Security Deep Dive

Why KERI is Necessary and Sufficient

KERI

https://keri.one
https://github.com/WebOfTrust

Samuel M. Smith Ph.D.
sam@keri.one

Resources

Resources:
https://keri.one/keri-resources/
KERI WebofTrust Community: (meetings, open source code Apache2, specification drafts)
https://github.com/WebOfTrust
https://github.com/WebOfTrust/keri
ToIP: (specifications OWF License) (New KERI Suite Working Group)
https://trustoverip.org/
https://wiki.trustoverip.org/display/HOME/ACDC+(Authentic+Chained+Data+Container)+Task+Force
https://trustoverip.github.io/tswg-cesr-specification/
https://trustoverip.github.io/tswg-keri-specification/
https://trustoverip.github.io/tswg-acdc-specification/
https://wiki.trustoverip.org/display/HOME/Trust+Spanning+Protocol+Task+Force
https://docs.google.com/document/d/1DsvAOGXlMFeE6tYlcaHlitoGLbWGfromRGrvR43zsgs/edit?tab=t.0
Adoptions:
GLEIF (ISO) vLEI: https://www.gleif.org/en/lei-solutions/gleifs-digital-strategy-for-the-lei/introducing-the-verifiable-lei-vlei
European Banking Authority
US Customs: webLEI/vLEI
healthKERI: https://healthkeri.com/
Provenant: https://provenant.net/
Kerion: https://kerion.one/
Cardano: Verdana https://www.veridian.id/

Hard Problems & Solutions

Moving Data Across Trust Domains.

Persistent Identifiers and Issuances with Rotatable Key State

Nearly Instantaneous Key Compromise Detectability

Nearly Instantaneous Key Compromise Recovery

Duplicity Evident Operation

No Shared Secrets

Sign Everything - No Relying Parties

True Zero-Trust

Resistance to Surprise Quantum Attack

Global Portability at Unlimited Scale

Key Event Receipt Infrastructure (KERI)

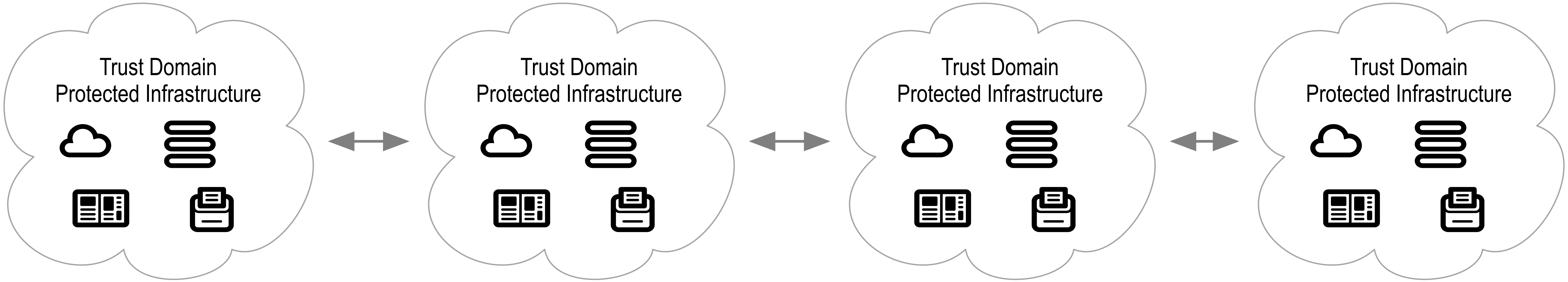
Composable Event Streaming Representation (CESR)

Authentic Chained Data Container (ACDC)

Trust Spanning Protocol (TSP)(SPAC)

Reputational Root-of-Trust:

GLEIF vLEI



Minimally Sufficient Means

No more than is *necessary*. No less than is *sufficient*.

Every part of KERI is necessary to protect against a set of known attacks.

Leaving out any part of KERI, therefore, exposes vulnerabilities.

This makes KERI's feature set *necessary* and *sufficient*.

"The assertion that a statement is a "necessary and sufficient" condition of another means that the former statement is true if and only if the latter is true. That is, the two statements must be either simultaneously true or simultaneously false." (https://en.wikipedia.org/wiki/Necessity_and_sufficiency)

A system using KERI is secure IFF (if and only if) all of KERI is employed.

