

# Authentic Chained Data Containers

ACDC

## Verifiable Credentials with Stuff

*Samuel M. Smith Ph.D.*

*Sam@prosapien.com*

*2022/04/26*

# Resources

## Documentation:

<https://github.com/trustoverip/tswg-acdc-specification>

[https://docs.google.com/presentation/d/1mO1EZa9BcjAjWEzw7DWi124uMfyNyDeM3HuajsGNoTo/edit#slide=id.g124effd16ae\\_0\\_164](https://docs.google.com/presentation/d/1mO1EZa9BcjAjWEzw7DWi124uMfyNyDeM3HuajsGNoTo/edit#slide=id.g124effd16ae_0_164)

<https://keri.one/keri-resources/>

<https://arxiv.org/abs/1907.02143> (KERI White Paper)

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

<https://wiki.trustoverip.org/display/HOME/ACDC+Authentic+Chained+Data+Container%29+Task+Force>

<https://github.com/WebOfTrust>

<https://github.com/WebOfTrust/keri>

# Basic ACDC Features

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

Leverages SAIDs (Self-Addressing Identifiers)

Schemas are SAIDified: 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 Exploitation (including statistical correlation)

# Basic ACDC

```
{  
  "v": "ACDC10JSON00011c_",
  "d": "EBdT3gIXOf2BBWNHdSXCJnFJL50uQPyM5K0neuniccM",
  "i": "did:keri:EmkPreYpZffK66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6Fkekw1Ojkggt",
  "s": "E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY11kFrn9y2PY",
  "e": "ERH3dCdoFOLe71ihEqcywJcnjtJtQIYPvAu6DZI13MOA",
  "r": "Ee71ihEqcywJcnjtJtQIYPvAu6DZI13MORH3dCdoFOLB",
}  
  
{  
  "v": "ACDC10JSON00011c_",
  "d": "EBdT3gIXOf2BBWNHdSXCJnFJL50uQPyM5K0neuniccM",
  "u": "0ANghkDaG7OY1wjADAE0qHcg",
  "i": "did:keri:EmkPreYpZffK66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6Fkekw1Ojkggt",
  "s": "E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY11kFrn9y2PY",
  "e": "ERH3dCdoFOLe71ihEqcywJcnjtJtQIYPvAu6DZI13MOA",
  "r": "Ee71ihEqcywJcnjtJtQIYPvAu6DZI13MORH3dCdoFOLB"
}
```

# ACDC Normative Field Labels

ACDC Field Labels

| Label | Title                     | Description   |
|-------|---------------------------|---|
| v     | Version String            | Regexable format: ACDCv <sub>w</sub> SSSShhhhh_ that provides protocol type, version, serialization type, size, and terminator.   |
| d     | Digest (SAID)             | Self-referential fully qualified cryptographic digest of enclosing map.   |
| i     | Identifier (AID)          | Semantics are determined by the context of its enclosing map.   |
| u     | UUID                      | Random Universally Unique IDentifier as fully qualified high entropy pseudo-random string, a salted nonce.  |
| ri    | Registry Identifier (AID) | Issuance and/or revocation, transfer, or retraction registry for ACDC.  |
| s     | Schema                    | Either the SAID of a JSON Schema block or the block itself.   |
| a     | Attribute                 | Either the SAID of a block of attributes or the block itself.   |
| A     | Attribute Aggregate       | Either the Aggregate of a selectively disclosable block of attributes or the block itself.  |
| e     | Edge                      | Either the SAID of a block of edges or the block itself.  |
| r     | Rule                      | Either the SAID a block of rules or the block itself.   |
| n     | Node                      | SAID of another ACDC as the terminating point of a directed edge that connects the encapsulating ACDC node to the specified ACDC node as a fragment of a distributed property graph (PG). |
| 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.  |

# Basic ACDC

```
{  
  "v": "ACDC10JSON00011c_",
  "d": "EBdT3gIXOf2BBWNHdSXCJnFJL50uQPyM5K0neuniccM",
  "i": "did:keri:EmkPreYpZfFk66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6Fkekw1Ojkggt",
  "s": "E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY11kFrn9y2PY",
  "e": "ERH3dCdoFOLe71ihEqcywJcnjtJtQIYPvAu6DZI13MOA",
  "r": "Ee71ihEqcywJcnjtJtQIYPvAu6DZI13MORH3dCdoFOLB",
}  
  
{  
  "v": "ACDC10JSON00011c_",
  "d": "EBdT3gIXOf2BBWNHdSXCJnFJL50uQPyM5K0neuniccM",
  "u": "0ANghkDaG7OY1wjADAE0qHcg",
  "i": "did:keri:EmkPreYpZfFk66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPM",
  "ri": "did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6Fkekw1Ojkggt",
  "s": "E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY11kFrn9y2PY",
  "e": "ERH3dCdoFOLe71ihEqcywJcnjtJtQIYPvAu6DZI13MOA",
  "r": "Ee71ihEqcywJcnjtJtQIYPvAu6DZI13MORH3dCdoFOLB"
}
```

# Public Attribute Section

```
{
  "a": {
    "d": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "i": "did:keri:EpZffK66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPmkPreYA",
    "score": 96,
    "name": "Jane Doe"
  }
}

{
  "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
      }
    ]
  }
}
```

# Private (Hidden) Attribute Section

```
{  
  "a":  
  {  
    "d": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",  
    "u": "0AwjaDAE0qHcgNghkDaG7OY1",  
    "i": "did:keri:EpZffK66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPmkPreYA",  
    "score": 96,  
    "name": "Jane Doe"  
  }  
}  
  
{  
  "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,  
      }  
    ]  
  }  
}
```

# Attribute Section

```
{  
  "a":  
  {  
    "d": "EgveY4-9Xg0cLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",  
    "u": "0AwjaDAE0qHcgNghkDaG7OY1",  
    "i": "did:keri:EpZffK66jpf3uFv7vk1XKhzBrAqjsKAn2EDIPmkPreYA",  
    "score": 96,  
    "name": "Jane Doe"  
  }  
}
```

# Edge Section

```
{  
  "e":  
  {  
    "d": "EerzwLIr9Bf7V_NHwY11kFrn9y2PgveY4-9XgOcLxUdY",  
    "boss":  
    {  
      "d": "E9y2PgveY4-9XgOcLxUdYerzwLIr9Bf7V_NHwY11kFrn",  
      "n": "EI13MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA",  
      "w": "high"  
    }  
  }  
}
```

# Edge Section

```
{  
  "e":  
  {  
    "d": "EerzwLIr9Bf7V_NHwY11kFrn9y2PgveY4-9XgOcLxUdY",  
    "boss":  
    {  
      "d": "E9y2PgveY4-9XgOcLxUdYerzwLIr9Bf7V_NHwY11kFrn",  
      "n": "EI13MORH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZA",  
      "w": "high"  
    }  
  }  
}
```

# Nested Edge Section with Operators

```
{  
  "e":  
  {  
    "d": "EerzwLIr9Bf7V_NHwY1lkFrn9y2PgveY4-9XgOcLx, UdY",  
    "o": "AND",  
    "boss":  
    {  
      "n": "EIl3MORH3dCdoFOLe71ihcqcywJcnjtJtQIYPvAu6DZA",  
      "o": ["NI2I", "NOT"],  
      "power": "high"  
    },  
    "baby":  
    {  
      "n": "EORH3dCdoFOLe71ihcqcywJcnjtJtQIYPvAu6DZAI13A",  
      "o": "I2I",  
      "power": "low"  
    },  
    "food":  
    {  
      "o": "OR",  
      "power": "med",  
      "plum":  
      {  
        "n": "EQIYPvAu6DZAI13AORH3dCdoFOLe71ihcqcywJcnjtJt",  
        "o": "NI2I"  
      },  
      "pear":  
      {  
        "n": "EJtQIYPvAu6DZAI13AORH3dCdoFOLe71ihcqcywJcnjt",  
        "o": "NI2I"  
      }  
    }  
  }  
}
```

# 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](https://en.wikipedia.org/wiki/Ricardian_contract)

The Ricardian Contract

the BowTie Model

Chain-link Confidentiality

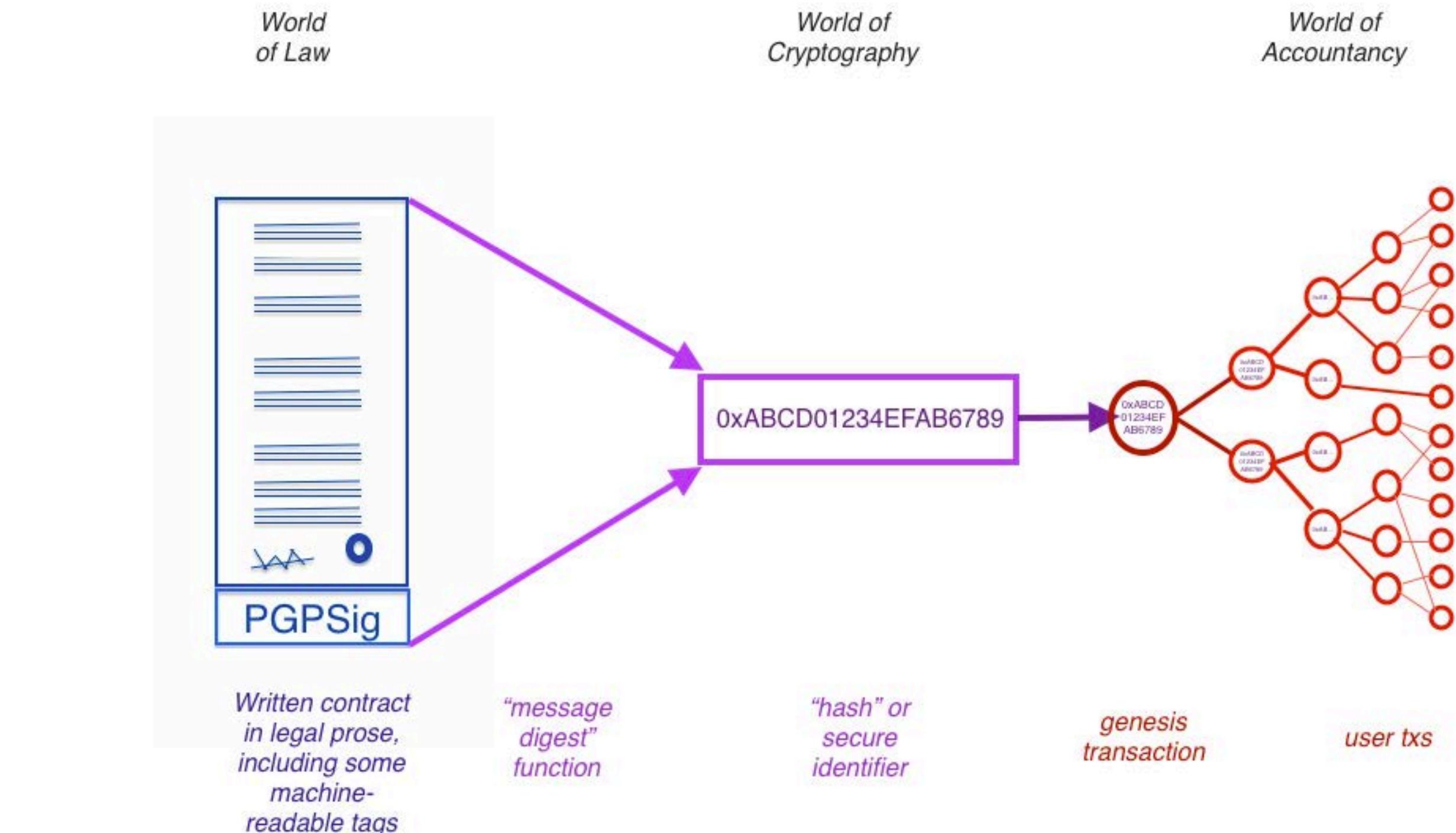
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2045818](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2045818)

World  
of Law

World of  
Cryptography

World of  
Accountancy

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



# Chain-Link Confidentiality

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2045818](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.

# Latent Accountability

Escrow

KYC

Contingent Enforcement and Recourse

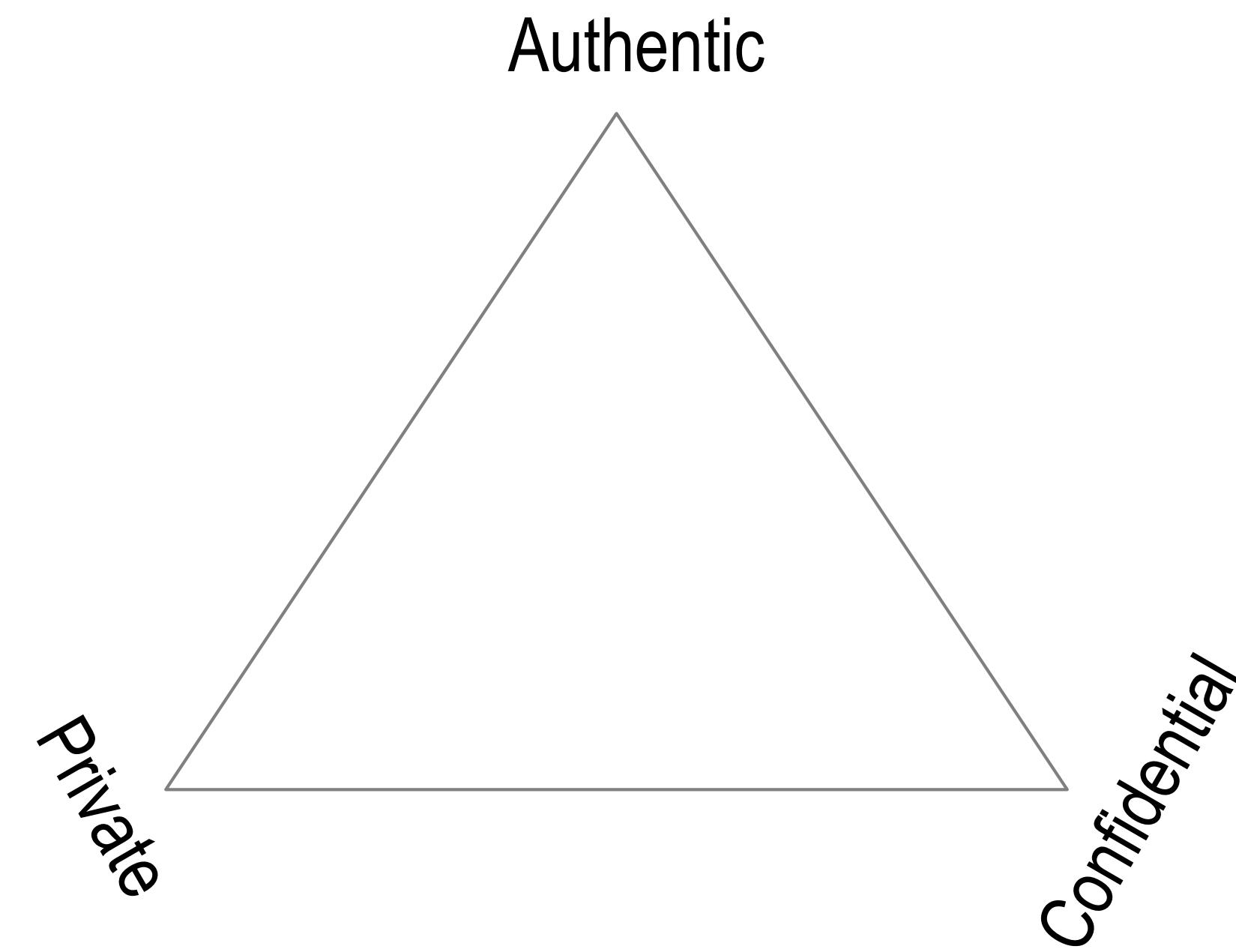
Bonding

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 in the conversation (non-content meta-data)

*Authenticity:*

about proving who said what in the conversation (secure attribution)

*Confidentiality:*

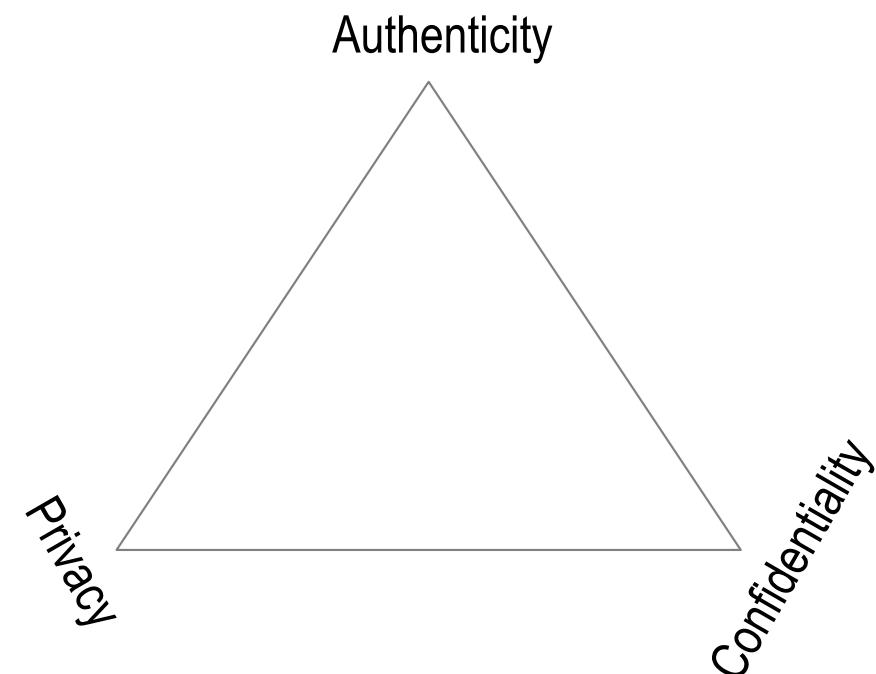
about control over the disclosure of what was said in the conversation (content data)

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.

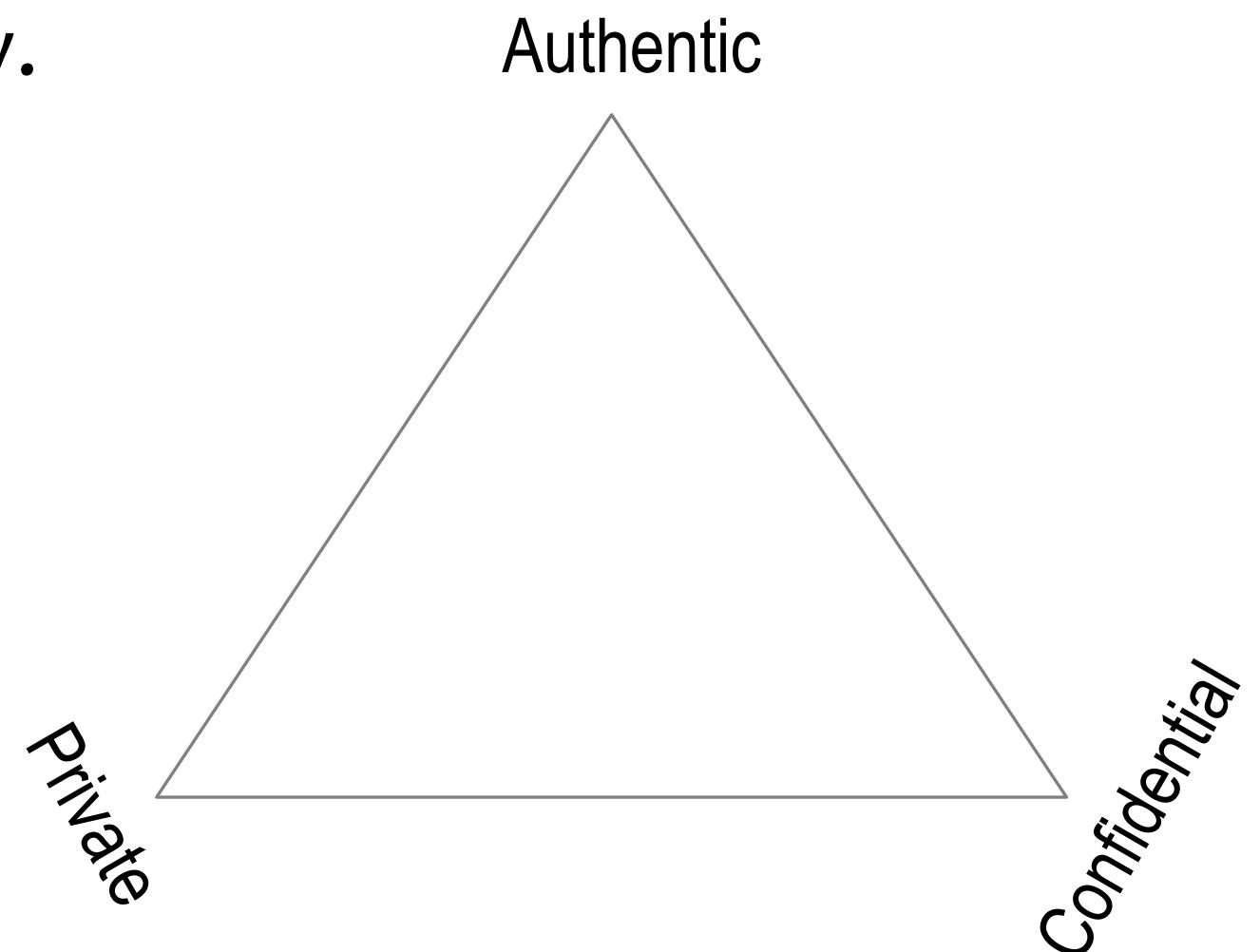
This means that technologically there is no way to prevent disclosure by any party to some third party.

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.



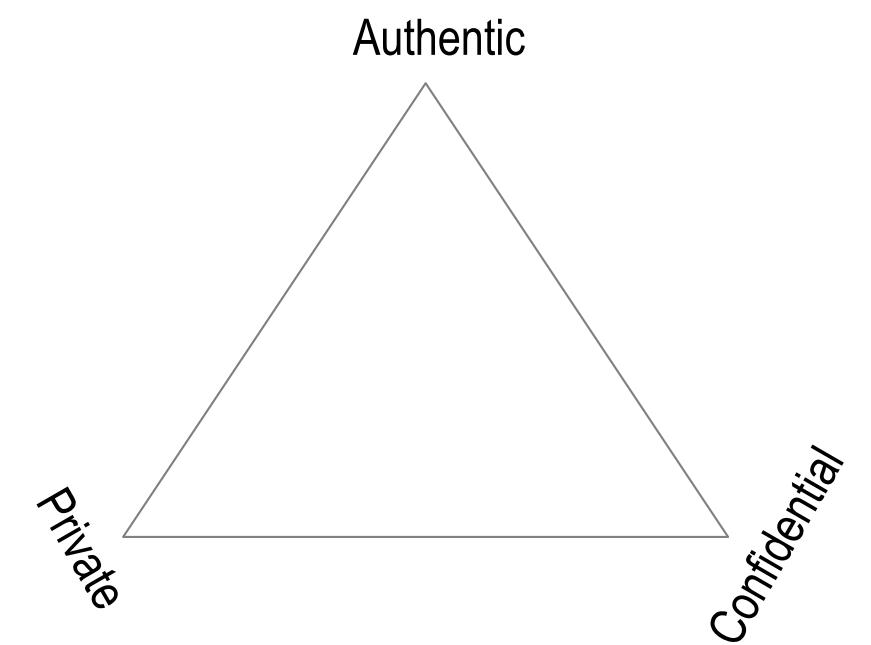
# Layering

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)

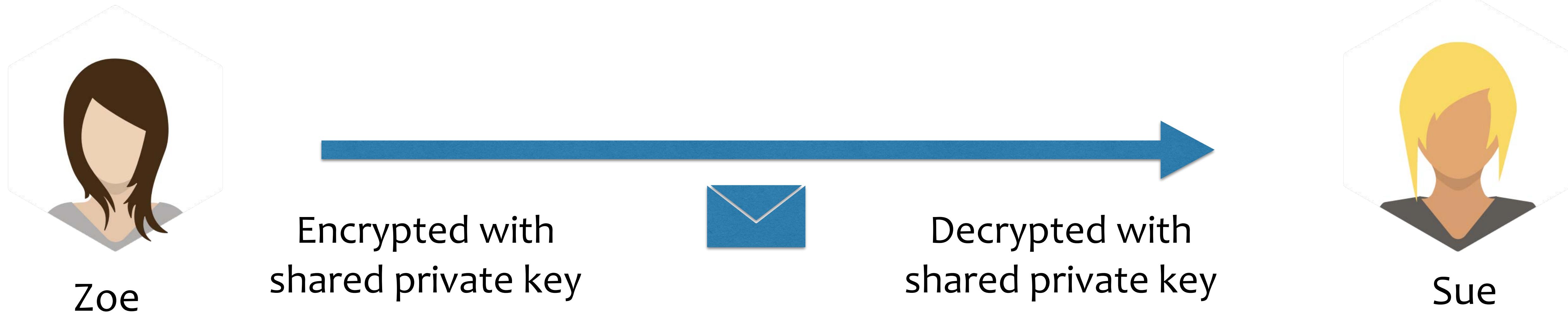
Data Supply Chains (Authentic data economy)

# Non-Repudiable Authenticity



Non-repudiable authenticity is zero-trust

# Repudiable Authenticity



Repudiable authenticity requires trust (is not zero-trust)

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



Encrypted with  
shared private key



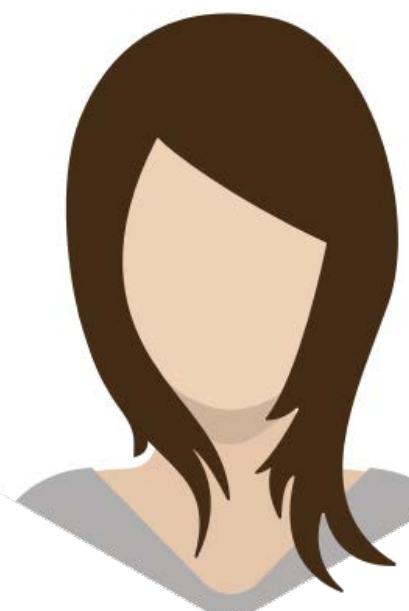
Decrypted with  
shared private key



Non-Repudiable authenticity has recourse.  
Best fits current business and regulatory eco-systems.

