Authentic Chained Data Container
ACDC

Verifiable Data Structures (Graphs)
that support
Authenticatable Attestations & Credentials

Samuel M. Smith Ph.D.
sam@prosapien.com
2022/09/15

Resources

Specification Documentation:

ACDC Internet Draft (ToIP - IETF)
https://github.com/trustoverip/tswg-acdc-specification

ACDC for Muggles
https://docs.google.com/presentation/d/1mO1EZa9BcjAjWEzw7DWi124uMfyNyDeM3HuajsGNoTo/edit#slide=id.ga411be7e84_0_0

Resources on KERI and ACDC
https://keri.one/keri-resources/

GLEIF vLEI Credentials (Global Legal Entity Identifier Foundation) ISO LEI specification
https://www.gleif.org/en/vlei/introducing-the-verifiable-lei-vlei

Community: (meetings, open source code, IETF internet drafts)

ACDC Working Group
https://wiki.trustoverip.org/display/HOME/ACDC+%28Authentic+Chained+Data+Container%29+Task+Force

Related Internet Drafts
https://github.com/WebOfTrust/keri

Important ACDC Features

ACDC is based on KERI so one gets all the features of KERI for free.

Leverage SAIDs (Self-Addressing Identifiers) and AIDs (Autonomic Identifiers)
    Based on white magic (dumb) crypto: digests, and digital signatures.
Leverages CESR (Composable Event Streaming Representation) to resolve the text vs binary tension.
JSON Schema (Type-is-schema, schema are immutable)
Chaining (property graph model)
Graduated Disclosure (compact, partial, private, selective)
Contractually Protected Disclosure (chain-link confidentiality, contingent disclosure)
    Protection against data exploitation from both statistical correlation and cryptographic correlation
Decentralized extensibility model
Zero-trust End-Verifiable

Basic ACDCs

Private Compact Variant


```
{
  "v":  "ACDC10JSON00011c_",
  "d":  "EAdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "u":  "0ABghkDaG7OY1wjaDAE0qHcg",
  "i":  "did:keri:EBkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:ECmRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":  "ED6jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":  "EEveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":  "EFH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":  "EG71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB"
}
```

Basic ACDC JSON Schema

Private Compact Variant

```
{
  "v":  "ACDC10JSON00011c_",
  "d":  "EAdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "u":  "0ABghkDaG7OY1wjaDAE0qHcg",
  "i":  "did:keri:EBkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:ECmRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":  "ED6jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":  "EEveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":  "EFH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":  "EG71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB"
}
```

Non-Composed JSON Schema

```
{
  "$id": "EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Compact Private ACDC",
  "description": "Example JSON Schema for a Compact Private ACDC.",
  "credentialType": "CompactPrivateACDCExample",
  "type": "object",
  "required":
  [
    "v",
    "d",
    "u",
    "i",
    "ri",
    "s",
    "a",
    "e",
    "r"
  ],
  "properties":
  {
    "v":
    {
      "description": "ACDC version string",
      "type": "string"
    },
    "d":
    {
     "description": "ACDC SAID",
      "type": "string"
    },
    "u":
    {
     "description": "ACDC UUID",
      "type": "string"
    },
    "i":
    {
      "description": "Issuer AID",
      "type": "string"
    },
    "ri":
    {
      "description": "credential status registry ID",
      "type": "string"
    },
    "s": {
      "description": "schema SAID",
      "type": "string"
    },
    "a": {
      "description": "attribute SAID",
      "type": "string"
    },
    "e": {
      "description": "edge SAID",
      "type": "string"
    },
    "r": {
      "description": "rule SAID",
      "type": "string"
    }
  },
  "additionalProperties": false
}
```

Uncompacted Private Attribute Section

```
{
  "a":
  {
    "d": "EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "u": "0AwjaDAE0qHcgNghkDaG7OY1",
    "i": "did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA",
    "score": 96,
    "name": "Jane Doe"
  }
}
```

Private Compact Variant

```
{
  "v":  "ACDC10JSON00011c_",
  "d":  "EAdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "u":  "0ABghkDaG7OY1wjaDAE0qHcg",
  "i":  "did:keri:EBkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:ECmRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":  "ED6jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":  "EEveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":  "EFH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":  "EG71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB"
}
```

Composed JSON Schema

```
{
  "a":
  {
    "description": "attribute section",
    "oneOf":
    [
      {
        "description": "attribute SAID",
        "type": "string"
      },
      {
        "description": "uncompacted attribute section",
        "type": "object",
        "required":
        [
          "d",
          "u",
          "i",
          "score",
          "name"
        ],
        "properties":
        {
          "d":
          {
            "description": "attribute SAID",
            "type": "string"
          },
          "u":
          {
            "description": "attribute UUID",
            "type": "string"
          },
          "i":
          {
            "description": "Issuee AID",
            "type": "string"
          },
          "score":
          {
            "description": "test score",
            "type": "integer"
          },
          "name":
          {
            "description": "test taker full name",
            "type": "string"
          }
        },
        "additionalProperties": false,
      }
    ]
  }
}
```

Edge Section


```
{
  "e":
  {
    "d": "EBrzwLIr9Bf7V_NHwY1lkFrn9y2PgveY4-9XgOcLxUdY",
    "boss":
    {
      "d": "EFy2PgveY4-9XgOcLxUdYerzwLIr9Bf7V_NHwY1lkFrn",
      "n": “EIl3MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA”,
      "s": “EFOLe71iheqcywJcnjtJtQIYPvAu6DZAIl3MORH3dCdo”,
    }
  }
}
```

Nested Edge Section with Operators


```
{
  "e":
  {
    "d": "EBrzwLIr9Bf7V_NHwY1lkFrn9y2PgveY4-9XgOcLx,UdY",
    "o": "AND",
    "boss":
    {
      "n": "EGl3MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA",
      "o": ["NI2I", "NOT"]
    },
    "baby":
    {
      "n": "EMRH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZAIl3A",
      "o": "I2I"
    },
    "food":
    {
      "o": "OR",
      "plum":
      {
        "n": "EHIYPvAu6DZAIl3AORH3dCdoFOLe71iheqcywJcnjtJt",
        "o": "NI2I"
      },
      "pear":
      {
        "n": "ECtQIYPvAu6DZAIl3AORH3dCdoFOLe71iheqcywJcnjt",
        "o": "NI2I"
      }
    }
  }
}
```

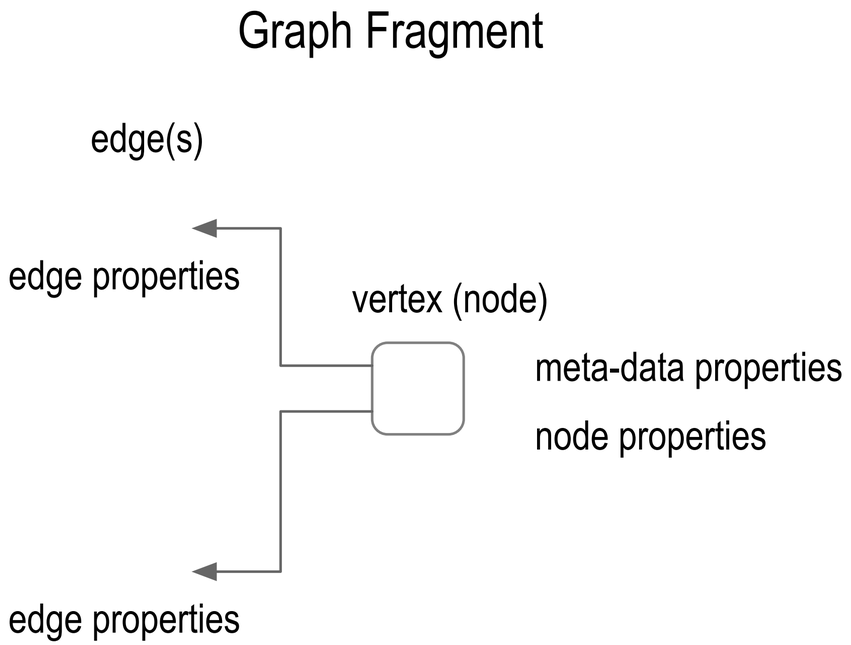
ACDC Normative Field Labels

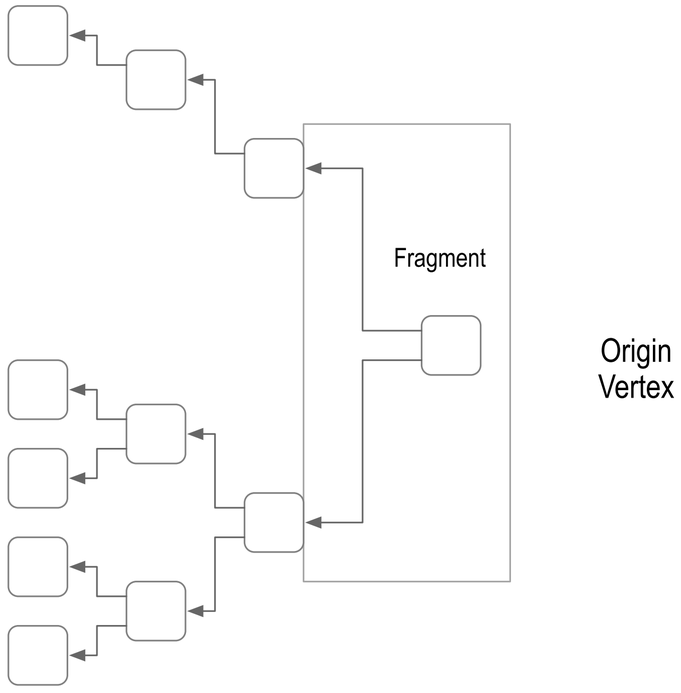
ACDC Field Labels

Label	Title	Description
		Top Level Fields
`v`	Version String	Regex-able format: `"ACDCvvSSSShhhhhh_"` that provides protocol type, protocol version, serialization type, serialized size, and terminator.
`d`	Digest (SAID)	Self-Addressing IDentifier. Self-referential fully qualified (agile) cryptographic digest of enclosing map in CESR format: `"EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM"`
`i`	Issuer Identifier (AID)	Autonomic IDentifier whose control authority is established via KERI verifiable key state in CESR format: `"EAkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM"`
`u`	UUID	Random Universally Unique IDentifier as fully qualified high entropy pseudo-random string, a salty nonce. Protection from rainbow table attack on private variants in CESR format: `"0ABghkDaG7OY1wjaDAE0qHcg"`
`ri`	Registry Identifier	Issuance and/or revocation, transfer, or retraction registry identifier, cryptographically derived from Issuer Identifier (AID-ish) in CESR format: `"ECmRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt"`
`s`	Schema	Either the SAID in CESR format block in CESR format or the block itself: `"ED6jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A"`
`a`	Attribute	Either the SAID of a block of attributes in CESR format or the block itself in CESR format: `"EEveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY"`
`A`	Attribute Aggregate	Either the Aggregate of a selectively disclosable block of attributes in CESR format or the block itself in CESR format: `"EHveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY"`
`e`	Edge	Either the SAID of a block in CESR format of edges or the block itself in CESR format:
`r`	Rule	Either the SAID a block of rules in CESR format or the block itself.
		Other Fields
`d`	Digest (SAID)	Self-Addressing IDentifier. Self-referential fully qualified (agile) cryptographic digest of enclosing map in CESR format: `"EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM"`
`i`	Identifier (AID)	Autonomic IDentifier context dependent whose control authority is established via KERI verifiable key state in CESR format, such as, Issuee Identifier,: `"EAkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM"`
`u`	UUID	Random Universally Unique IDentifier as fully qualified high entropy pseudo-random string, a salty nonce. Protection from rainbow table attack on private variants in CESR format: `"0ABghkDaG7OY1wjaDAE0qHcg"`
`n`	Node	SAID of another ACDC in CESR format as the terminating point (vertex) of a directed edge that connects the encapsulating ACDC node to the specified ACDC as a distributed property graph (PG) fragment:
`o`	Operator	Either unary operator on edge or m-ary operator on edge-group in edge section. Enables expressing of edge logic on edge subgraph.
`w`	Weight	Edge weight property that enables default property for directed weighted edges and operators on directed weighted edges.
`l`	Legal Language	Text of Ricardian contract clause.

Big Picture: What is an ACDC?

- **Decentralizable distributed verifiable data structure that is structurally constrained by immutable but composable JSON Schema.**
- **Each ACDC is universally uniquely referenced by its SAID.**
- **Authenticatable decentralizable distributed graph fragment that may be communicated securely.**
- **Graph fragments use SAIDs to (hash-chain) together without any expansion needed.**
- **The composition of graph fragments is an authenticatable verifiable graph data structure, i.e. a chained set of ACDCs (zero-trust end-verifiable security model)**
- **Verifiable data structures all the way down**





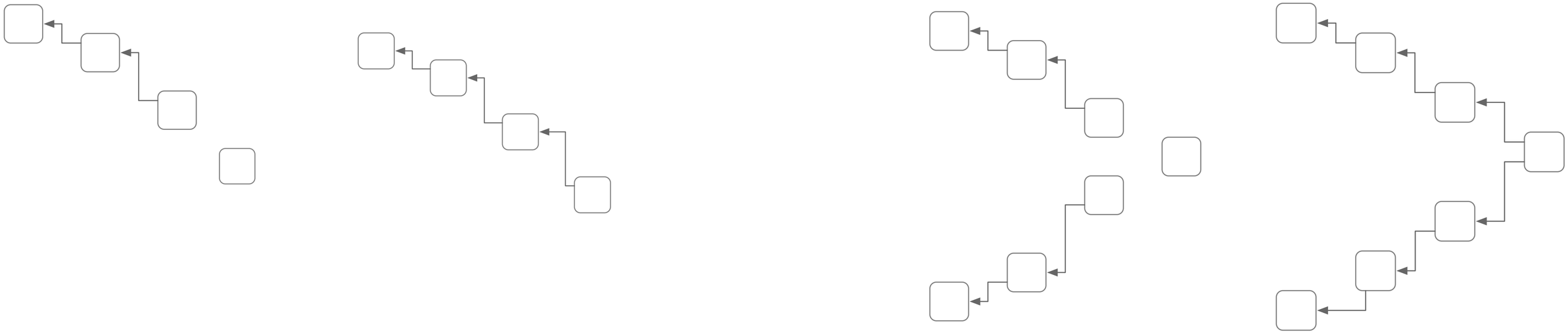
Chained ACDCs (graphs) Enable

Provenanced chains-of-custody of decentralizable authenticatable data attestations

Traceable data for data supply chains

Provenanced chains-of-authority for decentralizable authenticatable credentials

Verifiable delegated entitlements or authorizations

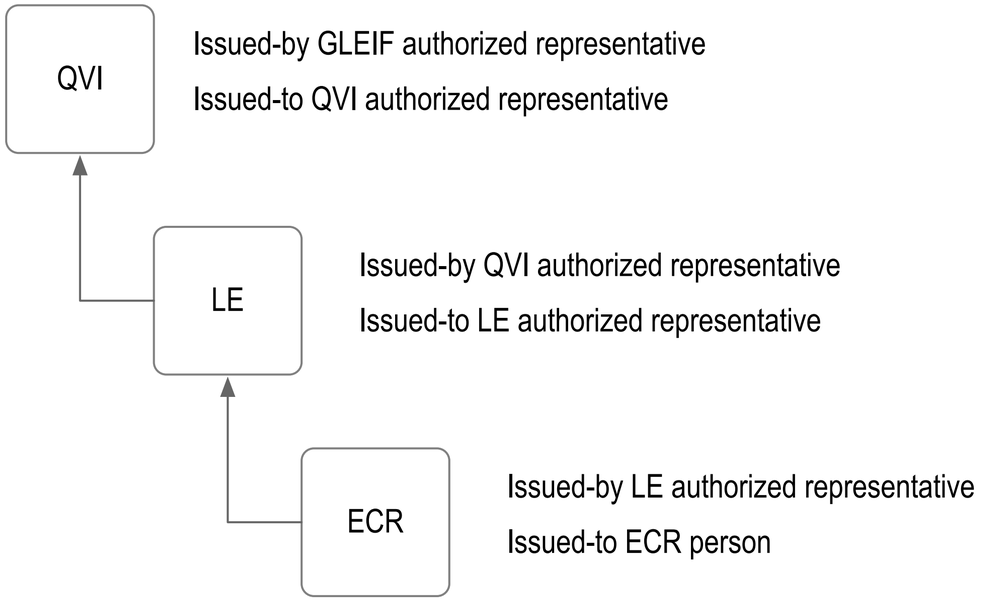


GLEIF vLEI Credential Example

Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Engagement Context Role (ECR) Credential

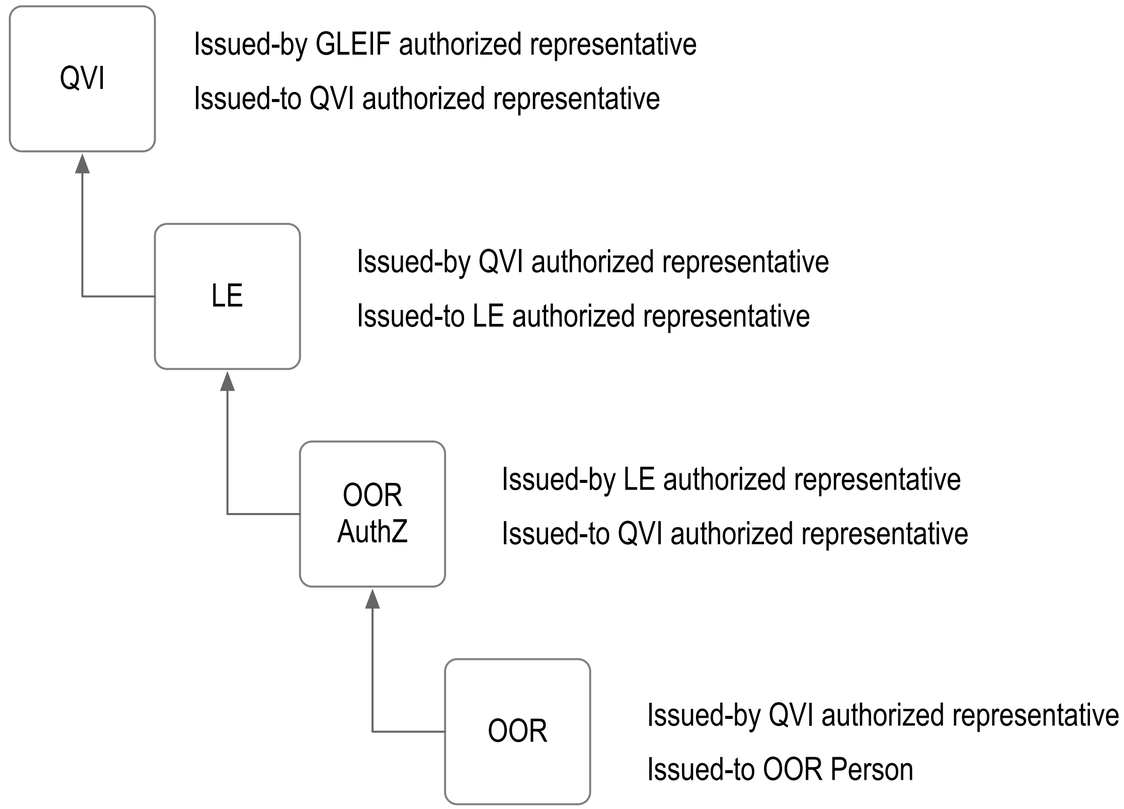


Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Official Organizational Role Authorization (OOR-AuthZ) Credential

Official Organizational Role (OOR) Credential



GLEIF vLEI Authorized Attestation Example

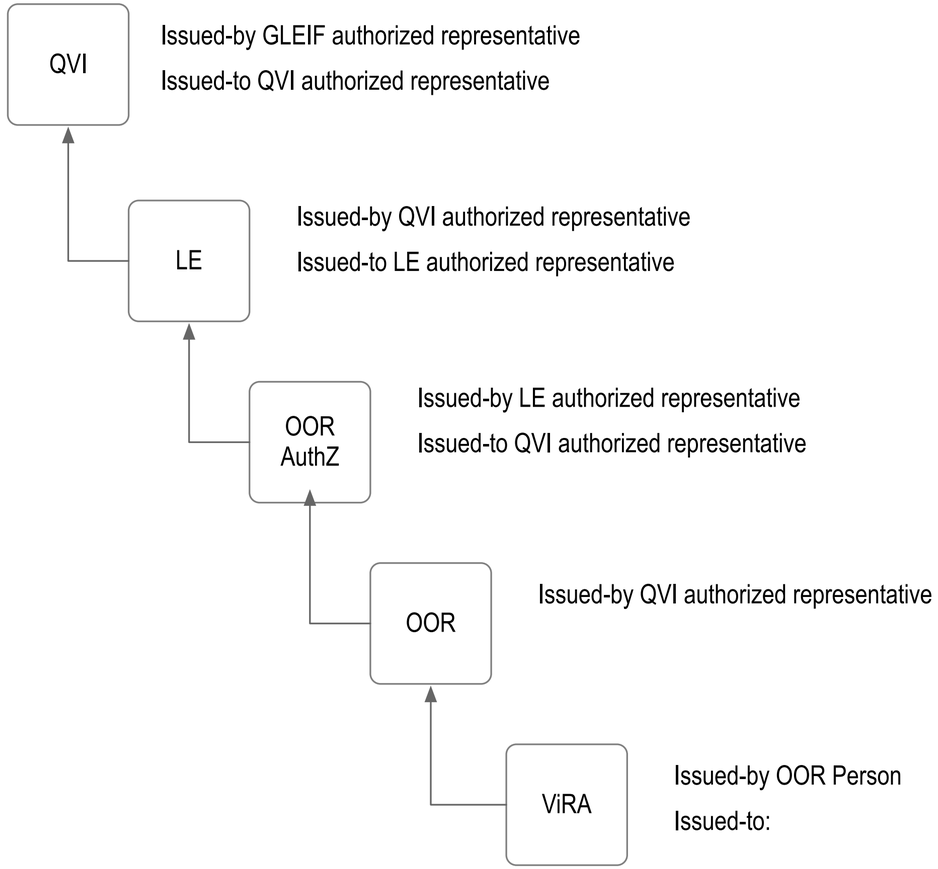
Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Official Organizational Role Authorization (OOR-AuthZ) Credential

Official Organizational Role (OOR) Credential

Verifiable IXBRL Report Attestation (ViRA)



GLEIF vLEI Credential Example: Schema Edge OOR-Auth

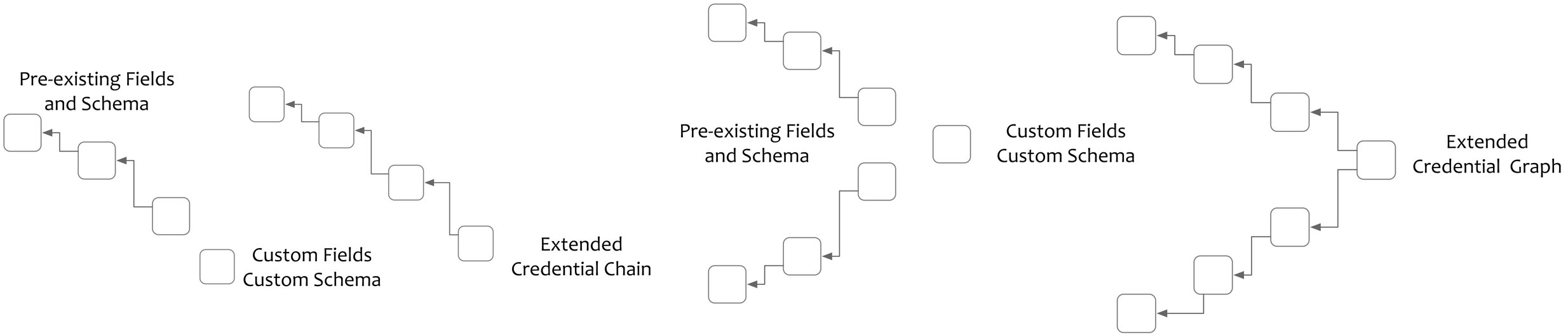
```
"e": {
      "description": "edges block",
      "properties": {
        "d": {
          "description": "said of edges block",
          "type": "string"
        },
        "le": {
          "description": "chain to legal entity vLEI credential",
          "properties": {
            "n": {
              "type": "string"
            },
            "s": {
              "type": "string",
              "description": "SAID of required schema of the credential pointed to by this node",
              "const": "EWJkQCFvKuyxZi582yJPb0wcwuW3VXmFNuvbQuBpgmIs"
            }
          },
          "additionalProperties": false,
          "required": [
            "n",
            "s"
          ],
          "type": "object"
        }
      },
      "additionalProperties": false,
      "required": [
        "d",
        "le"
      ],
      "type": "object"
    },
```

Append-to-Extend

Append-only verifiable data structures have strong security properties that simplify end-verifiability & foster decentralization.

Append-only provides permission-less extensibility by downstream issuers, presenters, and/or verifiers

- Each ACDC has a universally-unique content-based identifier with a universally-unique content-based schema identifier.
- Fully decentralized name-spacing.
- Custom fields are appended via chaining via one or more custom ACDCs defined by custom schema (type-is-schema).
- No need for centralized permissioned name-space registries to resolve name-space collisions.
- The purposes of a registry now become merely schema discovery or schema blessing for a given context or ecosystem.
- The reach of the registry is tuned to the reach of desired interoperability by the ecosystem participants.
- Human meaningful labels on SAIDs are local context only.
- Versioning is simplified because edges still verify if new schema are backwards compatible. (persistent data structure model)



ACDC Summary

Push the functionality envelope of “verifiable credentials” using minimally sufficient means tooling.