Anyone using a subset of KERI must prove that it is still "sufficiently" secure despite eliding a "*necessary*" protection measure.

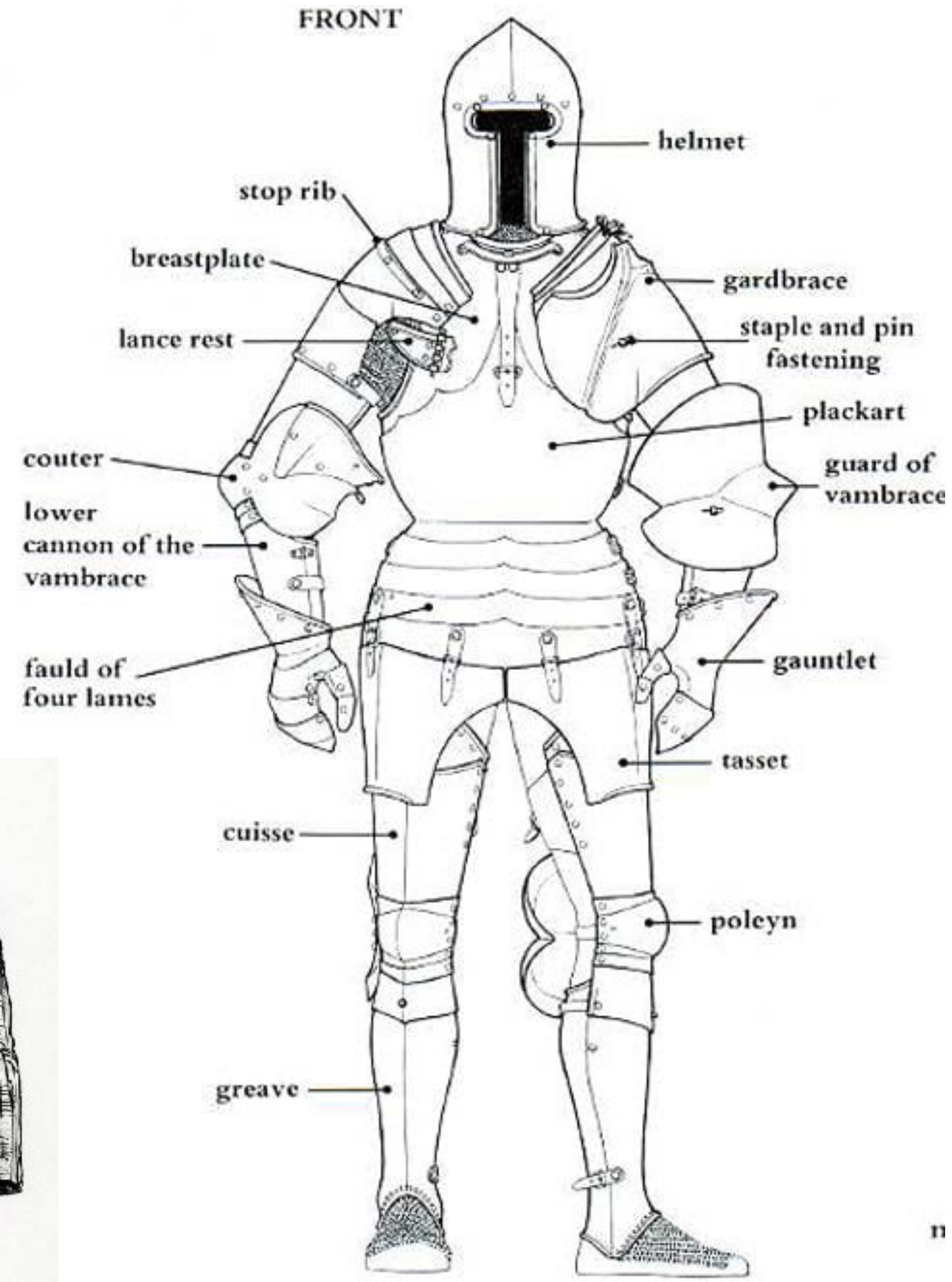
KERI Security leverages Combined Arms

“The *synchronized or simultaneous application of several arms to achieve an effect on the enemy that is greater than if each arm were used against the enemy in sequence.”* (https://www.specialtactics.me/s/MCDP-1-Warfighting.pdf) (https://hcss.nl/report/cognitive-effects-in-combined-arms-a-case-study-of-the-division-2025/)

Multiple mutually supporting threshold structures

Armor

Preventing *wounds* especially *fatal* *wounds*



Each component protects a vital area from injury.