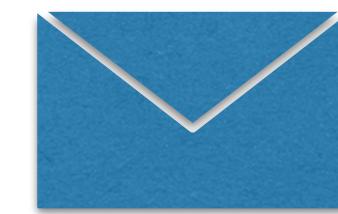
# Zero Knowledge Proof?

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

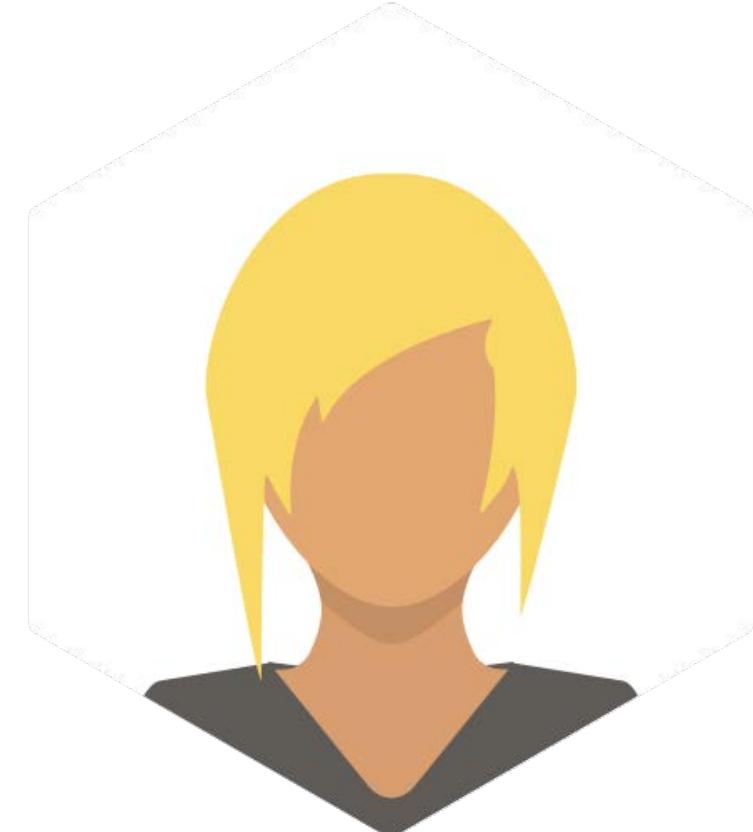


Zoe

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.



ZKP



Sue

ZKPoK (selective disclosure)

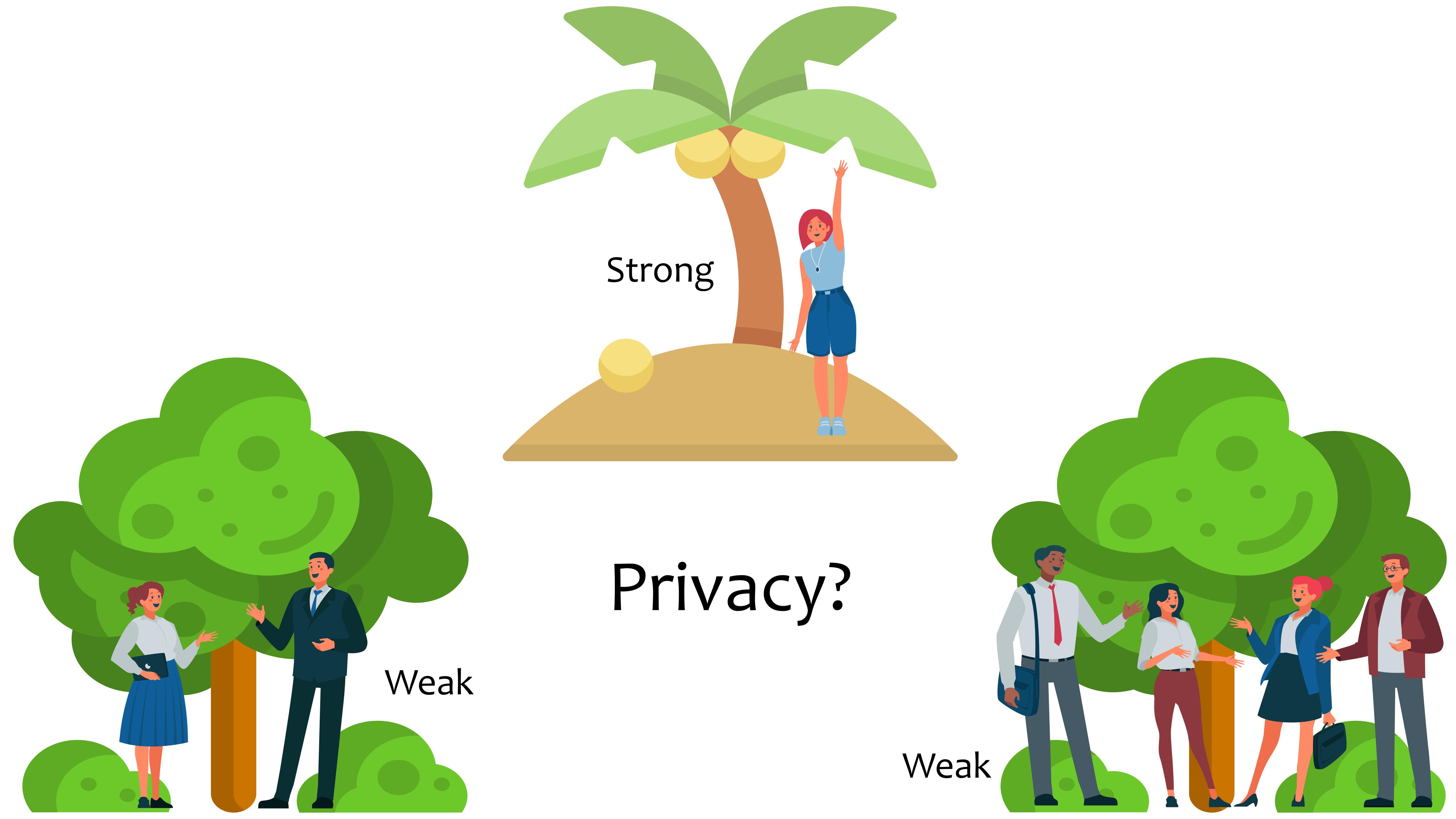
Zoe proves to Sue that a given statement is true without enabling Sue to prove to a third party that the statement is true and protecting Alice from Sue as a forger.

(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 Privacy

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

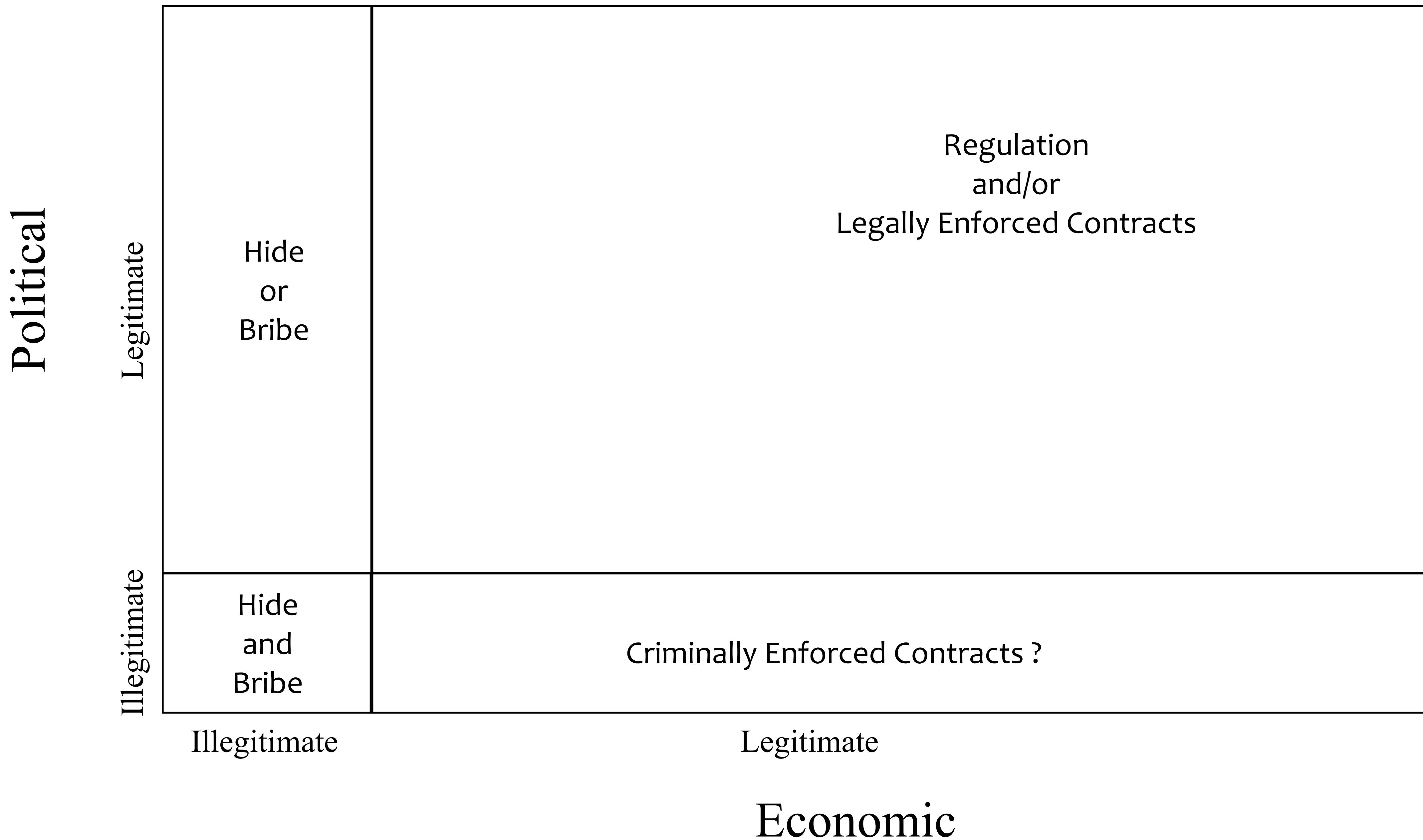
Super aggregators and state actors have effectively unlimited storage and compute capacity. Eventually all disclosed data will be at least statistically correlatable.

# Weak Privacy

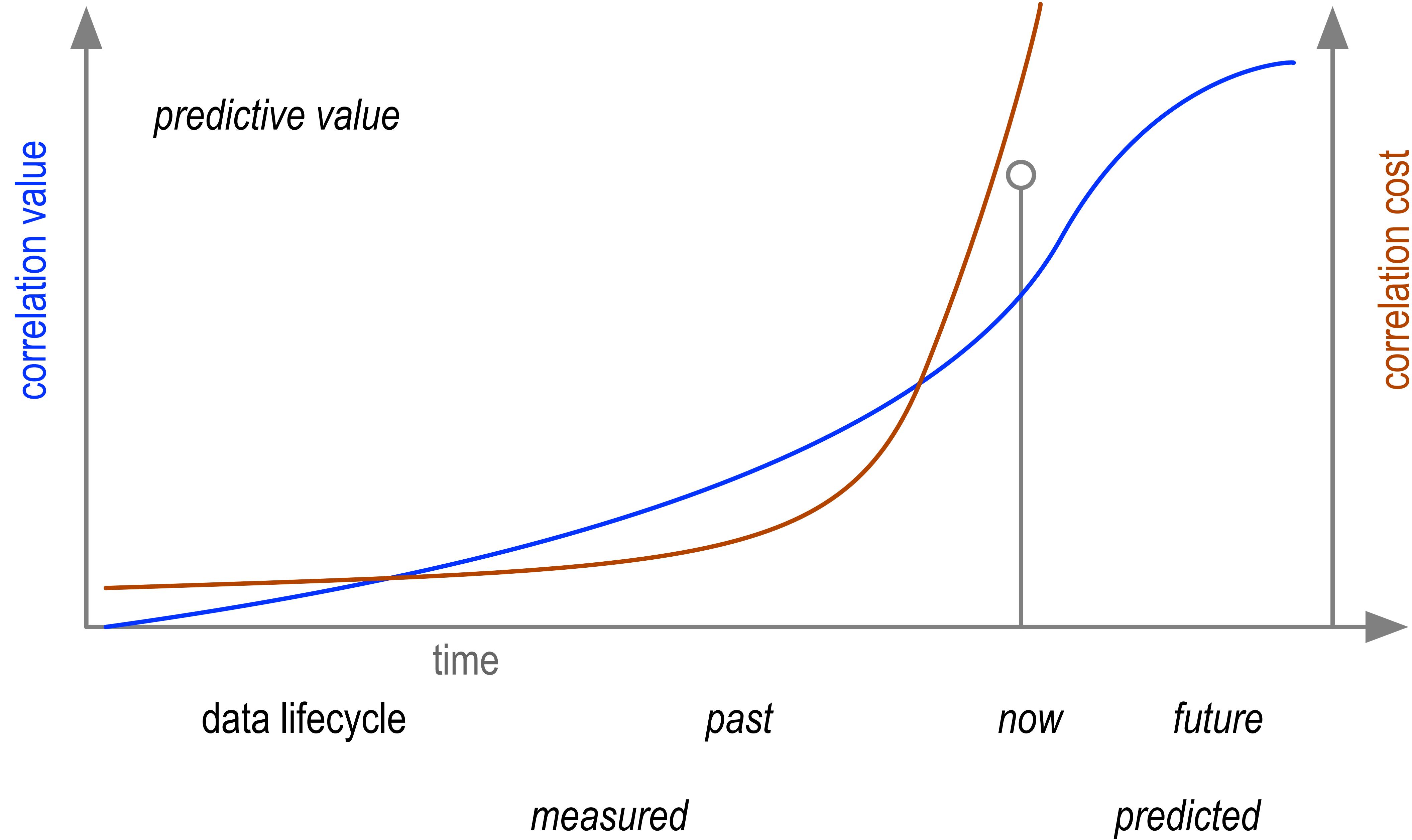
Definition: un-correlated interactions over bounded time and space.

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

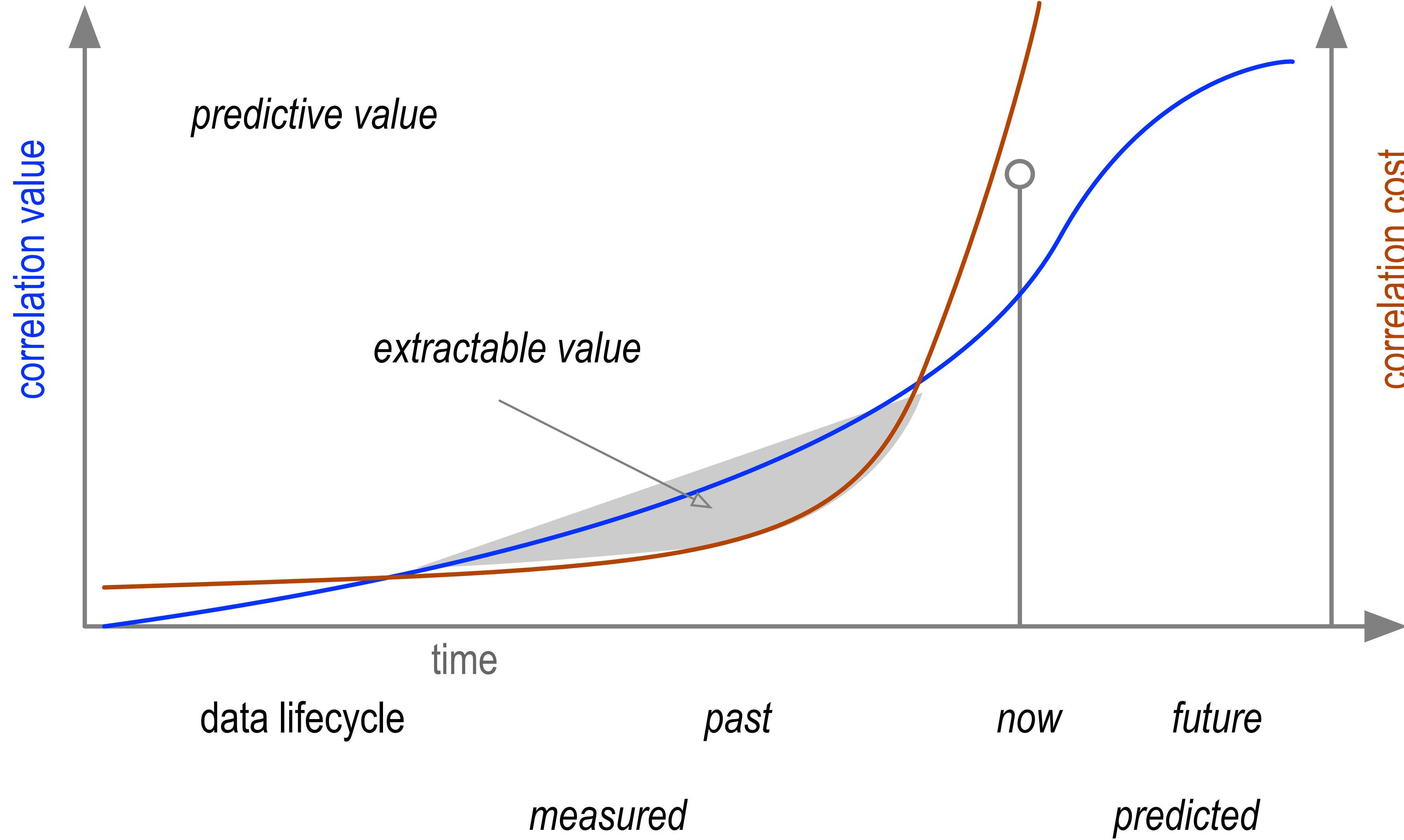
# Operating Regimes



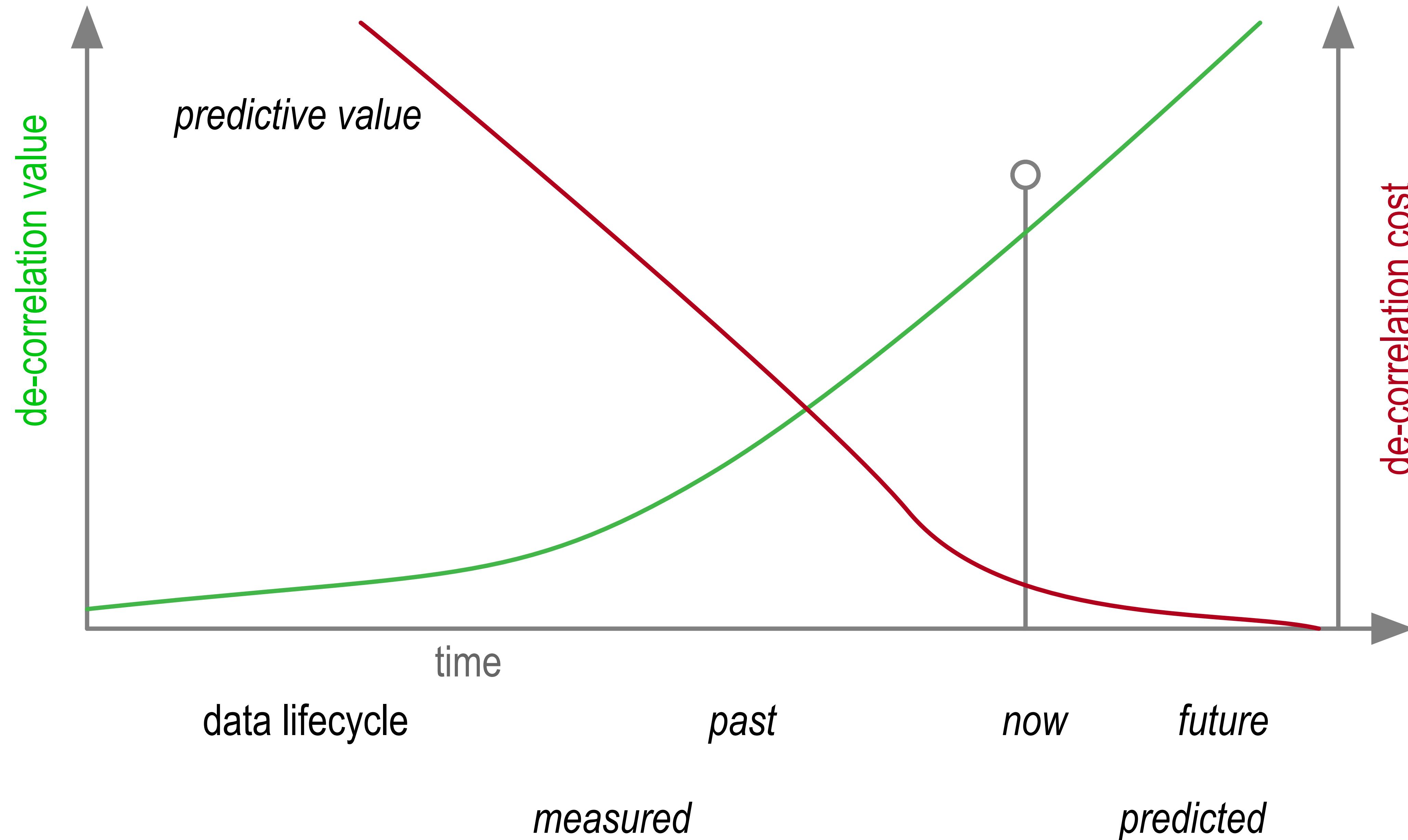
# Economics of Correlator



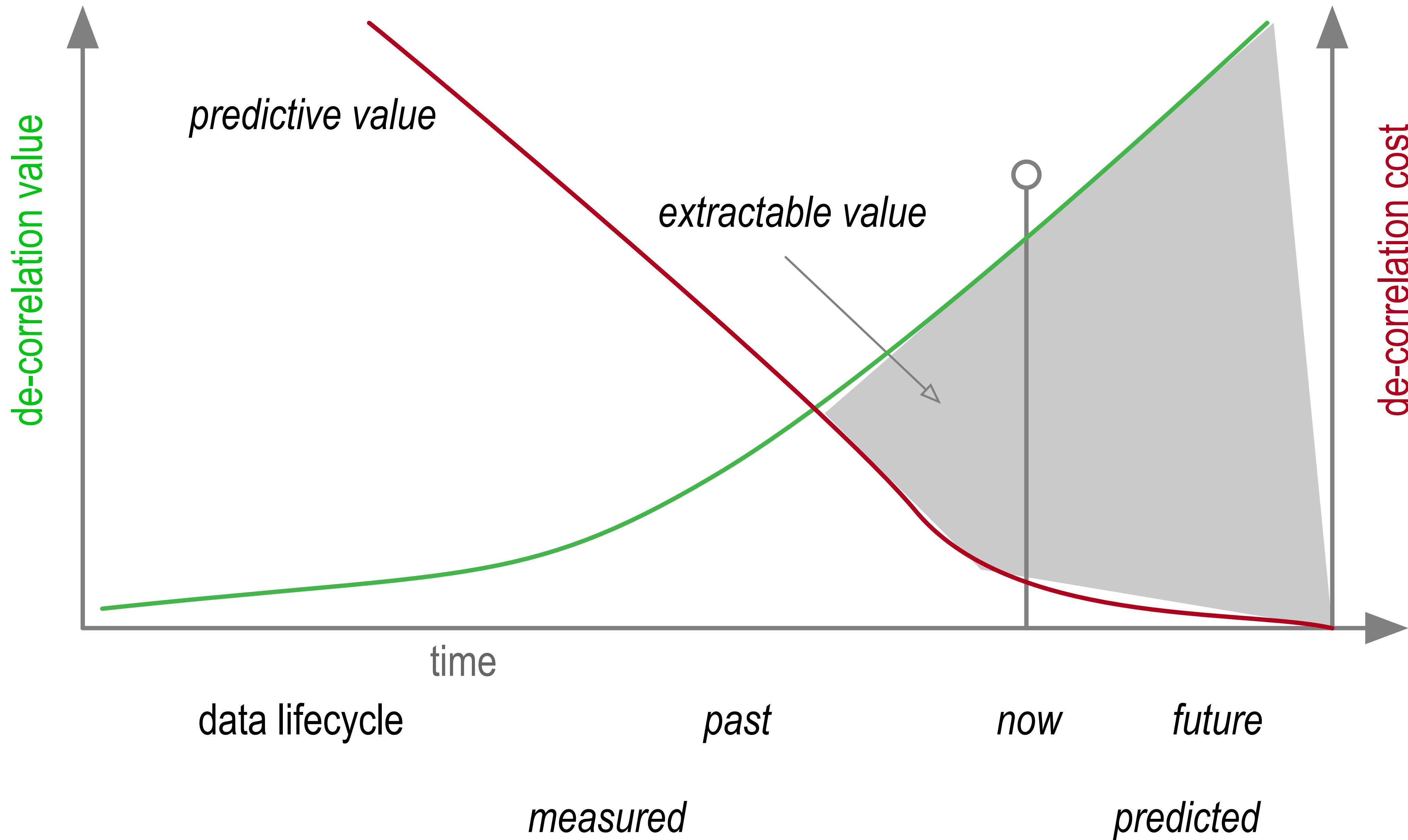
# Economics of Correlator: Value Extraction



# *Economics of De-correlator*



# Economics of De-correlator: Value Extraction



# Freedom *balanced*

Freedom from ...

exploitation (commercial)

intimidation (political)

censorship (political)

Freedom to ...

extract value (commercial)

build relationships (social)