Future looking features:

- SAIDS (agile self-referential content identifiers)
- Graduated Disclosure
- Contractually Protected Disclosure
- Chaining and Delegation
- Permissionless Extensibility
- Zero-Trust End Verifiability via Verifiable Data Structures
- Multiple Serializations
- Scalability (text and binary streaming support) via CESR self-framing composable primitives and groups of primitives

Minimally Sufficient Means

- Dumb Crypto (Digests and Digital Signatures)
- JSON and JSON Schema (composability features)
- Extensible Layered Model

Interoperability Through Layering

The ACDC/KERI stack is opinionated about security with very precise strong security properties

Fully decentralizable, distributable, zero-trust, end-verifiable mechanisms based on authenticatable extensible verifiable data structures (append-only hash chained signed)

Mashups of security features that do not cleanly separate security properties are antithetical to the ACDC/KERI stack.

Interoperable security first, then interoperable semantics for upper layers of the application stack.

ACDCs could be used as a secure conveyance for other representations that appear as an opaque payload in the ACDC.

ACDCs could be a blessed trust spanning layer for W3C VCs for those who want its security properties

ACDC Community Interoperability Ask

Enable one-to-one mappings between ACDCs and other data models and representations without forcing ACDCs to use syntax from other data models and representations.

ACDCs use JSON-Schema with JSON, CBOR, MGPK, and CESR serializations, compact labels & CESR Primitives.

CESR primitives can be mapped one-to-one to JWT primitives.

ACDC normative field labels can be mapped one-to-one to their equivalents in other representations.

Allow JSON Schema (ACDCs need composition operators from JSON Schema).

Not asking to replace JSON-LD but be allowed to co-exist with JSON-LD.

JSON is well just JSON. Please no MUST have non-JSON artifacts (@context).

Questions?

Least Disclosure

Principle of Least Disclosure

ACDCs are designed to satisfy the principle of least disclosure.

The system should disclose only the minimum amount of information about a given party needed to facilitate a transaction and no more.

Partial Disclosure

- Compactness
- Chain-link Confidentiality

Selective Disclosure

- Unbundling
- Bulk-issuance

Mechanisms:

- Compact via SAID over content
- Blinded via SAID over content with embedded UUID (salty-nonce)
- Unbundled via Aggregate of bundle of blinded content
- Uncorrelatable via bulk issued blinded content

Three Party Exploitation Model

First-Party = Discloser of data.
Second-Party = Disclosee of data received from First Party (Discloser).
Third-Party = Observer of data disclosed by First Party (Discloser) to Second Party (Disclosee).

Second-Party (Disclosee) Exploitation
implicit permissioned correlation.
    no contractual restrictions on the use of disclosed data.
explicit permissioned correlation.
    use as permitted by contract
explicit unpermissioned correlation with other second parties or third parties.
    malicious use in violation of contract

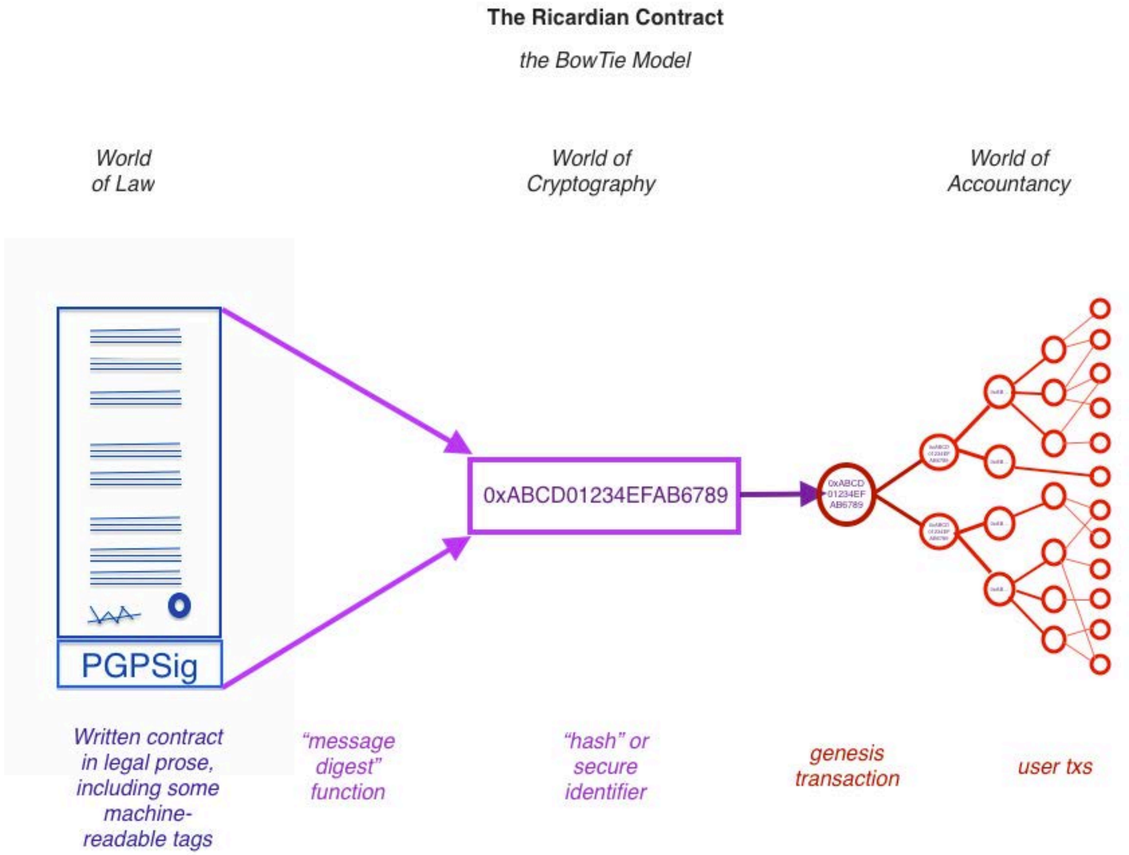
Third-Party (Observer) Exploitation
implicit permissioned correlation.
    no contractual restrictions on use of observed data.
explicit unpermissioned correlation via collusion with second parties.
    malicious use in violation of second party contract

Contractually Protected Disclosure

Ricardian Contracts:
https://en.wikipedia.org/wiki/Ricardian_contract

Chain-link Confidentiality
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2045818

Consent, Waiver, Terms-of-use, Remuneration, etc.



Chain-Link Confidentiality

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2045818

A chain-link confidentiality regime contractually links the disclosure of information to obligations to protect that information as the information moves downstream.

The system focuses on the relationships not only between the discloser of information and the initial recipient but also between the initial recipient and subsequent recipients.

Through the use of contracts, this approach links recipients of information as in a chain, with each subsequent recipient bound by the same obligation to protect the information.

These chain contracts contain at least three kinds of terms:

1) obligations and restrictions on the use of the disclosed information;

2) requirements to bind future recipients to the same obligations and restrictions; and

3) requirements to perpetuate the contractual chain.

This approach creates a system for the permissible dissemination of information.

It protects Disclosers by ensuring that the recipient's obligation to safeguard information is extended to third parties.

Contractual Exchange

Discloser provides a non-repudiable Offer with verifiable metadata (sufficient partial disclosure) which includes any terms or restrictions on use.

Disclosee verifies Offer against composed schema and metadata adherence to desired data.

Disclosee provides non-repudiable Accept of terms that are contingent on compliant disclosure.

Discloser provides non-repudiable Disclosure with sufficient compliant detail.

Disclosee verifies Disclosure using decomposed schema and adherence of disclosed data to Offer.

Disclosee may now engage in permissioned use and carries liability as a deterrent against unpermissioned use.

IPEX: Issuance & Presentation Exchange protocol

Presentation Exchange:

An exchange that provides disclosure of one or more ACDCs between a *Discloser* and a *Disclosee*.

A presentation exchange is the process by which authenticatable information may be exchanged between two parties, namely, the Discloser and Disclosee.

ACDC:

Type of data as issuance concretely defined by the ACDC specification.

Discloser:

An ACDC in a disclosure *is disclosed by* the *Discloser*.

Disclosee:

An ACDC in a disclosure *is disclosed to* the *Disclosee*.

Issuer:

An ACDC *is issued by* the *Issuer*. The *Issuer* identifier (AID) appears in the top level of the ACDC.

Issuee:

An ACDC is optionally *issued to* the *Issuee*. When present, the *Issuee* identifier (AID) appears at the top level of the attribute section or in the attribute list at the top level of the attribute aggregate section of the ACDC.

Each ACDC MUST have an *Issuer* and MAY have an *Issuee*

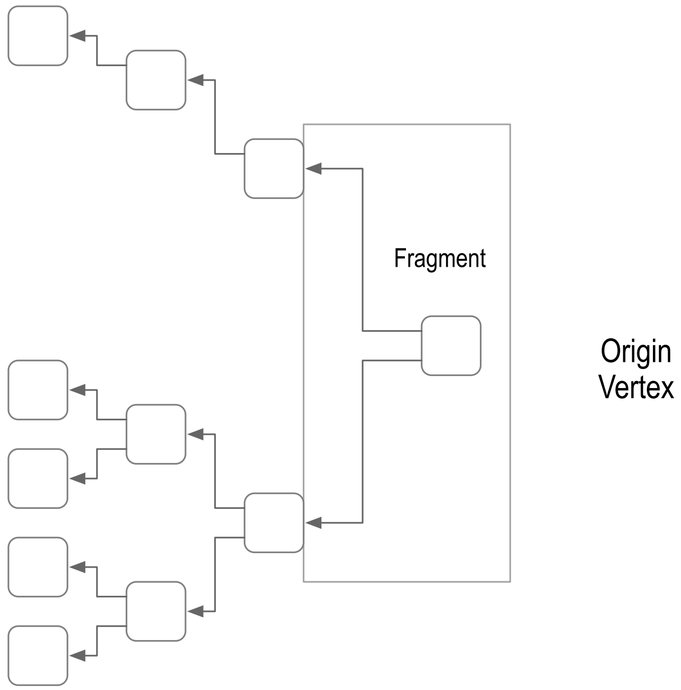
IPEX: Issuance & Presentation Exchange protocol

The set of ACDCs so disclosed in a presentation exchange MUST be chained.

This set of chained ACDCs define a directed acyclic graph (DAG) that MUST have at least one vertex and MAY have zero or more edges pointing to other vertices.

Each ACDC itself defines a graph fragment consisting of one vertex and zero or more directed edges.

Each directed edge contained in an ACDC points to a vertex contained in another ACDC. The ACDC that contains the origin vertex of the DAG is called the *origin* or *primary* ACDC of the presentation exchange.



IPEX: Issuance & Presentation Exchange protocol

The disclosure performed by a presentation exchange MAY be graduated and/or MAY be contractually protected.

Issuance Exchange:

A special case of a presentation exchange where the Discloser is the *Issuer* of the *origin* (*Primary*) ACDC of the DAG formed by the set of chained ACDCs so disclosed.

In an issuance exchange, when the *origin* ACDC has an Issuee, the *Disclosee* MAY also be the *origin* (*Primary*) ACDC's *Issuee*.

The Issuer MUST provide a signature on the SAID of the most compact variant defined by the schema of the ACDC. When more than one variant is defined by the schema via the oneOf composition operator for any top-level field, the most compact variant MUST appear as the first entry in the oneOf list. When only one variant of each top-level field is defined by the schema, that variant is therefore by definition the most compact variant.

ACDC analogy to Merkle Tree

The different variants of an ACDC form a hash tree (using SAIDs) that is analogous to a Merkle Tree.

Signing the top-level SAID of the compact version of the ACDC is equivalent to signing the Merkle Root of a Merkle Tree.

Different variants of an ACDC (SADs with SAIDs) correspond to different paths through a Merkle tree.

The process of verifying that a SAD via its SAID of a section is included in a schema authorized variant down from the top-level SAID is equivalent to a Merkle Tree proof of inclusion along a path in the Merkel Tree down from its Root.

This allows a single signature to provide proof of Issuance of the presentation of any schema authorized variants of the ACDC.

Proof-of-Issuance Proof-of-Disclosure

An Issuer MAY provide signatures of the SAIDS of other variants, as well as signatures of the SADs of other variants.

Proof of issuance is provided by disclosing the SAID of the most compact variant and a reference to the SEAL anchoring that SAID in either the KEL or TEL of the issuer.

Proof of disclosure is provided by disclosing the SAD of the most compact variant and then recursively disclosing the nested SADs of each of the top level sections of the most compact variant as needed for the promised disclosure.

Thus for any and all disclosed variants of an ACDC, the Disclosee need only verify the same proof of issuance as defined above and may need to verify the specific proof of disclosure for the given disclosed variant as defined above.

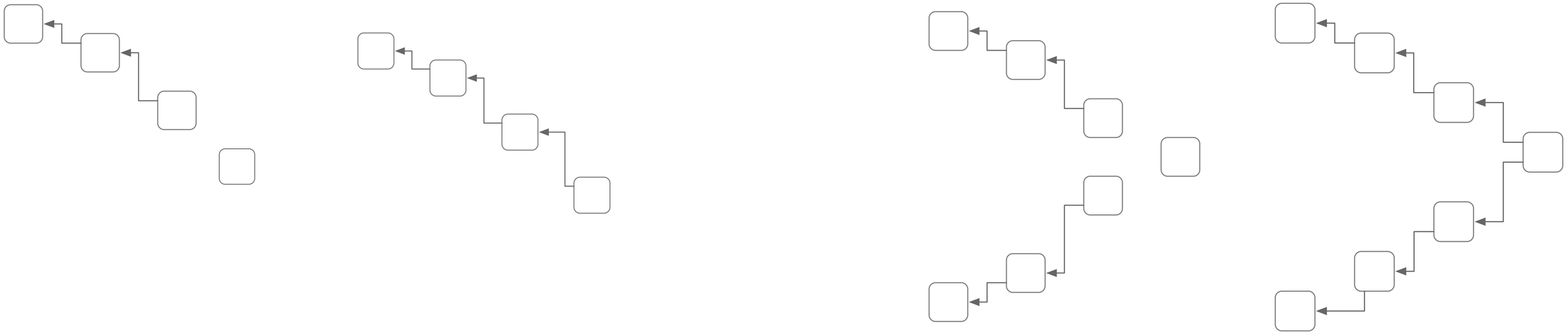
Append-to-Extend

Append-only verifiable data structures have strong security properties

Append-only simplifies end-verification

Append-only provides permission-less extensibility

Interoperable permissioned name-space registries are no longer needed with append-to-extend



Chain-Link Confidentiality

Disclosures via Presentations Exchanges may be contractually protected by Chain-Link Confidentiality (i.e a Chain-Link Confidential disclosure).

The chaining in this case is different from the chaining described above between Issuances in a DAG of chained Issuances. Chain-link confidentiality, in contrast, chains together a sequence of Disclosees.

Each Disclosee in the sequence in turn is the Discloser to the next Disclosee.

The terms-of-use of the original disclosure as applied to the original Disclosee MUST be applied by each subsequent Discloser to each subsequent Disclosee via each of the subsequent disclosures (presentation exchanges).

These terms-of-use typically constrain disclosure to only approved parties, i.e. imbue the chain of disclosures with some degree of confidentiality. These terms-of-use are meant to contractually protect the data rights of the original Issuer or Issuee of the data being disclosed.

Endorsers vs. Issuers

The ACDC Issuer is the securely attributable source of the ACDC. (Secure Attribution) that is designated in the Issuer field of the ACDC. The Issuer AID may be controlled by by set of controllers given multiple key-pairs with a threshold. Proof of Issuance requires the threshold be met.

An ACDC Endorser merely lends credibility (reputation) to an ACDC by signing the ACDC. The Endorser has its own AID with its own set set of controllers. The Endorser AID is not included in the ACDC.

Essentially a disclosure by the Discloser of an ACDC where the discloser is not the Issuer makes the Discloser by virtue of signing the disclosure, a type of Endorser. However in such a disclosure the purpose of the signing by the Discloser may be more than an endorsement such as to make a commitment with respect to a contractually protected disclosure.

When the Discloser is the Issuee then the signing of the Disclosure is not merely an endorsement but provides a temporal proof of control over the Issuee AID and could satisfy a “live” presentation requirement by the Disclosee.

Uncompacted Public Attribute Section

```
{
  "a":
  {
    "d": "EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "i": "did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA",
    "score": 96,
    "name": "Jane Doe"
  }
}
```

Public Compact Variant

```
{
  "v":  "ACDC10JSON00011c_",
  "d":  "EAdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "i":  "did:keri:EBkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:ECmRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":  "ED6jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":  "EEveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":  "EFH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":  "EG71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB",
}
```

Composed JSON Schema

```
{
  "a":
  {
    "description": "attribute section",
    "oneOf":
    [
      {
        "description": "attribute SAID",
        "type": "string"
      },
      {
        "description": "uncompacted attribute section",
        "type": "object",
        "required":
        [
          "d",
          "i",
          "score",
          "name"
        ],
        "properties":
        {
          "d":
          {
            "description": "attribute SAID",
            "type": "string"
          },
          "i":
          {
            "description": "Issuee AID",
            "type": "string"
          },
          "score":
          {
            "description": "test score",
            "type": "integer"
          },
          "name":
          {
            "description": "test taker full name",
            "type": "string"
          }
        },
        "additionalProperties": false
      }
    ]
  }
}
```

Attribute Section


```
{
  "a":
  {
    "d": "EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "u": "0AwjaDAE0qHcgNghkDaG7OY1",
    "i": "did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA",
    "score": 96,
    "name": "Jane Doe"
  }
}
```

Edge Section


```
{
  "e":
  {
    "d": "EerzwLIr9Bf7V_NHwY1lkFrn9y2PgveY4-9XgOcLxUdY",
    "boss":
    {
      "d": "E9y2PgveY4-9XgOcLxUdYerzwLIr9Bf7V_NHwY1lkFrn",
      "n": "EIl3MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA",
      "w": "high"
    }
  }
}
```

Latent Accountability

Escrow
- KYC
- Contingent Enforcement and Recourse

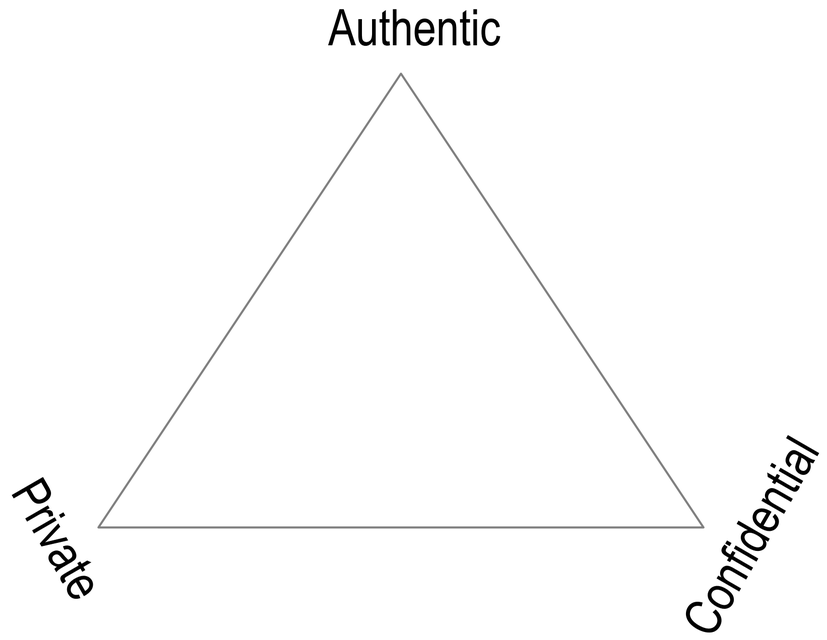
Bonding

Bounties

Example Use Cases

PAC Theorem

A conversation may be two of the three, *private*, *authentic*, and *confidential* to the same degree, but not all three at the same degree.



Trade-offs required!

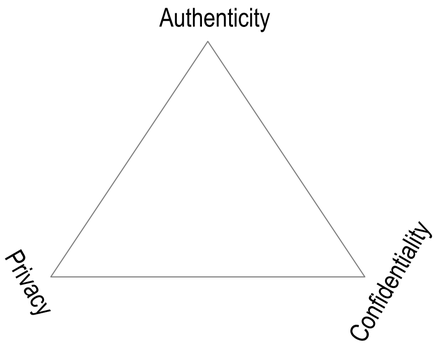
Definitions

Private:
The parties to a conversation are only known by the parties to that conversation.
Authentic:
The origin and content of any statement by a party to a conversation is provable to any other party.
Confidential:
All statements in a conversation are only known by the parties to that conversation.

Privacy:
about control over the disclosure of who participated is in the conversation (non-content meta-data)
Authenticity:
about proving who said what in the conversation (secure attribution)
Confidentiality:
about control over the disclosure of what was said in the conversation (content data)

Relatively weak legal protection for non-content (supoena)
Relatively strong legal protection for content (search warrant)

https://www.lawfareblog.com/relative-vs-absolute-approaches-contentmetadata-line
https://www.pogo.org/analysis/2019/06/the-history-and-future-of-mass-metadata-surveillance/



Proving Authenticity

Non-repudiable Proof:

a statement's author cannot successfully dispute its authorship

Asymmetric key-pair digital signature

Repudiable Proof:

a statement's author can successfully dispute its authorship

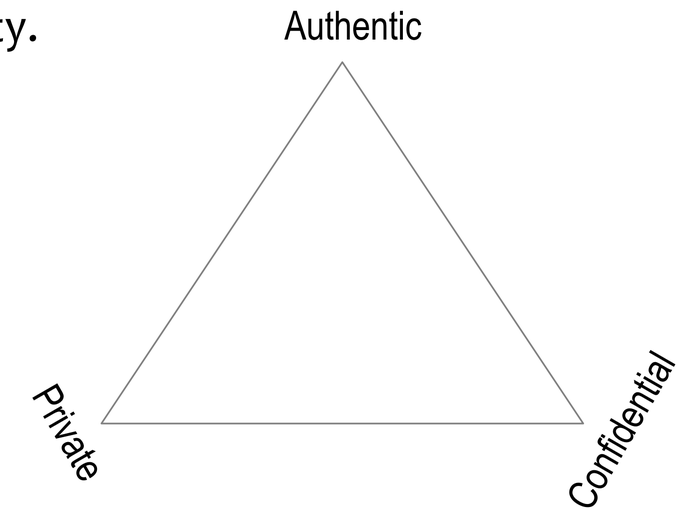
DH shared symmetric key-pair encryption (auth crypt)

Shared secret makes every verifier a potential forger

Trade-offs

Private:
The parties to a conversation are only known by the parties to that conversation.
Authentic:
The origin and content of any statement by a party to a conversation is provable to any other party.
Confidential:
All statements in a conversation are only known by the parties to that conversation.

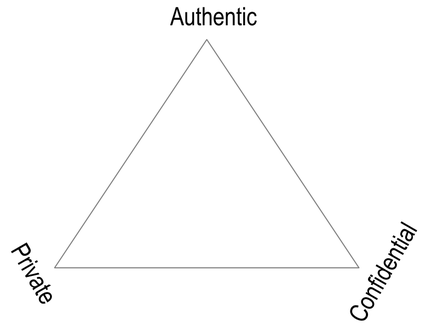
Non-repudiation means any party to conversation can proof to any other party exactly what was said by whom.
This means that technologically there is no way to prevent disclosure by any party to some third party.
We can incentivize confidentiality by imposing a liability on the parties to the disclosure set before disclosure occurs.
Enforcement of that liability will usually necessarily violate privacy but not confidentiality.
Real world value often requires transitivity.
Transitive value transfer will violate complete privacy.



Layering

A communication system can layer the different properties in different orders thereby imposing a priority on each property.

Authenticity
Confidentiality
Privacy



Why Solve the Secure Attribution Problem

Secure attribution of any communication to its source

Authentic communication

Authentic interactions based on secure attribution of all statements by participants:

- Data Provenance (Verifiable authenticity of data
- Data Supply Chains (Authentic data economy)

Non-Repudiable Authenticity

Zoe

Signed with private key

Verified with public key

Sue

Non-repudiable authenticity is *zero-trust*

Repudiable Authenticity

Zoe

Encrypted with
shared private key

Decrypted with
shared private key

Sue

Repudiable authenticity requires trust (is not zero-trust)

Non-Repudiable Authenticity Is Legally Binding.
Repudiable Authenticity Is Not Legally Binding.

Zoe

Encrypted with
shared private key

Decrypted with
shared private key

Sue

Non-Repudiable authenticity has recourse.

Best fits current business and regulatory eco-systems.

Zero Knowledge Proof?

Zoe

Authentic ZKP: Is the information proven in a repudiable or non-repudiable manner?

ZKP

Sue

one party (the prover) can prove to another party (the verifier) that a given statement is true,
without conveying any information apart from the fact that the statement is indeed true.

ZKPoK (selective disclosure)

Zoe proves to Sue that a given statement is true without enabling Sue to prove to a third party that the statement is true and protecting Alice from Sue as a forger.

(plausible deniability, repudiability with forgery protection)

ZKP (selective disclosure)

Zoe proves an element of a bundle of information to Sue without disclosing any other element of the bundle

(non-repudiable or repudiable but if repudiable may not be protected from forgery)

Other “non-ZKP” like methods can perform a non-repudiable selective disclosure

Strong

Weak

Privacy?

Weak

Strong Privacy

Definition: un-correlated interactions over unbounded time and space.