Remove even one component and the adversary will target that area to the exclusion of all else.

Survivability

Susceptibility: Likelihood of being attacked

Vulnerability: Likelihood and extent of exploit given an attack

Recoverability: Likelihood and extent of rectifying the exploit

The "trap" of confusing low susceptibility with low vulnerability

Most systems are vulnerable to attack many or not susceptible to attack.

Susceptibility is about the economics of attack.

Is the reward worth the risk?

Malicious or Compromised Controller

Public AID used for verifiable issuances. (Indirect Mode)

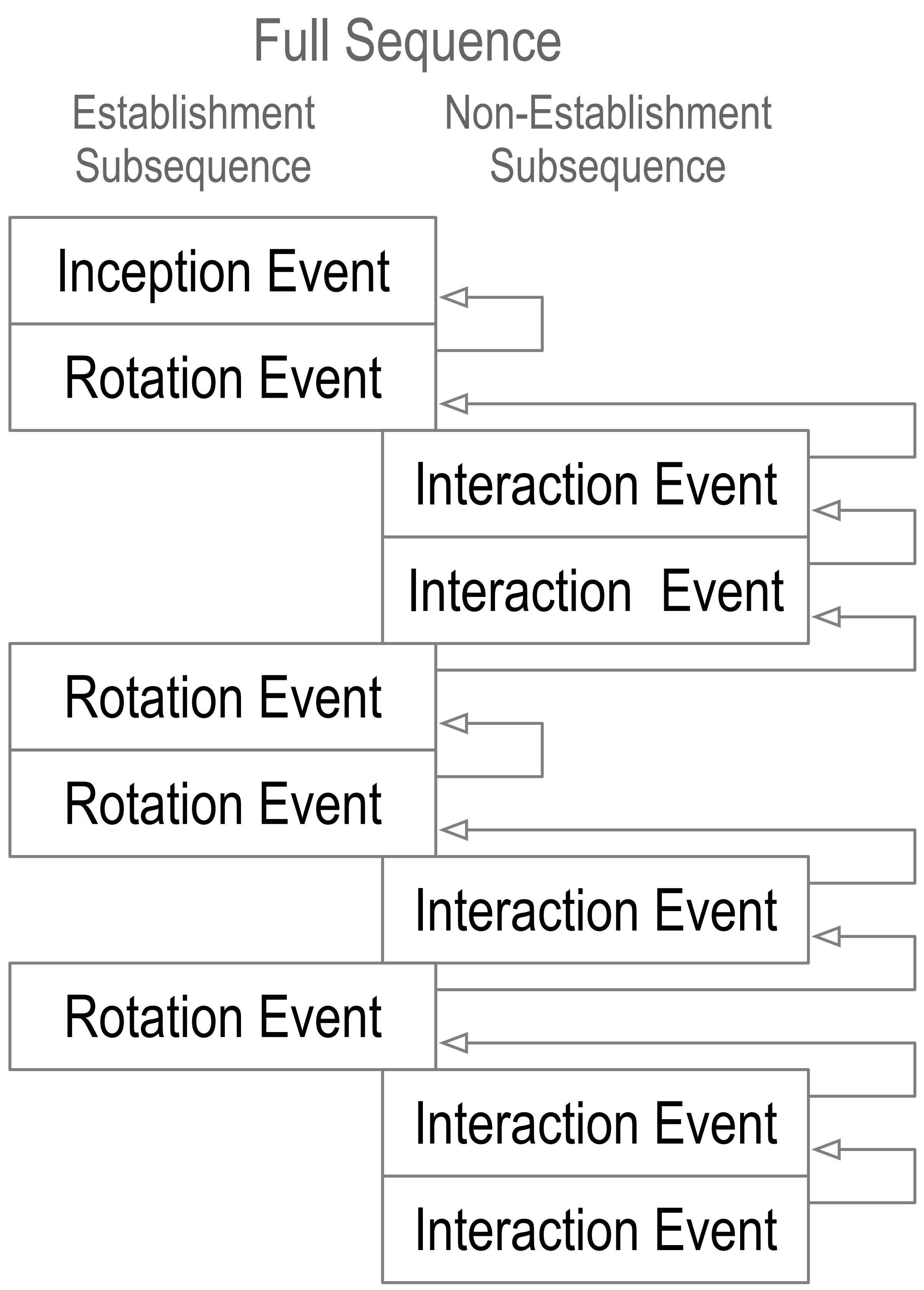
Where are the blockchain skeptics when you need them?