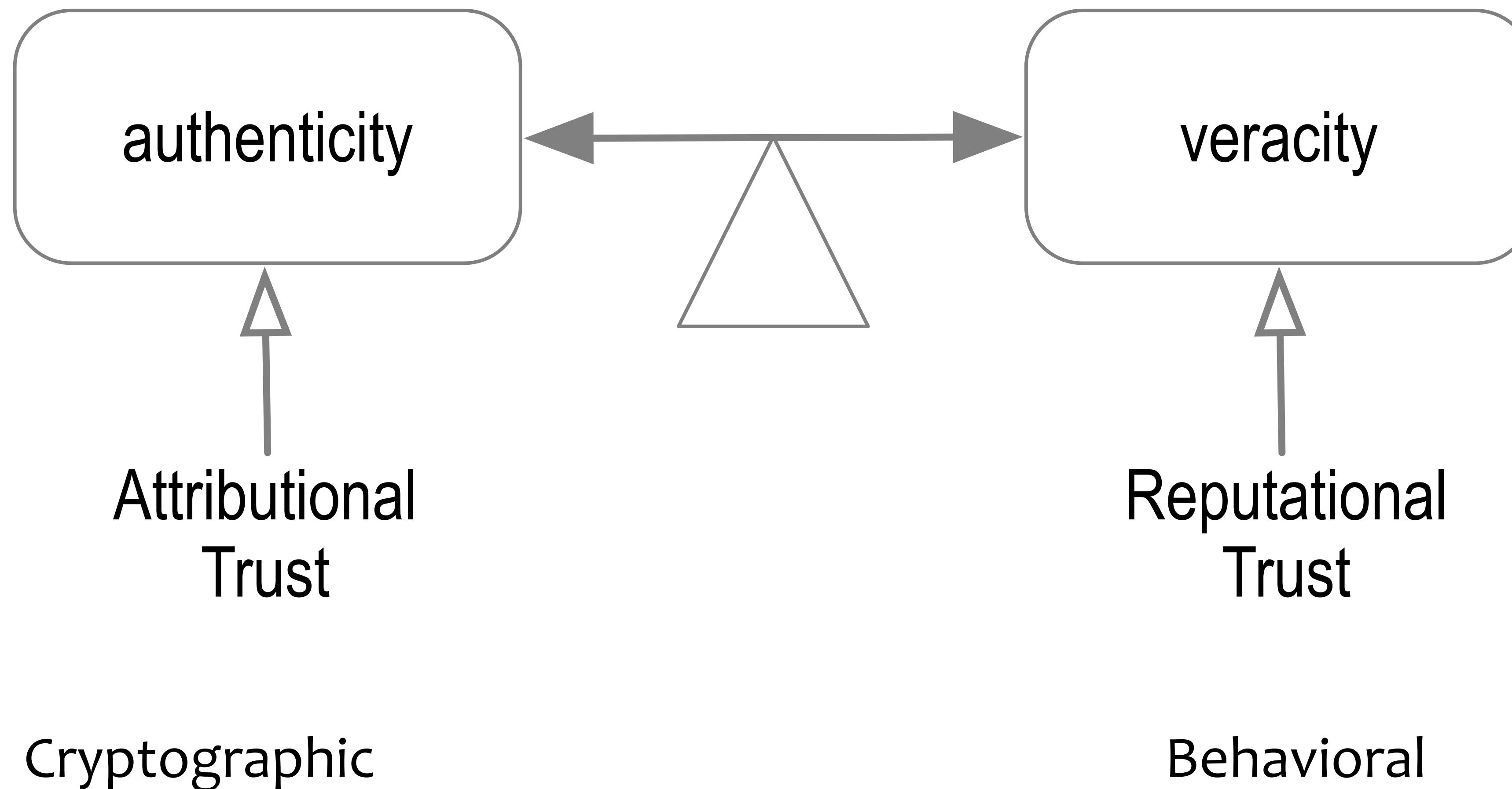
build community (political)

possibility of erasure = possibility of censorship

anonymity = loss-of-value from attribution

fairness = loss of privacy from attribution

# Trust Balance



# Unified Identifier Model

*AID*: Autonomic Identifier (primary)

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

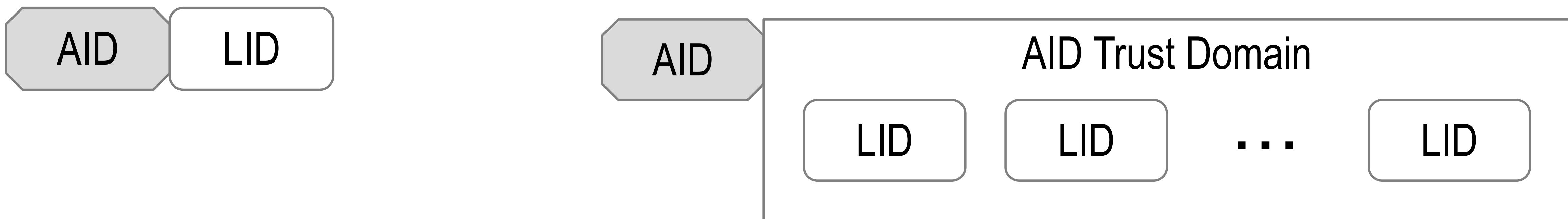
secure, decentralized, portable, universally unique

*LID*: 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  $AID \mid LID$  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:

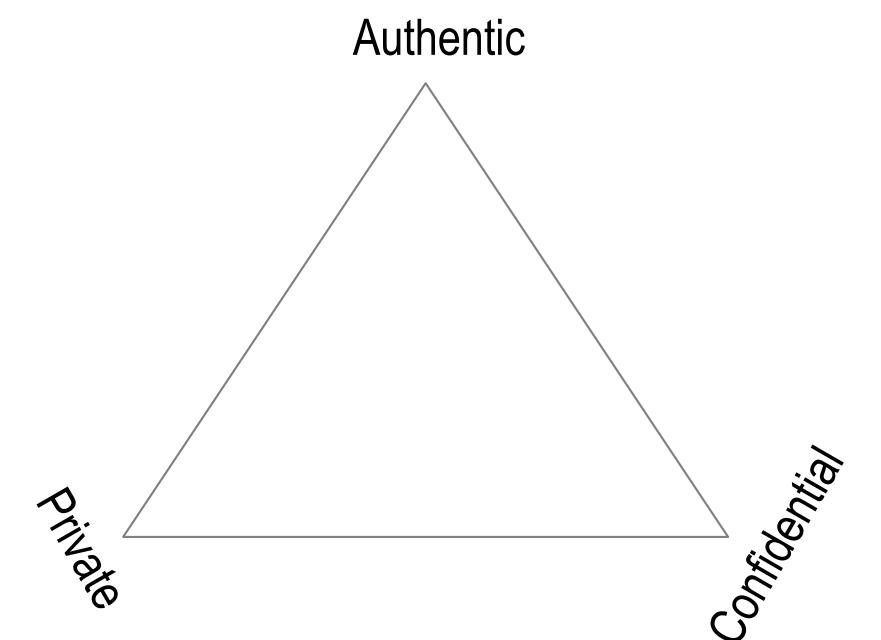
Interactive:

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)

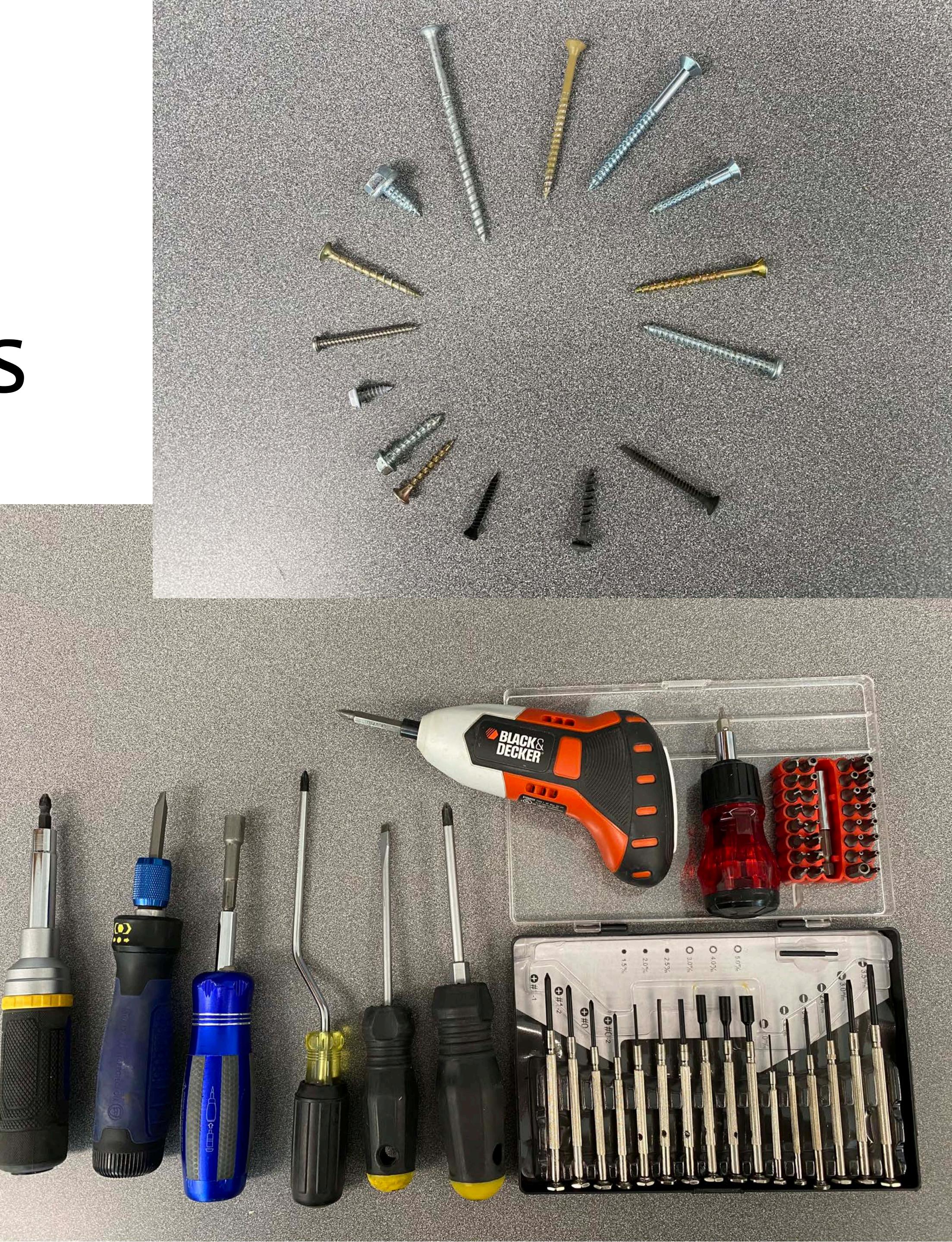
- Accept if no prior record.

- 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!

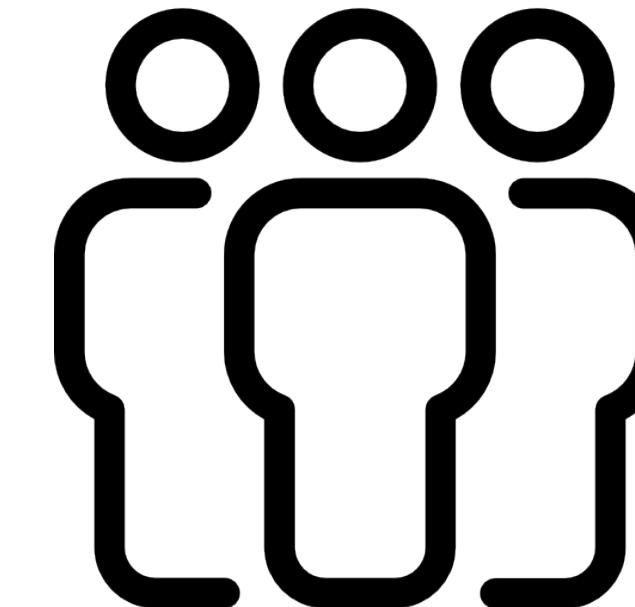
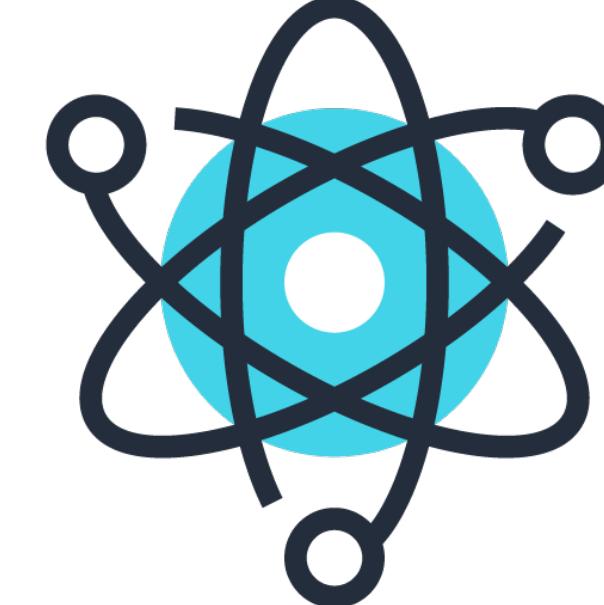
# Entropy Derived Tools

Cryptographic one-way functions ...

hashes, ECC scalar multiplication...

digital signatures, ZKPs ...

its

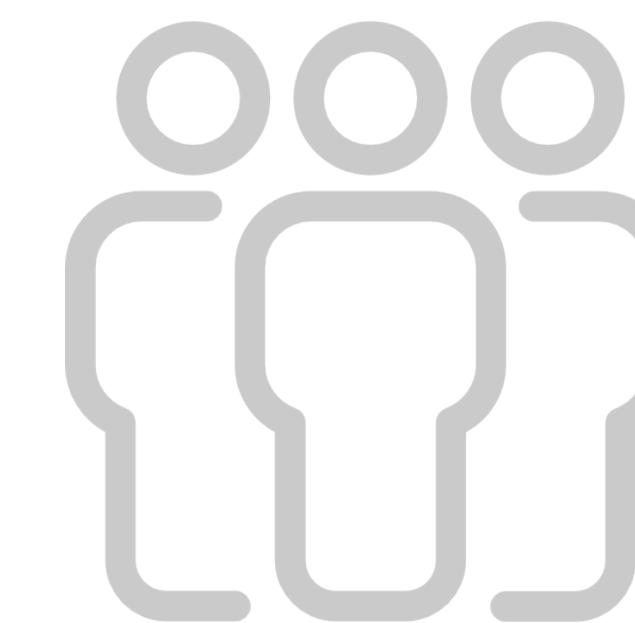


control



correlation

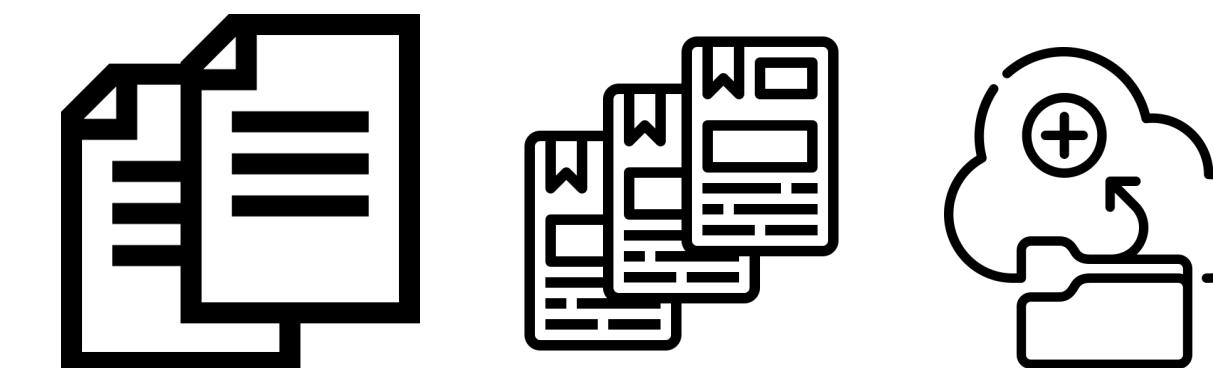
bits



cryptographic  
pseudonymous identifiers

control

attribution



# Tripartite Authentic Data (VC) Model

Issuer: Source of the VC. Creates (issues) and signs VC

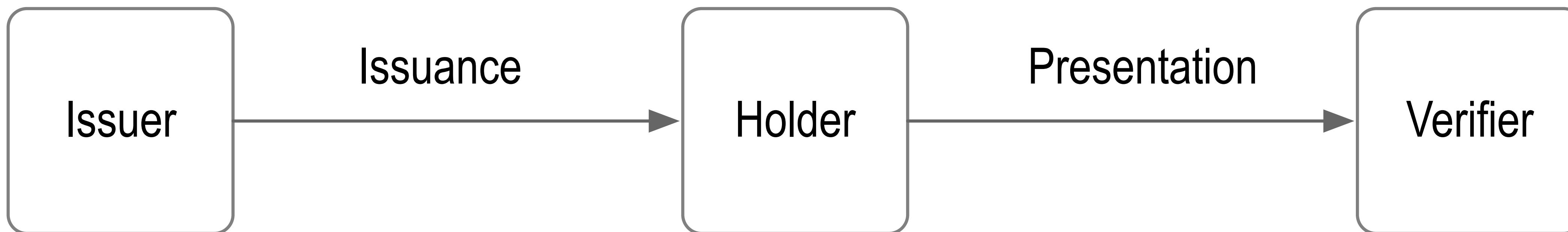
Holder: Usually the target of the VC. The holder is the “*issuee*” that receives the VC and holds it for its own use.

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.

Issuer-Holder-Verifier Model

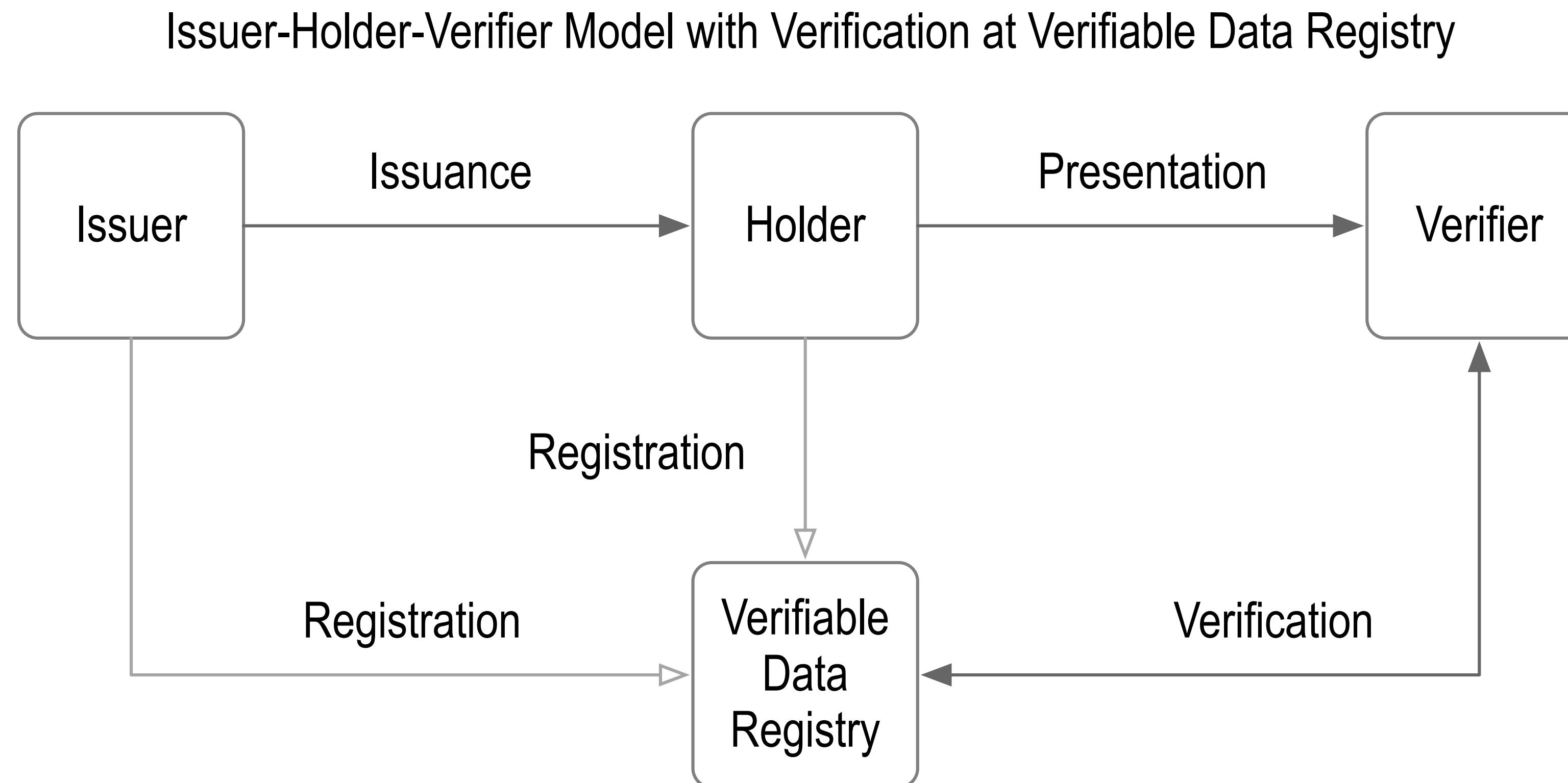


# Tripartite Authentic Data (VC) Model with VDR

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.

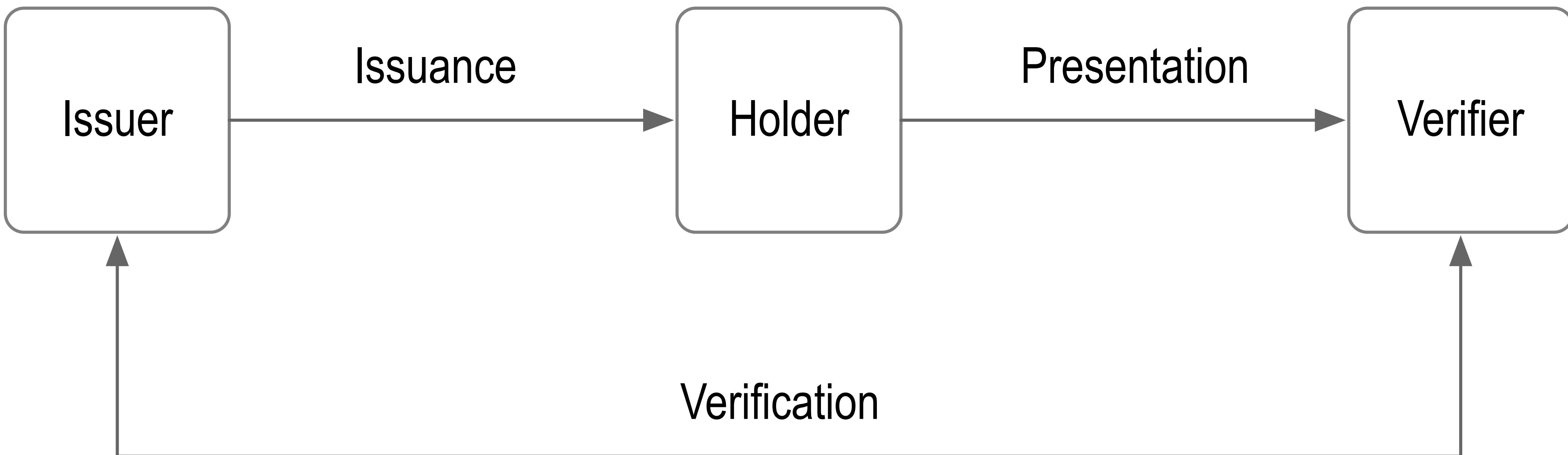
Each controller of a DID registers that DID on a VDR so that a verifier can determine the authoritative key pairs for any signatures.