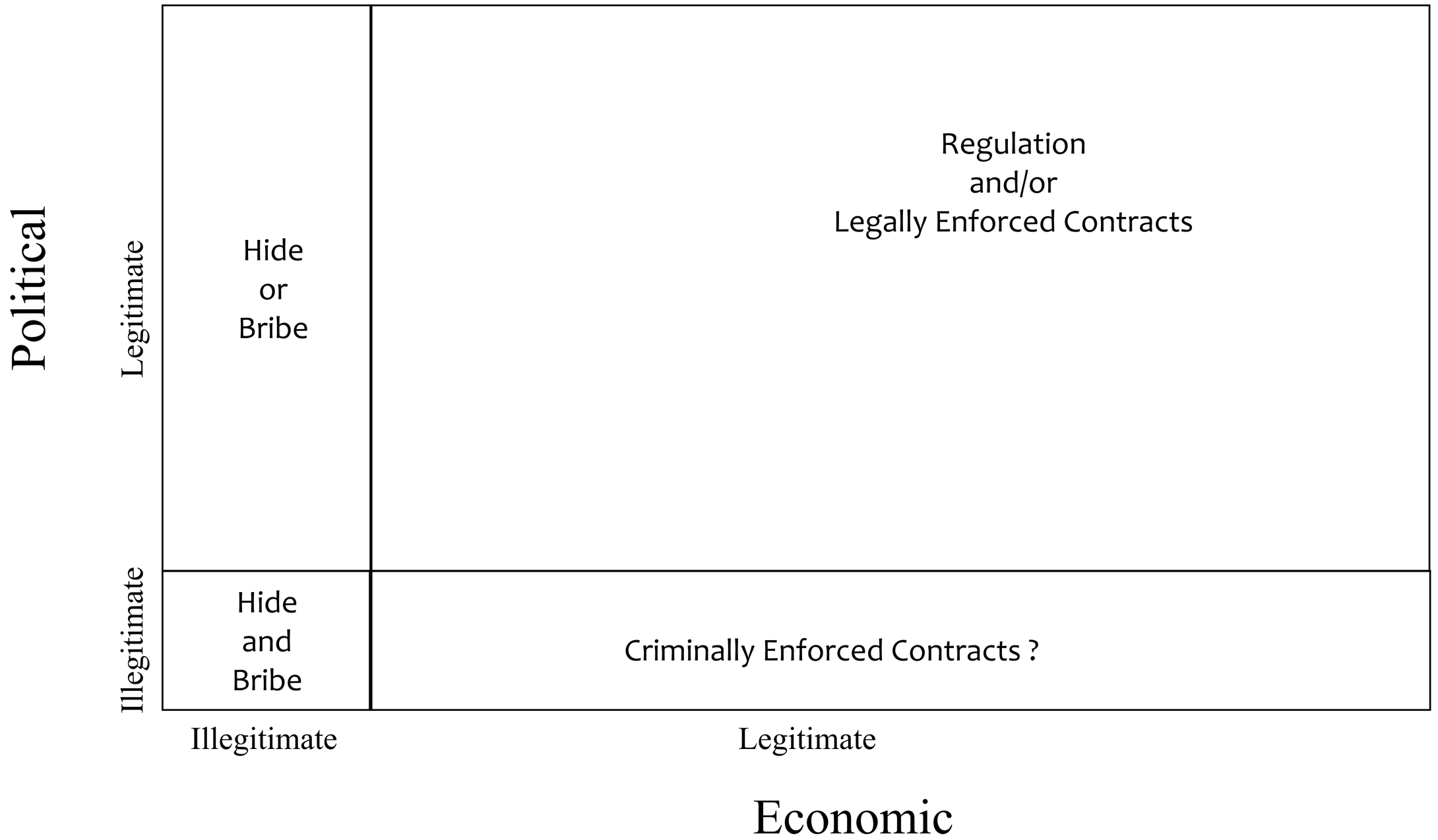
Super aggregators and state actors have effectively unlimited storage and compute capacity. Eventually all disclosed data will be at least statistically correlatable.

Weak Privacy

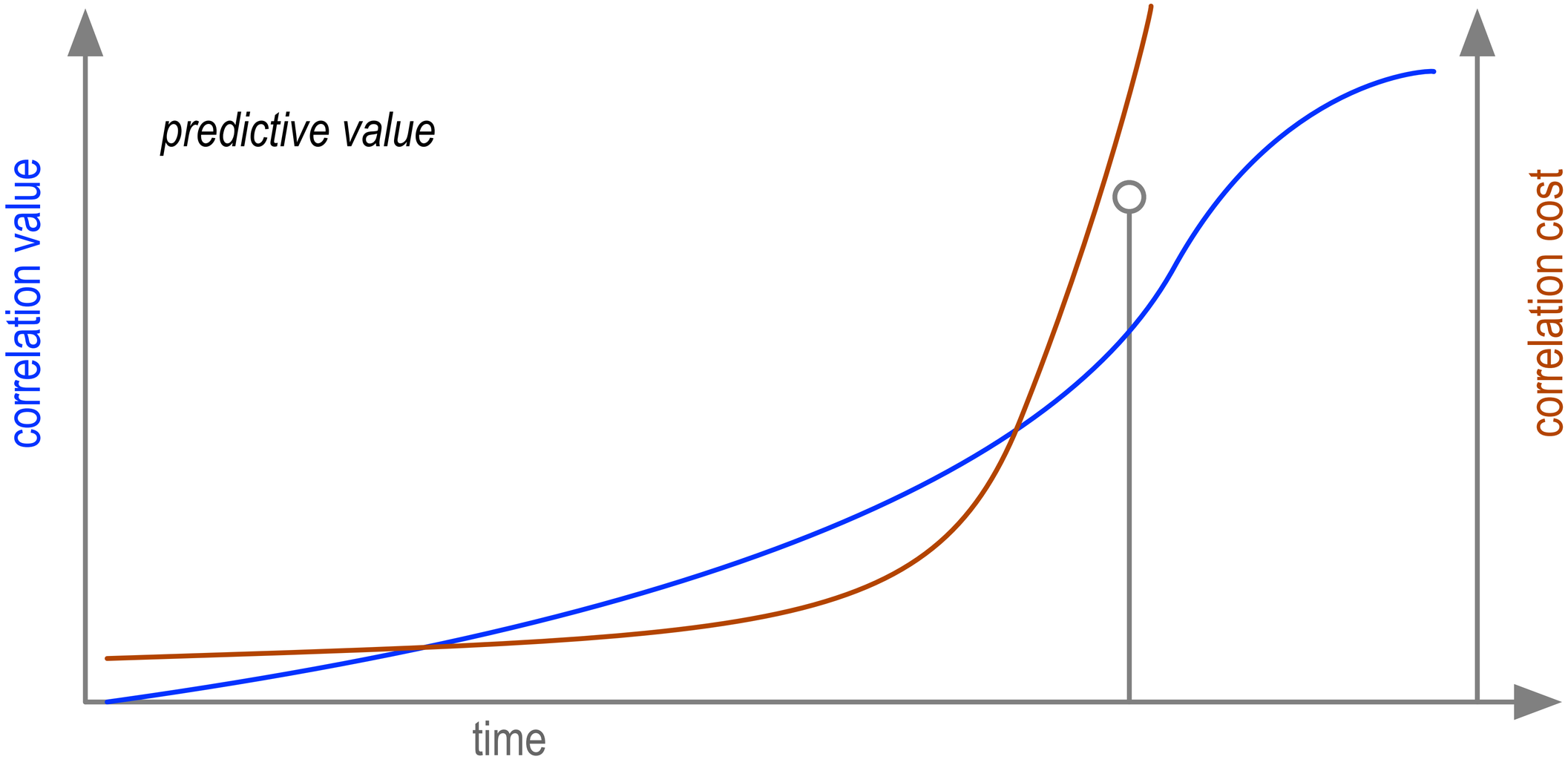
Definition: un-correlated interactions over bounded time and space.

when the cost of correlation exceeds the value of correlation the data will be un-correlated.

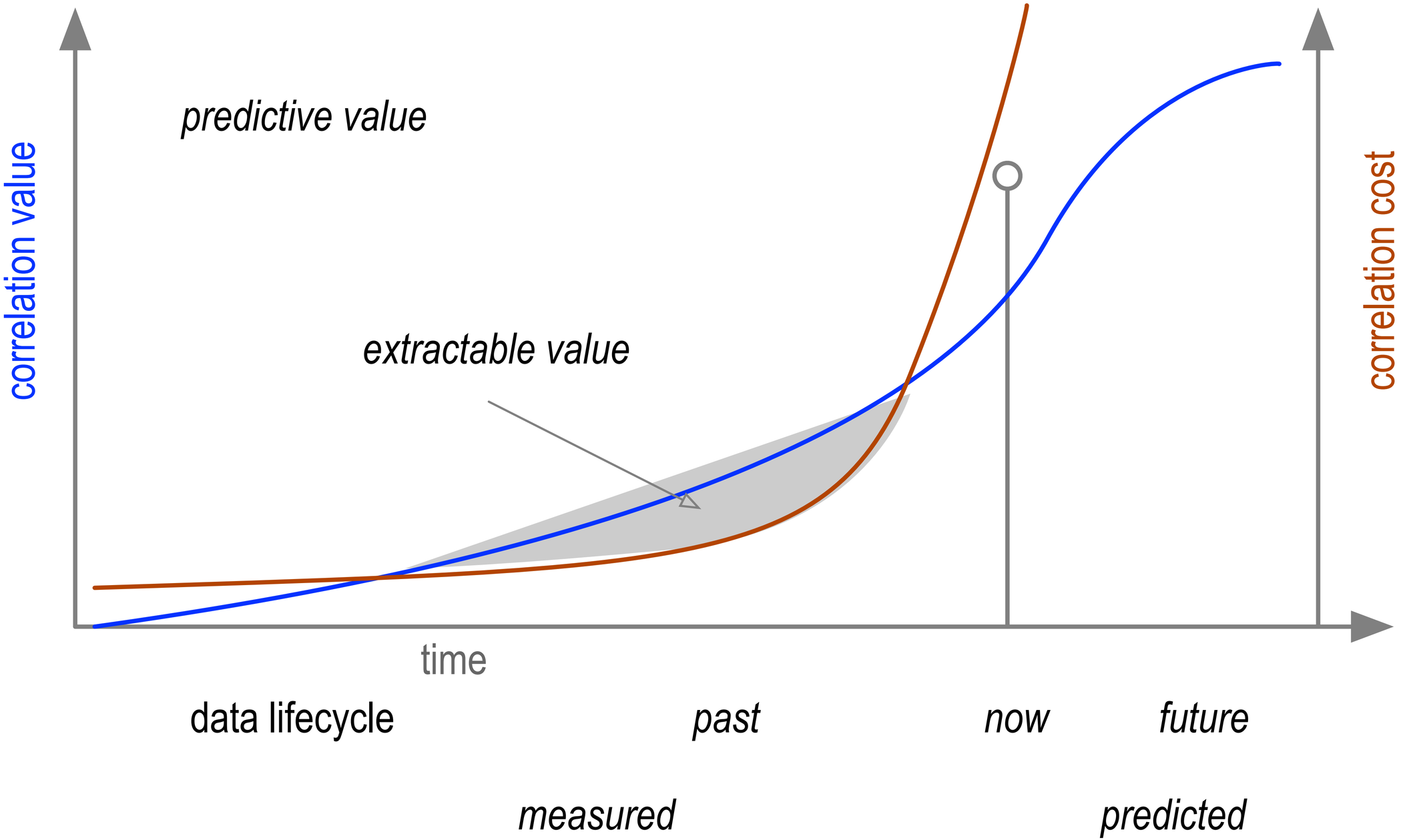
Operating Regimes



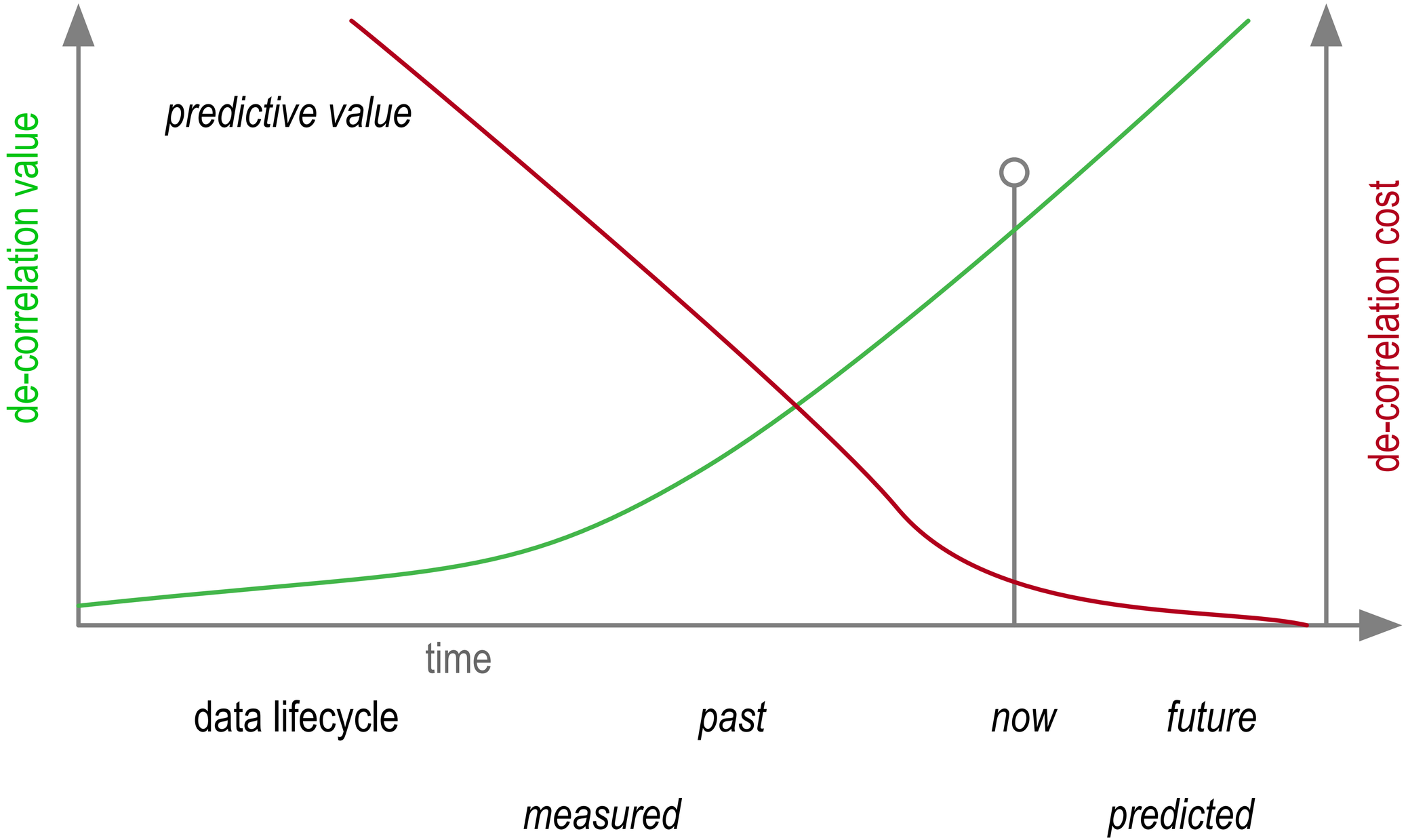
Economics of Correlator



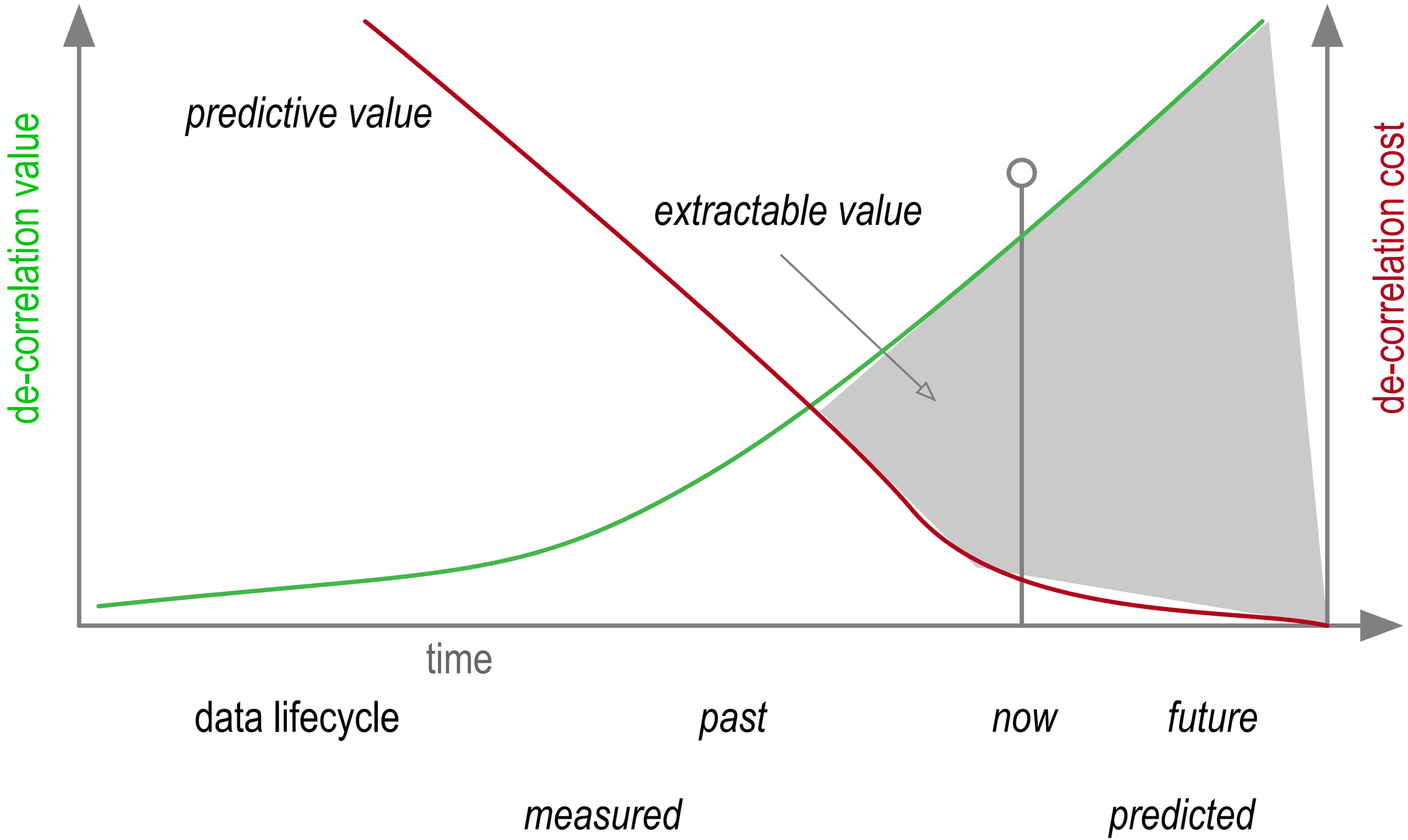
Economics of Correlator: Value Extraction



Economics of De-correlator



Economics of De-correlator: Value Extraction



Freedom

balanced

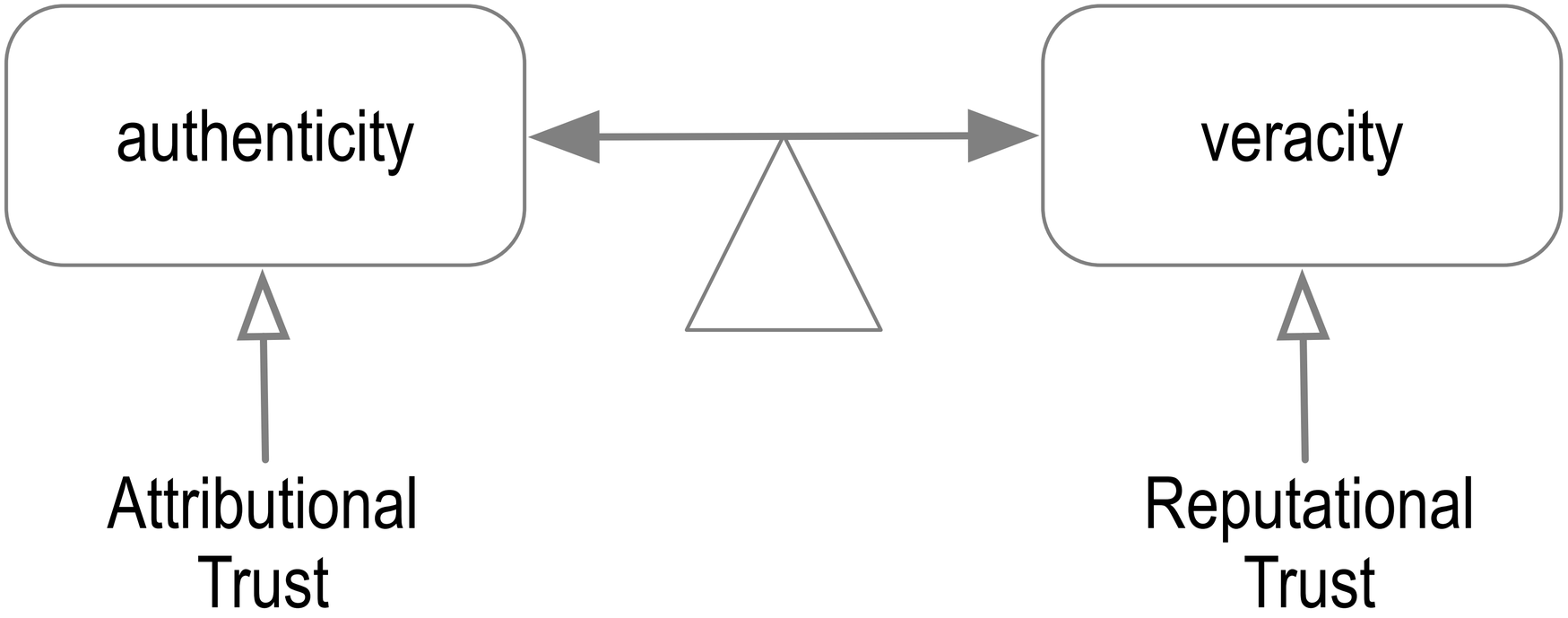
Freedom from …	Freedom to …
exploitation (commercial)	extract value(commercial)
intimidation (political)	build relationships (social)
censorship (political)	build community (political)

possibility of erasure = possibility of censorship

anonymity = loss-of-value from attribution

fairness = loss of privacy from attribution

Trust Balance



Cryptographic

Behavioral

Unified Identifier Model

AID: Autonomic Identifier (primary)

self-managing self-certifying identifier with cryptographic root of trust

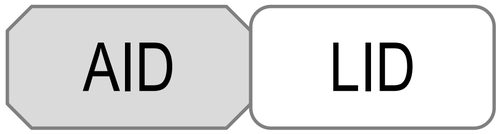
secure, decentralized, portable, universally unique

L*ID*: Legitimized Human Meaningful Identifier (secondary)

legitimized within trust domain of given AID by a verifiable authorization from AID controller

authorization is verifiable to the root-of-trust of AID

Forms *AID*|*LID* couplet within trust domain of AID



AID|LID Couplet

625.127C125r

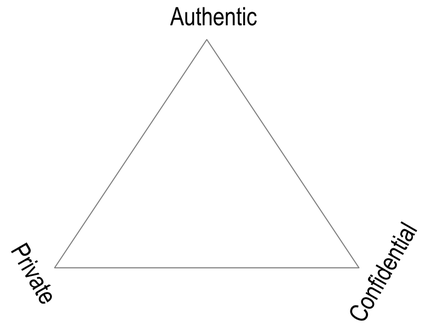
EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148|625.127C125r

Background

BADA (Best Available Data Acceptance) Policy

Authentic Data:
- Two primary attacks:
  - Replay attack:
    - Mitigation: Monotonicity
  - Deletion attack:
    - Mitigation: Redundancy

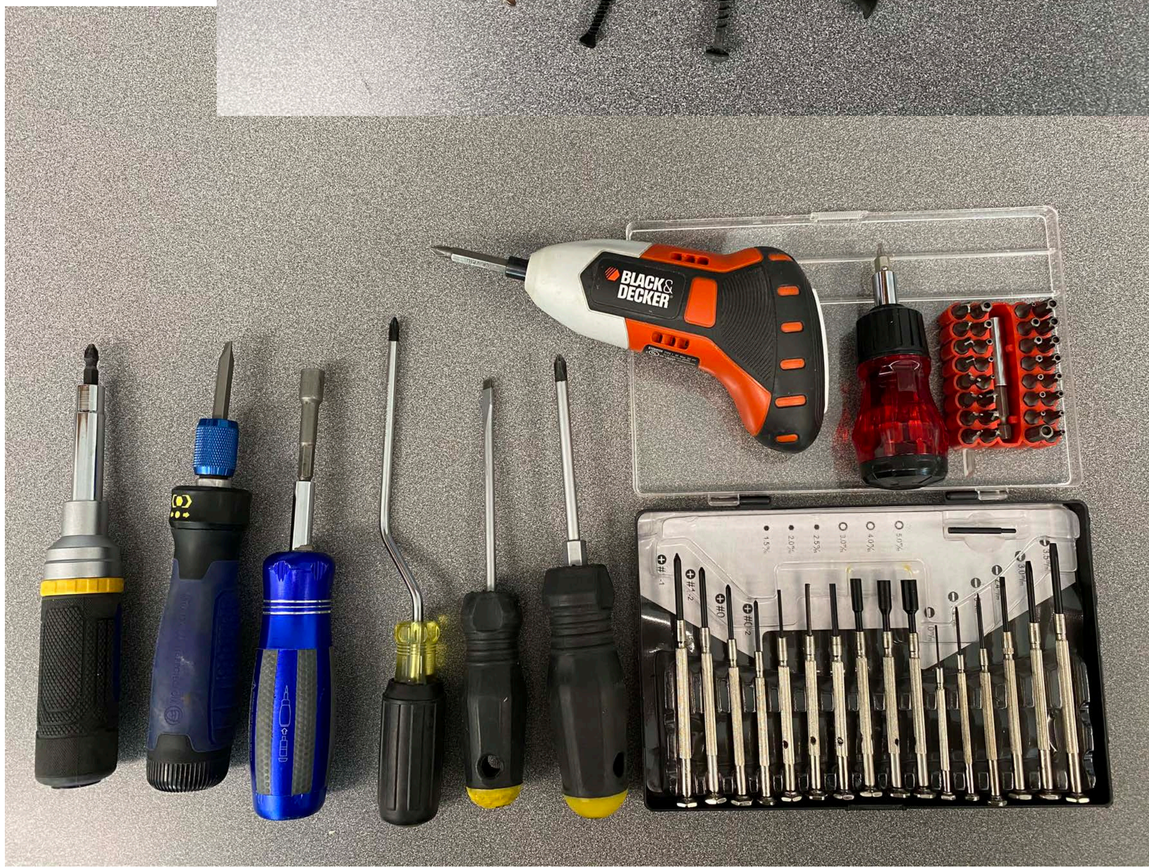
Replay Monotonicity:
- Interactive:
  - Nonce
- Non-interactive:
  - Memory (sequence number, date-time stamp, nullification)
  - More scalable



RUN off the CRUD

Client-Server API or Peer-to-Peer.
Create, Read, Update, Delete (CRUD)
Read, Update, Nullify (RUN)
Decentralized control means server never creates only client. Client (Peer) updates server (other Peer) always for data sourced by Client (Peer). So no Create.
Non-interactive monotonicity means we can't ever delete.
So no Delete. We must Nullify instead. Nullify is a special type of Update.
Ways to Nullify:
    null value
    flag indicating nullified
Rules for Update : (anchored to key state in KEL)
    Accept if no prior record.
    Accept if anchor is later than prior record.
Rules for Update: (signed by keys given by key state in KEL, ephemeral identifiers have constant key state)
    Accept if no prior record.
    Accept if key state is later than prior record.
    Accept if key state is the same and date-time stamp is later than prior record.