What parts of KERI Infrastructure are controlled by a malicious AID controller?

This may include the network over which a validator communicates.

How to protect a validator from a malicious controller.

Inconsistency and Duplicity



inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

Internal vs. External Inconsistency

Internally inconsistent log = not verifiable.

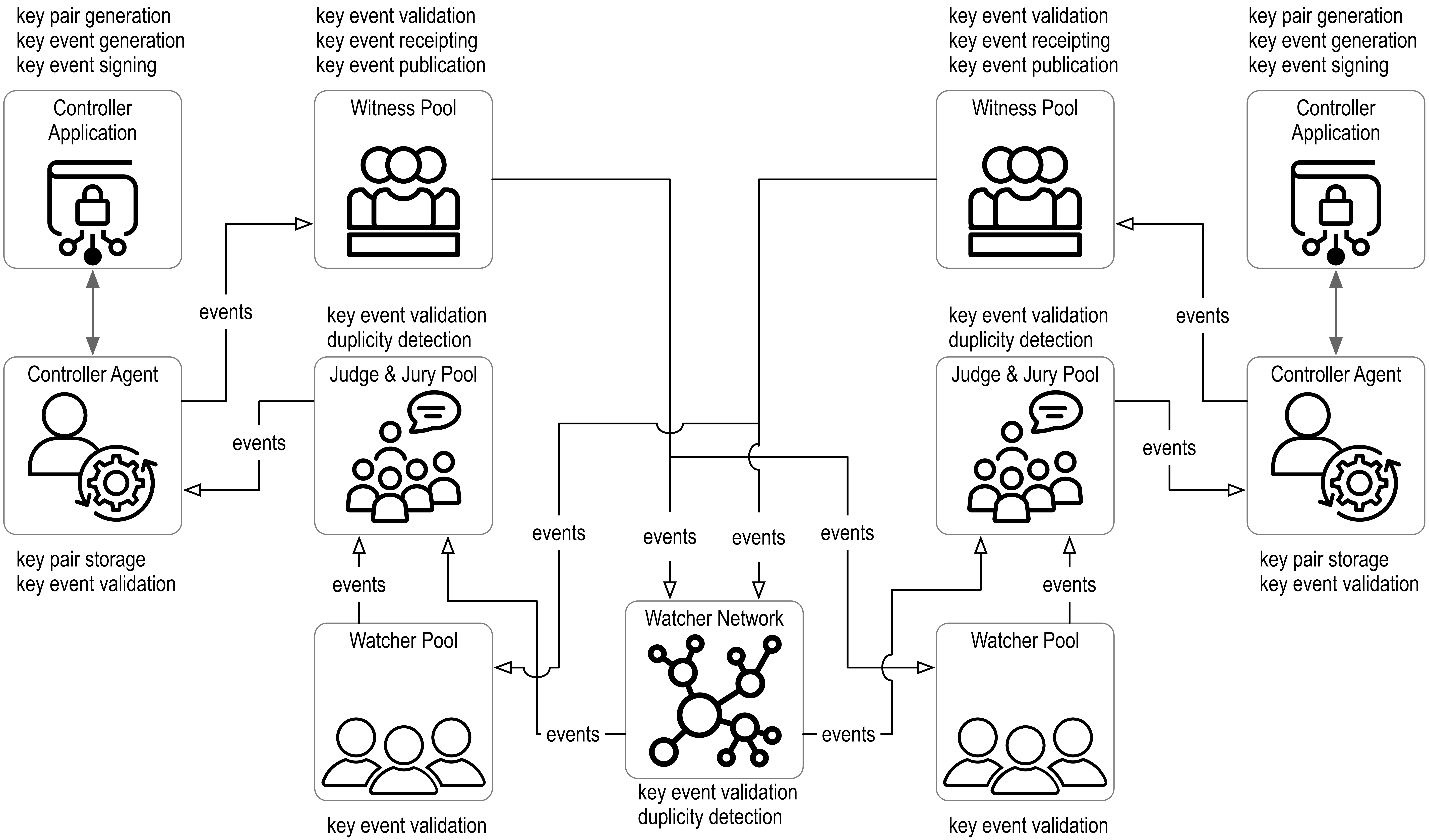
Log verification from a self-certifying cryptographic root-of-trust protects against internal inconsistency.