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



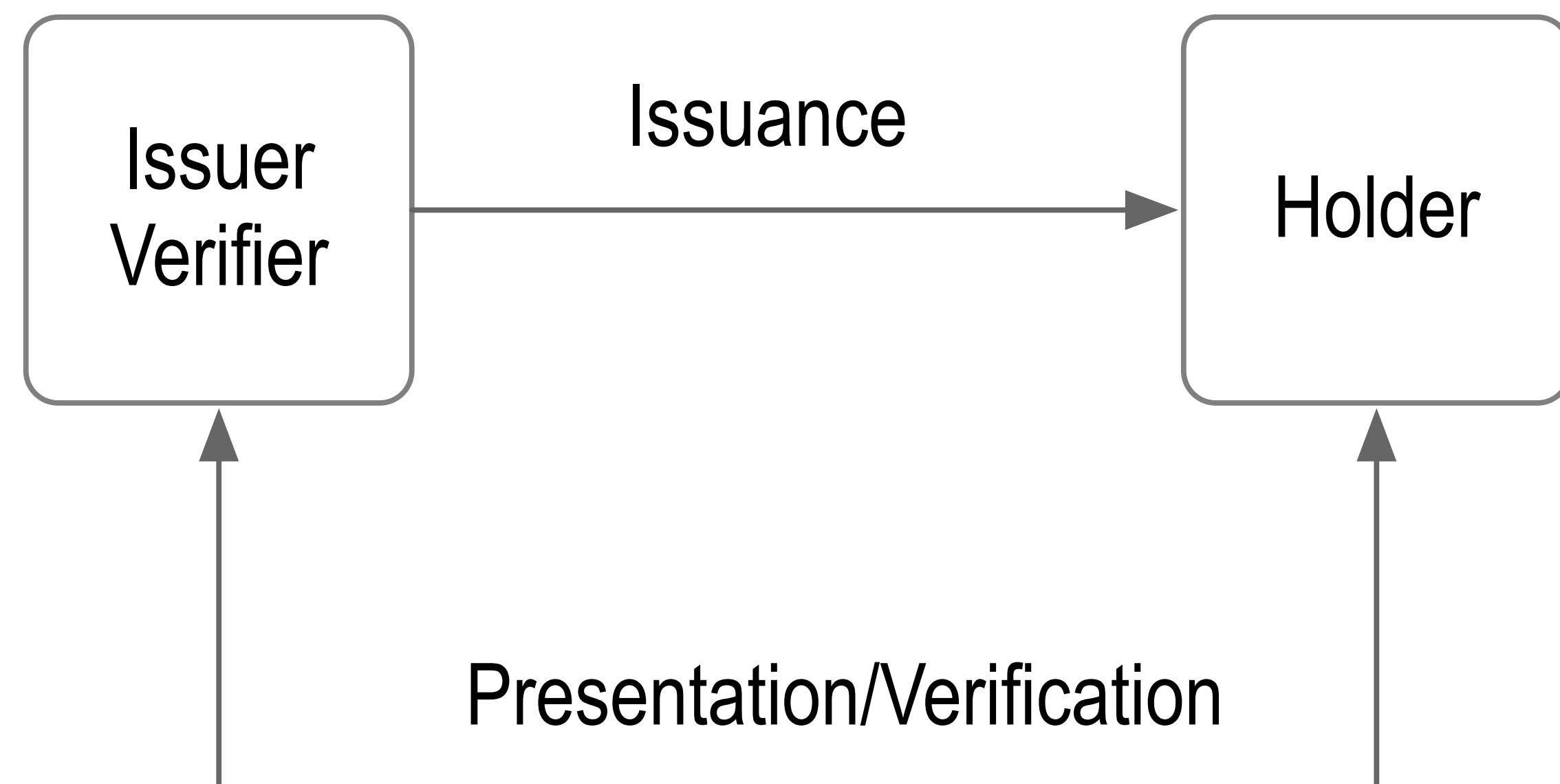
# Tripartite without VDR

Issuer-Holder-Verifier Model with Verification at Issuer

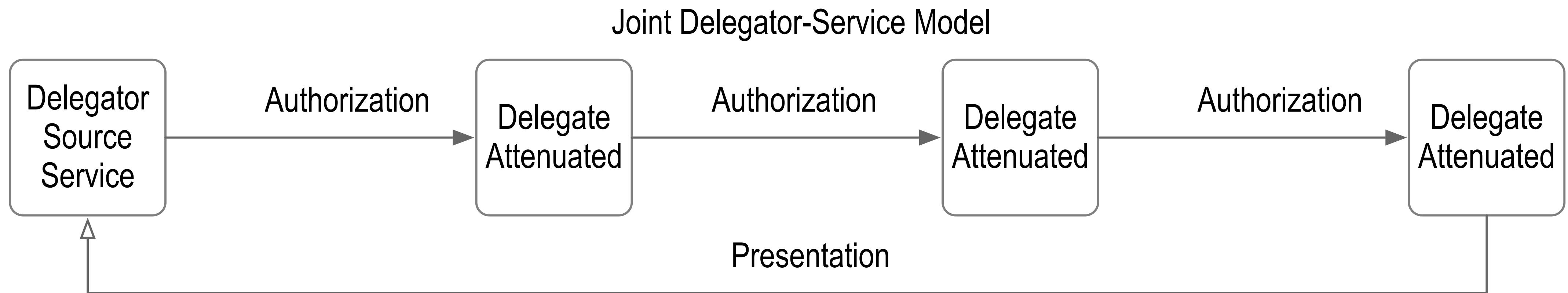


# Bipartite Model

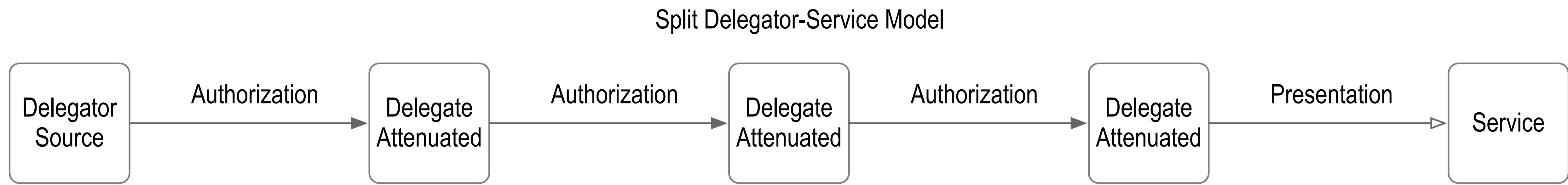
Issuer-Holder Model with Verification at Issuer



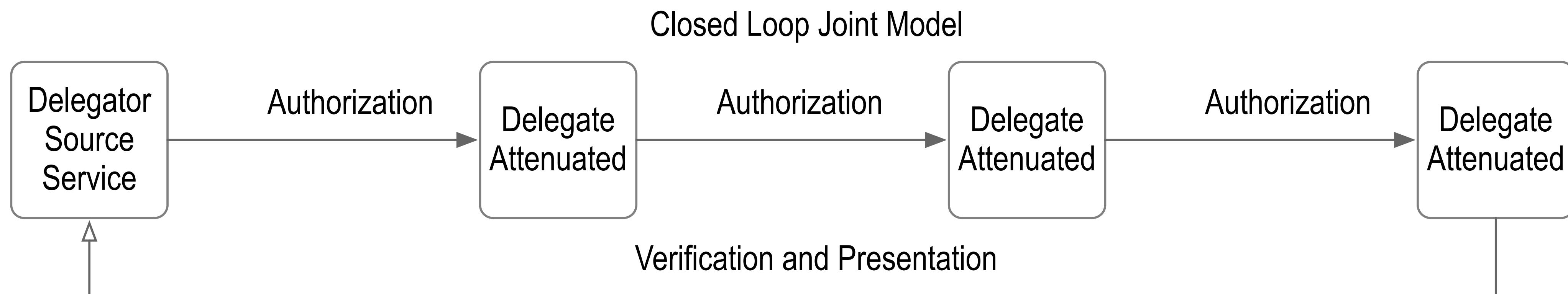
# Joint Delegator-Service Model



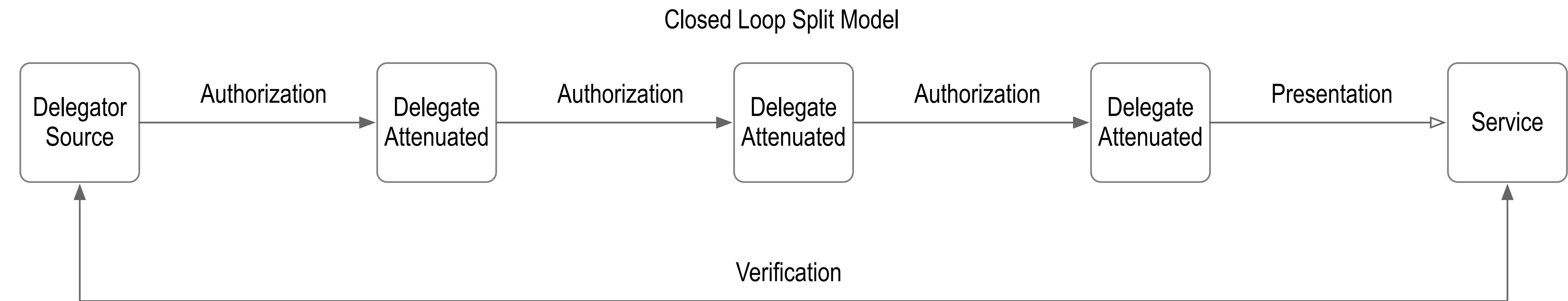
# Split Delegator-Service Model



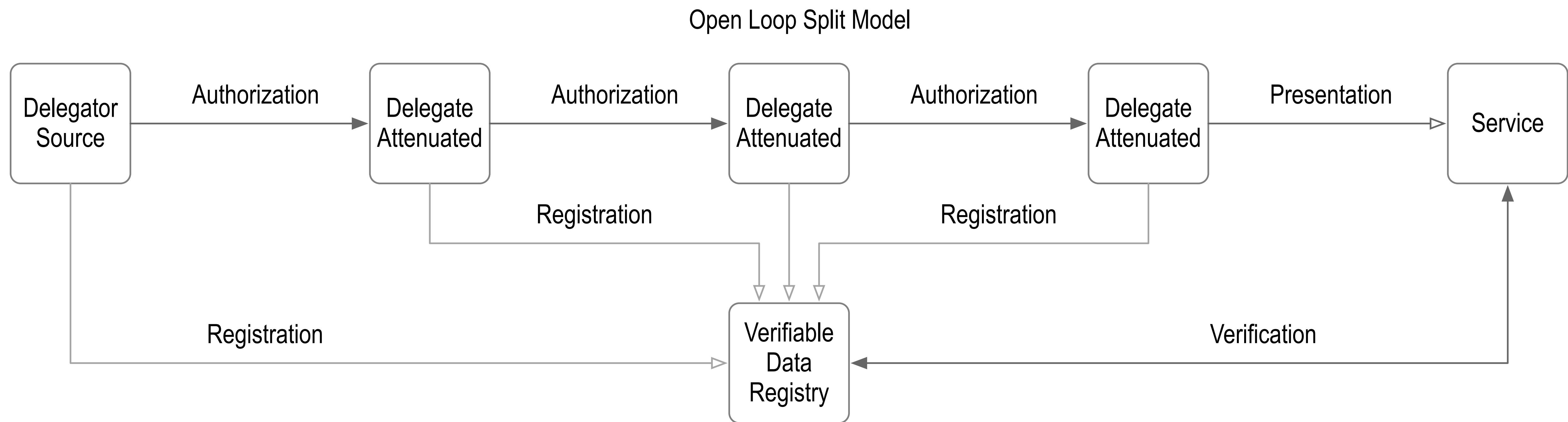
# Closed Loop Joint Model



# Closed Loop Split Model



# Open Loop Split Model

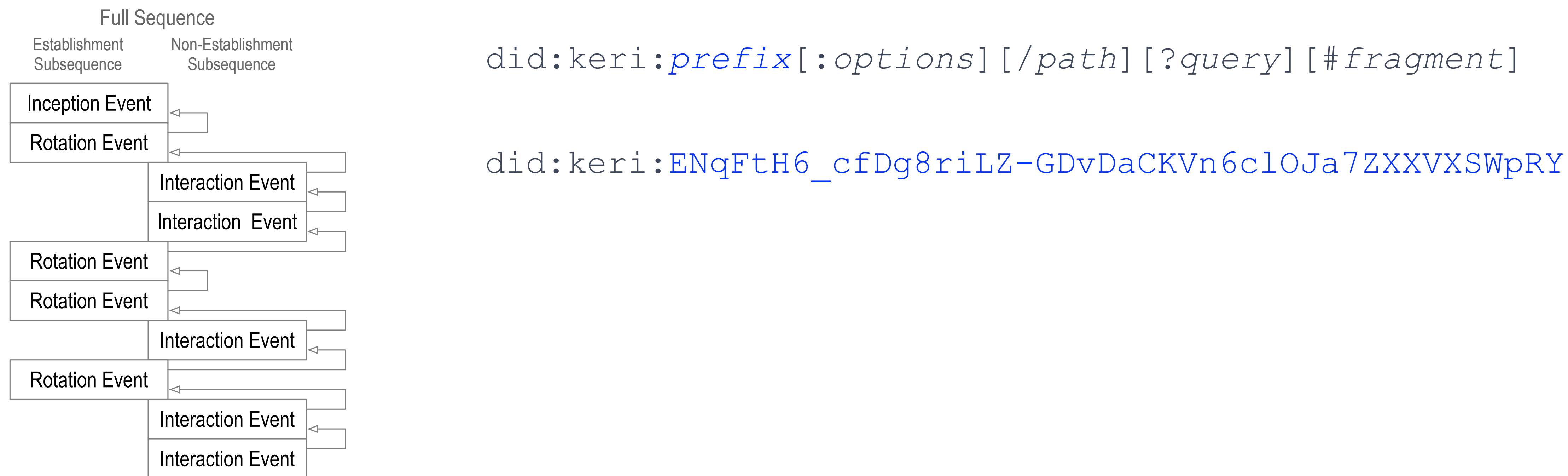


# KERI VDRs vs. Shared Ledger VDRs

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.



# 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.

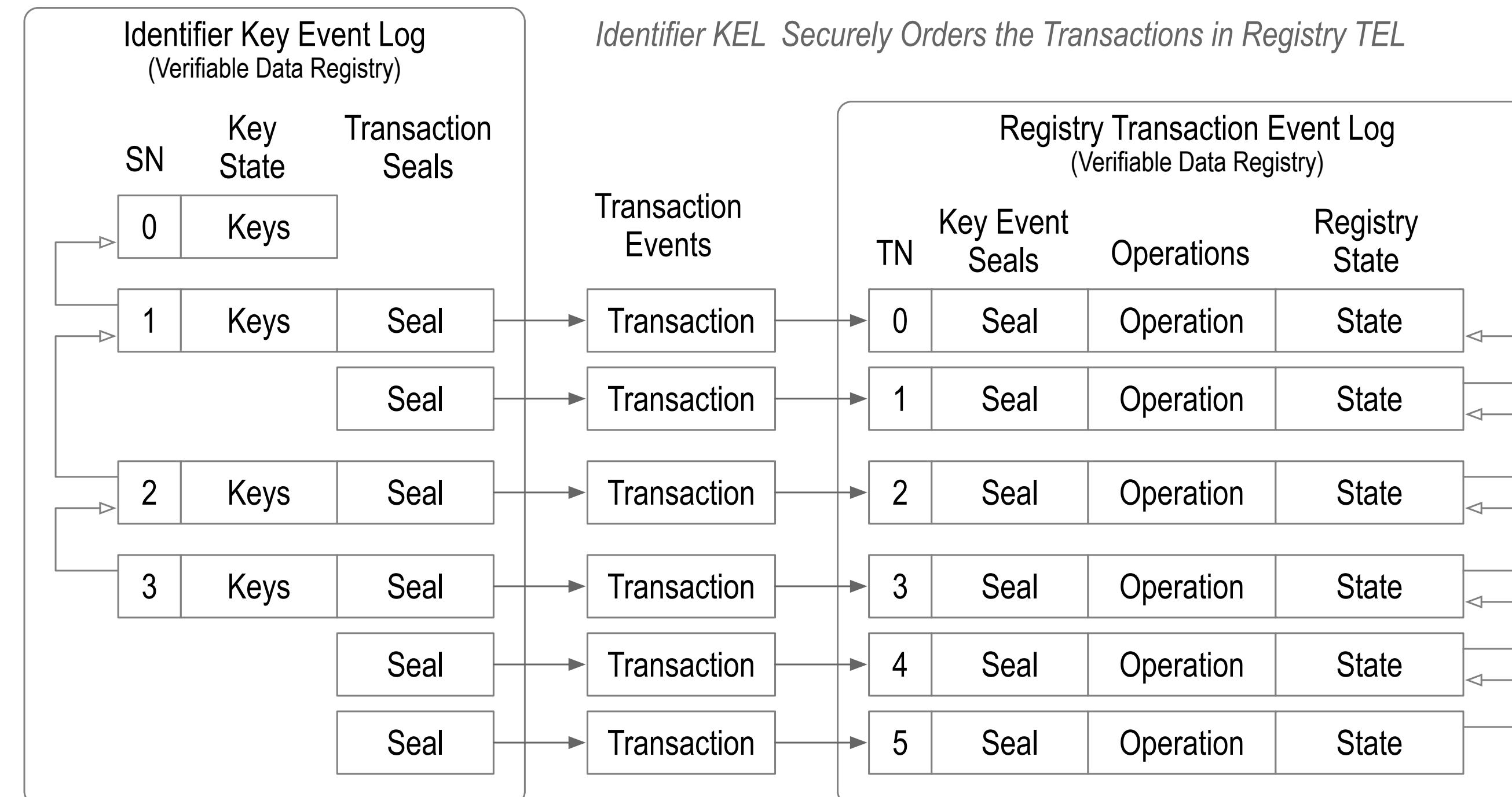
The transactions likewise contain a reference seal back to the key event authorizing the transaction.

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.



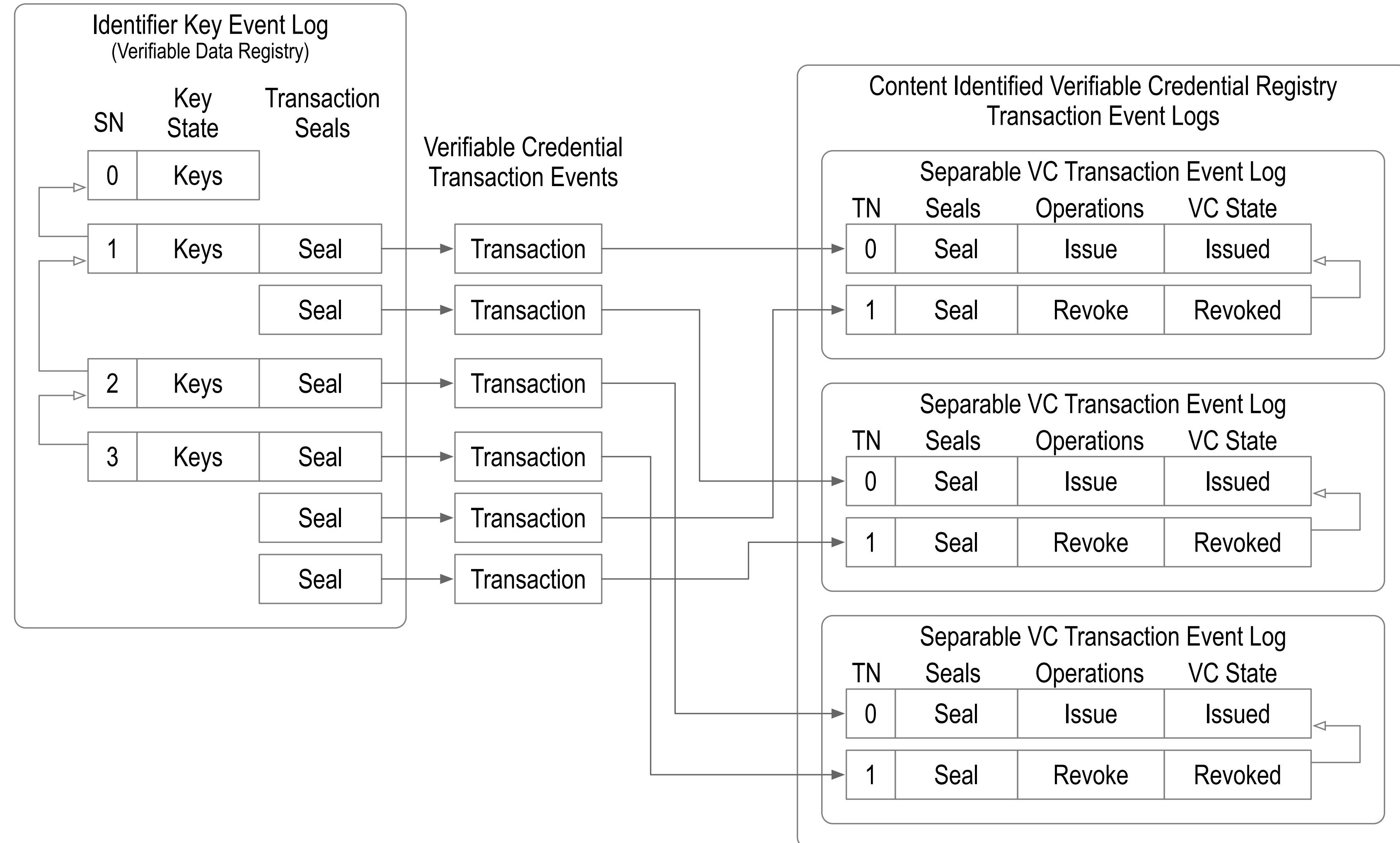
# Registry with Separable VC Issuance-Revocation TELs

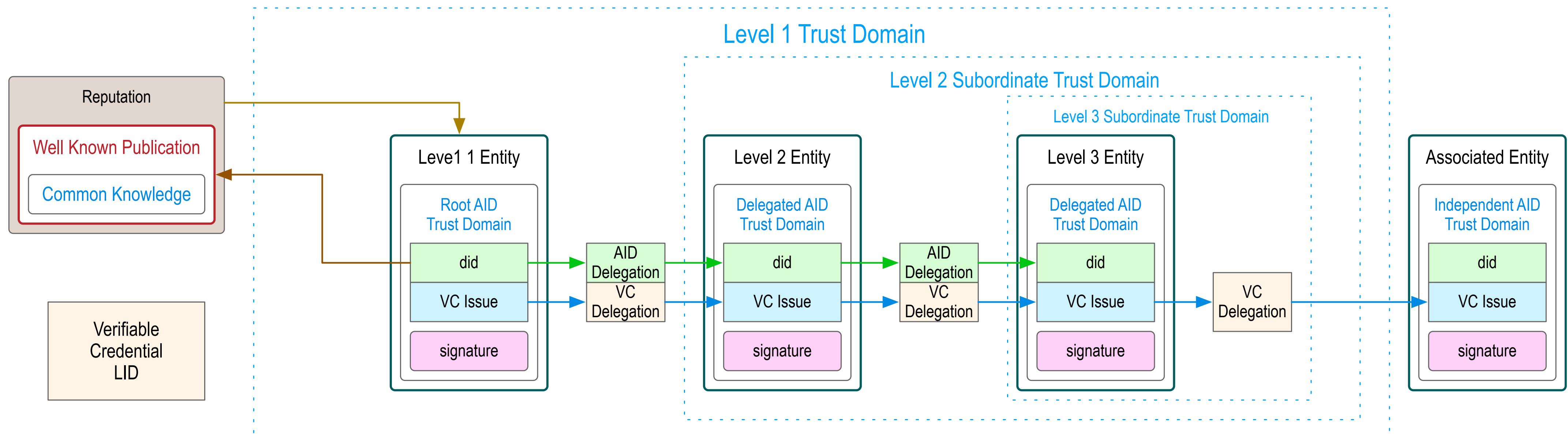
Each VC may be uniquely identified with a SAID.

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

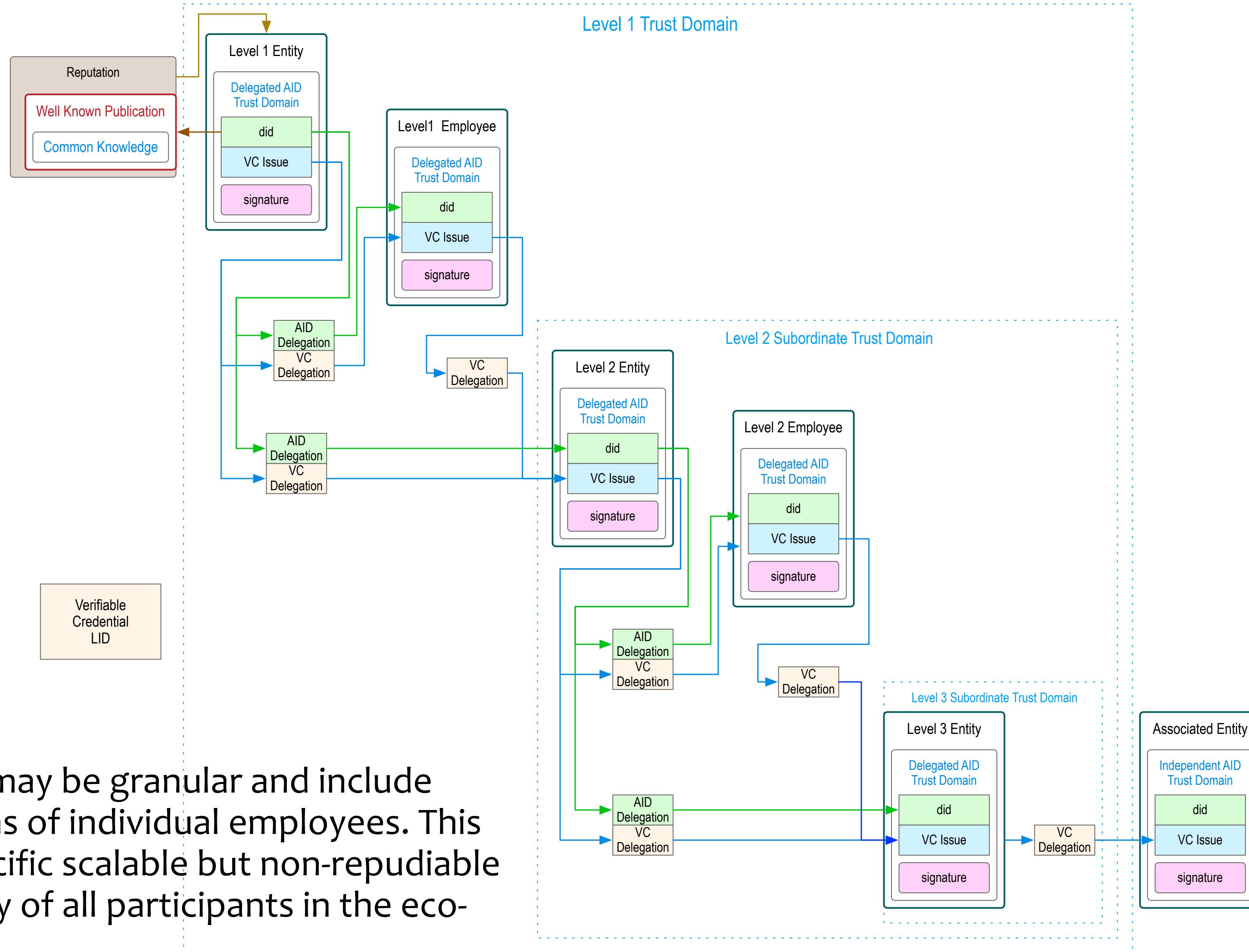




Each level of delegation forms a nested trust domain that is protected by the level above. This increases ultimate security while enabling higher performance event issuance in lower layers.