Toolkits

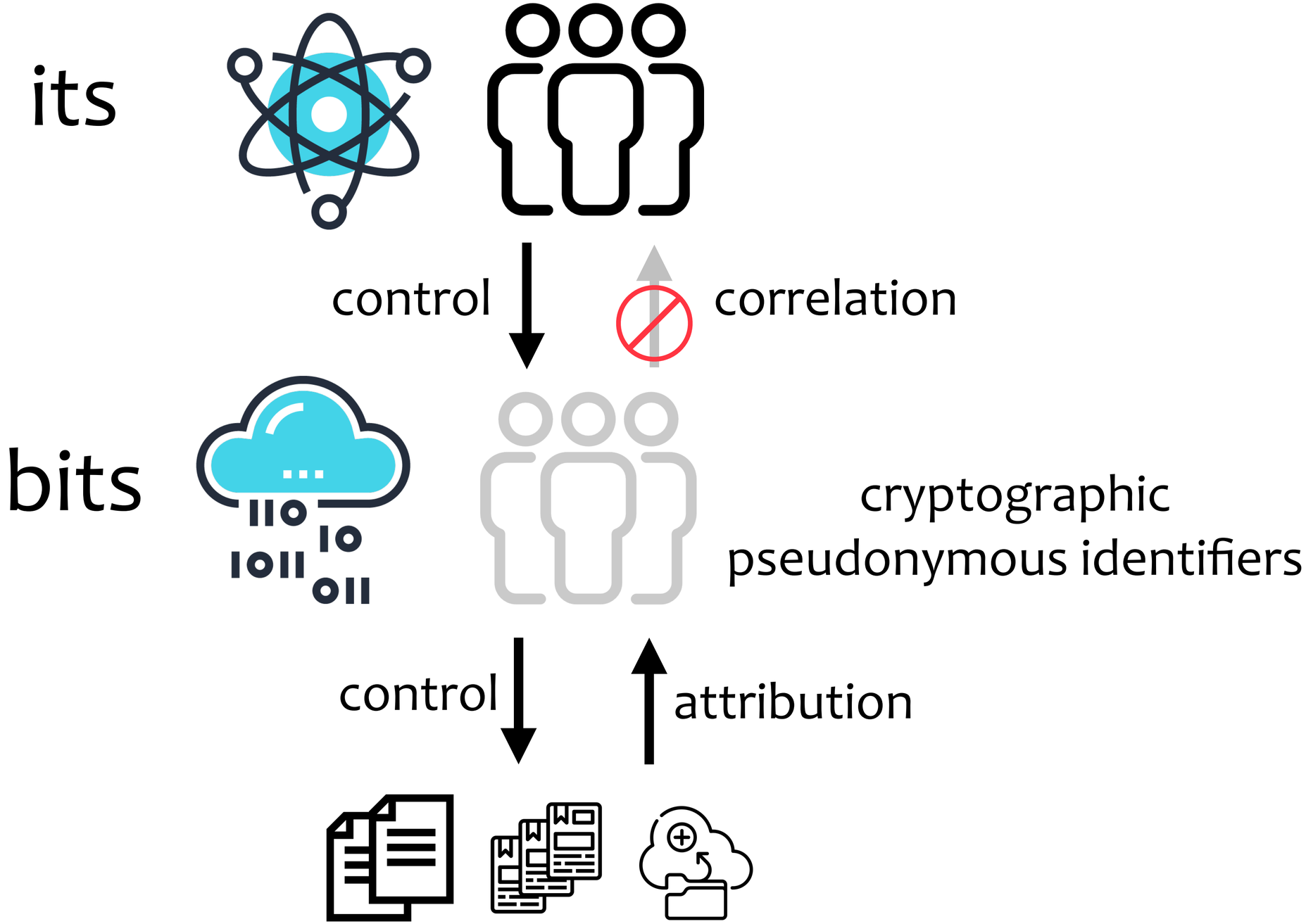


Only have one set of tools for truly secure data control!

Entropy Derived Tools

Cryptographic one-way functions …

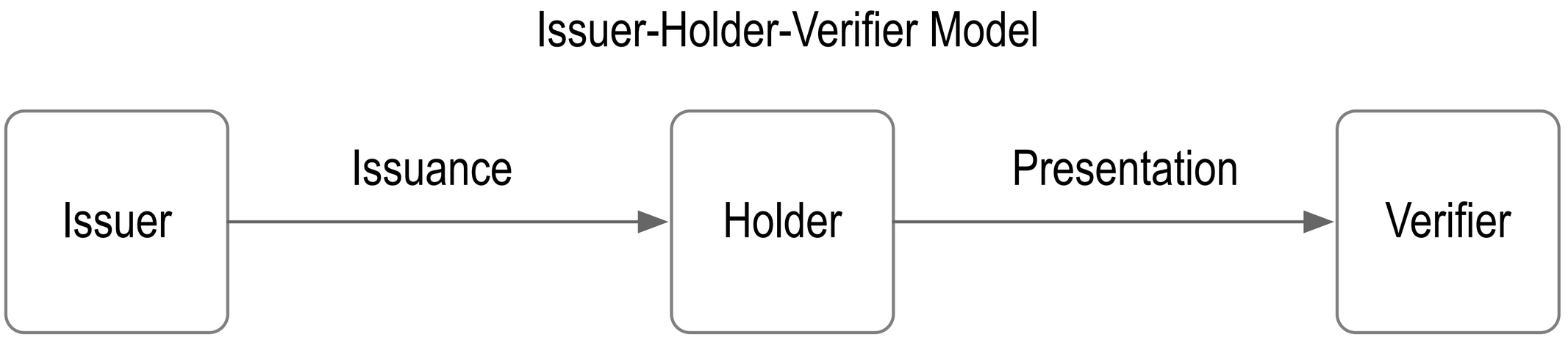
hashes, ECC scalar multiplication…
digital signatures, ZKPs …


its
control
correlation
bits
cryptographic
pseudonymous identifiers
control
attribution

Tripartite Authentic Data (VC) Model

Issuer: Source of the VC. Creates (issues) and signs VC
Holder: Usually the target of the VC. The holder is the "*issuee*" that receives the VC and holds it for its own use.
Verifier: Verifies the signatures on the VC and authenticates the holder at the time of presentation

The issuer and target each have a DID (decentralized identifier).
The DIDs are used to look-up the public key(s) needed to verify signatures.

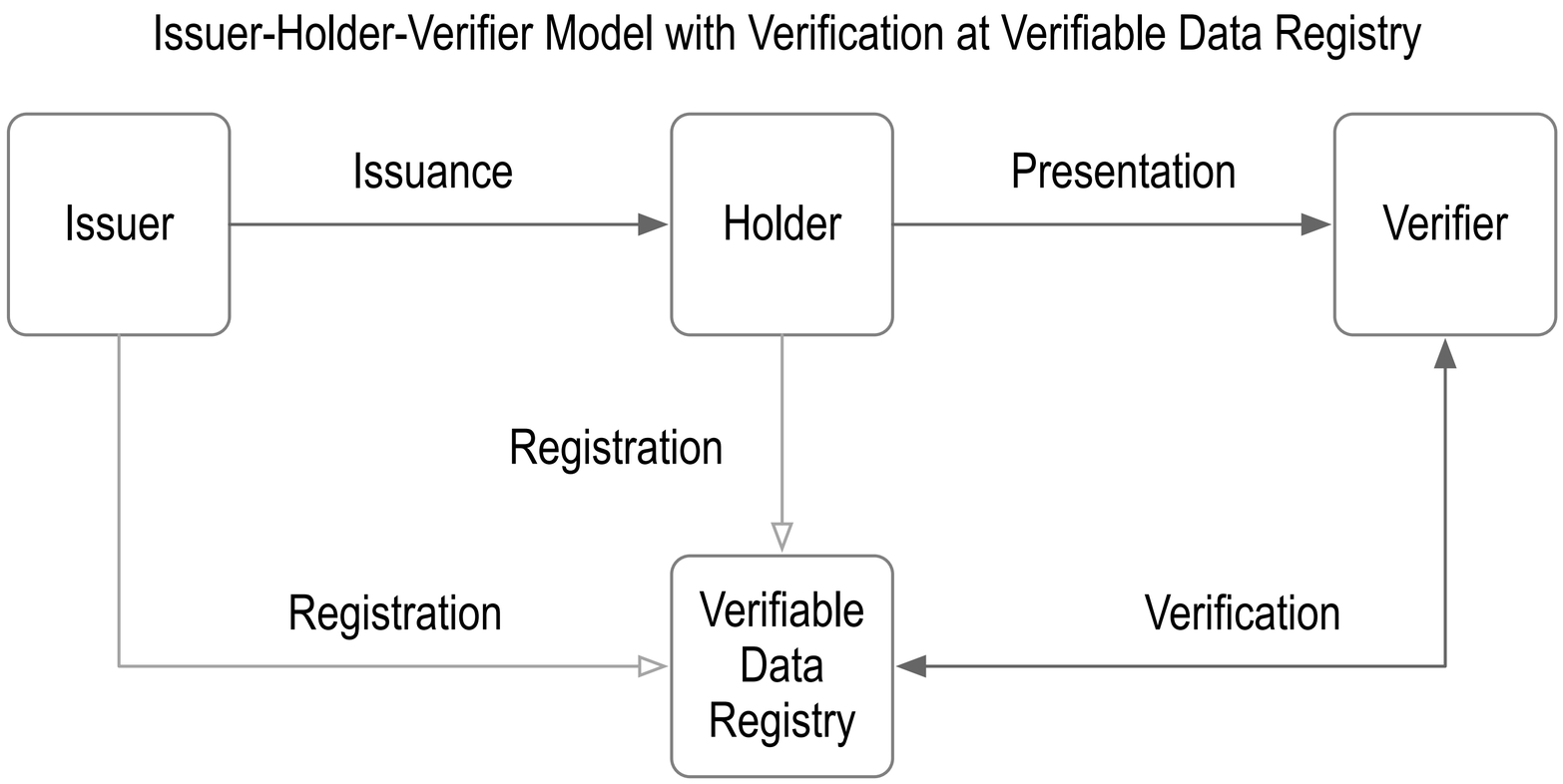


Tripartite Authentic Data (VC) Model with VDR

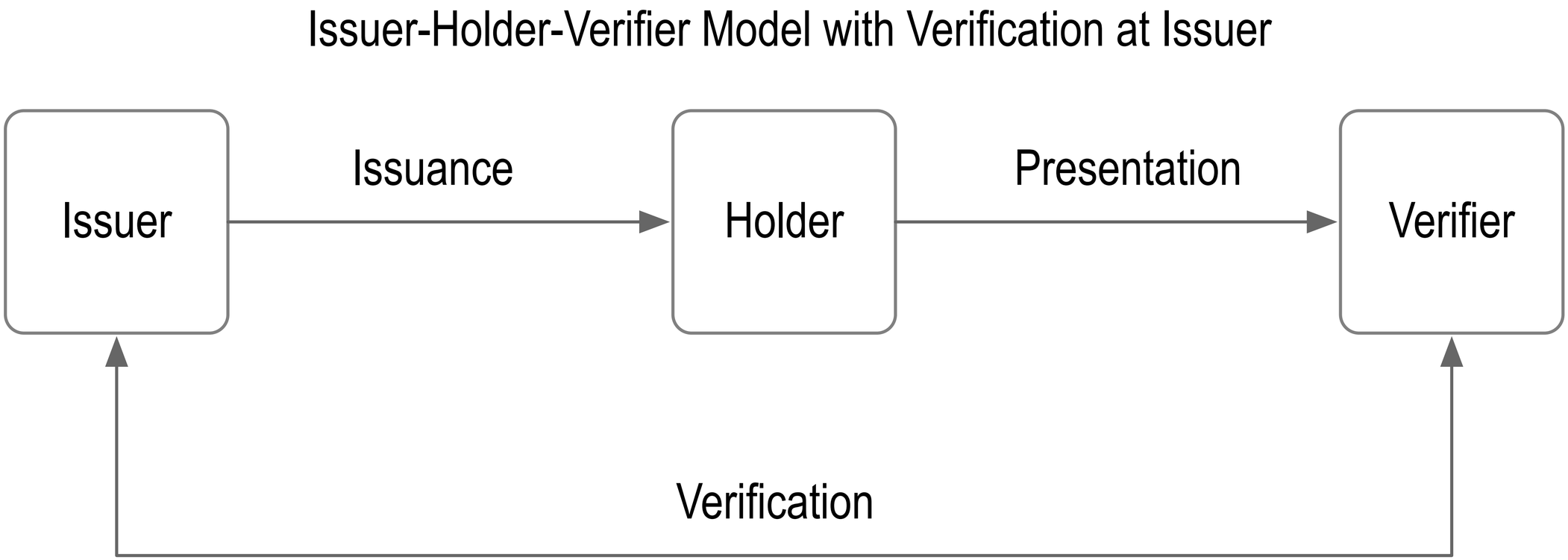
Verifiable Data Registry (VDR) enables decentralized but interoperable discovery and verification of authoritative key pairs for DIDs in order to verify the signatures on VCs. A VDR may also provide other information such as data schema or revocation state of a VC.

Each controller of a DID registers that DID on a VDR so that a verifier can determine the authoritative key pairs for any signatures.

We call this determination, *establishment of control authority* over a DID.

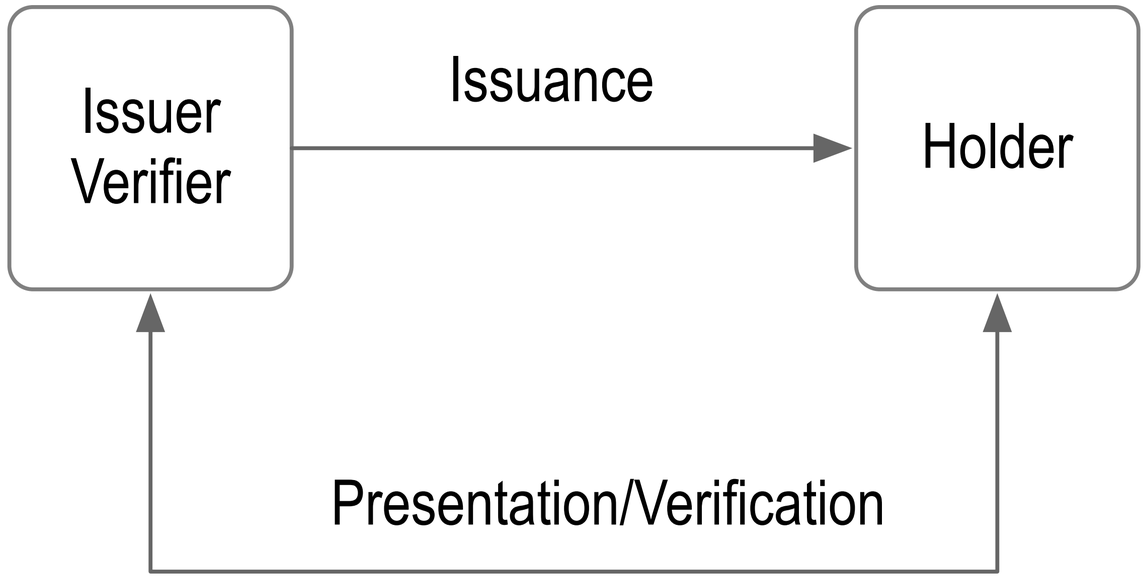


Tripartite without VDR

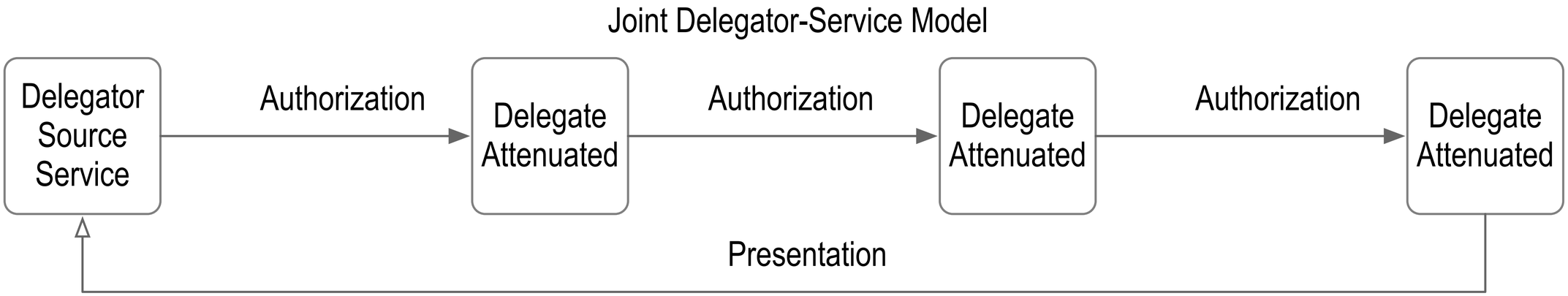


Bipartite Model

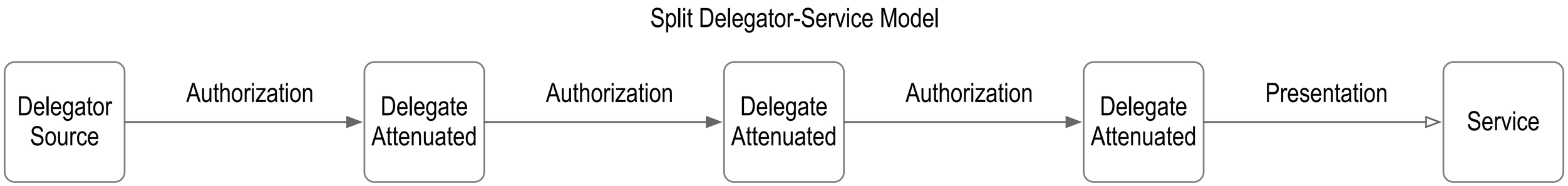
Issuer-Holder Model with Verification at Issuer



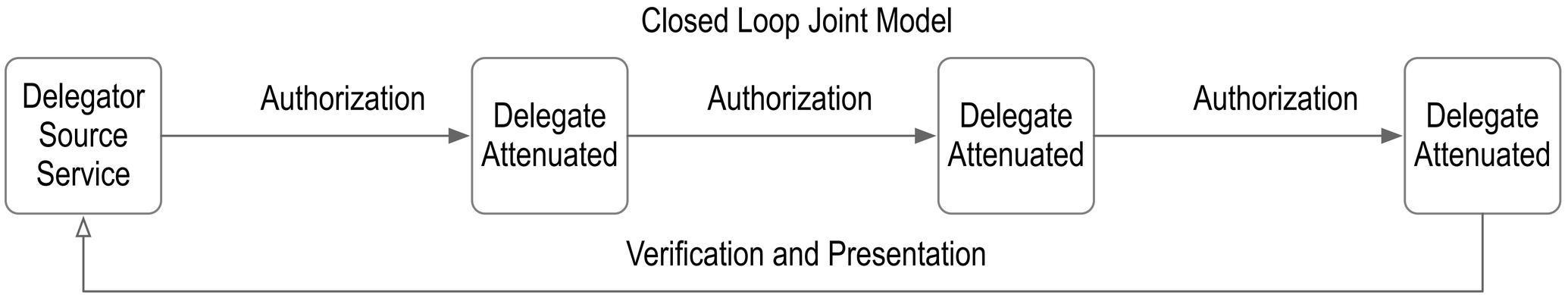
Joint Delegator-Service Model



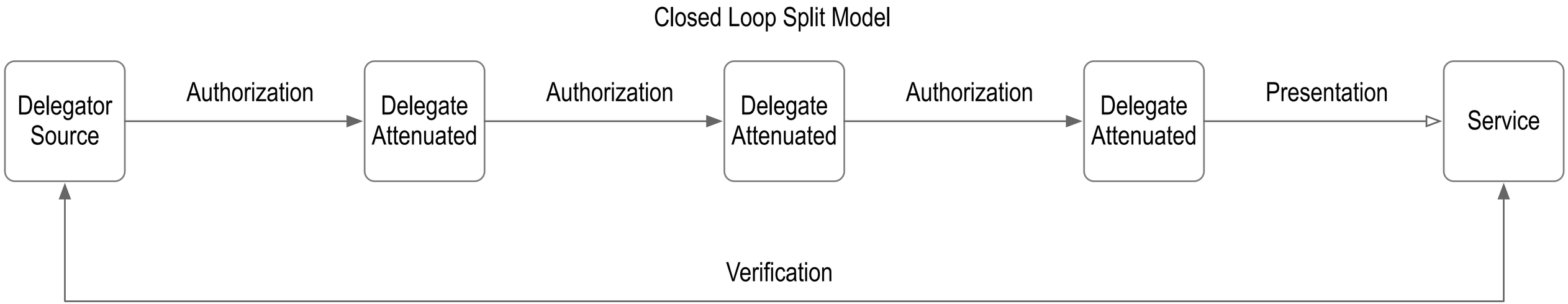
Split Delegator-Service Model



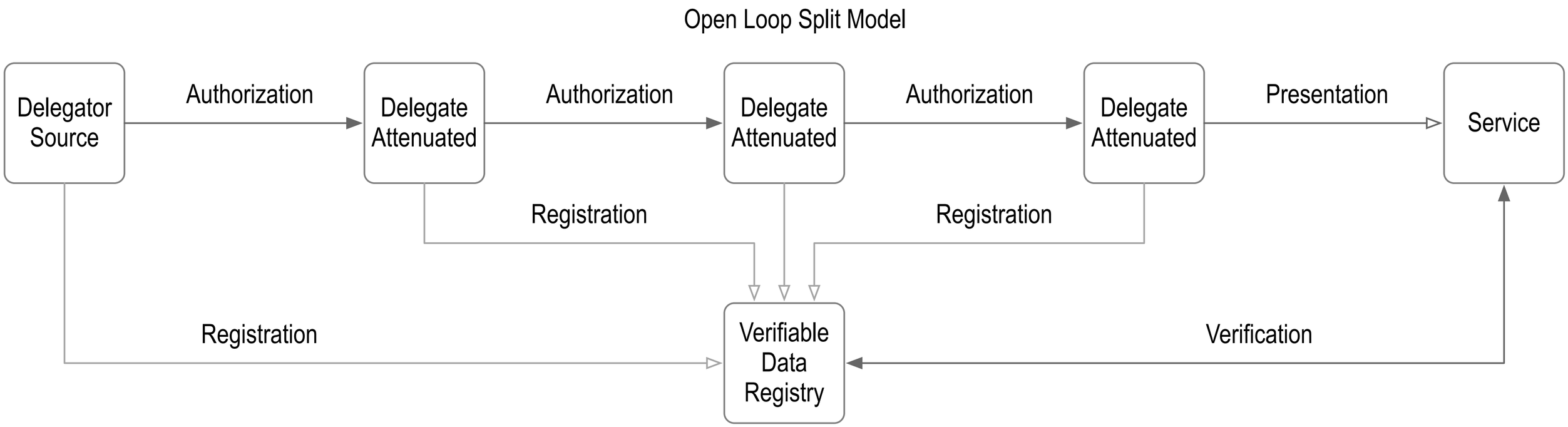
Closed Loop Joint Model



Closed Loop Split Model



Open Loop Split Model

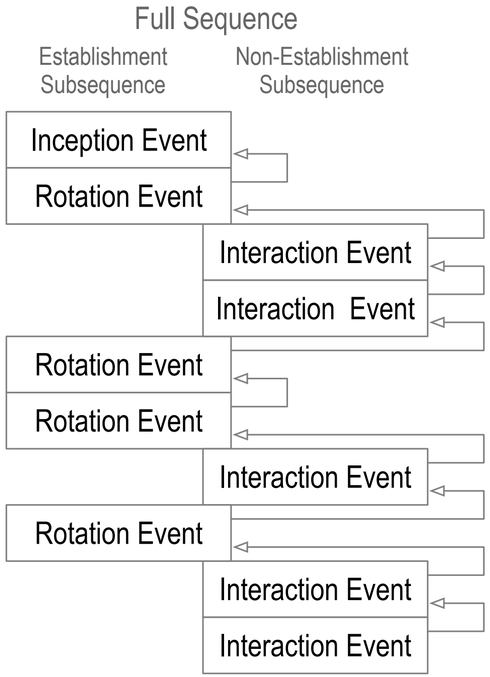


KERI VDRs vs. Shared Ledger VDRs

Most DID methods use a shared ledger (commonly referred to as a *blockchain*) for their VDR. Typically, in order to interoperate all participants must use the same shared ledger or support multiple different DID methods. There are currently over 70 DID methods. Instead GLEIF has chosen to use KERI based DID methods. KERI stands for Key Event Receipt Infrastructure. KERI based VDRs are ledger independent, i.e. not locked to a given ledger. This provides a path for greater interoperability without forcing participants in the vLEI ecosystem to use the same shared ledger.