Externally inconsistent logs. Two different logs for the same identifier, both verifiable = duplicity.

Duplicity detection protects against external inconsistency.

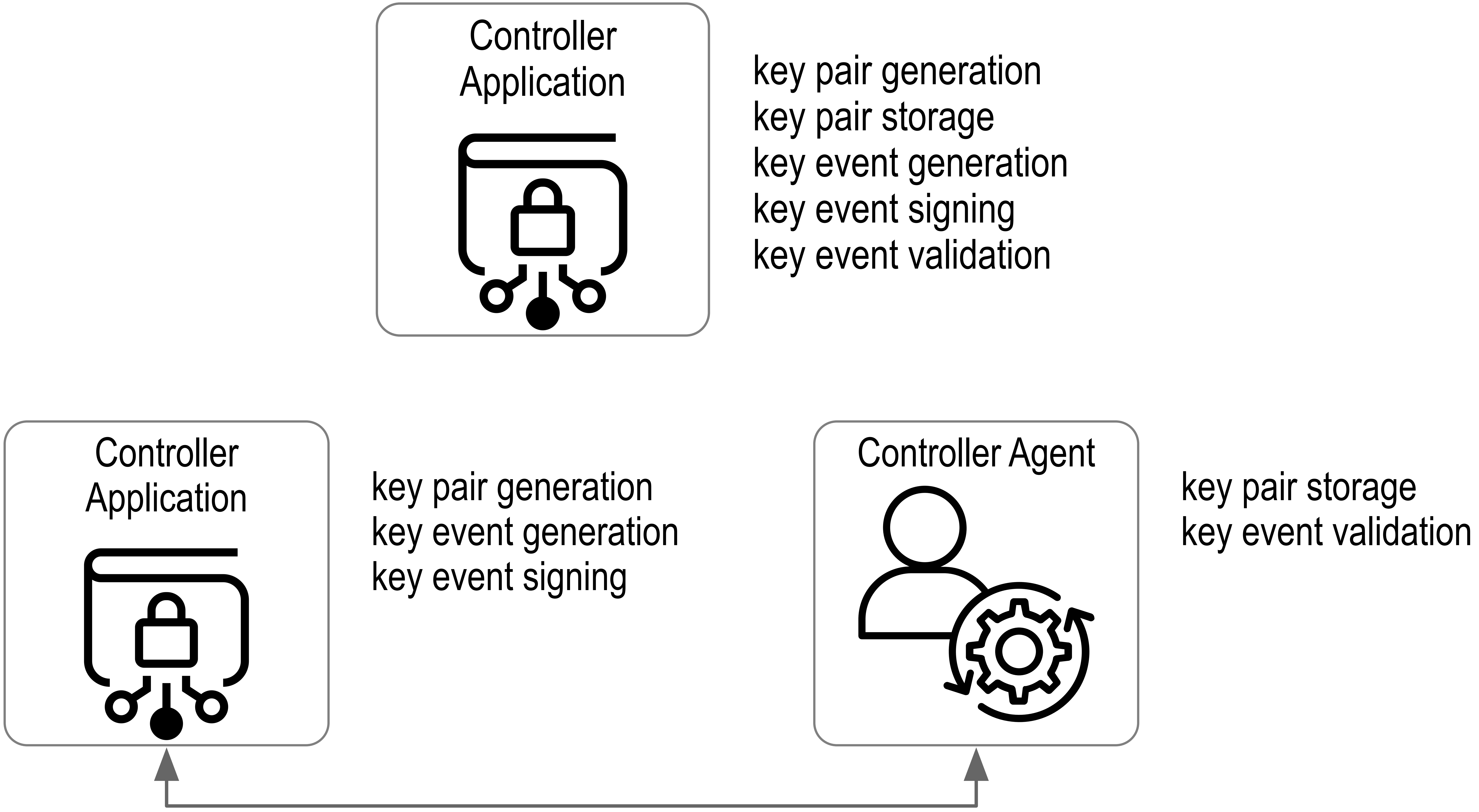