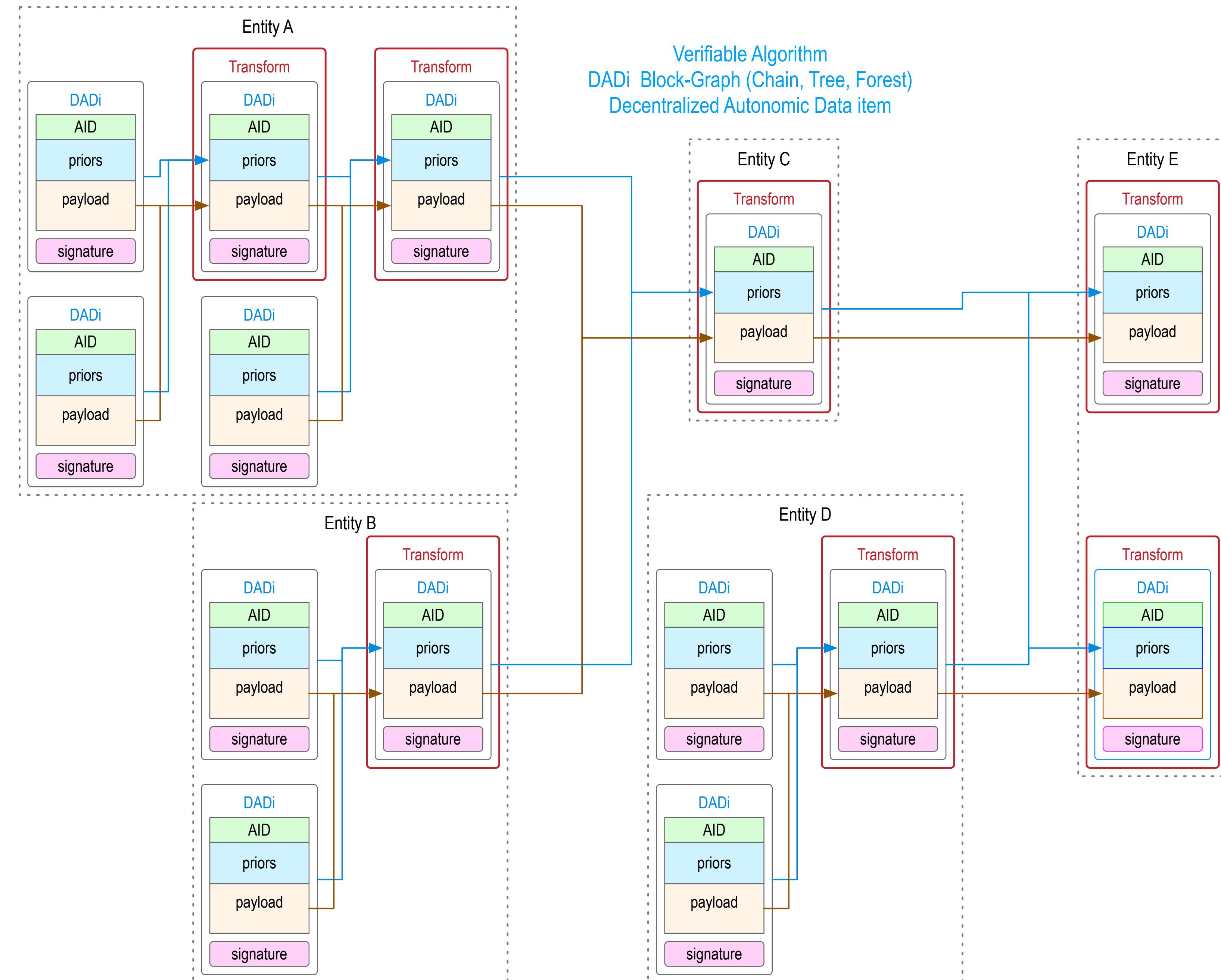
The Level 1 entity AID provides the root-of-trust for the whole ecosystem. This enables secure decentralized interoperability.

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 ecosystem.

**Verifiable Algorithm**  
**DADi Block-Graph (Chain, Tree, Forest)**  
**Decentralized Autonomic Data item**



# Identifier System Security

Authentic transmission of data may be verified using an identity system security overlay.

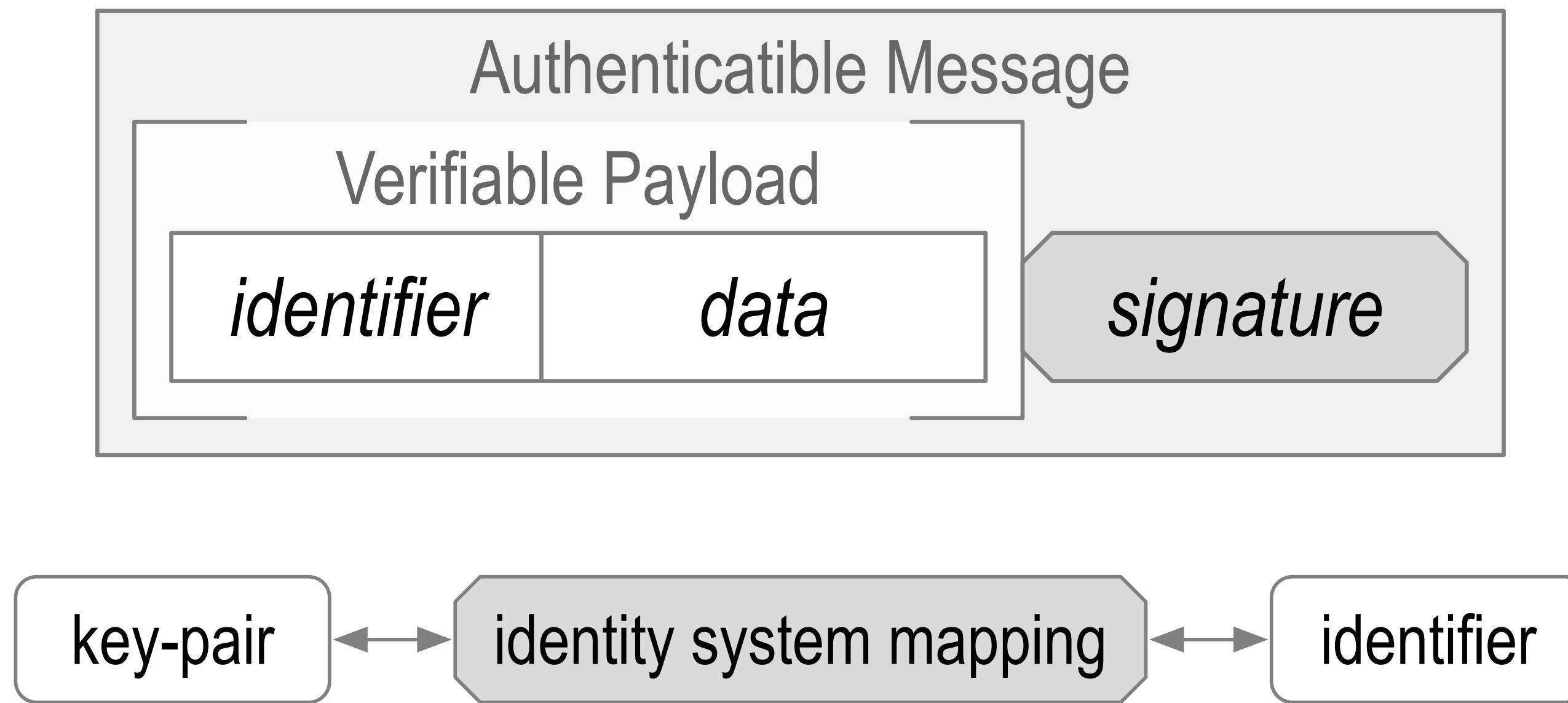
This overlay maps cryptographic key-pairs to identifiers.

When those identifiers are self-certifying they are derived via cryptographic one-way functions from the key pairs.

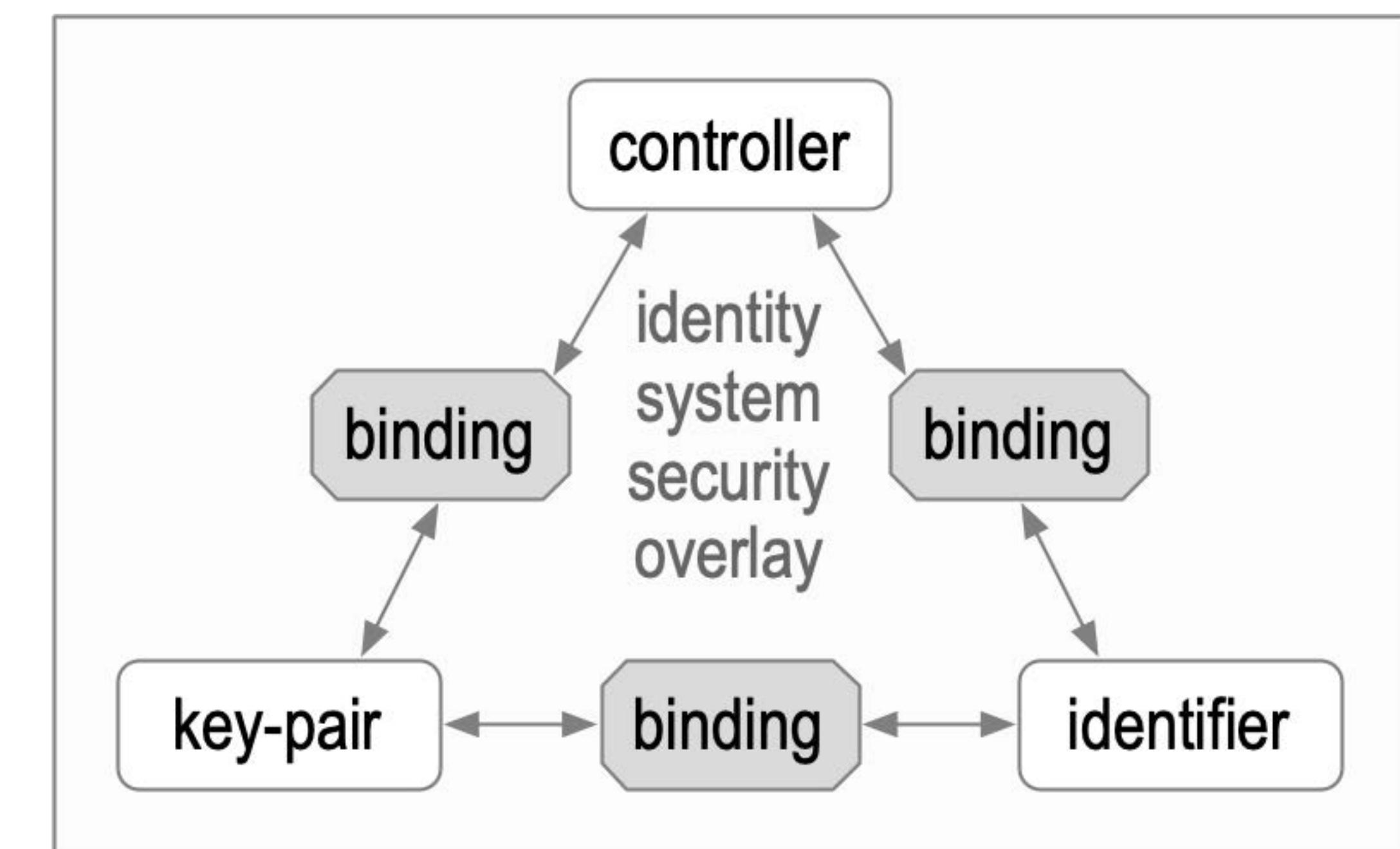
This provides a self-certifying identifier with a cryptographic root-of-trust.

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](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](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](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](mailto:sam@keri.one)

<https://keri.one>



# User Permissioned (web-of-trust) Percolated Discovery

## Invasion-Percolation Graph Theory for attack resistance

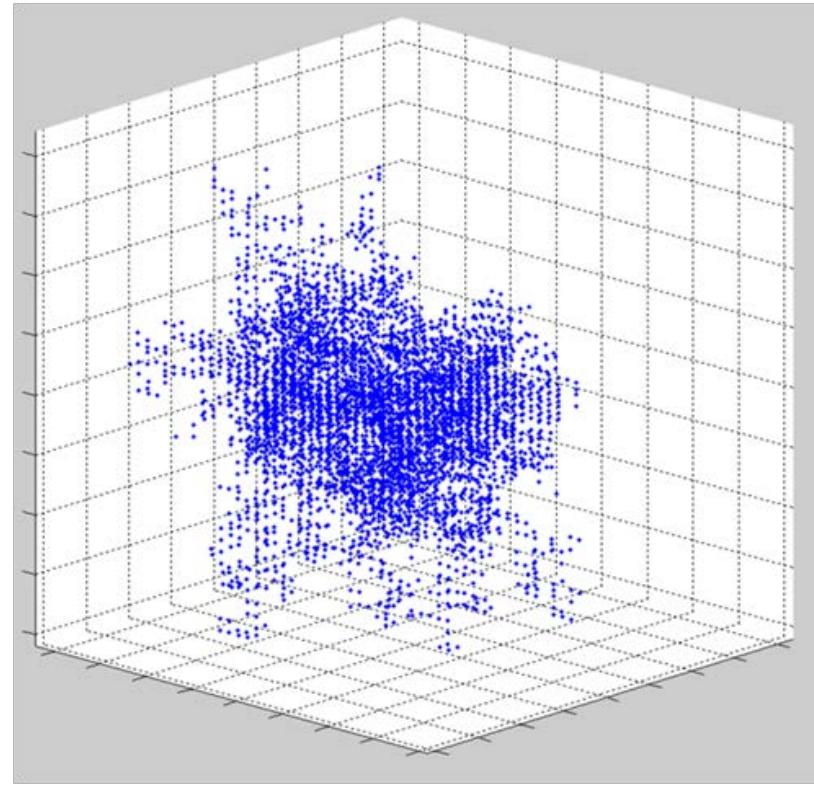
[https://en.wikipedia.org/wiki/Percolation\\_theory](https://en.wikipedia.org/wiki/Percolation_theory)

[https://en.wikipedia.org/wiki/First\\_passage\\_percolation](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 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! Vacuumous 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](.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:

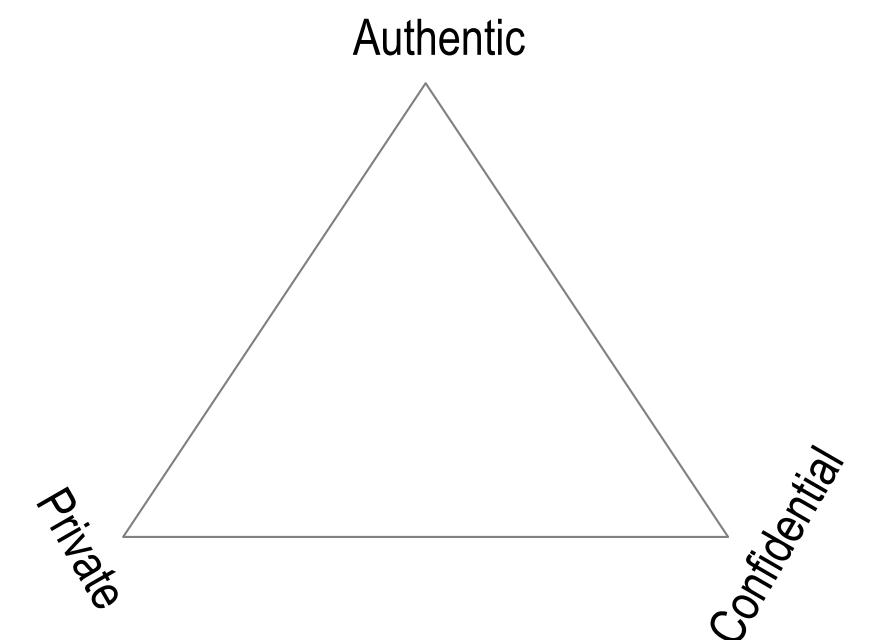
Interactive:

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

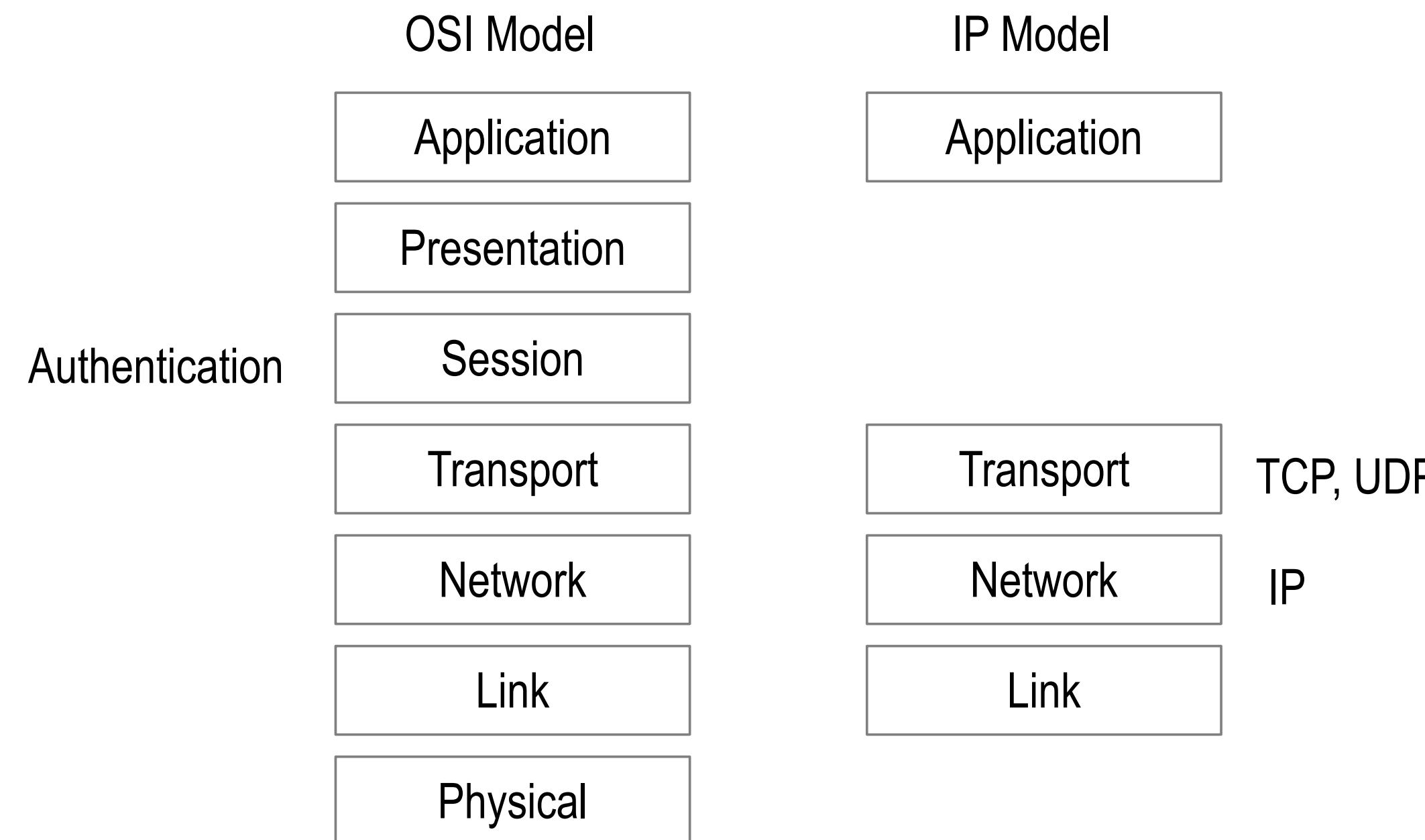
Datetime stamped BADA authorization by CID of EID in Role (Update)

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.

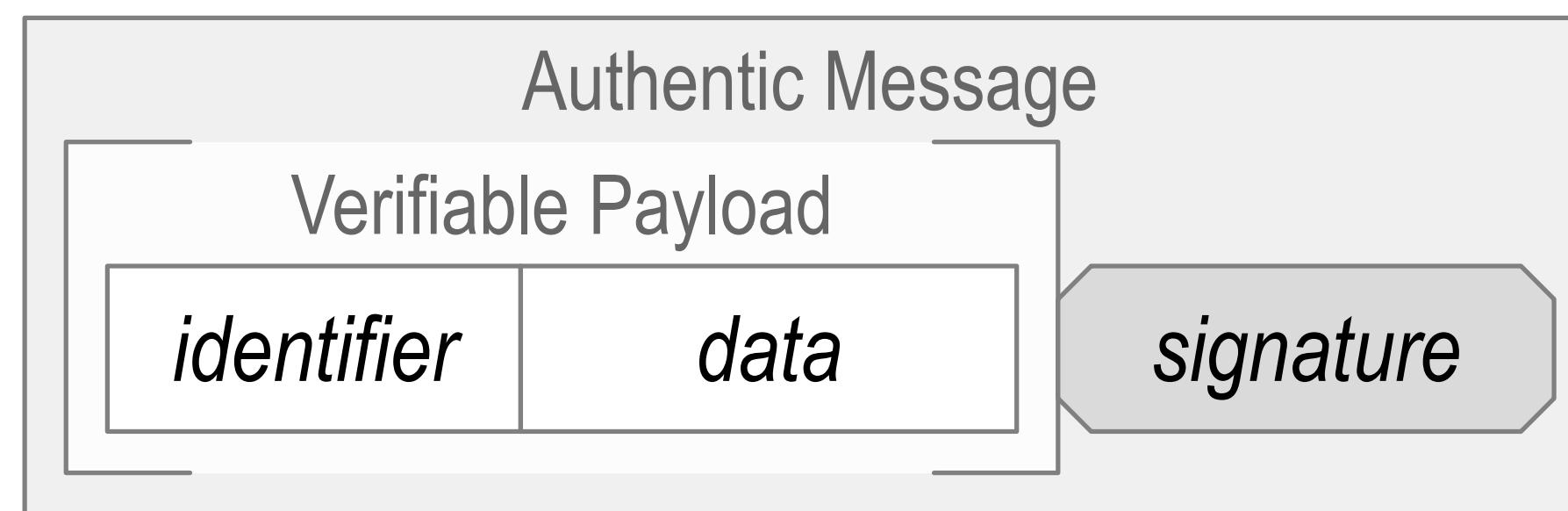
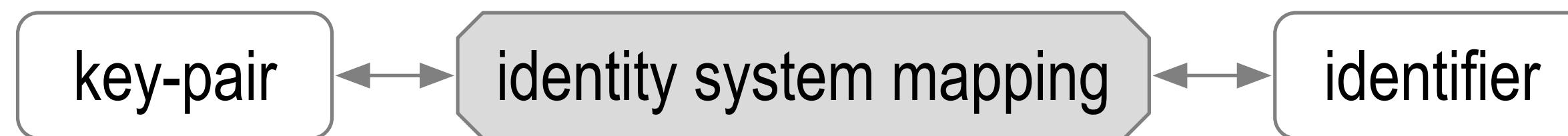


Instead ...

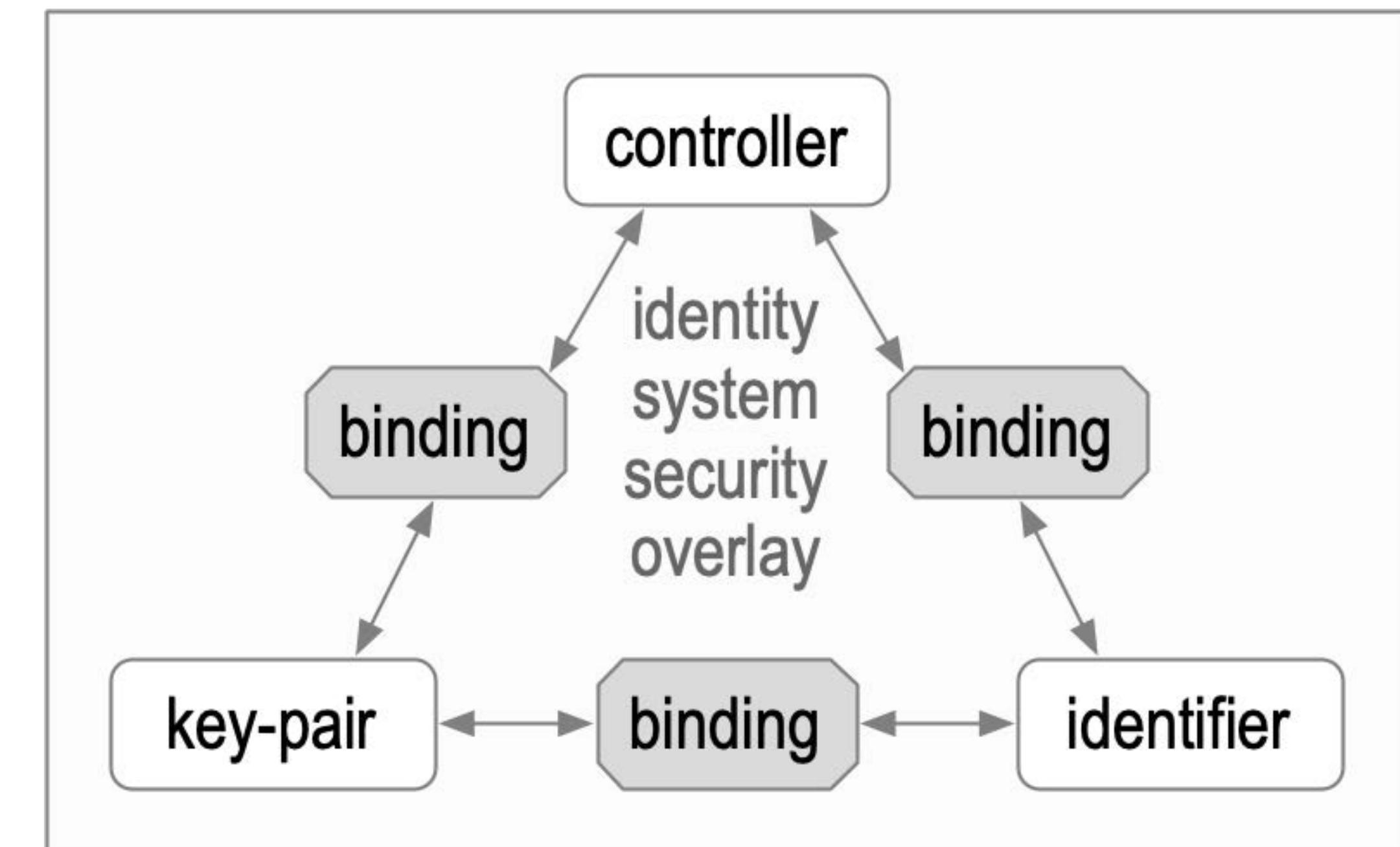
We use *bolt-on* identity system security overlays.  
(DNS-CA ...)

# Identity System Security Overlay

Establish authenticity of IP packet's message payload.