A KERI VDR is called a key event log (KEL). It is a cryptographically verifiable signed hash chained data structure, a special class of verifiable data structure. Each KERI based identifier has its own dedicated KEL. The purpose of the KEL is to provide proof of the establishment of control authority over an identifier. This provides cryptographically verifiable proof of the current set of authoritative keys for the identifier. KERI identifiers are long cryptographic pseudo random strings of characters. They are self-certifying and self-managing.

A KERI identifier is abstractly called an Autonomic Identifier (AID) because it is self-certifying and self-managing. A KERI DID is one concrete implementation of a KERI AID. The same KERI prefix may control multiple different DIDs as long as they share the same prefix.



`did:keri:`*`prefix`*`[:`*`options`*`][/`*`path`*`][?`*`query`*`][#`*`fragment`*`]`

`did:keri:ENqFtH6_cfDg8riLZ-GDvDaCKVn6clOJa7ZXXVXSWpRY`

KERI Identifier KEL VDR *Controls* Verifiable Credential Registry TEL VDR

A KERI KEL for a given identifier provides proof of authoritative key state at each event. The events are ordered. This ordering may be used to order transactions on some other VDR such as a Verifiable Credential Registry by attaching anchoring seals to KEL events.

Seals include cryptographic digest of external transaction data.

A seal binds the key-state of the anchoring event to the transaction event data anchored by the seal.

The set of transaction events that determine the external registry state form a log called a Transaction Event Log (TEL).

Transactions are signed with the authoritative keys determined by the key state in the KEL with the transaction seal.

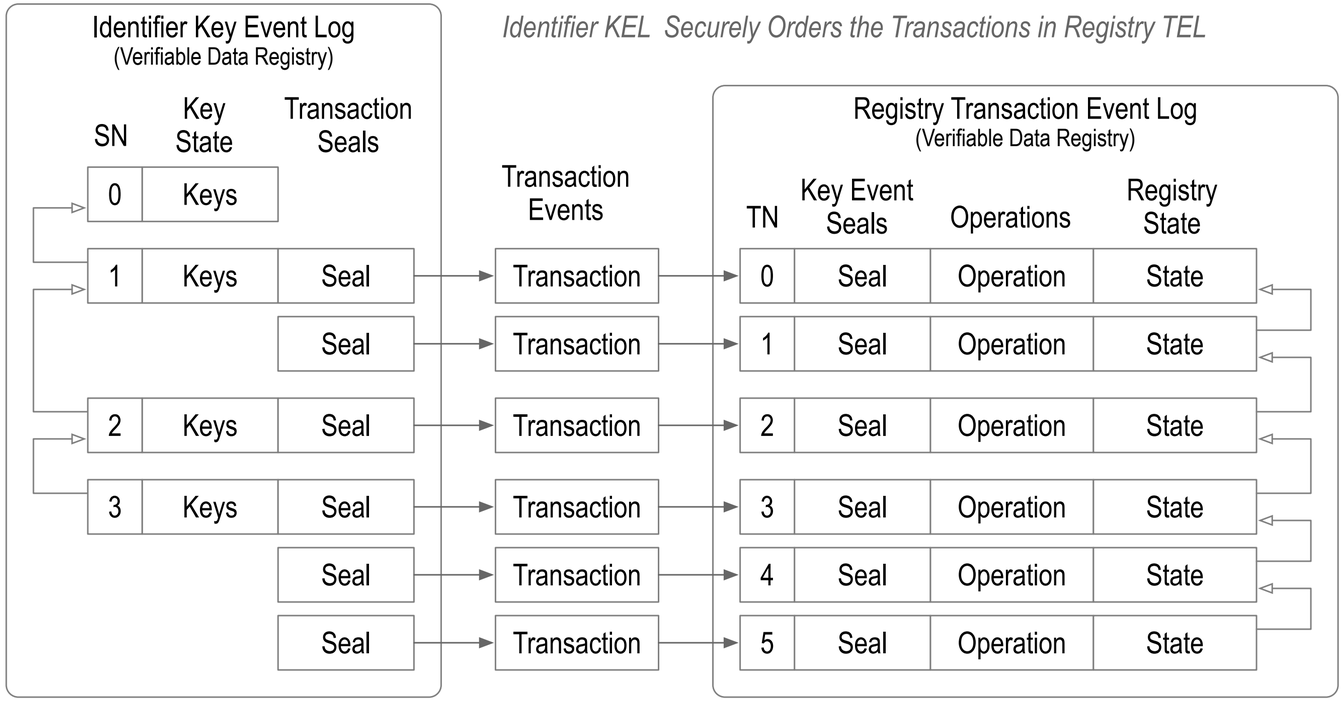
The transactions likewise contain a reference seal back to the key event authorizing the transaction.

This setup enables a KEL to control a TEL for any purpose. This includes what are commonly called “smart contracts”.

The TEL provides a cryptographic proof of registry state by reference to the corresponding controlling KEL.

Any validator may therefore cryptographically verify the authoritative state of the registry.

In the case of the vLEI the associated TEL controls a vLEI issuance and revocation registry.



seal = proof of authenticity

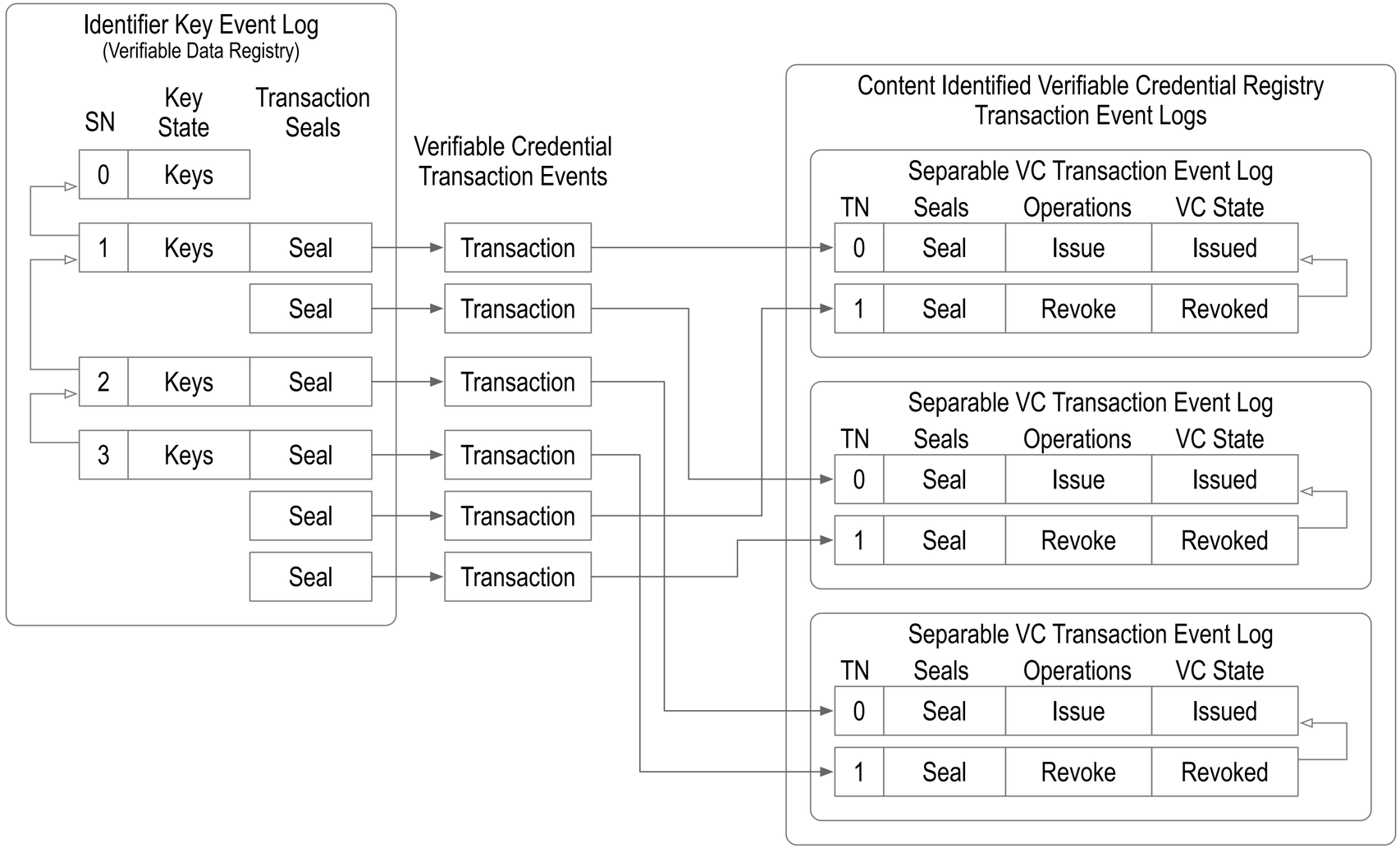
Registry with Separable VC Issuance-Revocation TELs

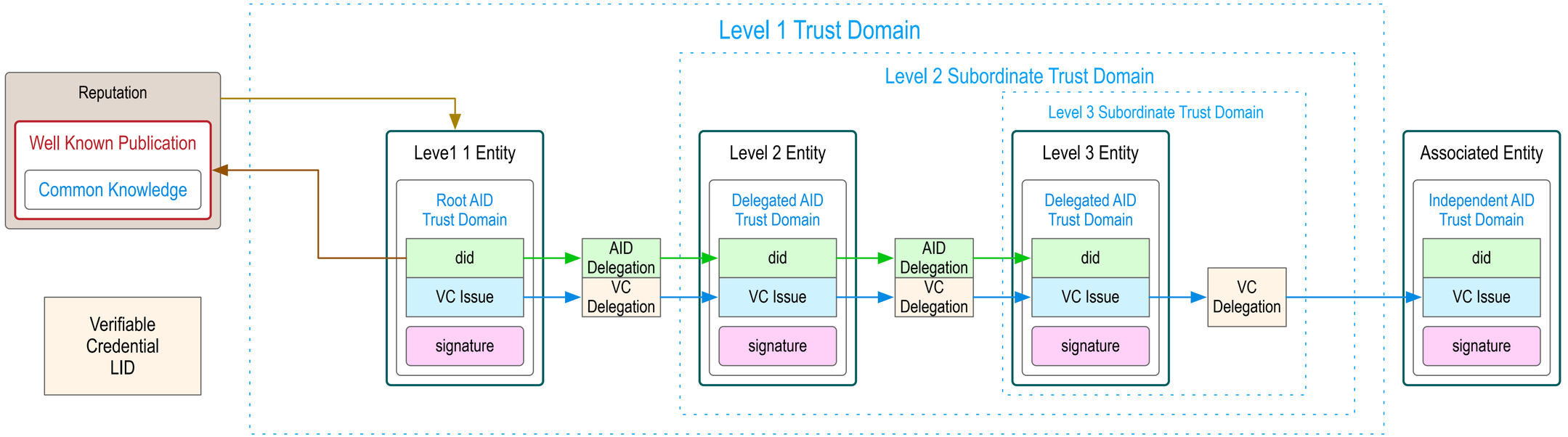
Each VC may be uniquely identified with a SAID.

Each VC also has a uniquely identified issuer using a KERI AID.

This combination enables a separable registry of VC issuance-revocation state.

The state may employ a cryptographic aggregation (such as an accumulator) for enhanced privacy

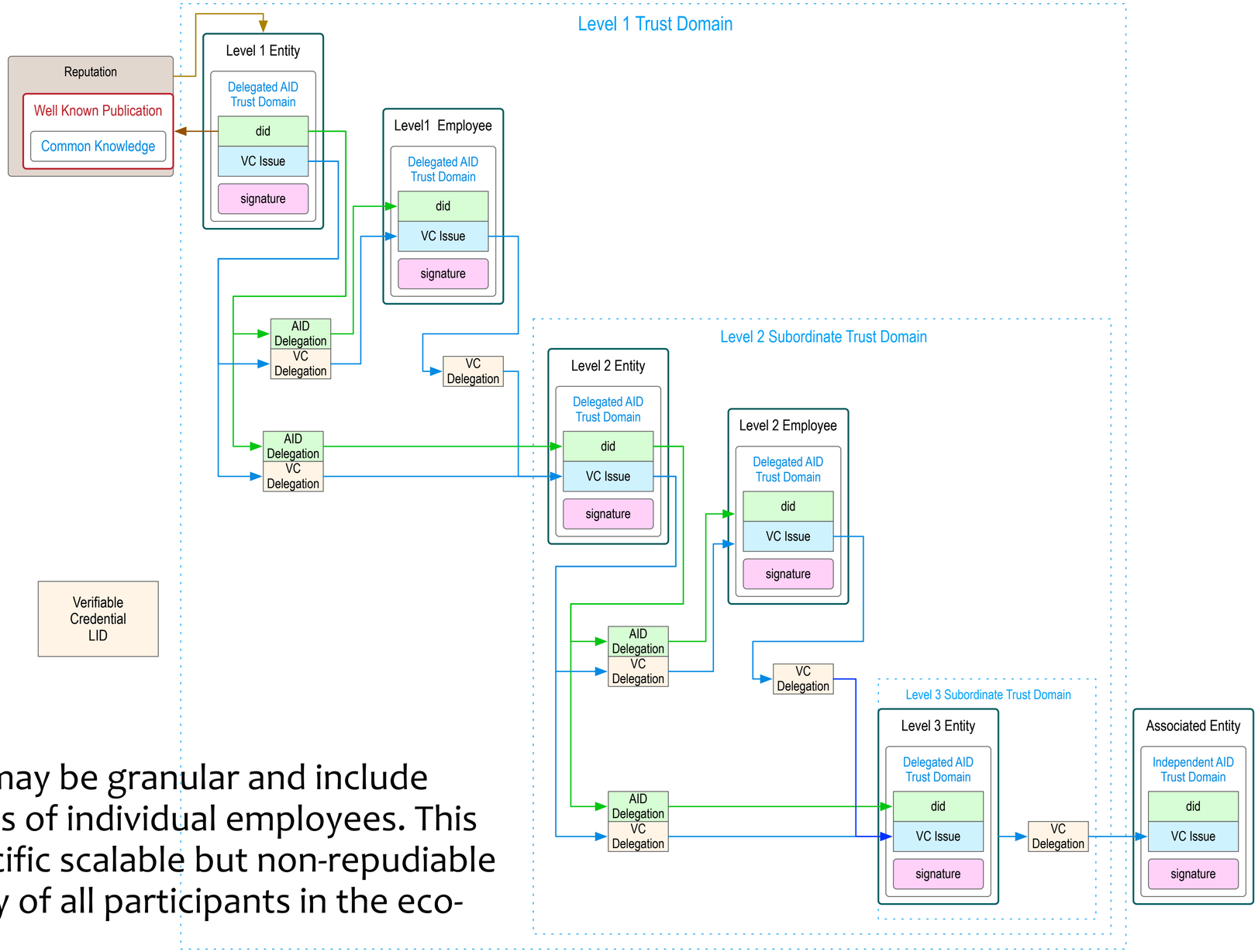




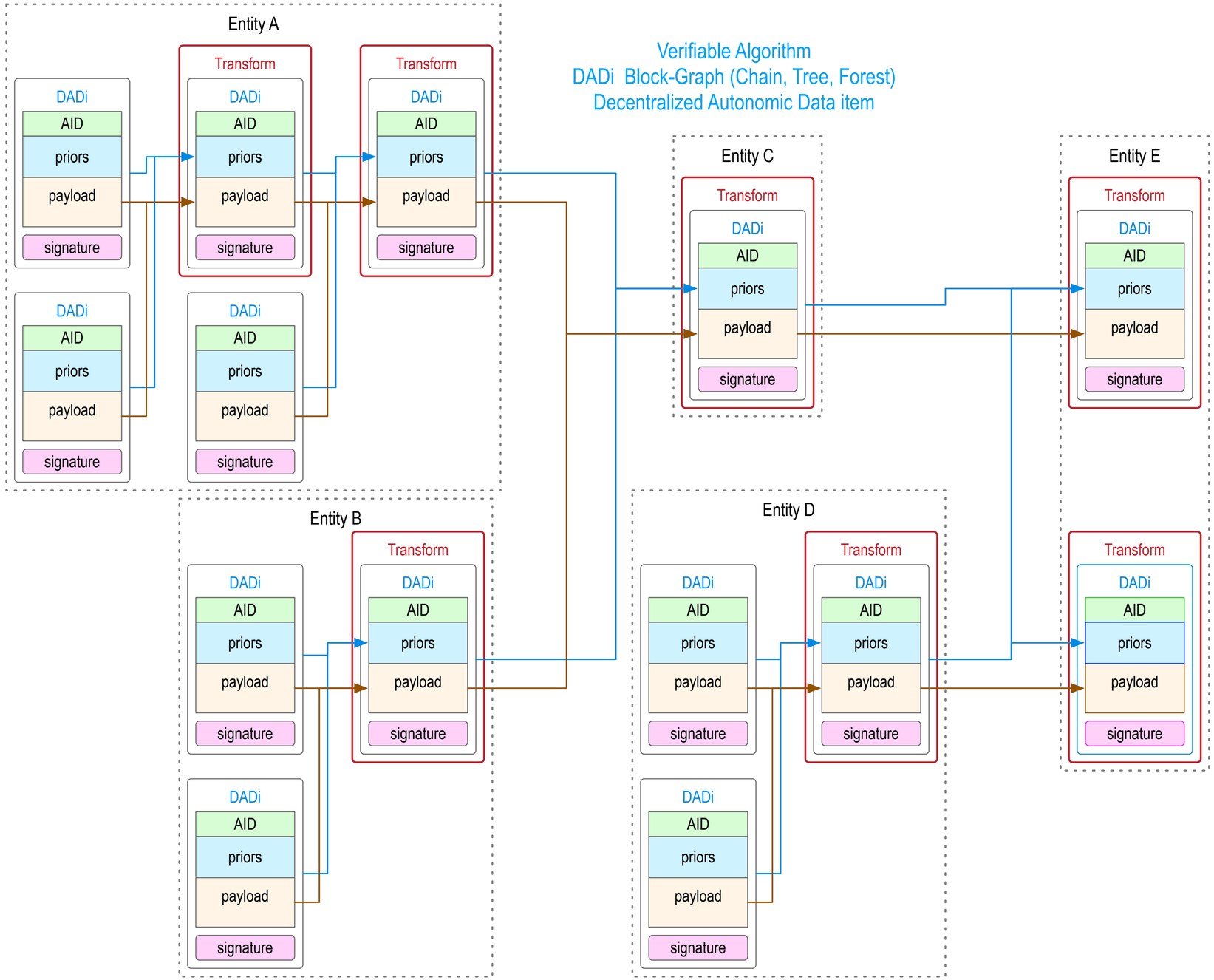
Each level of delegation forms a nested trust domain that is protected by the level above.
This increases ultimate security while enabling higher performance event issuance in lower layers.

The Level 1 entity AID provides the root-of-trust for the whole ecosystem. This enables secure decentralized interoperability.

Each trust domain may make delegations of both identifiers and verifiable credentials to a subordinate trust domain. These delegations provide revocable authorizations.

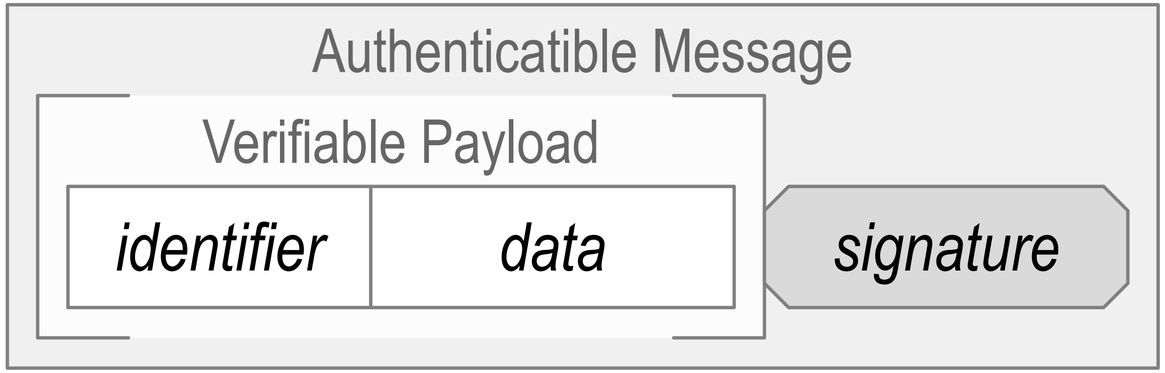


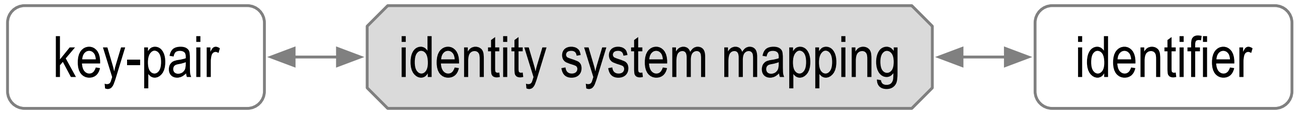
Delegations may be granular and include authorizations of individual employees. This provides specific scalable but non-repudiable accountability of all participants in the eco-system.


Verifiable Algorithm
DADi Block-Graph (Chain, Tree, Forest)
Decentralized Autonomic Data item
Entity A
Entity B
Entity C
Entity D
Entity E
Transform
DADi
AID
priors
payload
signature

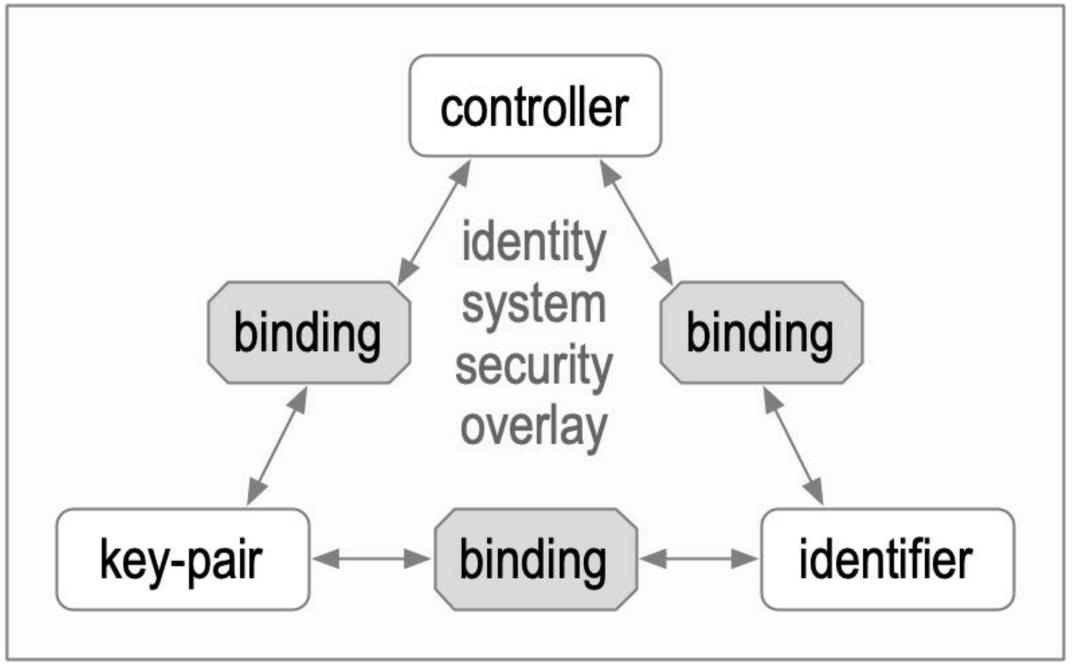
Identifier System Security

Authentic transmission of data may be verified using an identity system security overlay.
This overlay maps cryptographic key-pairs to identifiers.
When those identifiers are self-certifying they are derived via cryptographic one-way functions from the key pairs.
This provides a self-certifying identifier with a cryptographic root-of-trust.
A key event log (KEL) provide support for secure key rotation without changing the identifier.
Message authenticity is provided by verifying signatures to the authoritative keys pairs for the identifier included in the message.





