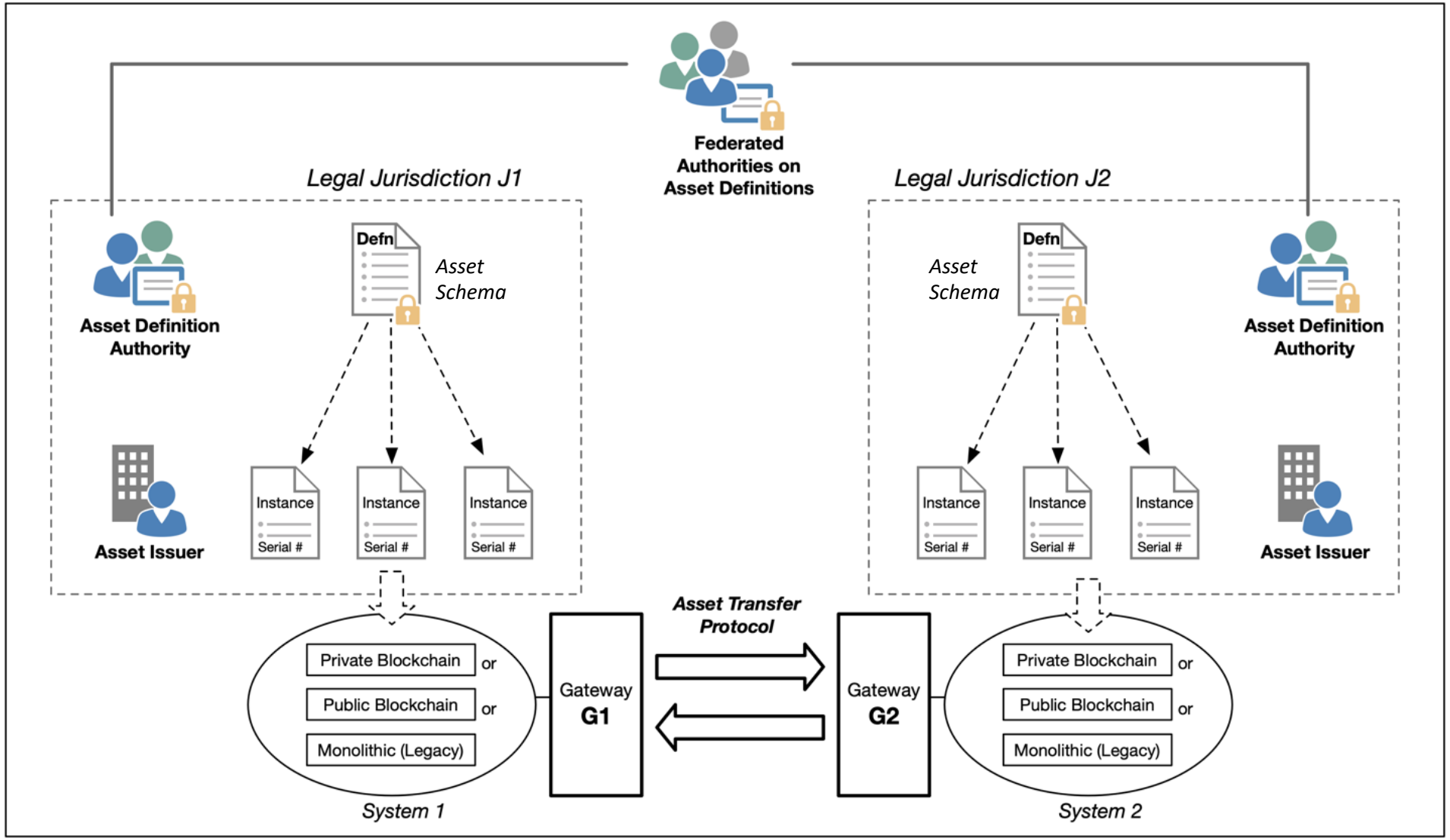
Asset Schemas (aka Asset profiles)

DENIS AVRILIONIS – COMPELLIO

SATP WG

IETF 118

ASSET DEFINITION



ASSET SCHEMAS

Asset Schema

General Definition of the structure of an asset schema

Specialisation

Bill of Lading schema

Supply chain

Tokenized commodity schema

Finance / OTC trading

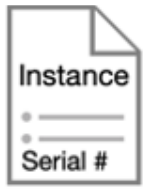
Tokenized DNS domain schema

Internet Domain Management

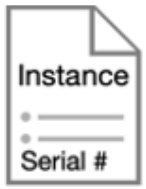
...

Industry-specific asset schemas

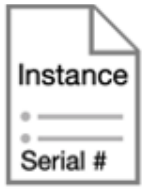
Instantiation



Bill of Lading #XYZ



Tokenized Gold for custodian YZT



Tokenized domain For example.com

Asset Instances managed by a given Network (system)

Asset Schema – Data Requirements

- Identification
  - On-chain (token) unique ID – Network-specific ID
  - Off-chain unique ID – Off-Network (LegacyID, PhysicalID,…)
  - Normative identification
    - Code (e.g. LU0001234567)
    - Code Kind (e.g. ISIN)
    - Descripition (e.g. Prospectus/KID/ReferenceDoc/…)
- Asset Attribute Data Structure (schema level – e.g. using of JSON LD and JSON Schema)
- Capabilities
  - Transfer-ability
  - Trade-ability
  - Extinguish-ability
  - Tax-ability
  - Collateralise-ability
  - Confiscate-ability
- Ownership
  - CreatorID
  - ProcessorID
  - DistributorID
  - CustodianID
  - OwnerID
- Validity/Compliance
  - AssetDefinitionAuthorityID
  - Issuance Date
  - Expiration Date
  - AuditEvidence
  - AuditorID
- Jurisdiction
  - ownerJurisdictionScope
  - processorJurisdictionScope
  - creatorJurisdictionScope

Asset Instance – Processing Requirements

Asset Instances must:

- Be Identifiable: asset instances have universal IDs
- Be Typed: asset instances must contain an explicit reference to their *assetSchemaID*
- Be Trackable: asset instances shall contain references to the previous ID in case of transfer
- Be “Serialisable stable”:

```
assetInstance = deserialise( serialise( assetInstance ) )
```

- Be “Persistent stable”:

```
serialise( assetInstance ) = fetch( persist( serialise( assetInstance ) ) )
```

- Be “Hash stable”:

```
hash ( serialise( assetInstance ) ) = H,
where H is identical in computing language and technological stack
```

→Possible implementation: Serialisation w/ JCS (RFC8785), Signing w/ JWS (RFC7515)

Optional: An *Asset Discovery Service* may be implemented as a global directory of assets and asset schemas