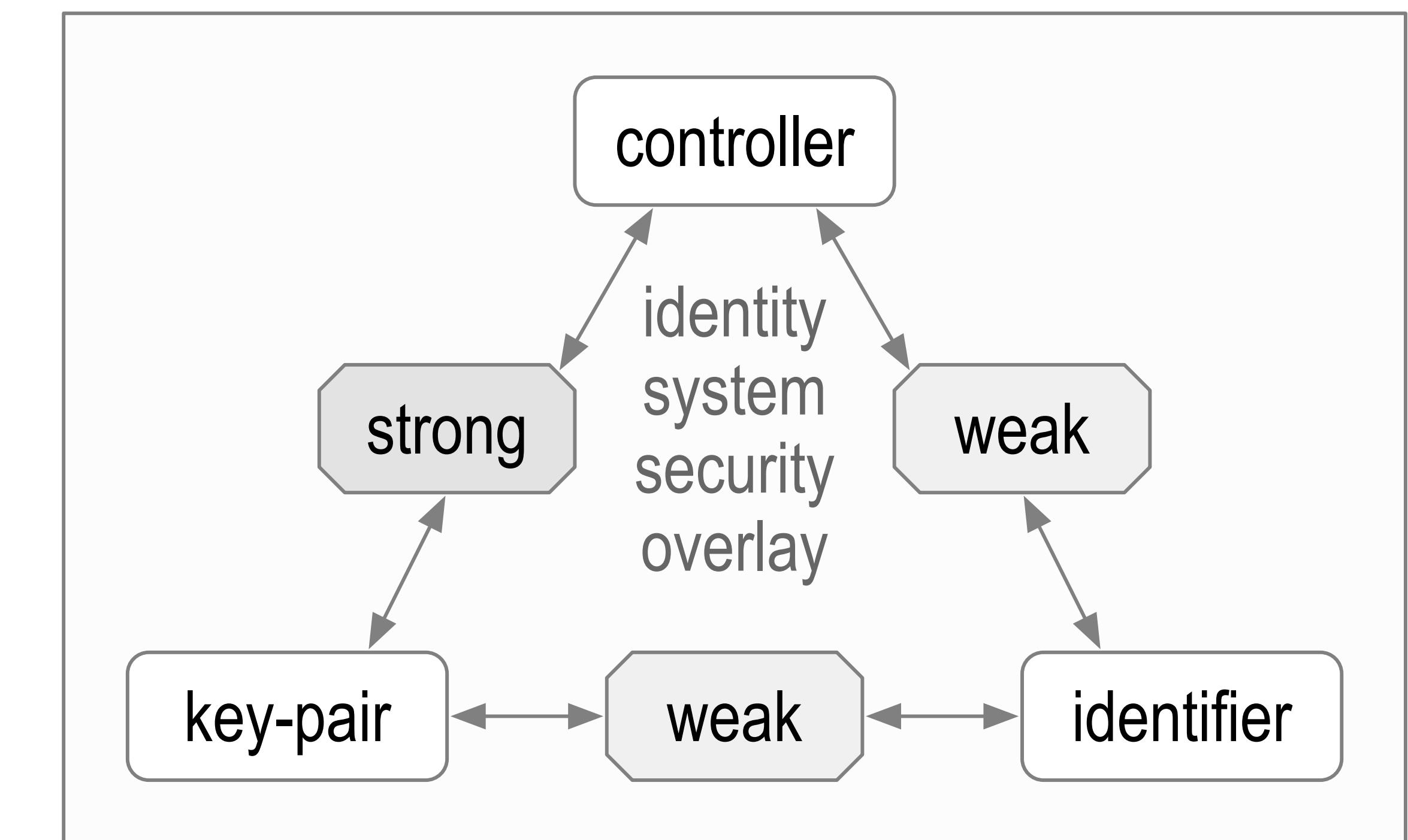
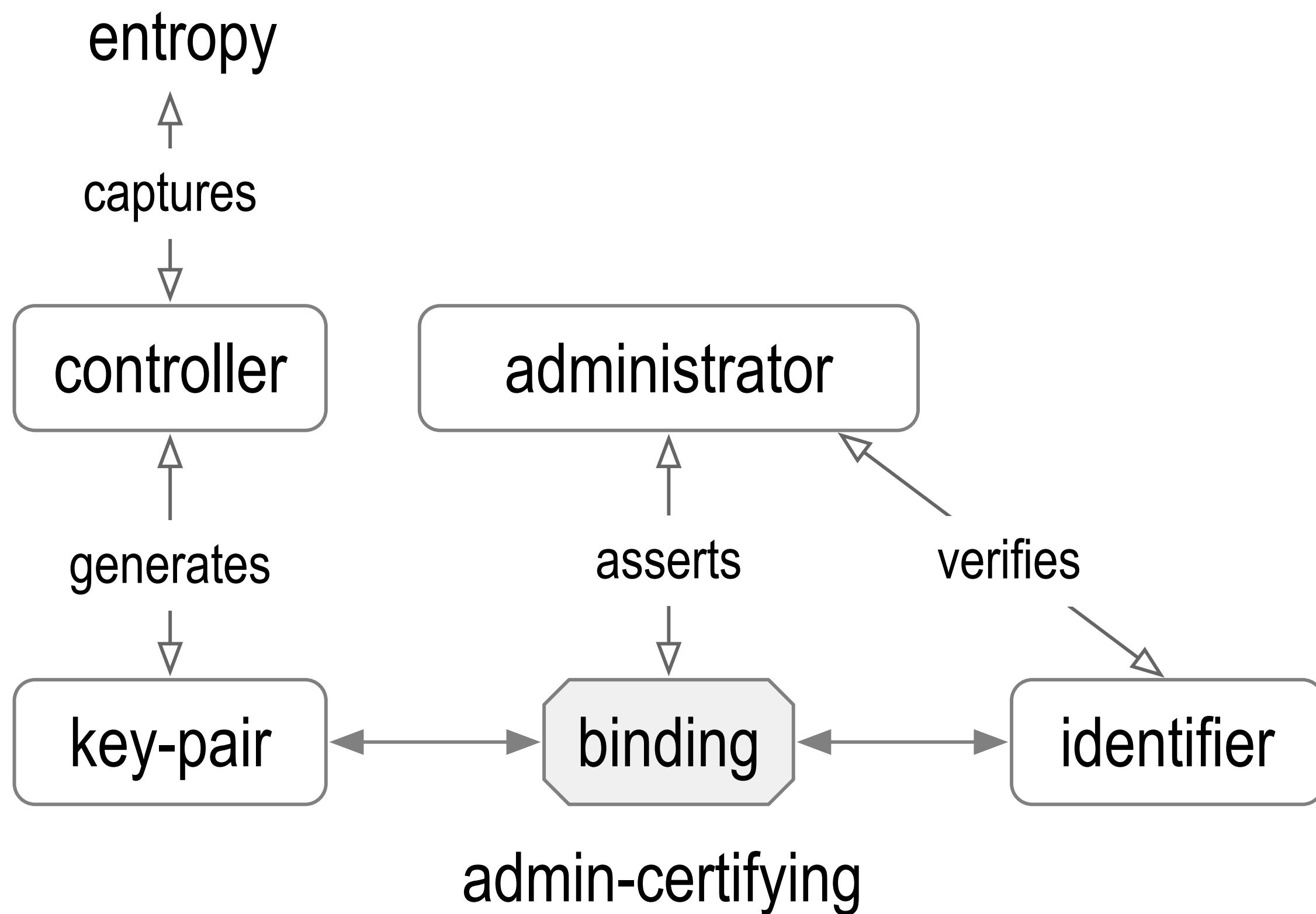


The overlay's security is contingent  
on the mapping's security.



Identifier Issuance

# Administrative Identifier Issuance and Binding

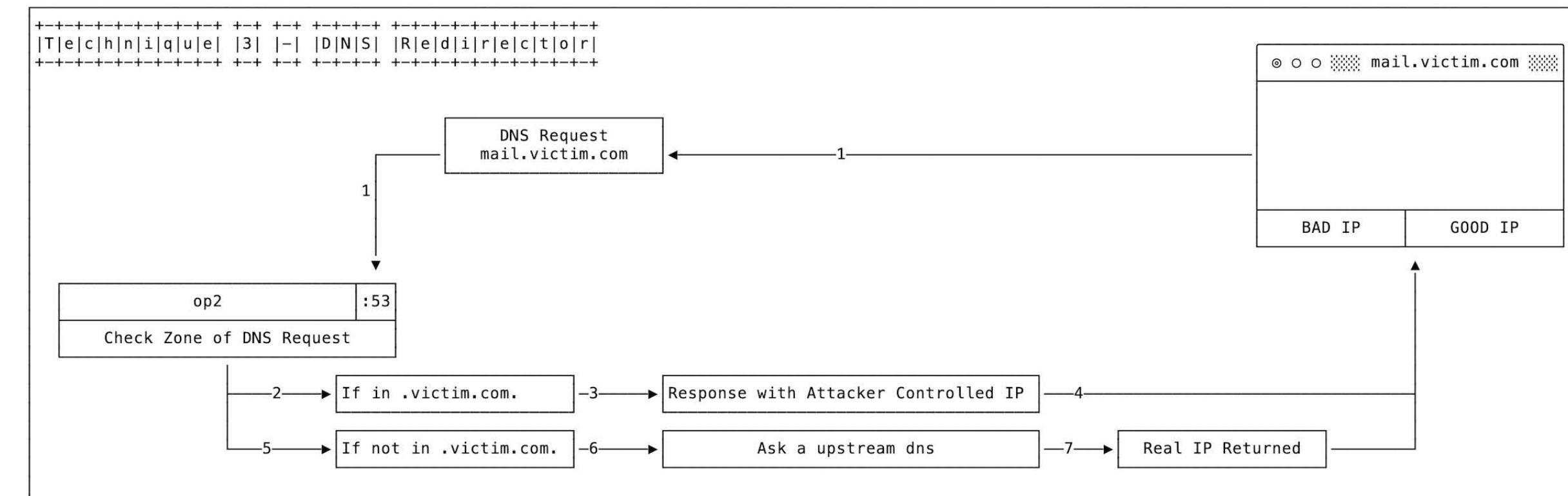
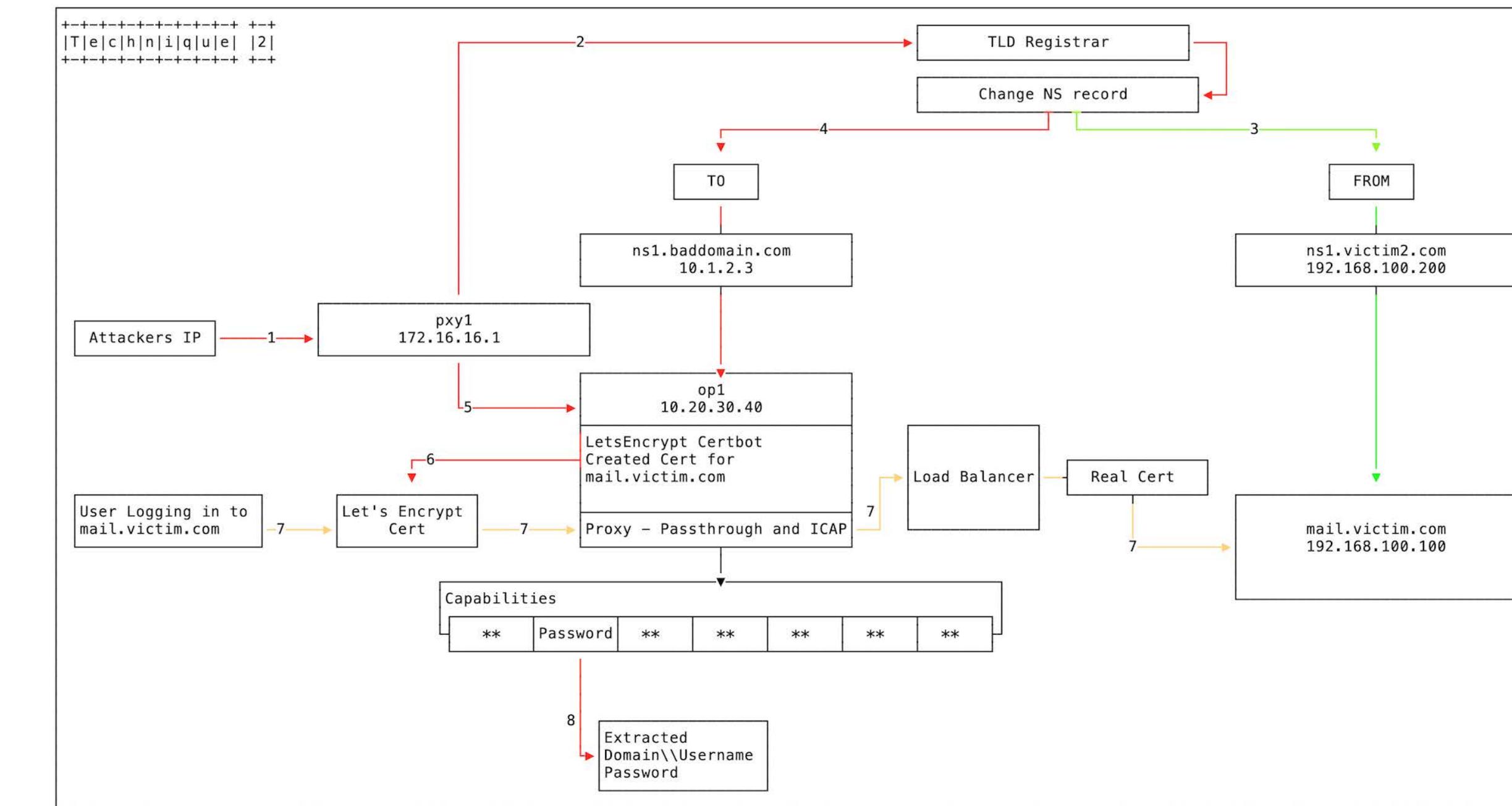
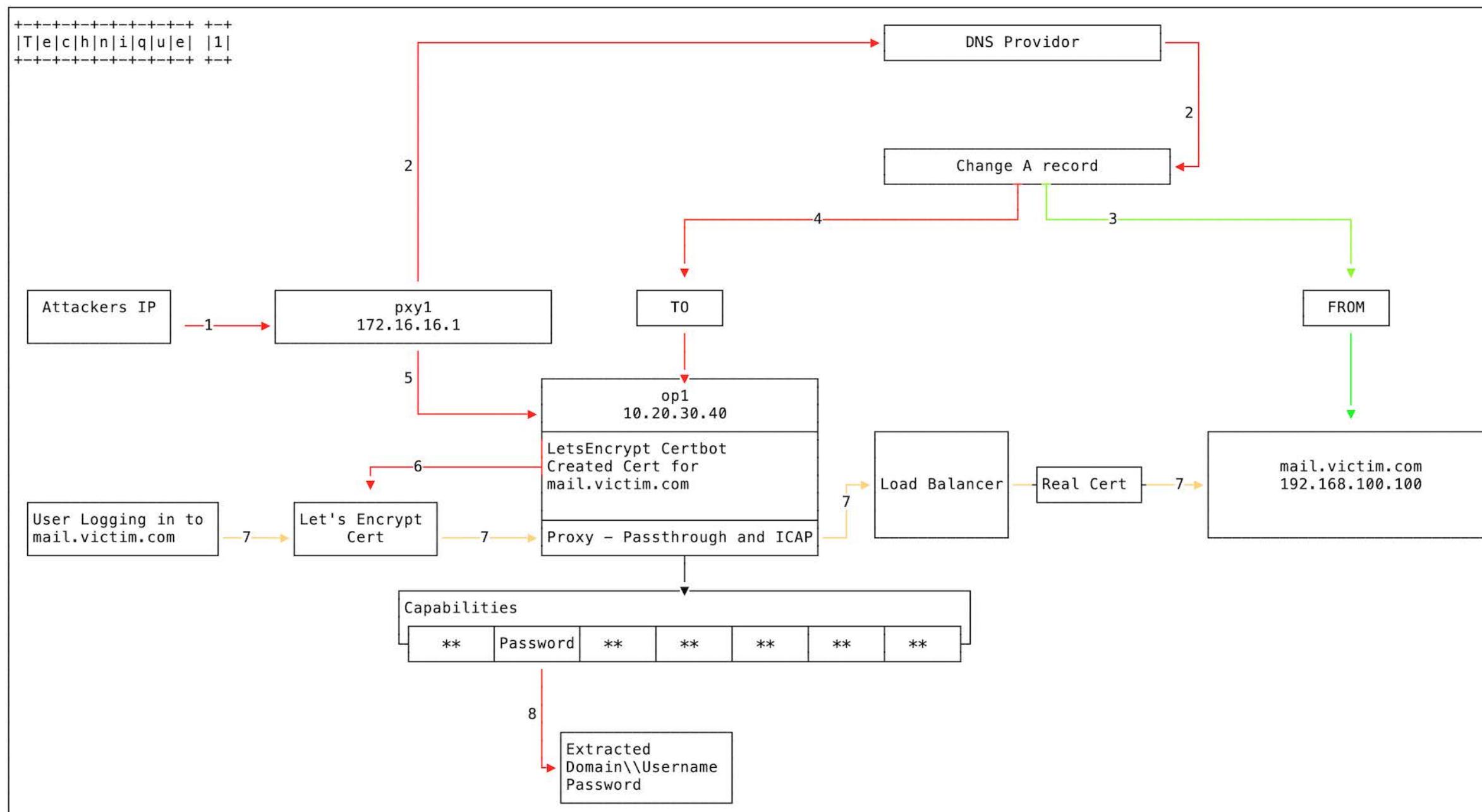


Admin-Certifying Identifier Issuance

# 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/>



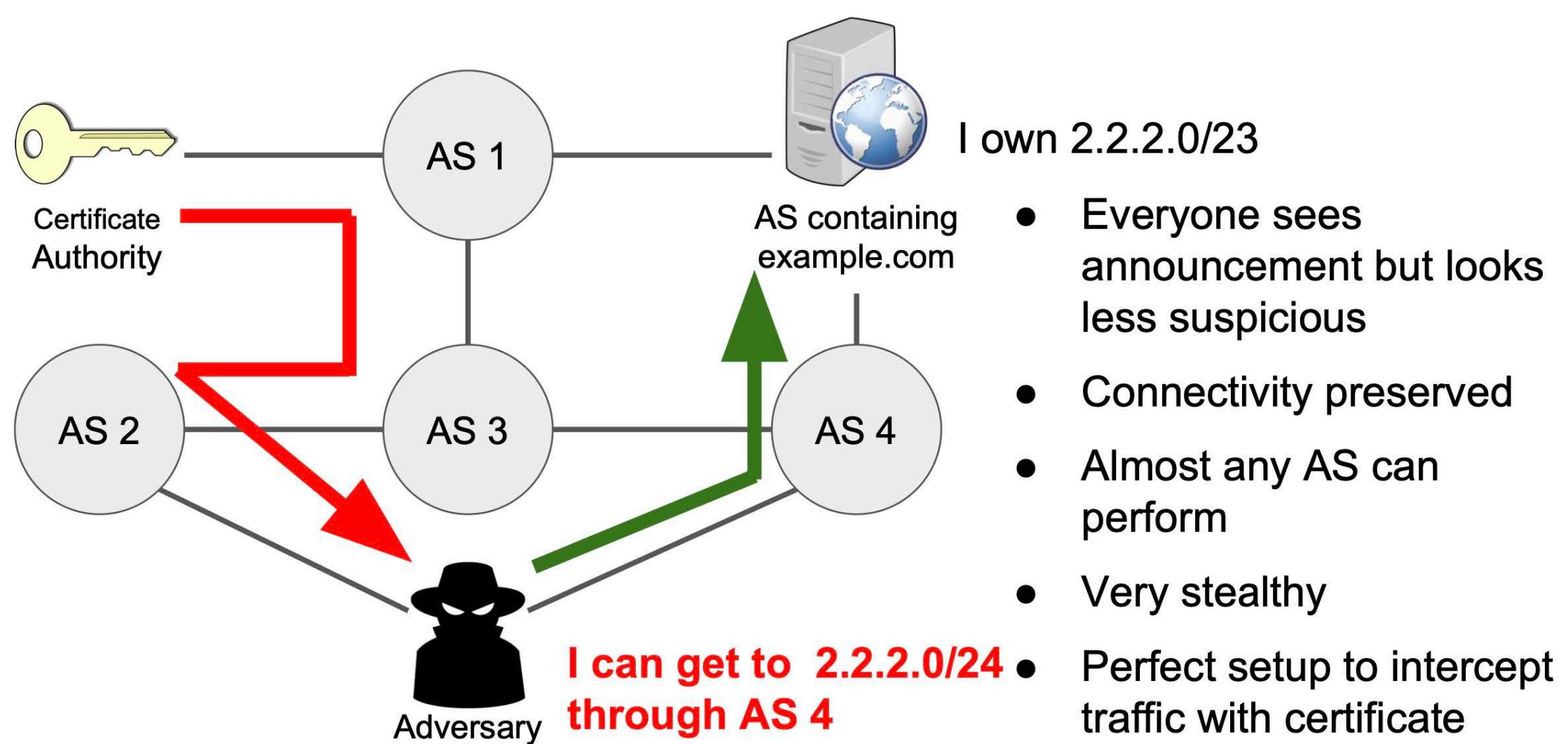
# BGP Hijacking: AS Path Poisoning

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>

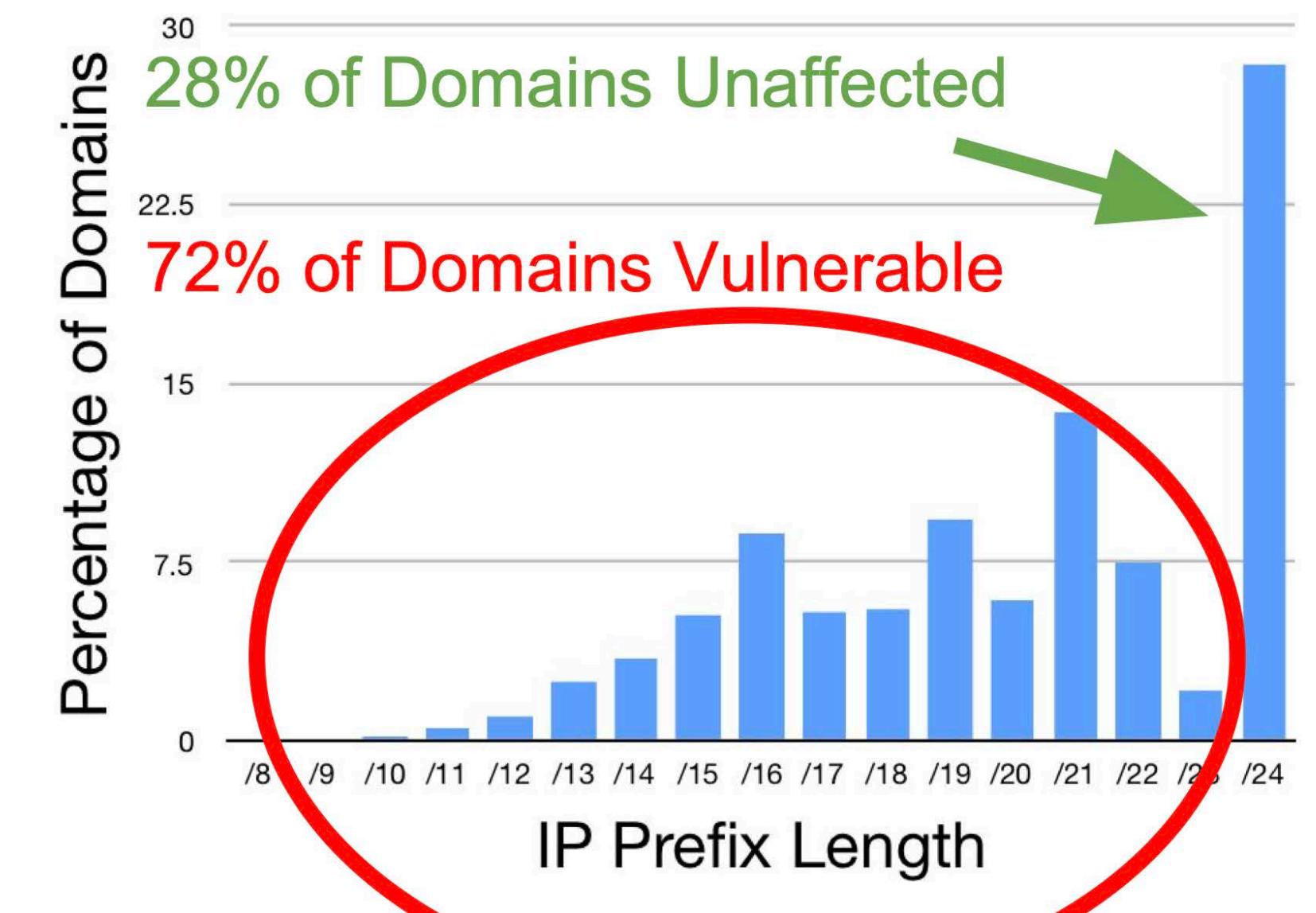
Gavrichenkov, A., “Breaking HTTPS with BGP Hijacking,” BlackHat, 2015 <https://www.blackhat.com/docs/us-15/materials/us-15-Gavrichenkov-Breaking-HTTPS-With-BGP-Hijacking-wp.pdf>