The overlay's security is contingent on the mapping's security.



Identifier Issuance

Background References

Self-Certifying Identifiers:

Girault, M., "Self-certified public keys," EUROCRYPT 1991: Advances in Cryptology, pp. 490-497, 1991
https://link.springer.com/content/pdf/10.1007%2F3-540-46416-6_42.pdf

Mazieres, D. and Kaashoek, M. F., "Escaping the Evils of Centralized Control with self-certifying pathnames," MIT Laboratory for Computer Science,
http://www.sigops.org/ew-history/1998/papers/mazieres.ps

Kaminsky, M. and Banks, E., "SFS-HTTP: Securing the Web with Self-Certifying URLs," MIT, 1999
https://pdos.csail.mit.edu/~kaminsky/sfs-http.ps

Mazieres, D., "Self-certifying File System," MIT Ph.D. Dissertation, 2000/06/01
https://pdos.csail.mit.edu/~ericp/doc/sfs-thesis.ps

TCG, "Implicit Identity Based Device Attestation," Trusted Computing Group, vol. Version 1.0, 2018/03/05
https://trustedcomputinggroup.org/wp-content/uploads/TCG-DICE-Arch-Implicit-Identity-Based-Device-Attestation-v1-rev93.pdf

Autonomic Identifiers:

Smith, S. M., "Open Reputation Framework," vol. Version 1.2, 2015/05/13
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/open-reputation-low-level-whitepaper.pdf

Smith, S. M. and Khovratovich, D., "Identity System Essentials," 2016/03/29
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/Identity-System-Essentials.pdf

Smith, S. M., "Decentralized Autonomic Data (DAD) and the three R's of Key Management," Rebooting the Web of Trust RWOT 6, Spring 2018
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/DecentralizedAutonomicData.pdf

Smith, S. M., "Key Event Receipt Infrastructure (KERI) Design and Build", arXiv, 2019/07/03 revised 2021
https://arxiv.org/abs/1907.02143

Smith, S. M., "Key Event Receipt Infrastructure (KERI) Design", 2020/04/22
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI_WP_2.x.web.pdf

Stocker, C., Smith, S. and Caballero, J., "Quantum Secure DIDs," RWOT10, 2020/07/09
https://github.com/WebOfTrustInfo/rwot10-buenosaires/blob/master/final-documents/quantum-secure-dids.pdf

Smith, S. M., "Universal Identifier Theory", 2020/10/23
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/IdentifierTheory_web.pdf

Certificate Transparency:

Laurie, B., "Certificate Transparency: Public, verifiable, append-only logs," ACMQueue, vol. Vol 12, Issue 9, 2014/09/08
https://queue.acm.org/detail.cfm?id=2668154

Google, "Certificate Transparency,"
http://www.certificate-transparency.org/home

Laurie, B. and Kasper, E., "Revocation Transparency,"
https://www.links.org/files/RevocationTransparency.pdf

Internet Safety with KERI

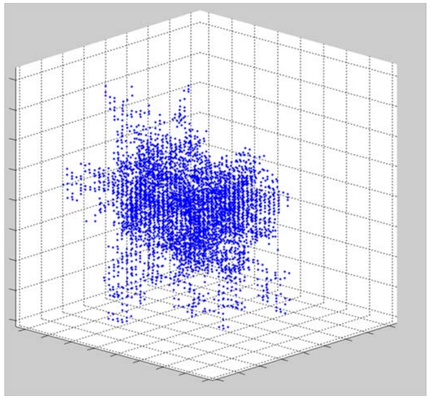
Invasion Percolation Discovery
OOBIs (Out-Of-Band-Introductions)
Spanning Trust Layer

Samuel M. Smith Ph.D.
IIW 20201 B
sam@keri.one
https://keri.one

KERI

User Permissioned (web-of-trust) Percolated Discovery

Invasion-Percolation Graph Theory for attack resistance



https://en.wikipedia.org/wiki/Percolation_theory

https://en.wikipedia.org/wiki/First_passage_percolation

http://www.physics.purdue.edu/flow/MMproject/Wilkinson1983.pdf

https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.103.018701

The Square and the Tower: Networks and Power. Niall Ferguson 2018

Percolation Theory uses graph theory to model the rate and extent of information flow by

pair-wise or n-wise sharing of information. No global lookup. Weak and Strong Links etc.

If network enables percolation and is spanned then all information is eventually available everywhere

Primary Result (Invasion-Percolation):

Eventually information fills (invades) all honest nodes in the graph whenever "capillary force" (authenticity) is greater for good information over bad information.

User permissioning means honest nodes self-isolate dishonest-nodes.

Each honest user forms identity graph of other honest nodes it interacts with that forms web-of-trust anchoring percolation discovery network.

User Permission Percolated Discovery

Insight: Need-to-know just-in-time discovery (NTK-JIT)

Issuer may provide upon demand at issuance all information an Issuee (*Holder) needs to verify the issuance. Now Holder has discovered by percolation what it needs-to-know (NTK) just-in-time (JTK) to verify.*

Holder now may provide upon demand at presentation all information any verifier needs to to verify the presentation. Now verifier has discovered by percolation what it needs-to-know (NTK) just-in-time (JTK) to verify. This includes all the percolated discovery from Issuer to Holder.

Likewise the Verifier may imbue on a NTK-JIT basis any subsequent use of that information with all the percolated discovery information it already received from the Holder plus any other information the Verifier needs to contribute.

KERI End-Verifiability means zero-trust in the percolation path.

Discovery becomes an availability not a security problem.

User Permissioned Percolated Discovery

SPED (Speedy Percolated Endpoint Discovery)

Privacy preserving or public discovery as needed

User permissioned & totally decentralized

Replaces or Augments User Permissioned DHT

Watcher Network may provide super Nodes for aggregated discovery if desirable

End-to-end verifiability means any discovery source is as good as any other.

End verifiable “truth” is still true from whatever source it may have come.

This enables secure bootstrap of discovery from any source on a NTK JIT basis.

No need for a globally trusted discovery bootstrap resolver

Zero Trust Percolated Discovery

Primary Discovery Data are Endpoints of KERI Components:

Controllers, Agents, Backers (Witness, Registrar), Watchers, Jurors, Judges, Forwarders

Endpoint is URL: IP Scheme, Host, Port, Path etc

Data Model for Securely Managing EndPoint Data

Controller (Principal AID)

Authorizes a Component to act as Player in Role

Player is AID of Component Controller

Role is purpose or function such as Watcher

Zero Trust Data as Authorization in context of KERI KeyState

ACDC Issue Revoke Reissue model

RUN model (Read, Update, Nullify)

Anchored or Signed with replay and deletion attack protection

Safe Internet Use

Minimally Sufficient Means

Leverage existing internet but safely, with end-verifiability

Internet DNS/CA is out-of-band w.r.t. KERI security

Use DSN/CA for out-of-band introductions w.r.t. KERI only, not authentication

Use IP addresses (128.187.16.184) for communication

OOBI (Out-Of-Band-Introduction)

How to use DNS safely! Vacuuous discovery of service endpoints.

Basic

```
https://hackmd.io/MxTAIBQTRkWU4-w140tNuA
```

OOBI = Url and AID Simple enough for QR Code

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Variant: Use query string to label endpoint to be discovered.

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=watcher&name=eve
https://example.com/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=witness
```

Well-Known Variant:

```
/.well-known/keri/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Result of well-known request is target URL or redirection

```
https://example.com/witness/witmer  (redirection)
http://8.8.5.5:8080/witness/witmer  (public IP)
http://10.0.5.15:8088/witness/witmer  (private IP)
```

Any OOBI may forward to another OOBI.

This is safe because the eventual endpoint is end-verifiable (authenticated).

OOBI (Out-Of-Band-Introduction)

Verbose OOBI Multi-OOBI

```
{
          "v" : "KERI10JSON00011c_",
          "t" : "rpy",
          "d": "EZ-i0d8JZAoTNZH3ULaU6JR2nmwyvYAfSVPzhzS6b5CM",
          "dt": "2020-08-22T17:50:12.988921+00:00",
          "r" : "/oobi/witness",
          "a" :
          {
             "urls":  ["http://example.com/watcher/watson", "http://example.com/witness/wilma"]
             "aid":  "EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM"
          }
}
```

Special Route Path

```
{
          "v" : "KERI10JSON00011c_",
          "t" : "rpy",
          "d": "EZ-i0d8JZAoTNZH3ULaU6JR2nmwyvYAfSVPzhzS6b5CM",
          "dt": "2020-08-22T17:50:12.988921+00:00",
          "r" : "/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM/watcher",
          "a" :
          {
             "eid": "BrHLayDN-mXKv62DAjFLX1_Y5yEUe0vA9YPe_ihiKYHE",
             "scheme": "http",
             "url":  "http://example.com/watcher/wilma",
          }
}
```

Bare URL as Self or Blind OOBI

A bare URL but no AID may be used as a bare OOBI for blind or self introductions.

Querying that bare URL (OOBI) may return or result in a default target OOBI or default target endpoint reply.

This provides a mechanism for self-introduction, self OOBI (SOOBI) or blind-introduction, blind OOBI (BOOBI) .

http://8.8.5.7:8080/oobi

http://localhost:8080/oobi```

http://8.8.5.7:8080/oobi?role=controller&name=eve

http://localhost:8080/oobi?role=controller&name=eve

By default the result of get request to this OOBI URL could be another OOBI with an AID that is the `self` AID of the node providing the bare OOBI endpoint or the actual authenticatable `self` endpoint with its AID or a default set of authenticatable endpoints.

Useful to bootstrap components in an infrastructure where the target URLs do not use a public DNS address but use instead something more secure like an explicit public IP address or a private IP or private DNS address.

A self introduction provides a bootstrap mechanism similar to a hostname configuration file with the exception that in the OOBI case the AID is not in the configuration file just the bare OOBI URL and the given node queries that bare OOBI to get the target endpoint AID. This allows bootstrap using bare IP addresses in systems where the IP infrastructure is more securely managed than public DNS or where some other Out-Of-Band-Authentication (OOBA) mechanism is used in concert.

Blind OOBI

Because the OOBI does not expose an AID, the resultant response when querying the OOBI may depend on other factors such as the source IP of the querier (requester) and/or another out-of-band-authentication (OOBA) mechanism. This supports private bootstrap of infrastructure.

Of course one could argue that this is just kicking the can down the road but IP addresses are correlatable and a blind OOBI can leverage IP infrastructure for discovery when used in combination with some other OOBA mechanism without unnecessary correlation.

Onion Routing with Blind OOBI

did-comm with Blind OOBI

Attack Protection

Replay Attack: Replay *of Authenticated (signed) Data*

TEL (ACDC) VDR Issue Revoke (kel anchored tel events) Heavyweight

Non TEL based: Best Available Data Model (BADA)

KEL anchored ordered data

KeyState-DateTime of signature ordered data.

Deletion Attack

Total erasure a security problem (GDPR flaw)

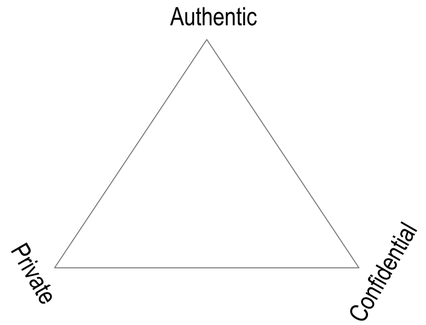
Once erased any stale authenticated data acting as authorization may be replayed without detection.

Mitigation for Deletion attack are redundant signed copies (eventually consistent DB)

BADA (Best Available Data Acceptance) Policy

Authentic Data:
- Two primary attacks:
  - Replay attack:
    - Mitigation: Monotonicity
  - Deletion attack:
    - Mitigation: Redundancy

Replay Monotonicity:
- Interactive:
  - Nonce
- Non-interactive:
  - Memory (sequence number, date-time stamp, nullification)
  - More scalable



BADA Rules

Update is included in or anchored to AID's key-state in KEL:

Rules for Acceptance of update :

Accept if no prior record.

Accept if update's anchor is later than prior record's anchor.

Update is signed by AID, but the update itself is not included in or anchored to AID's KEL:

1) Ephemeral AID whose key-state is fixed (no KEL needed)
2) Persistent AID whose key-state is provided by KEL

Rules for Acceptance of update :

If no prior record.

Accept if signature verifies against any key-state.

If prior record.

Compare key-state of the update's verified signature against key-state of prior record's verified signature.

Accept If update's key-state is later (in KEL) than prior record's key-state.

Accept if update's and prior record's key-states are the same

& update's date-time is later than prior record's date-time.

RUN off the CRUD

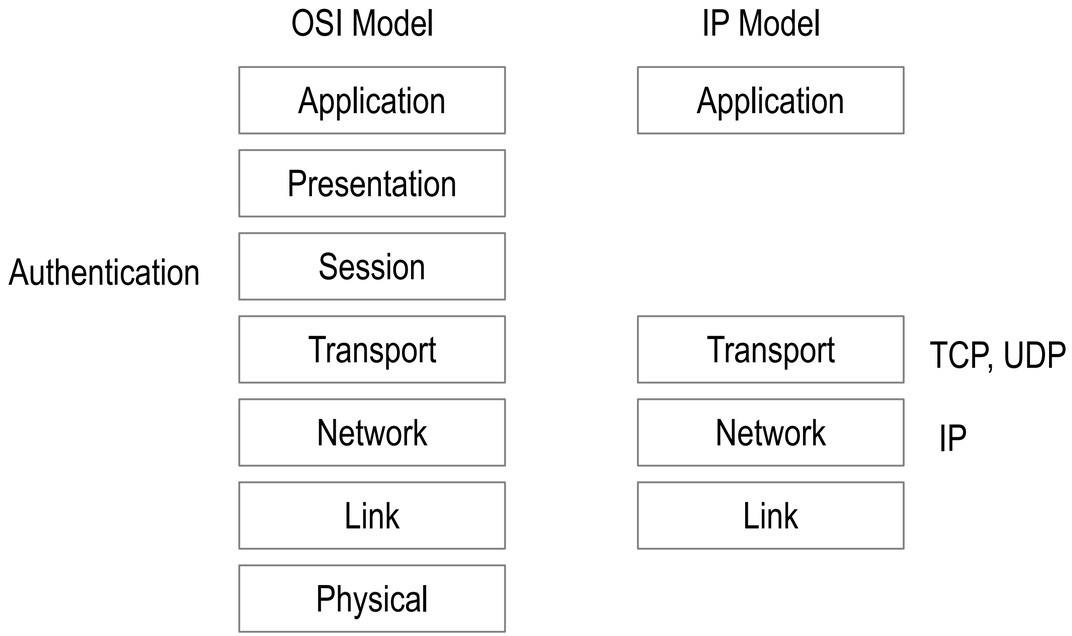
Client-Server API or Peer-to-Peer.
Create, Read, Update, Delete (CRUD)
Read, Update, Nullify (RUN)
Decentralized control means server never creates only client. Client (Peer) updates server (other Peer) always for data sourced by Client (Peer). So no Create.
Non-interactive monotonicity means we can't ever delete.
So no Delete. We must Nullify instead. Nullify is a special type of Update.
Ways to Nullify:

- null value
- flag indicating nullified

EndPoint Disclosure

Datetime stamped BADA authorization by CID of EID in Role (Update)
Datetime stamped BADA deauthorization by CID of EID in Role (Nullify)
Datetime stamped BADA authorization by EID of URL for scheme (Update).
Datetime stamped BADA deauthorization by EID of URL for scheme (Nullify)

The Internet Protocol (IP) is *bro-ken* because it has no *security* layer.

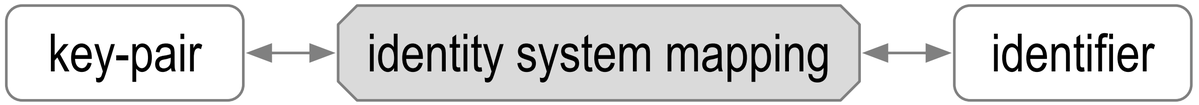


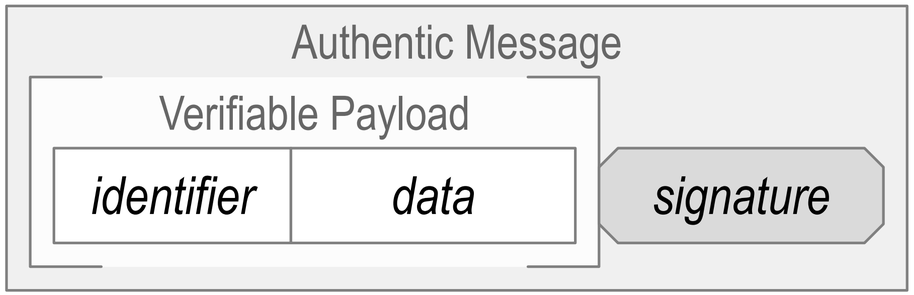
Instead …

We use *bolt-on* identity system security overlays. (DNS-CA …)

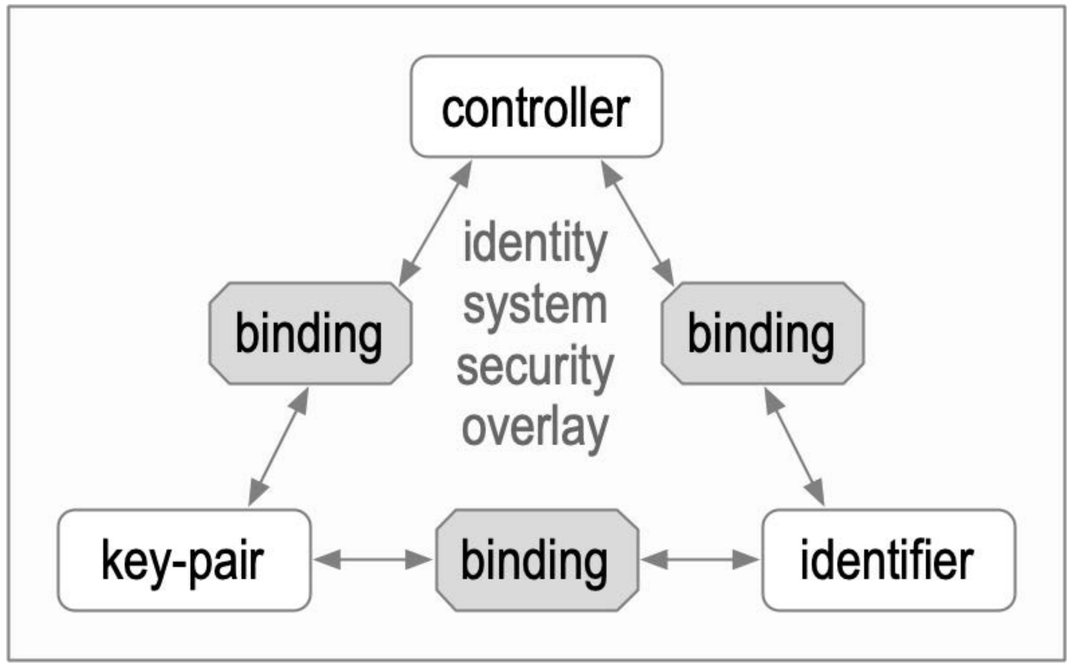
Identity System Security Overlay

Establish authenticity of IP packet's message payload.



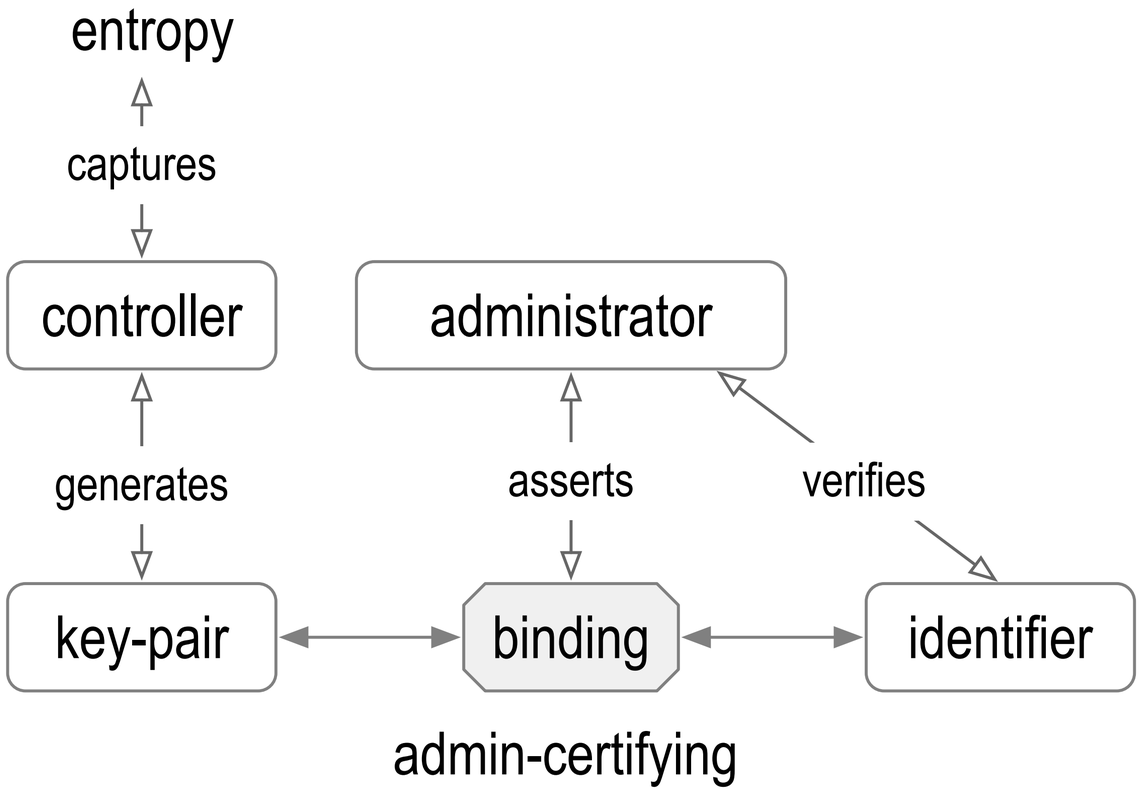


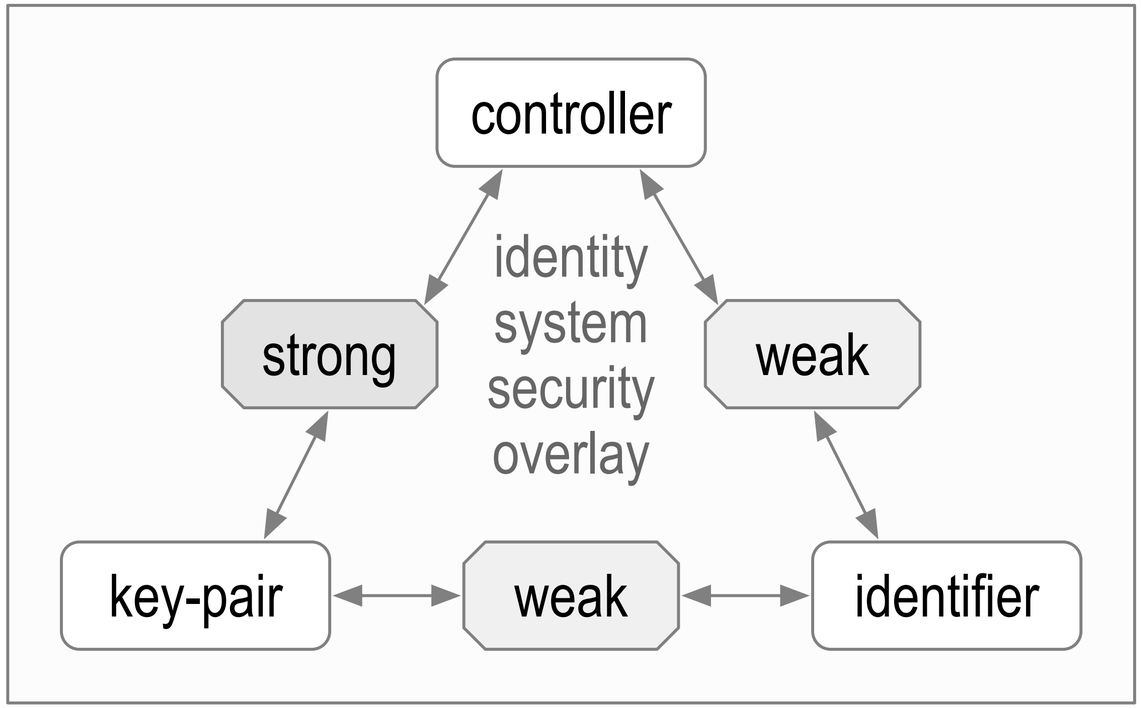
The overlay's security is contingent on the mapping's security.



Identifier Issuance

Administrative Identifier Issuance and Binding



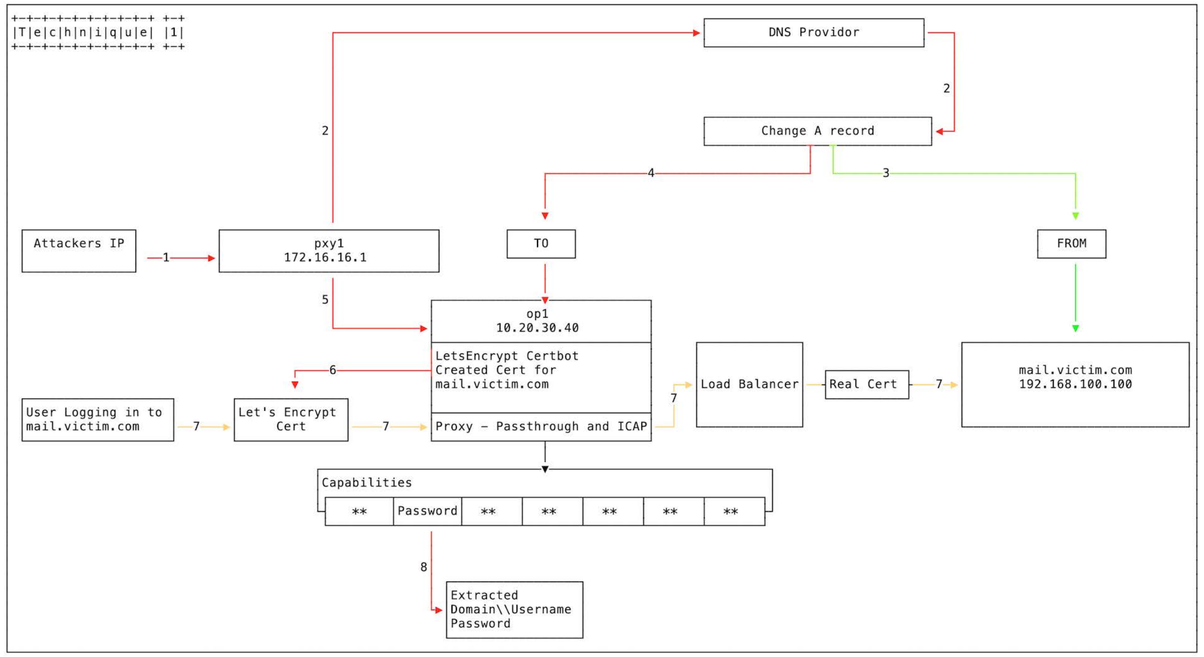


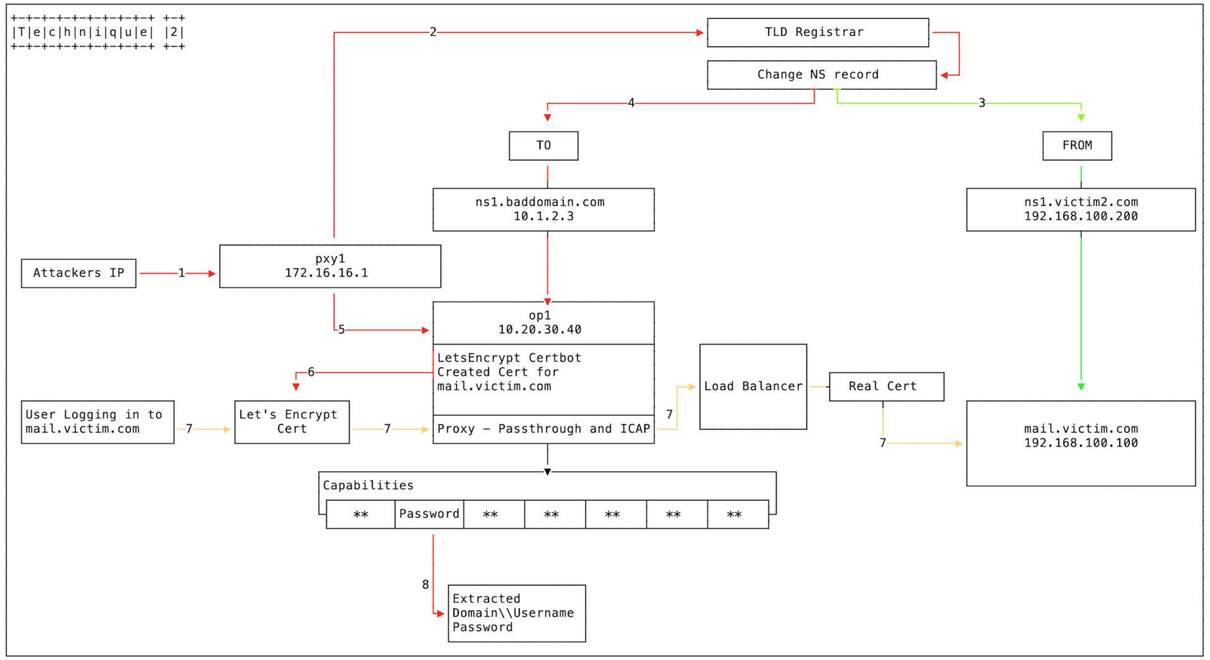
Admin-Certifying Identifier Issuance

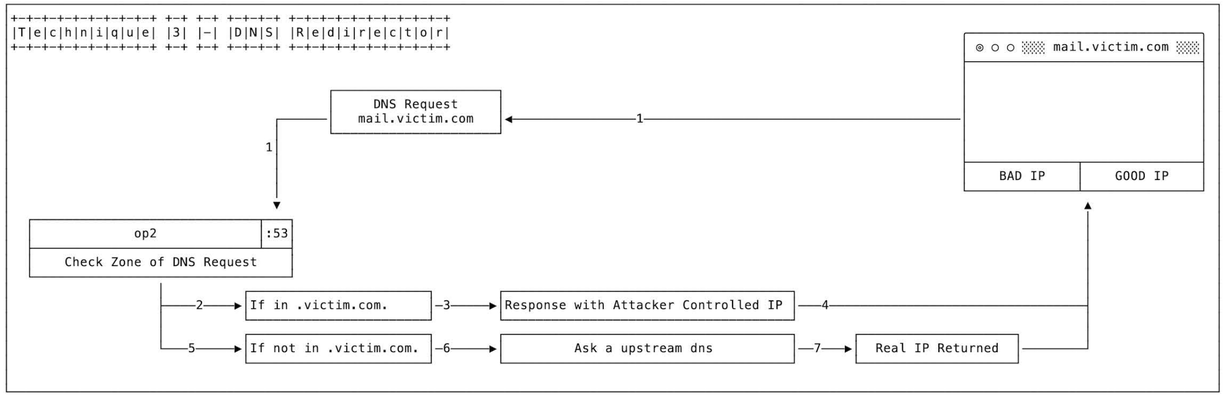
DNS Hijacking

A DNS hijacking is occurring at an unprecedented scale. Clever tricks allows attackers to obtain valid TLS certificate for hijacked domains.

https://arstechnica.com/information-technology/2019/01/a-dns-hijacking-wave-is-targeting-companies-at-an-almost-unprecedented-scale/







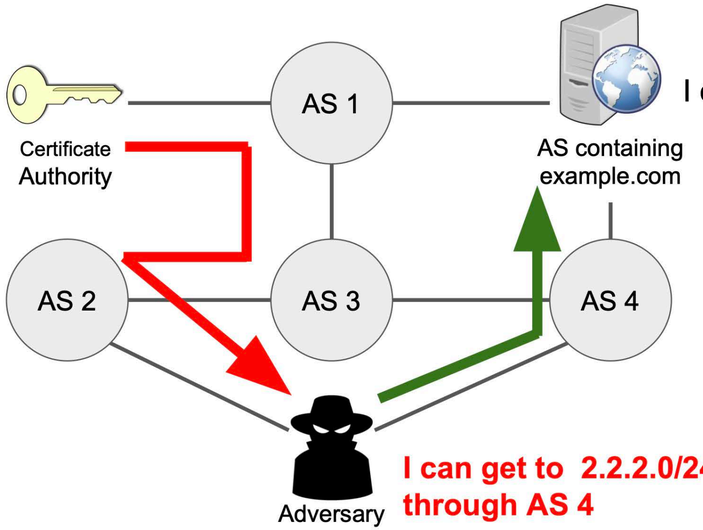
BGP Hijacking: AS Path Poisoning

Spoof domain verification process from CA. Allows attackers to obtain valid TLS certificate for hijacked domains.

Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J. and Mittal, P., "Bamboozling certificate authorities with {BGP}," vol. 27th {USENIX} Security Symposium, no. {USENIX} Security 18, pp. 833-849, 2018 https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee

Gavrichenkov, A., "Breaking HTTPS with BGP Hijacking," BlackHat, 2015 https://www.blackhat.com/docs/us-15/materials/us-15-Gavrichenkov-Breaking-HTTPS-With-BGP-Hijacking-wp.pdf

AS path poisoning

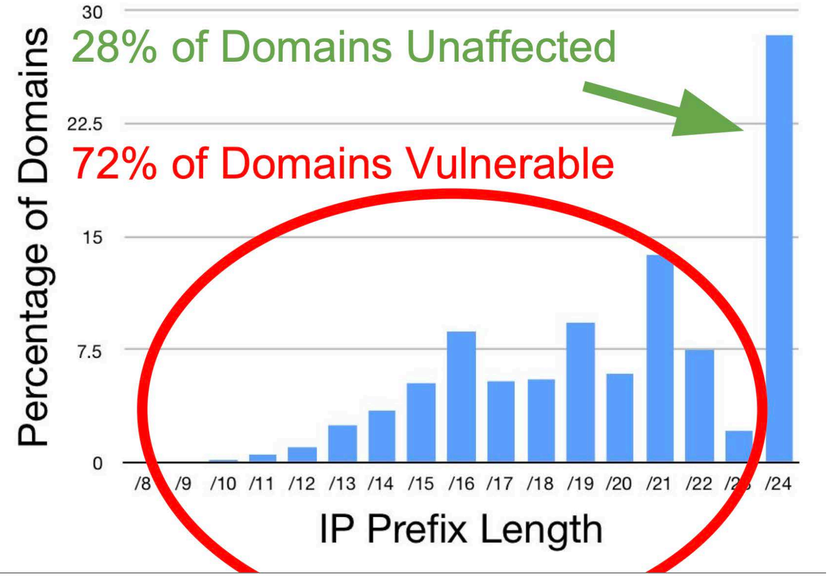


I own 2.2.2.0/23

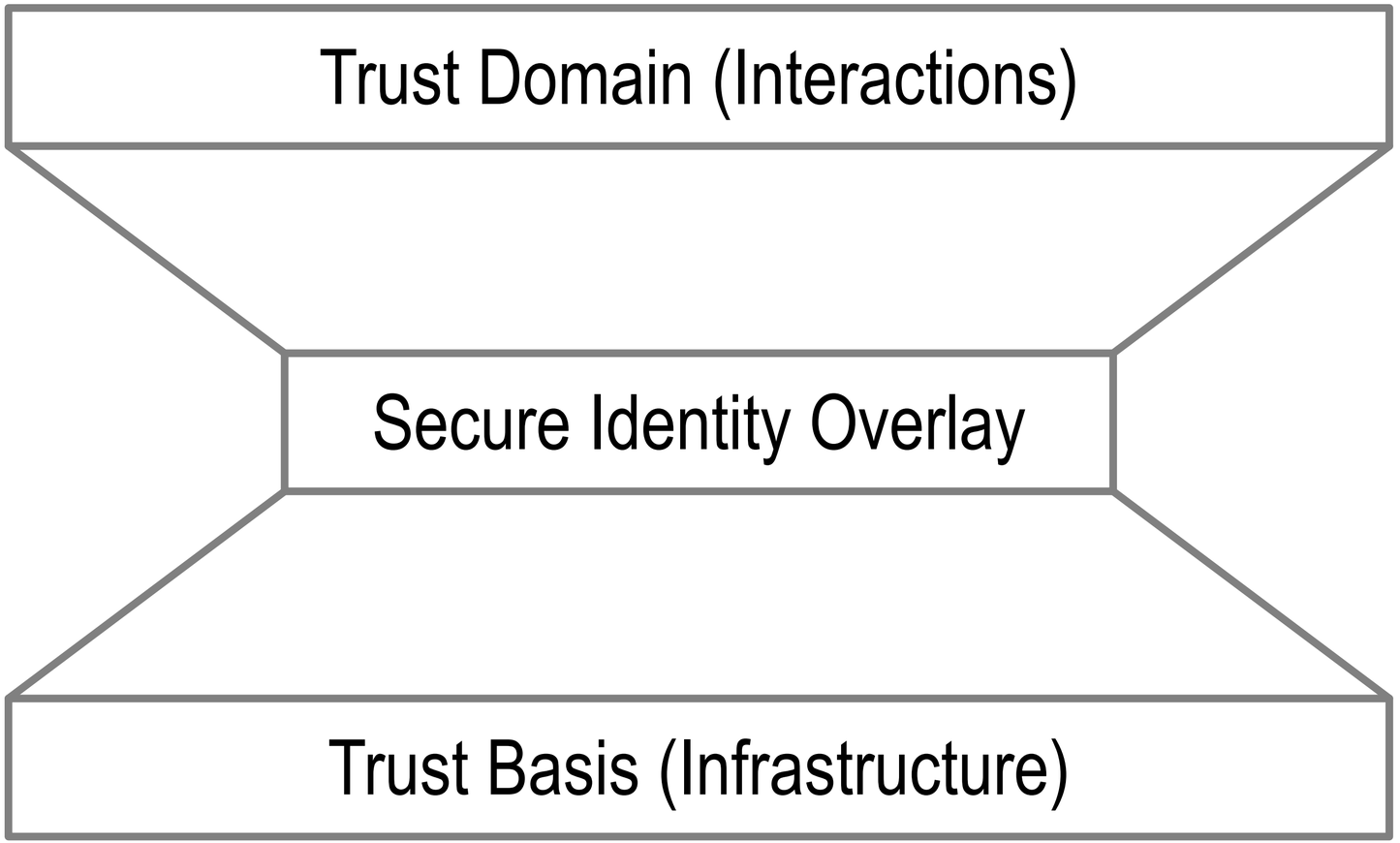
- Everyone sees announcement but looks less suspicious
- Connectivity preserved
- Almost any AS can perform
- Very stealthy
- Perfect setup to intercept traffic with certificate

Vulnerability of domains: sub-prefix attacks

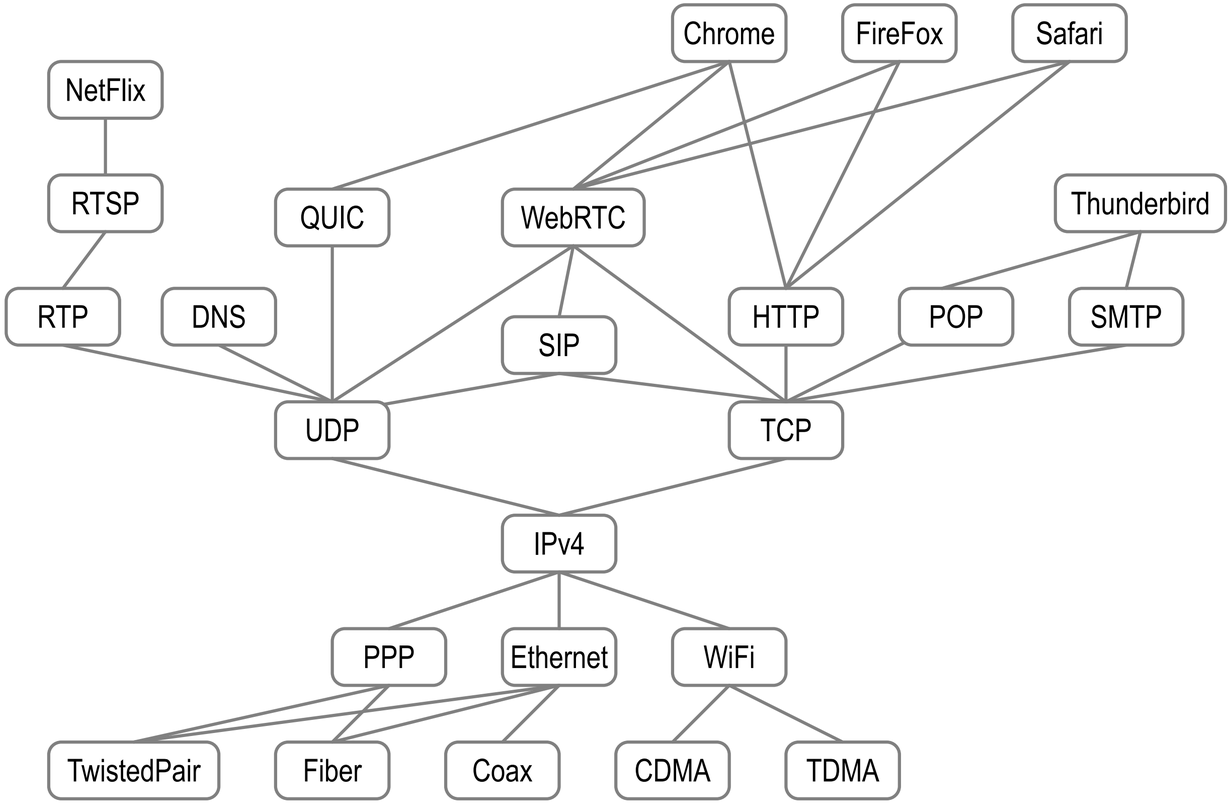
- Any AS can launch
- Only prefix lengths less than /24 vulnerable (filtering)

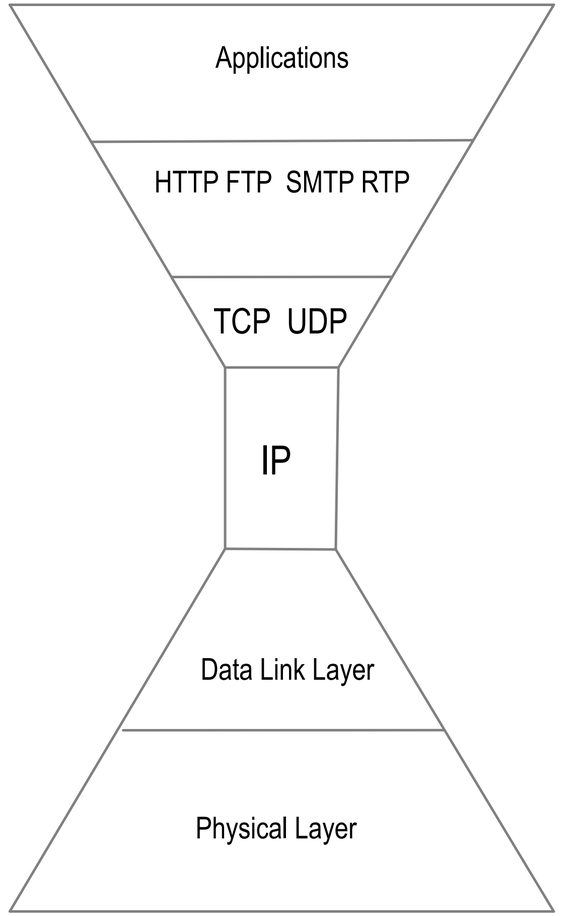


Identity System Security Overlay



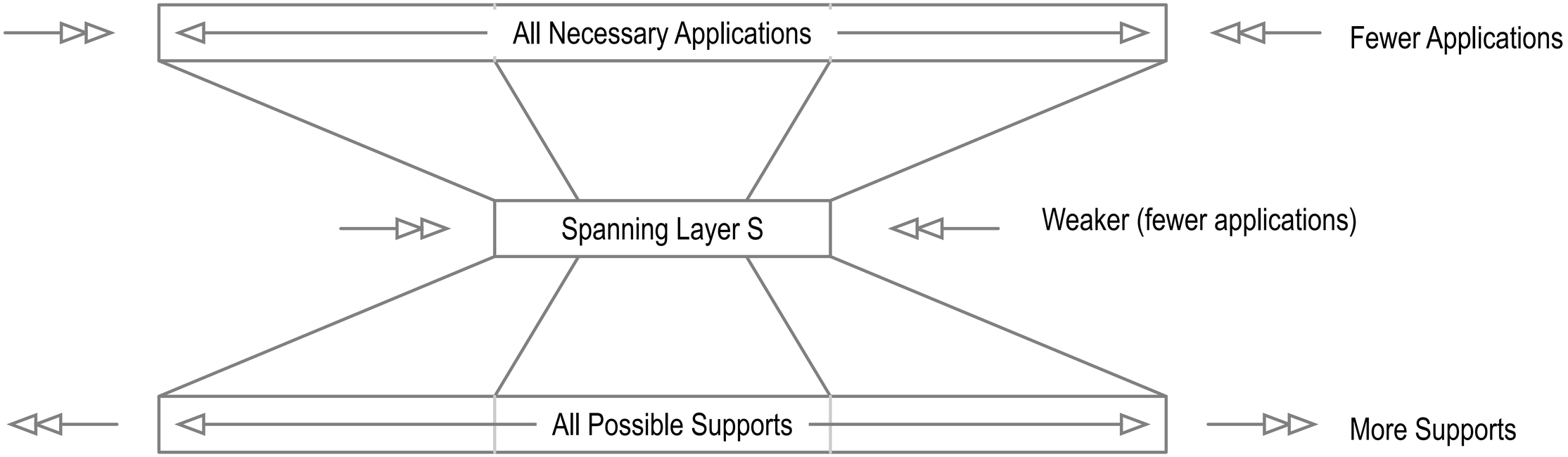
Spanning Layer





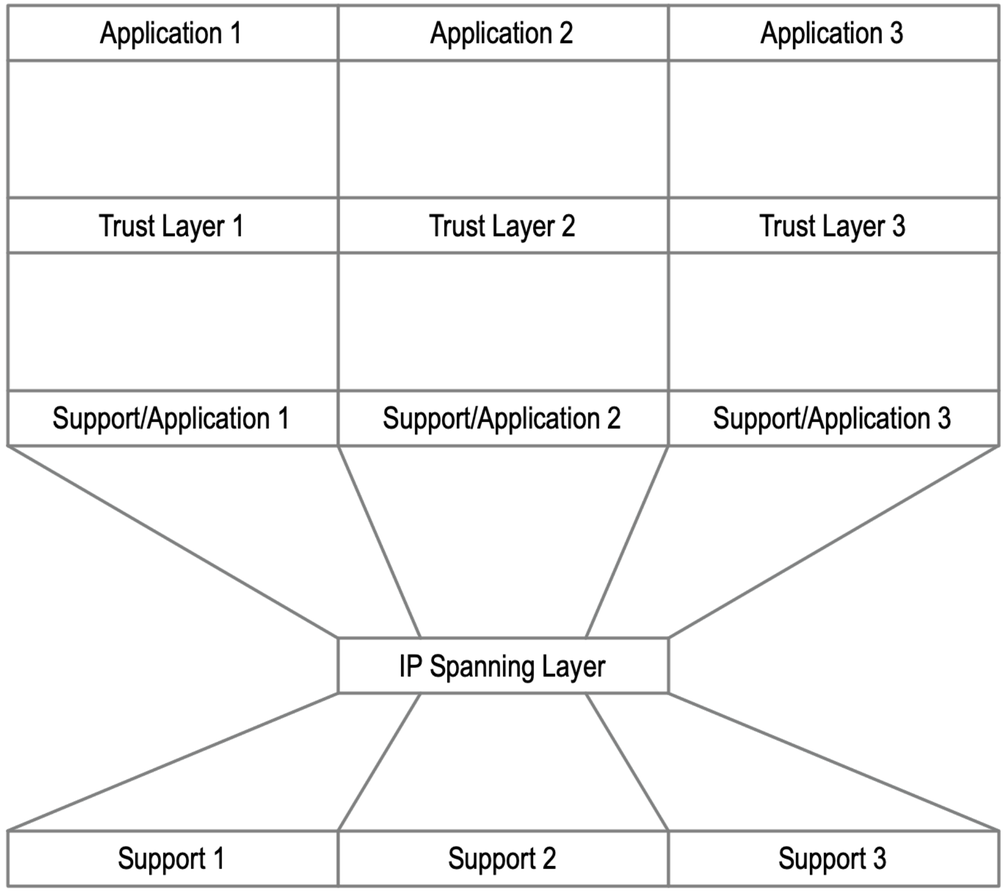
https://web.archive.org/web/20050415042854/http://www.csd.uch.gr/~hy490-05/lectures/Clark_interoperation.htm