## AS path poisoning

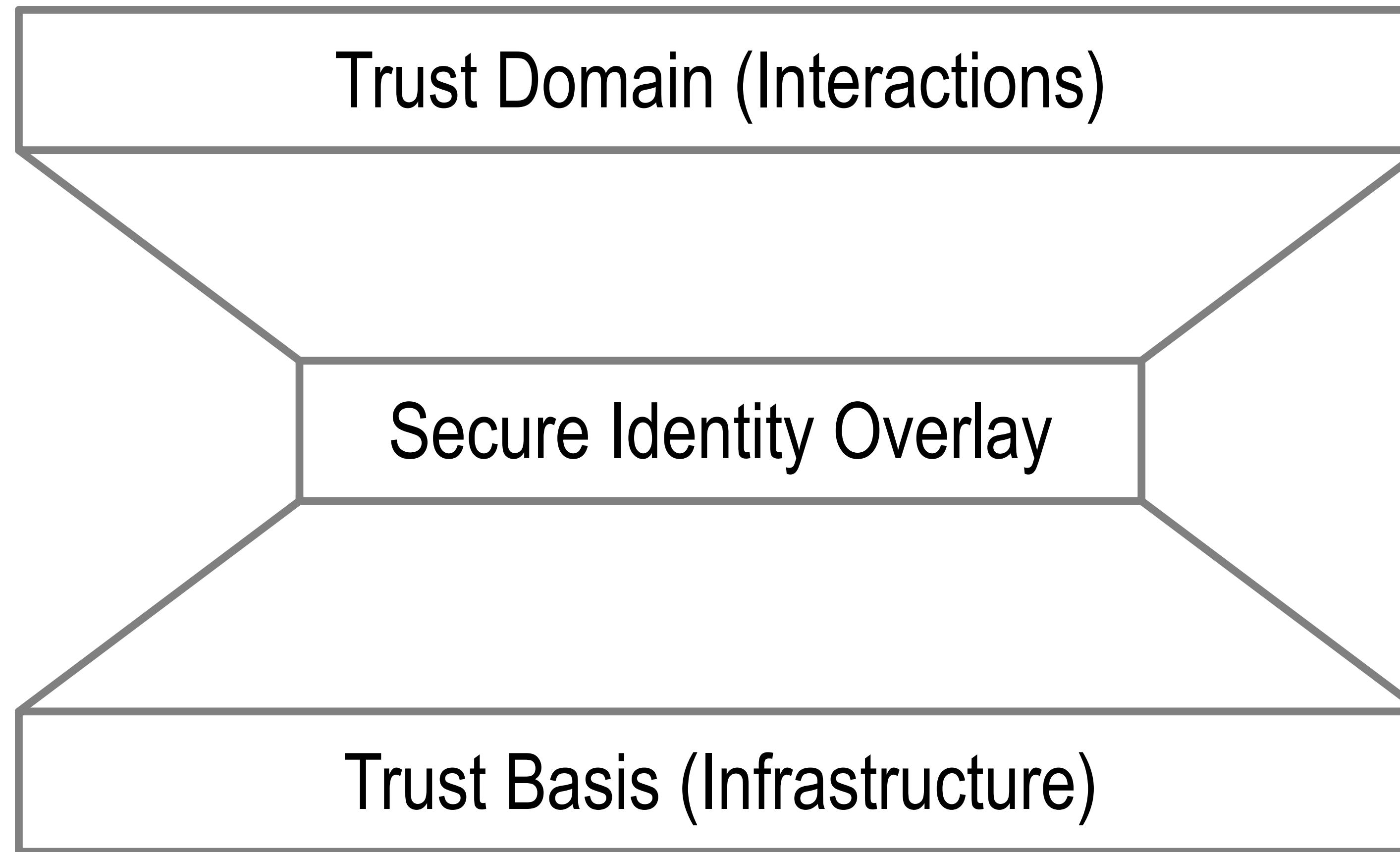


## Vulnerability of domains: sub-prefix attacks

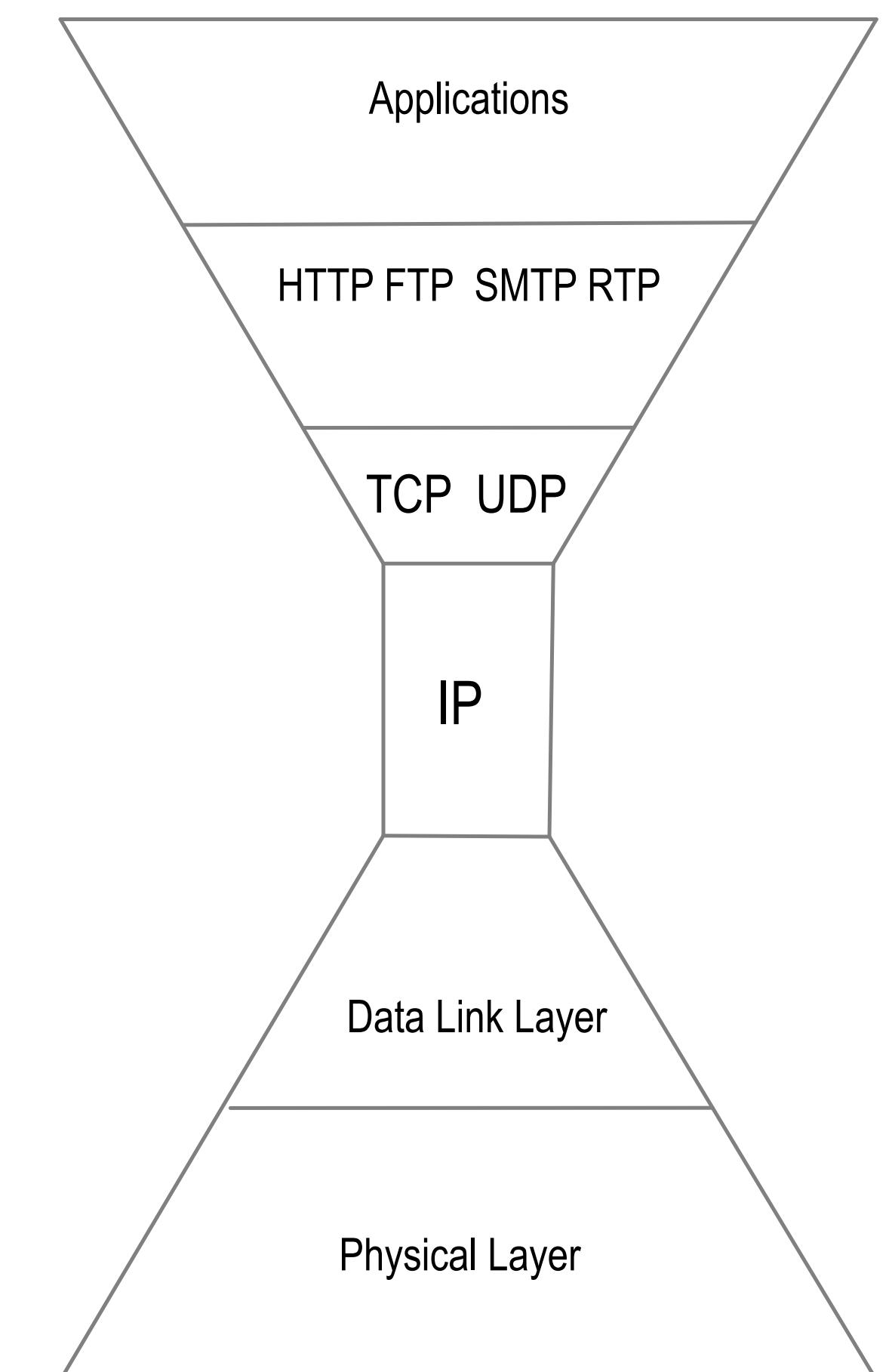
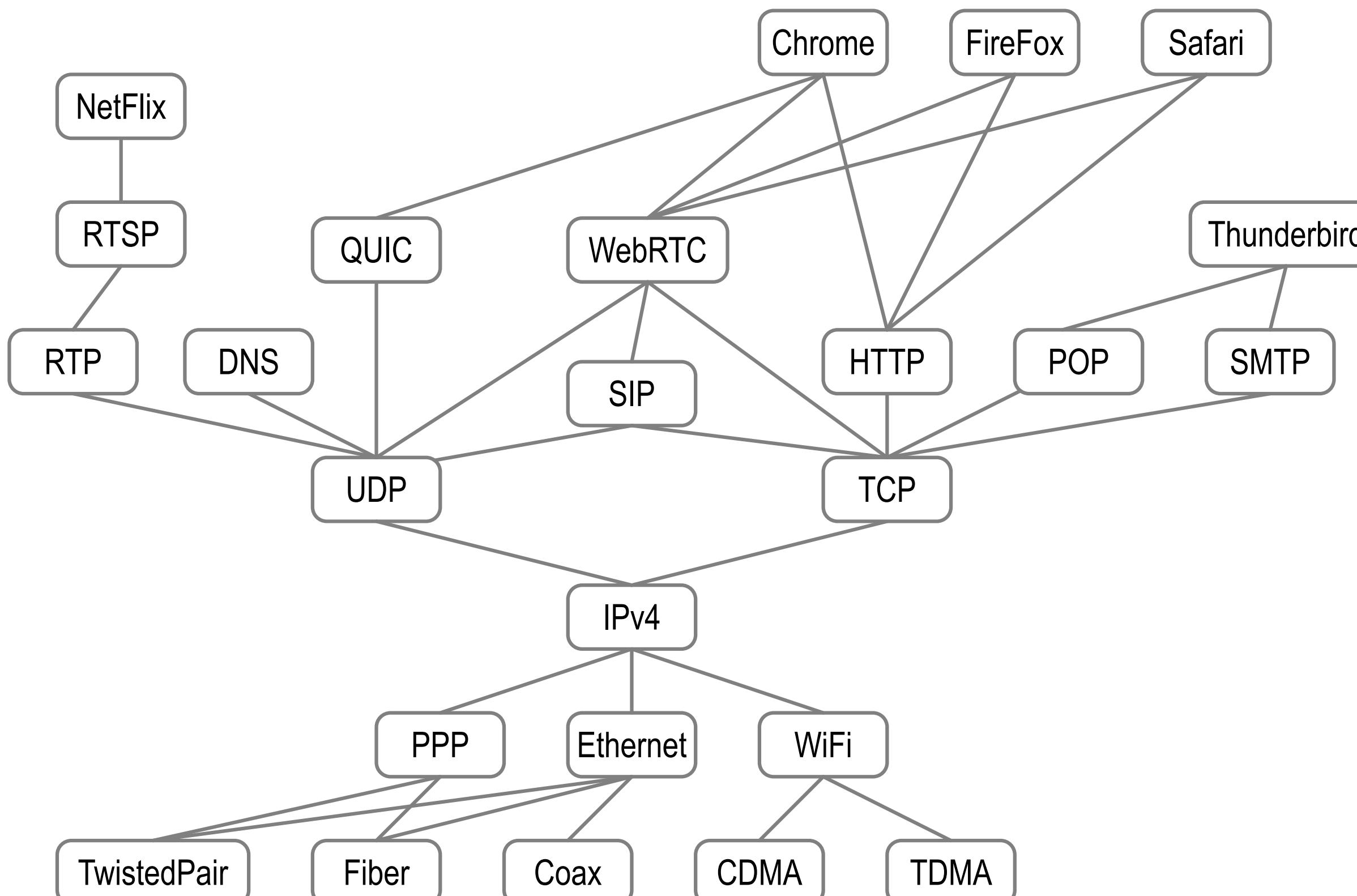
- Any AS can launch
- Only prefix lengths less than /24 vulnerable (filtering)



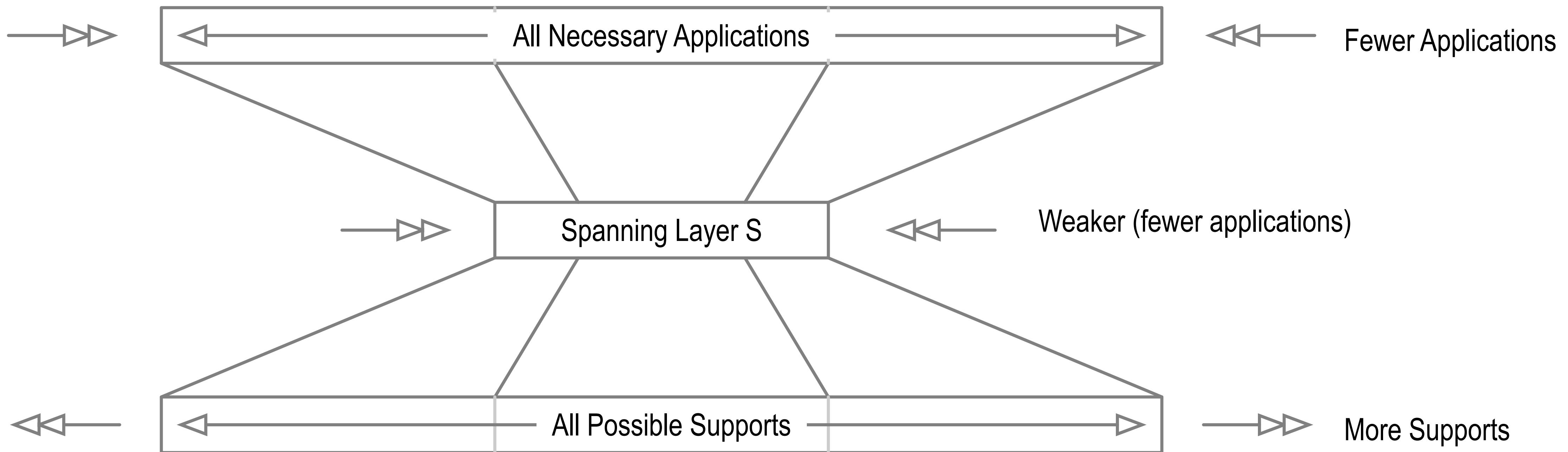
# Identity System Security Overlay



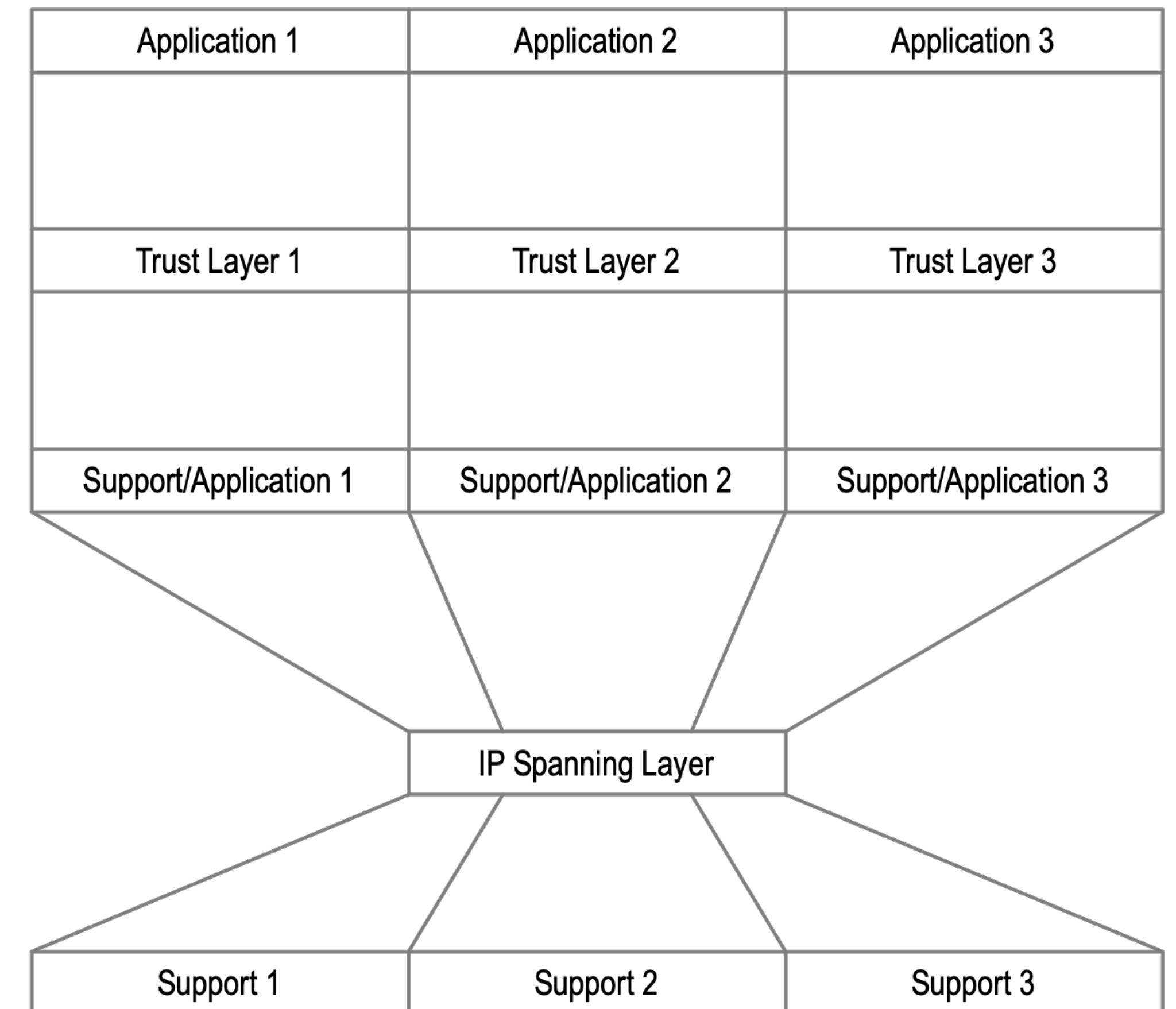
# Spanning Layer



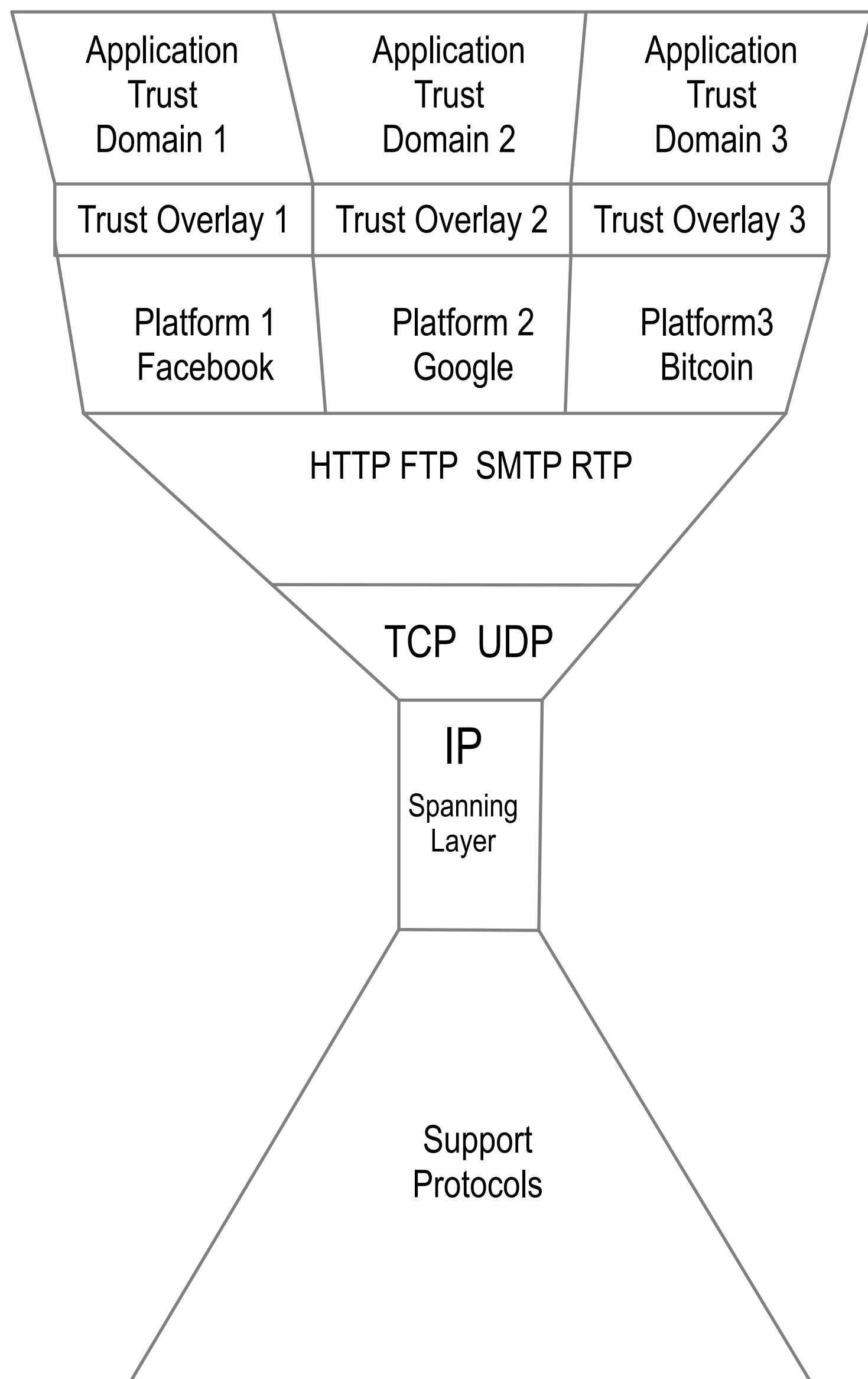
# Hourglass



# Platform Locked Trust

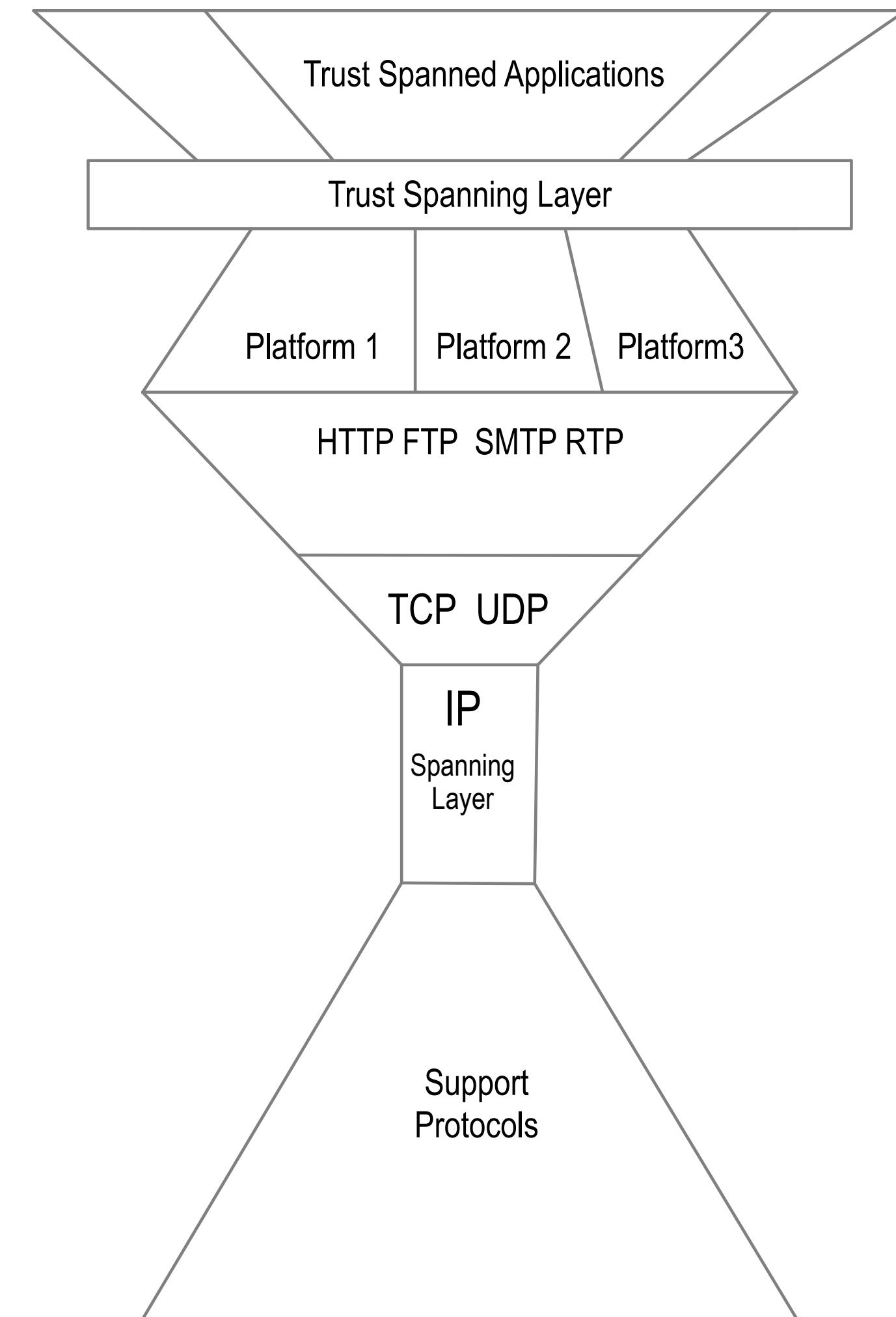
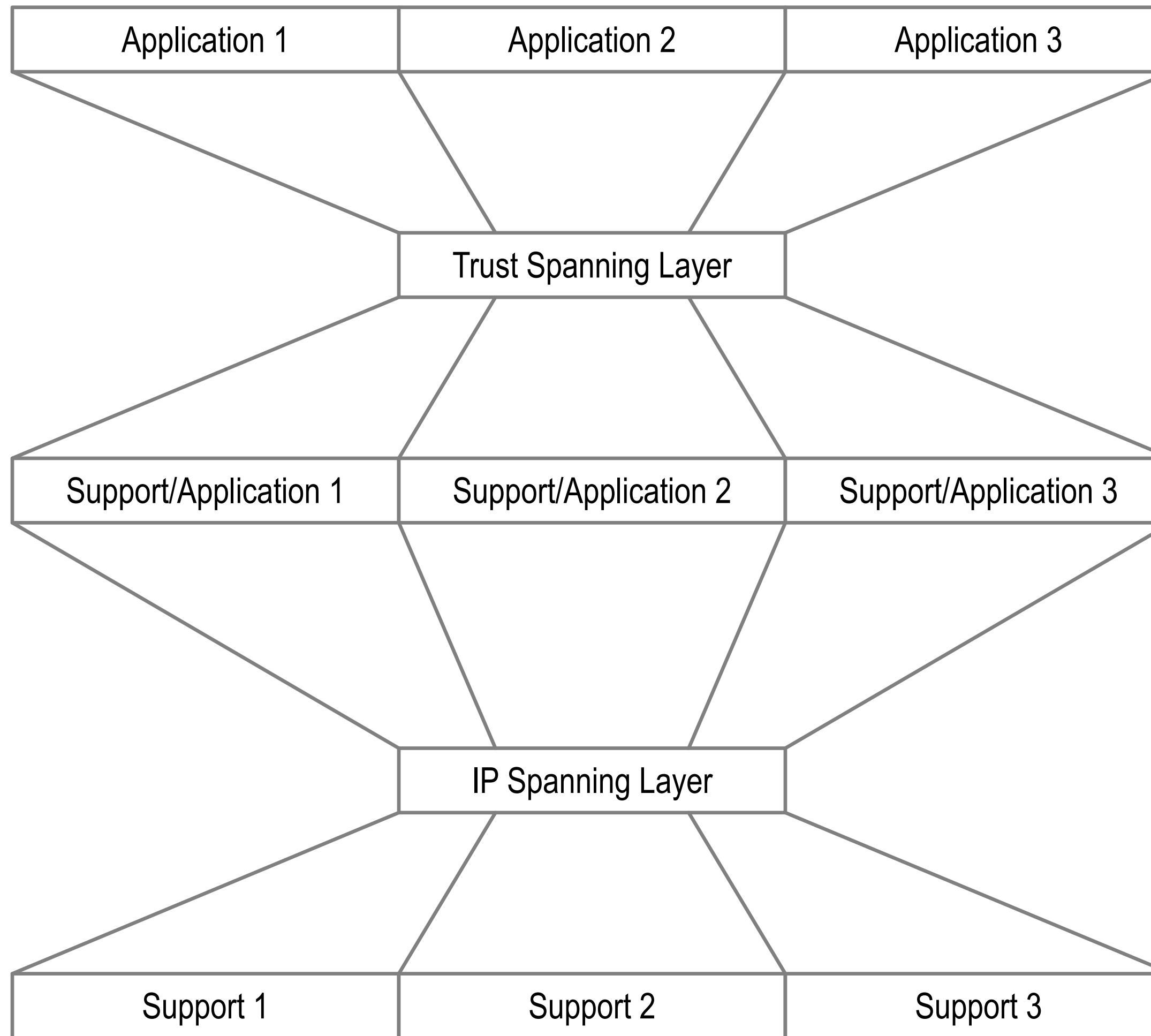


## Trust Domain Based Segmentation



Each trust layer only spans platform specific applications  
Bifurcated internet trust map  
No *spanning* trust layer

# Solution: Waist and Neck



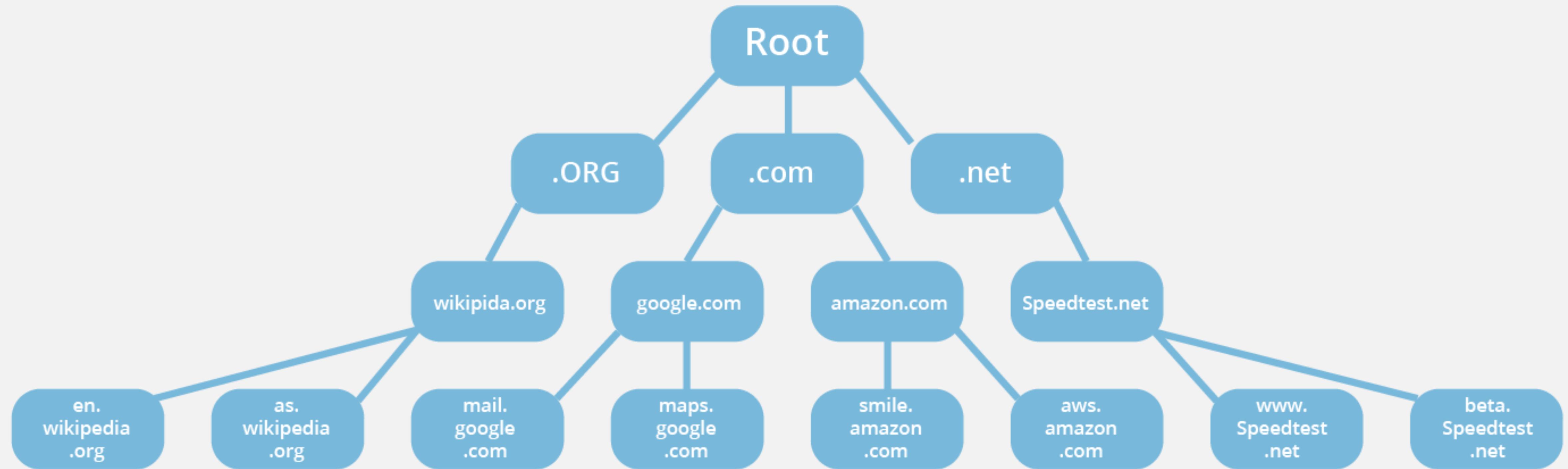
# Background

# Discovery

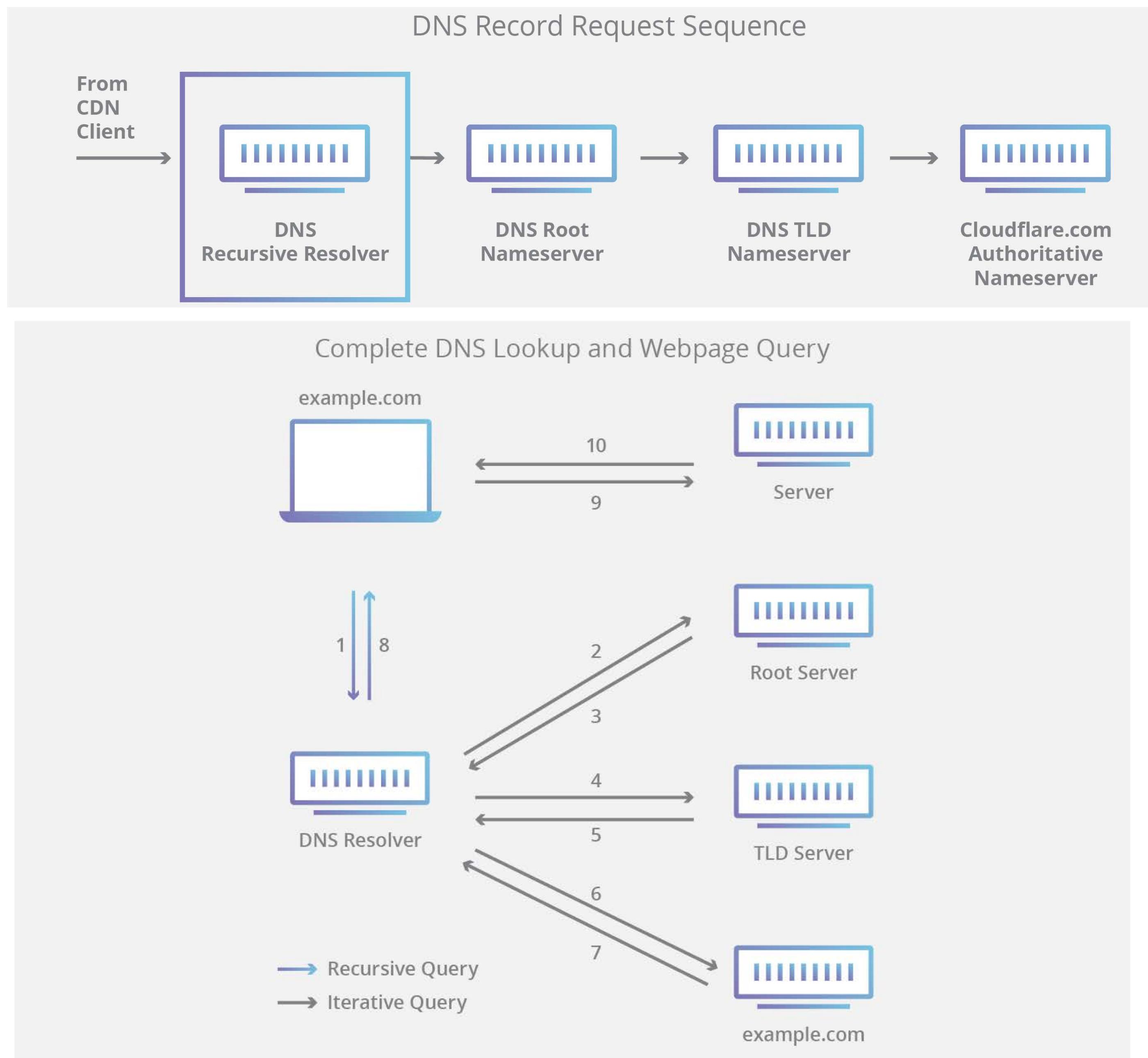
Ledger Based

Non-Ledger Based

# DNS “Hierarchical” Discovery



# DNS “Hierarchical” Discovery

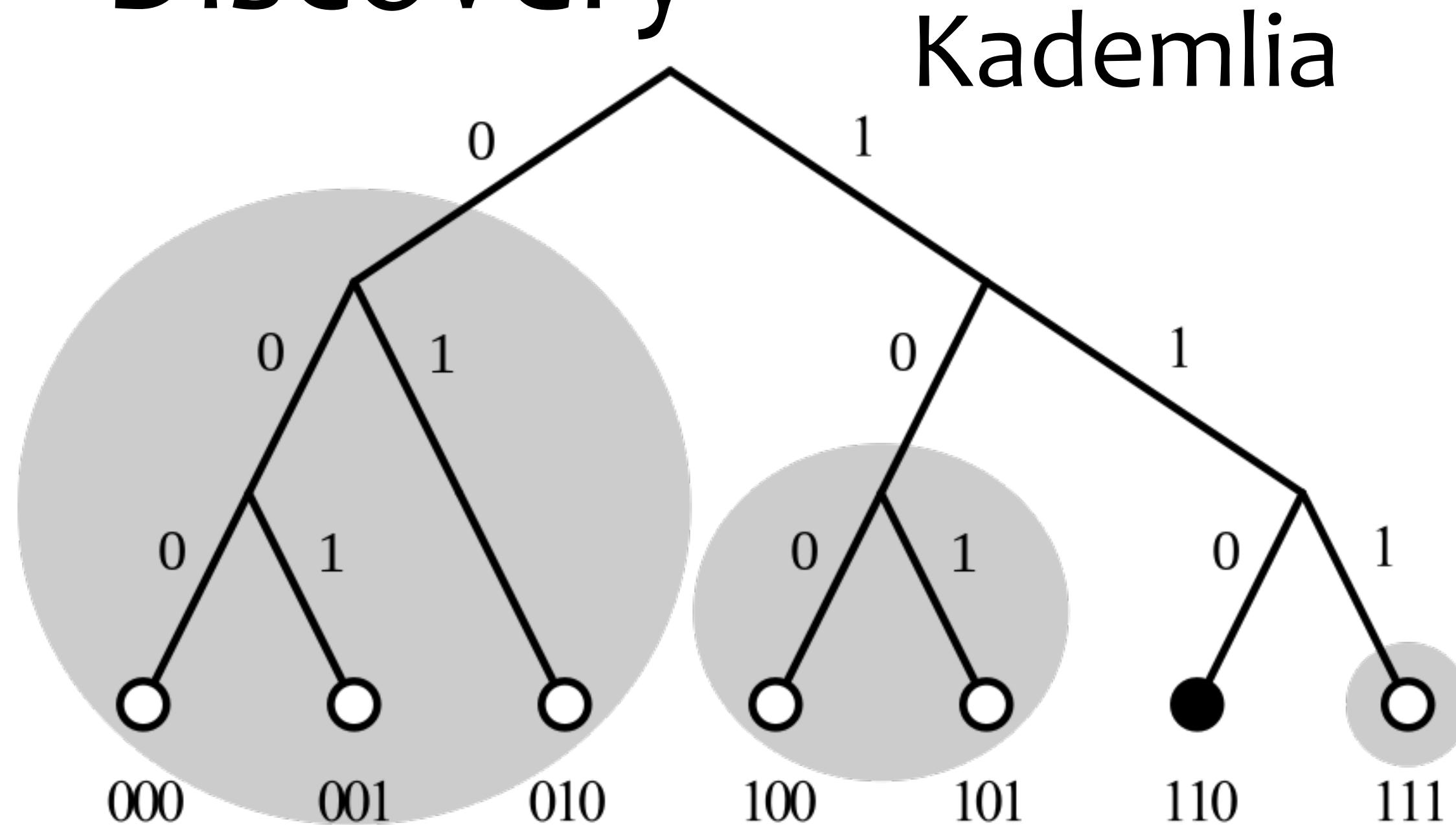
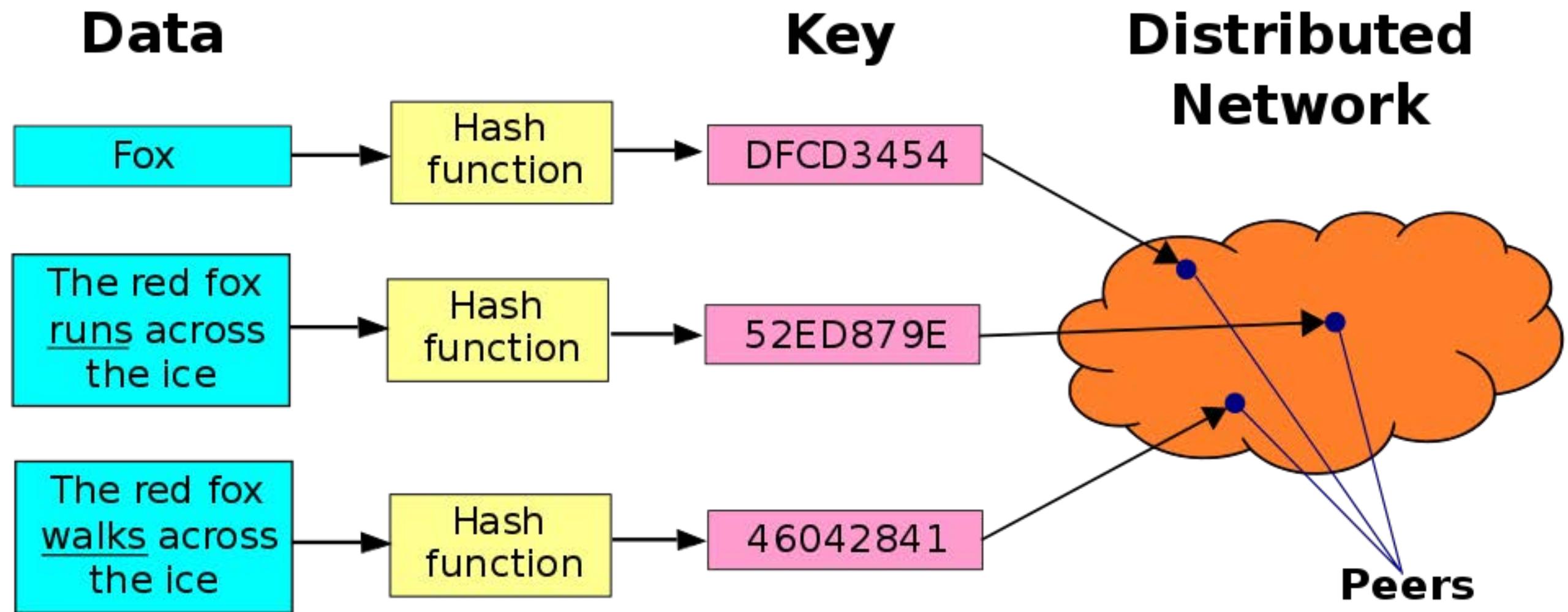


\$ORIGIN example.com.

```

@      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



# 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

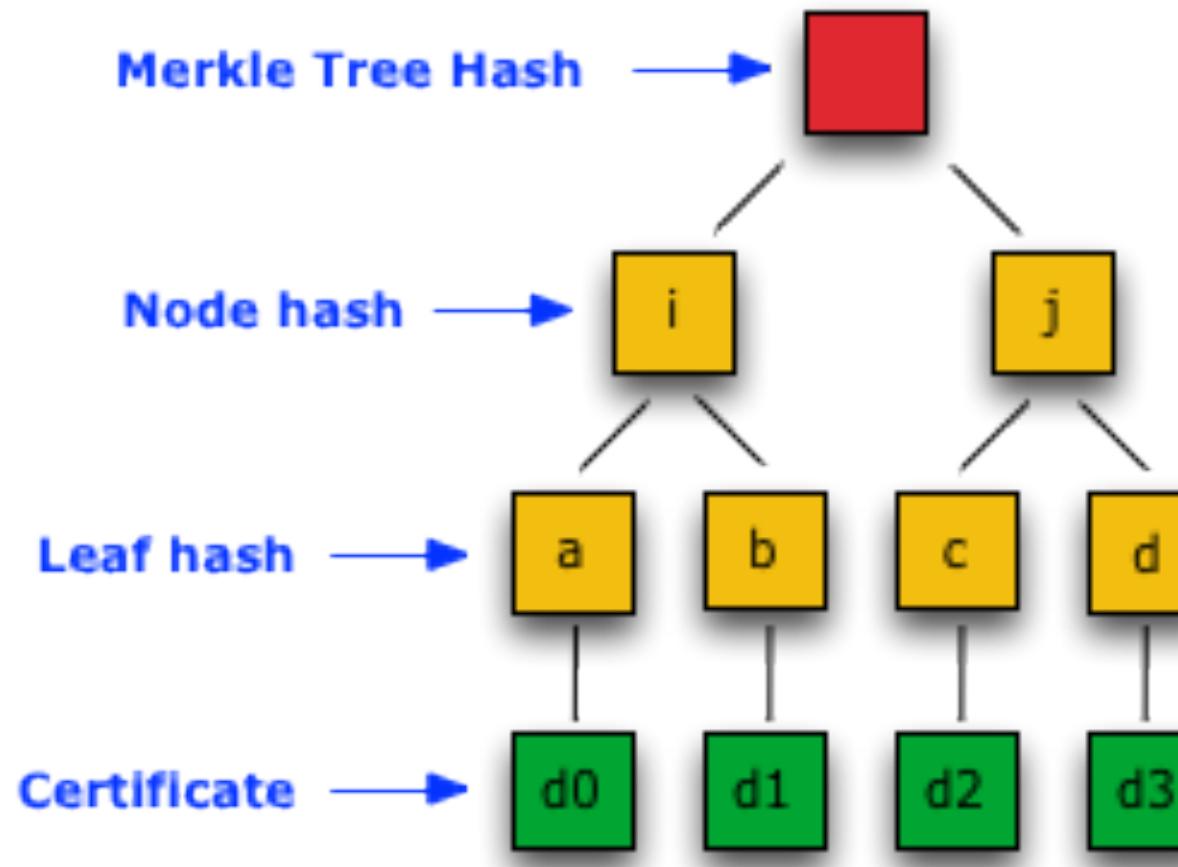
End-verifiable duplicity detection = Ambient verifiability of duplicity

Event log is third party infrastructure but zero trust because it is verifiable.

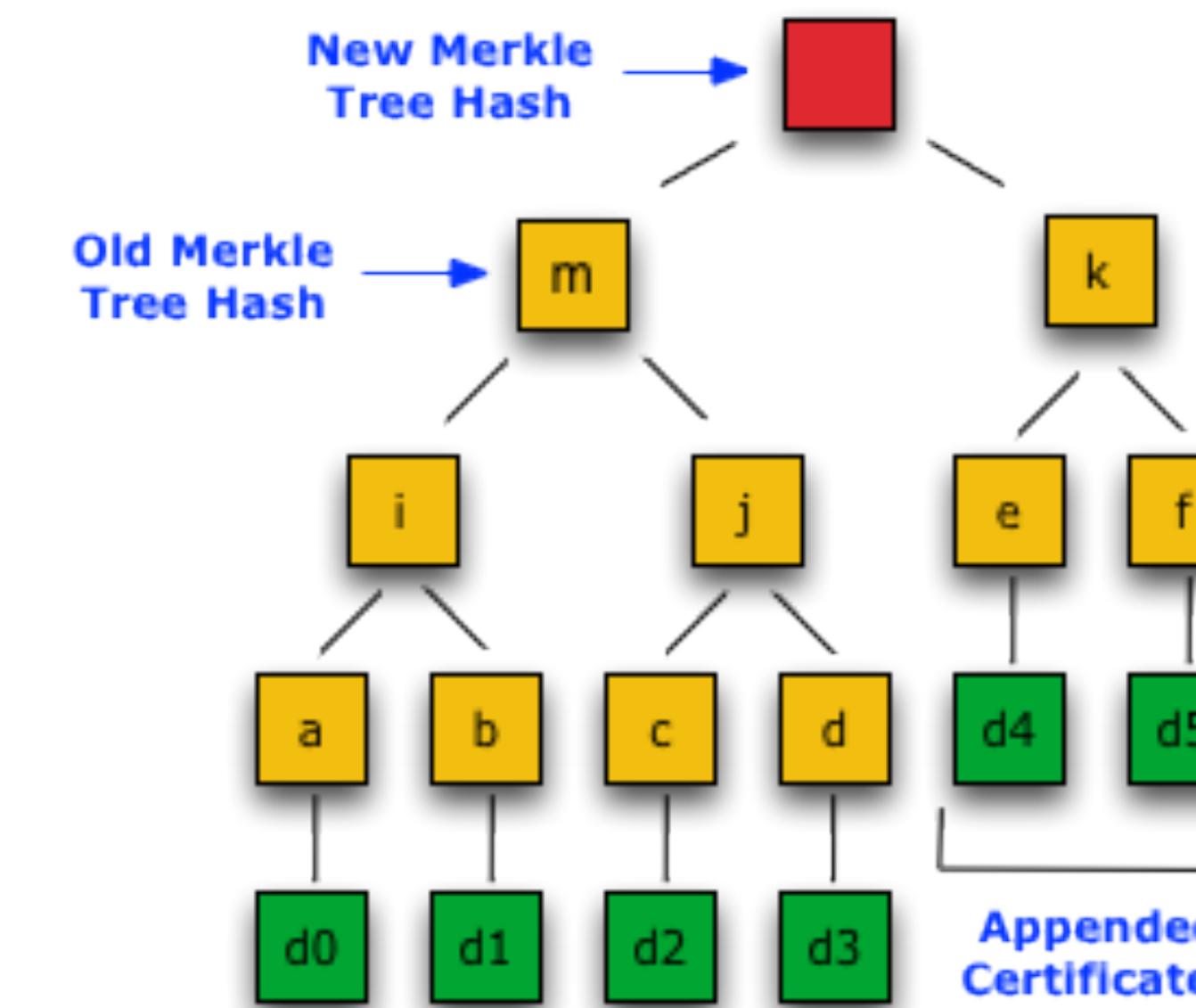
Sparse Merkle Trees for revocation of certificates

# Certificate Transparency Solution

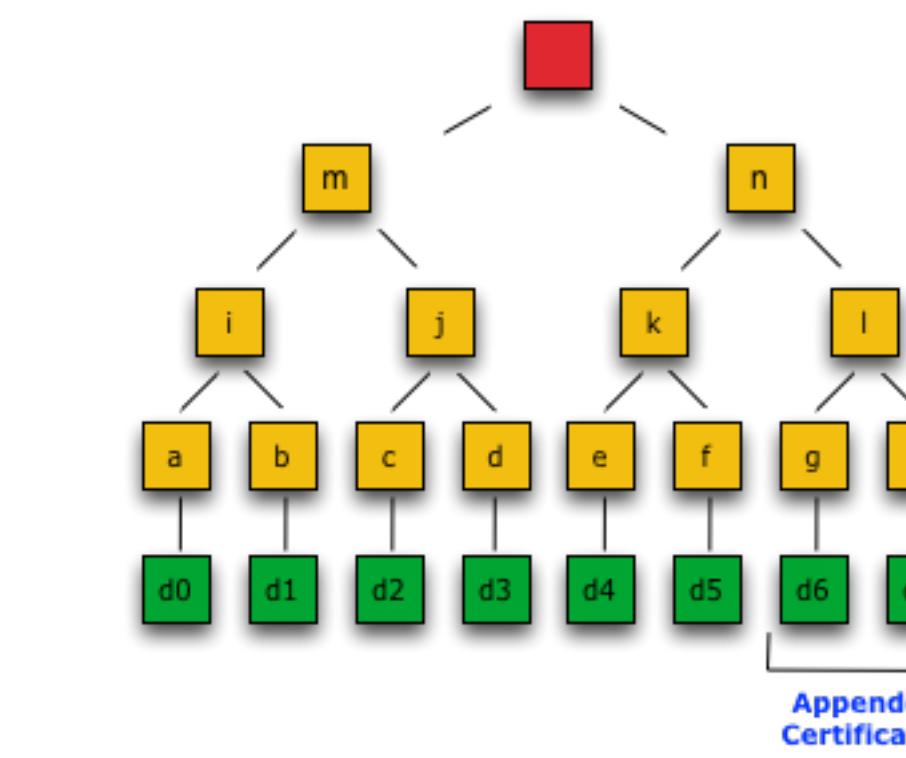
Public end-verifiable append-only event log with consistency and inclusion proofs  
End-verifiable duplicity detection = ambient verifiability of duplicity  
Event log is third party infrastructure but it is not trusted because logs are verifiable.  
Sparse Merkle trees for revocation of certificates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



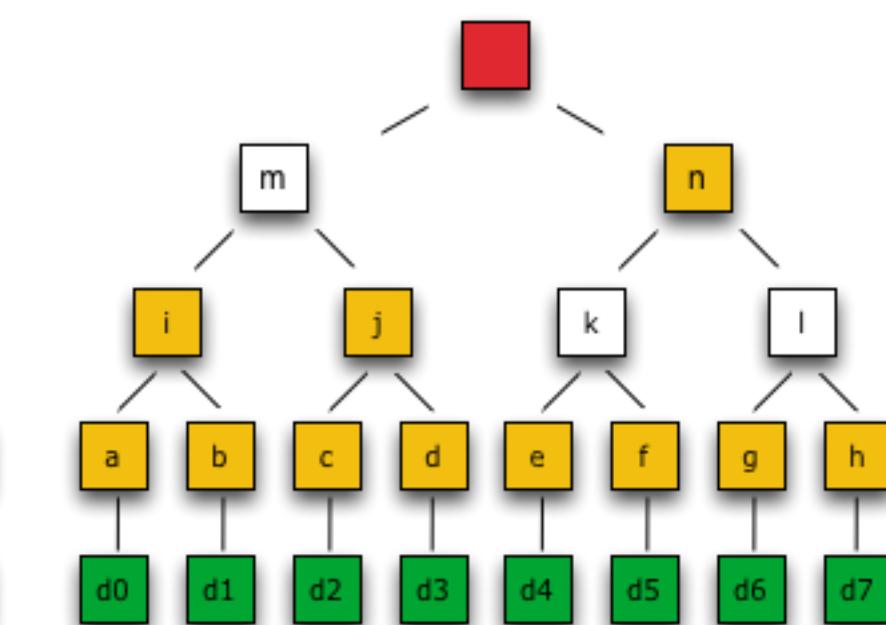
**Figure 1**



**Figure 2**



**Figure 3**



**Figure 4**