Hourglass

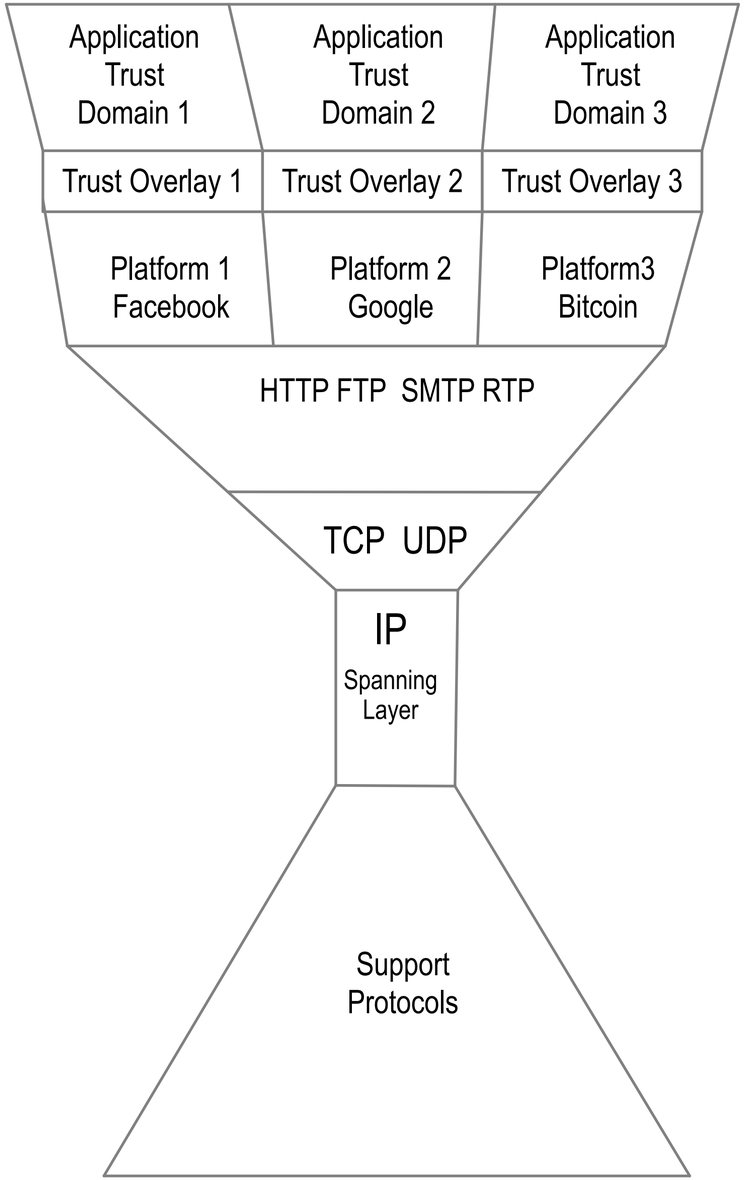


https://cacm.acm.org/magazines/2019/7/237714-on-the-hourglass-model/fulltext

Platform Locked Trust



Trust Domain Based Segmentation

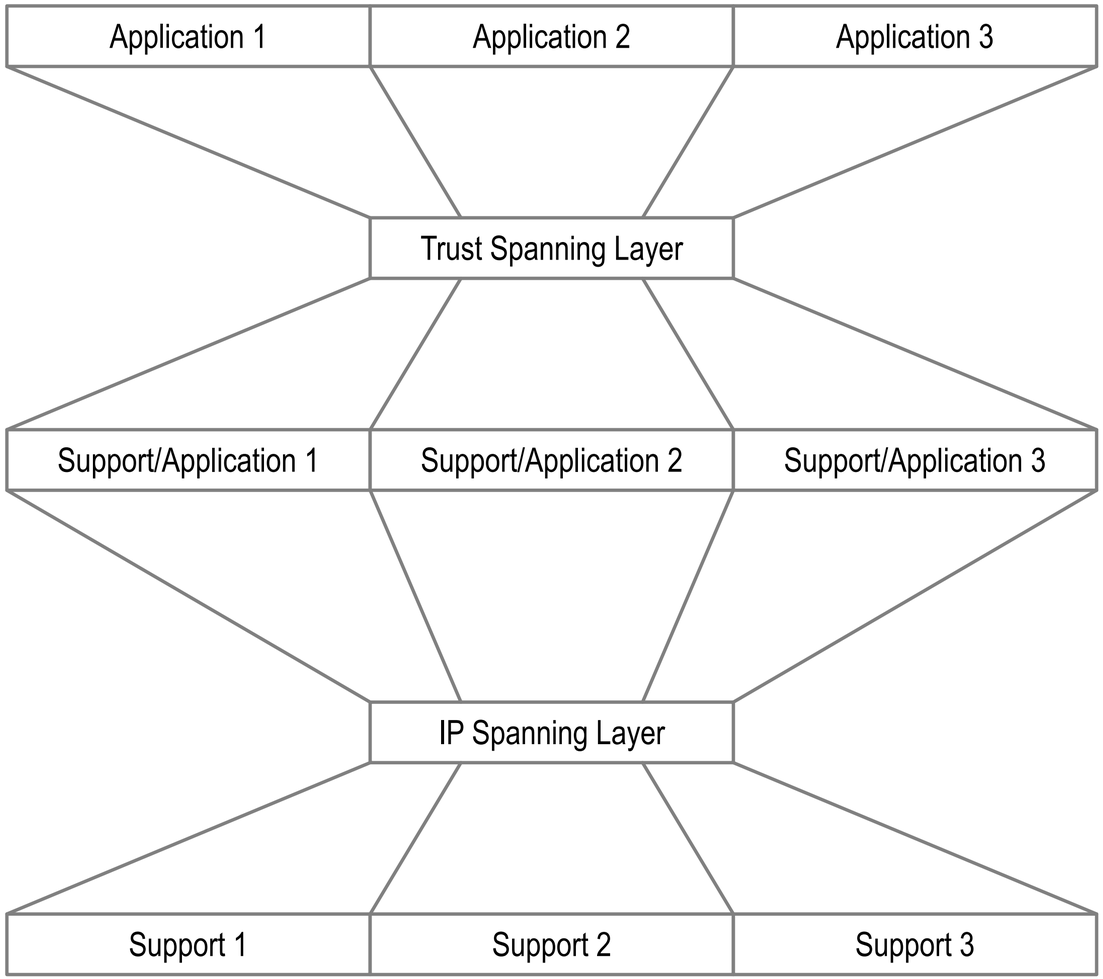


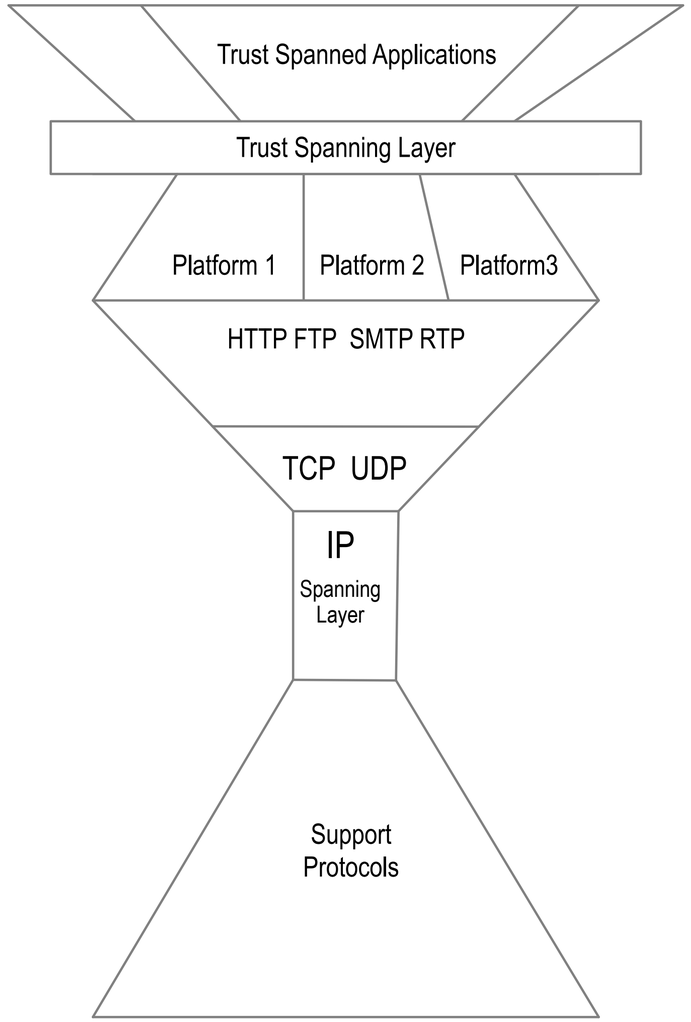
Each trust layer only spans platform specific applications

Bifurcated internet trust map

No *spanning* trust layer

Solution: Waist and Neck





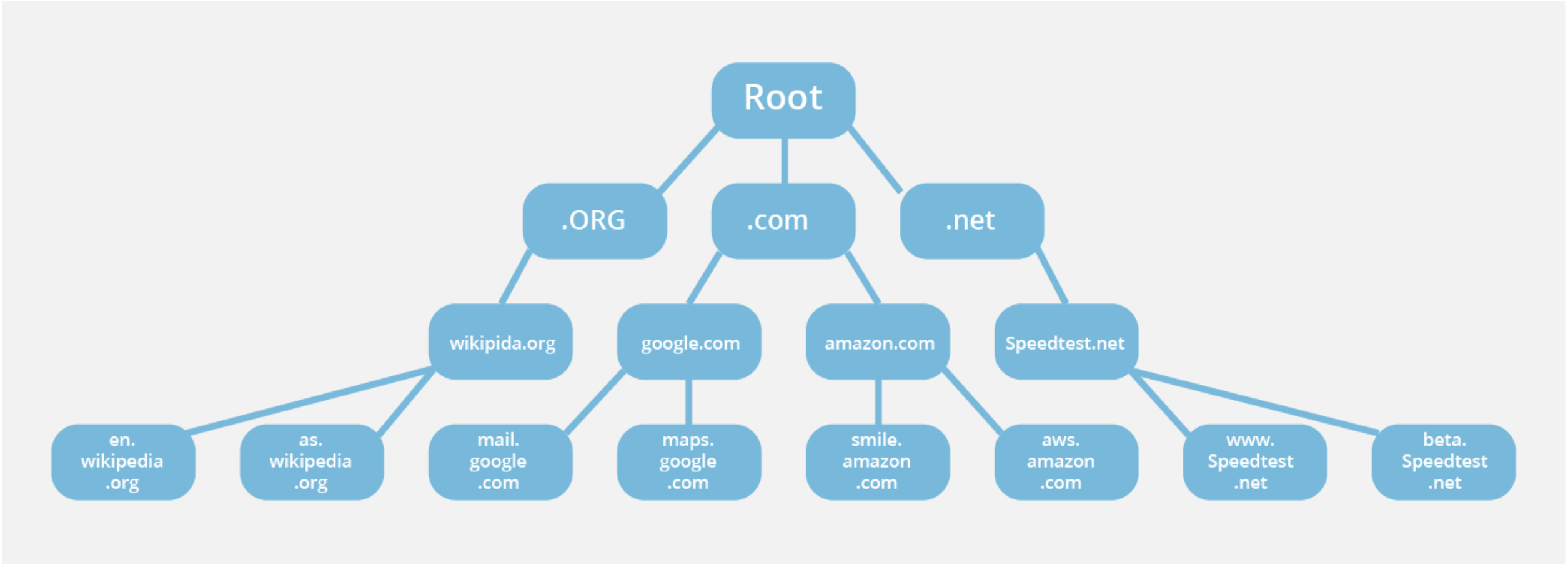
Background

Discovery

Ledger Based

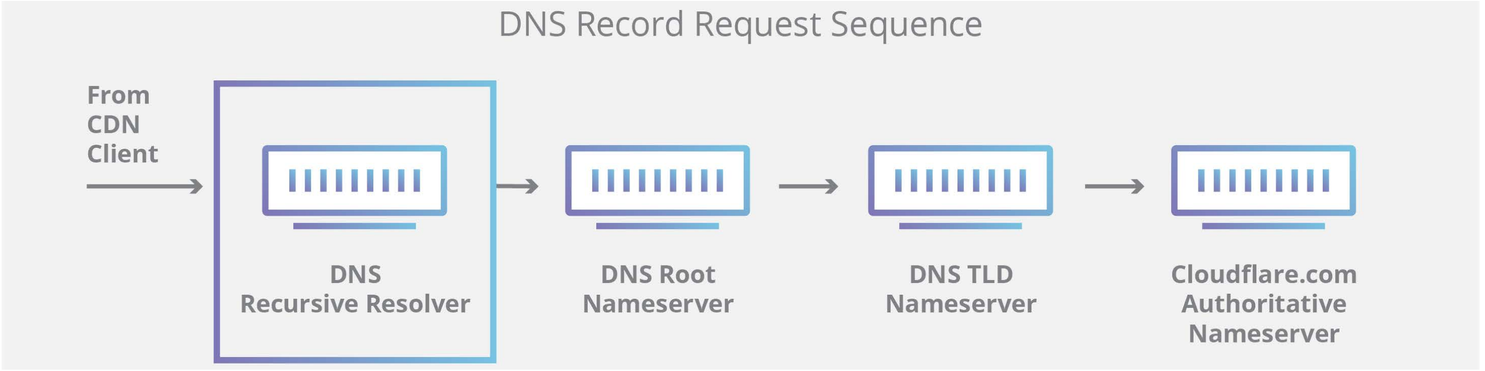
Non-Ledger Based

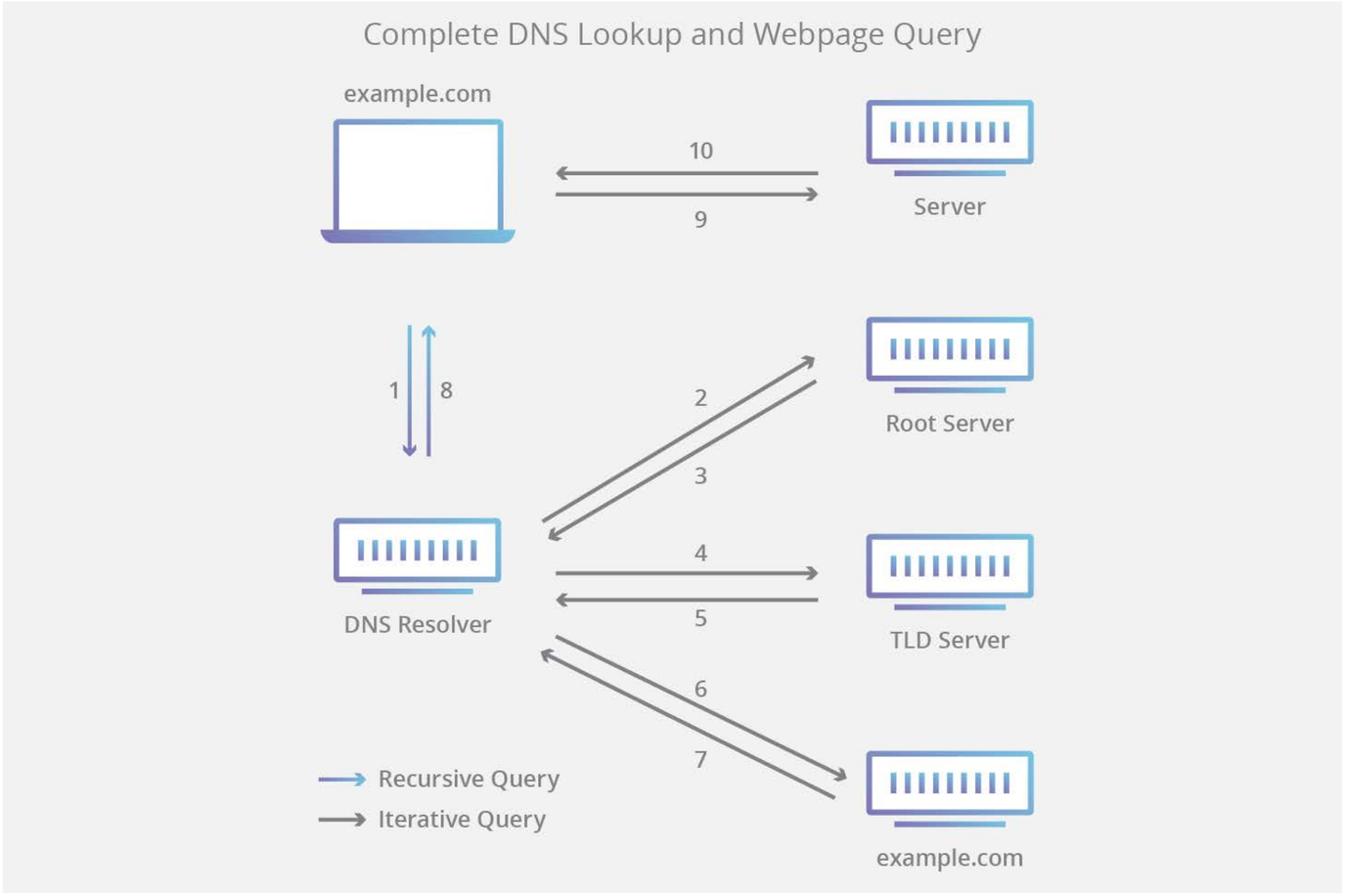
DNS “Hierarchical” Discovery



Diagrams from https://www.cloudflare.com/learning/dns/what-is-dns/

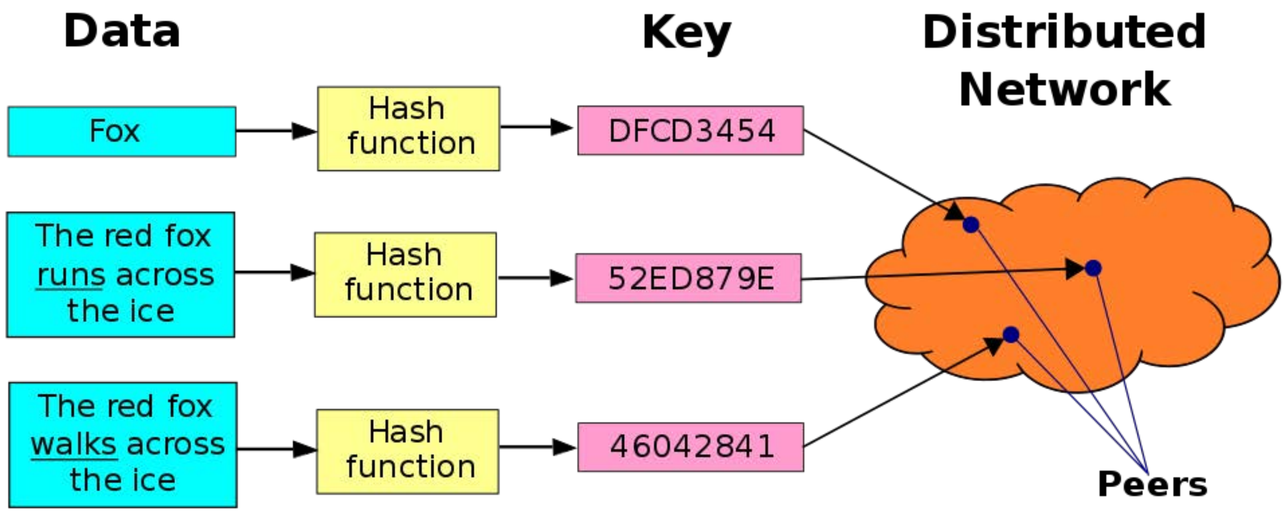
DNS “Hierarchical” Discovery





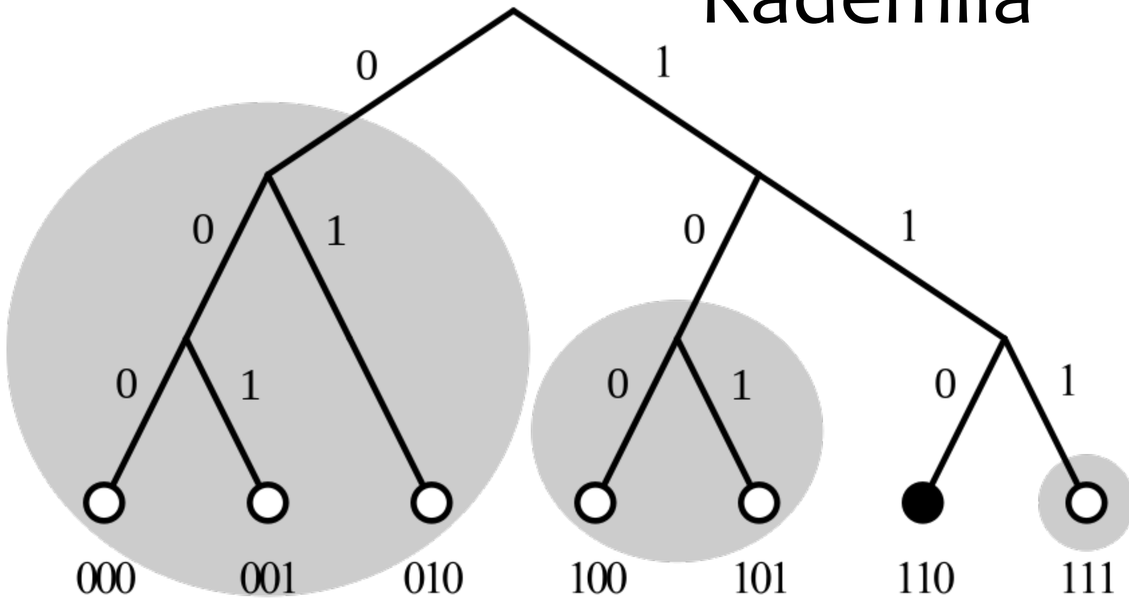

```
$ORIGIN example.com.
                        @               3600 SOA   ns1.p30.oraclecloud.net. (
                        zone-admin.dyndns.com.     ; address of responsible party
                        2016072701              ; serial number
                        3600                    ; refresh period
                        600                     ; retry period
                        604800                  ; expire time
                        1800                    ) ; minimum ttl
                        86400 NS    ns1.p68.dns.oraclecloud.net.
                        86400 NS    ns2.p68.dns.oraclecloud.net.
                        86400 NS    ns3.p68.dns.oraclecloud.net.
                        86400 NS    ns4.p68.dns.oraclecloud.net.
                        3600 MX     10 mail.example.com.
                        3600 MX     20 vpn.example.com.
                        3600 MX     30 mail.example.com.
                        60 A        204.13.248.106
                        3600 TXT    "v=spf1 includespf.oraclecloud.net ~all"
                        mail            14400 A     204.13.248.106
                        vpn               60 A      216.146.45.240
                        webapp            60 A      216.146.46.10
                        webapp            60 A      216.146.46.11
www                     43200 CNAME example.com.
```

DHT “Distributed” Discovery



Diagrams from https://en.wikipedia.org/wiki/Distributed_hash_table

Kademlia



Diagrams from https://en.wikipedia.org/wiki/Kademlia

Certificate Transparency Problem

“The solution the computer world has relied on for many years is to introduce into the system trusted third parties (CAs) that vouch for the binding between the domain name and the private key. The problem is that we've managed to bless several hundred of these supposedly trusted parties, any of which can vouch for any domain name. Every now and then, one of them gets it wrong, sometimes spectacularly.”

Pinning inadequate
Notaries inadequate
DNSSec inadequate
All require trust in 3rd party compute infrastructure that is inherently vulnerable

Certificate Transparency: (related EFF SSL Observatory)
Public end-verifiable append-only event log with consistency and inclusion proofs
End-verifiable duplicity detection = Ambient verifiability of duplicity
Event log is third party infrastructure but zero trust because it is verifiable.
Sparse Merkle Trees for revocation of certificates

Certificate Transparency Solution

Public end-verifiable append-only event log with consistency and inclusion proofs
End-verifiable duplicity detection = ambient verifiability of duplicity
Event log is third party infrastructure but it is not trusted because logs are verifiable.
Sparse Merkle trees for revocation of certificates
(related EFF SSL Observatory)

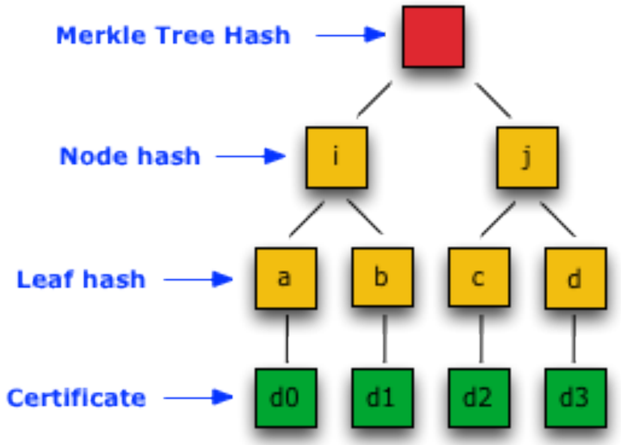


Figure 1

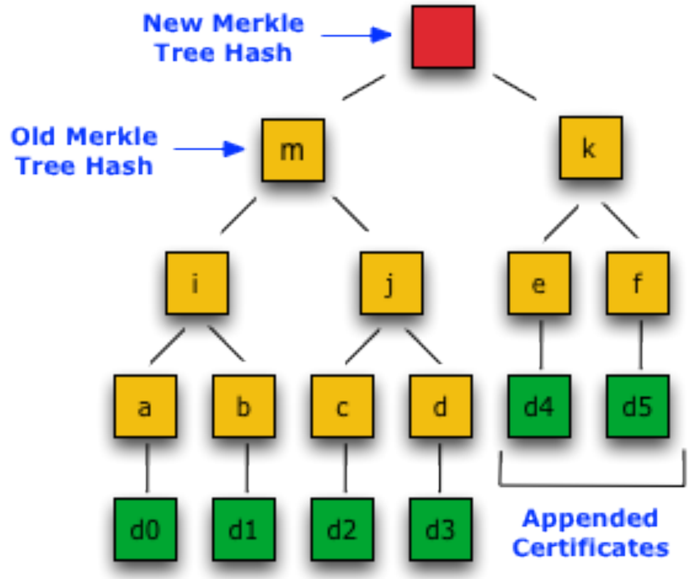


Figure 2

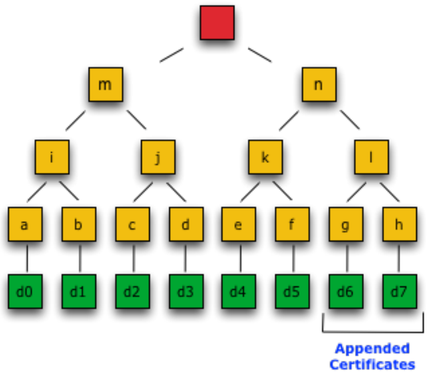


Figure 3

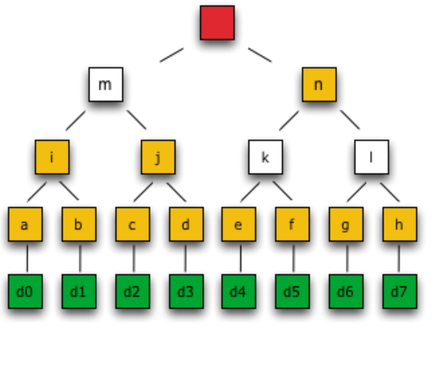


Figure 4