KERI provides duplicity evident DKMI

KERI Ecosystem Components: Witnesses and Watchers, Advanced Indirect Mode



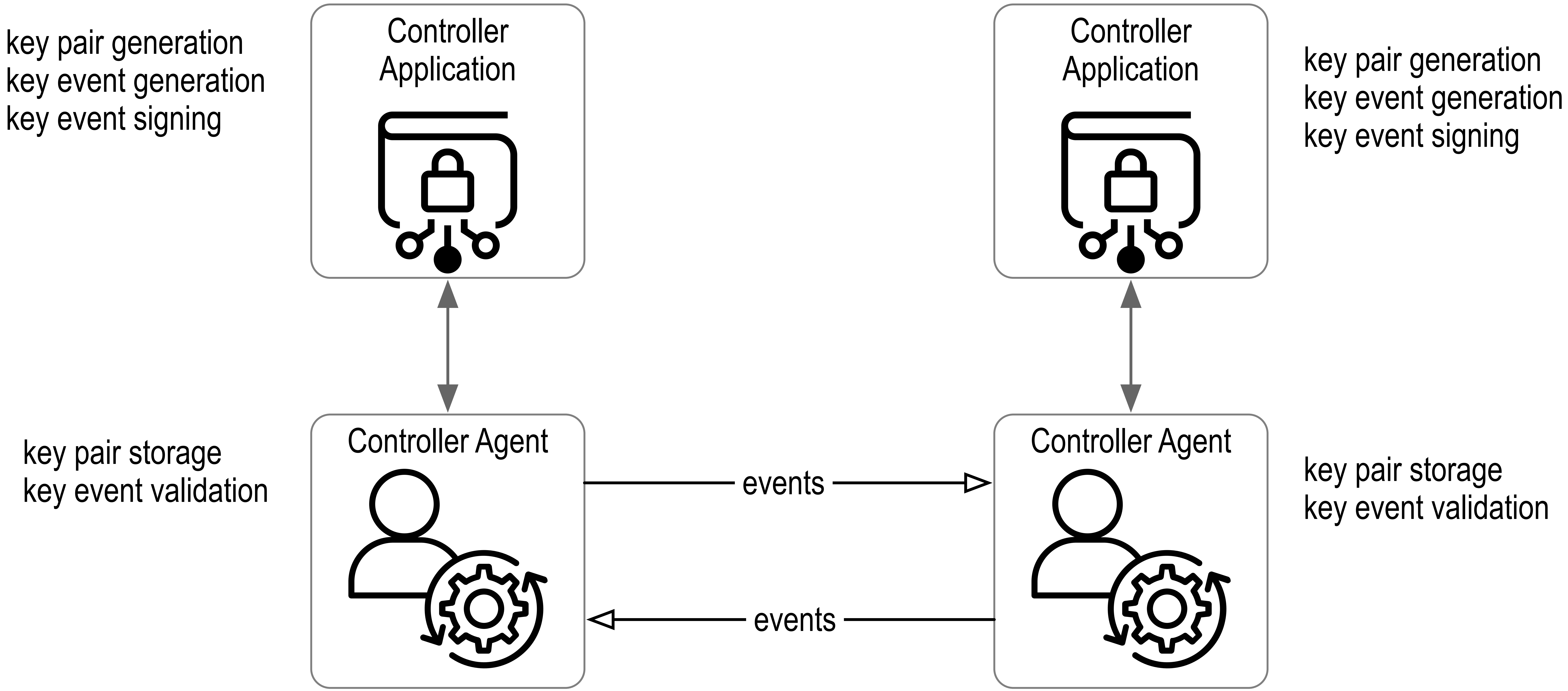
Ambient Verifiability

KERI Ecosystem Components: Controller Application and Agents

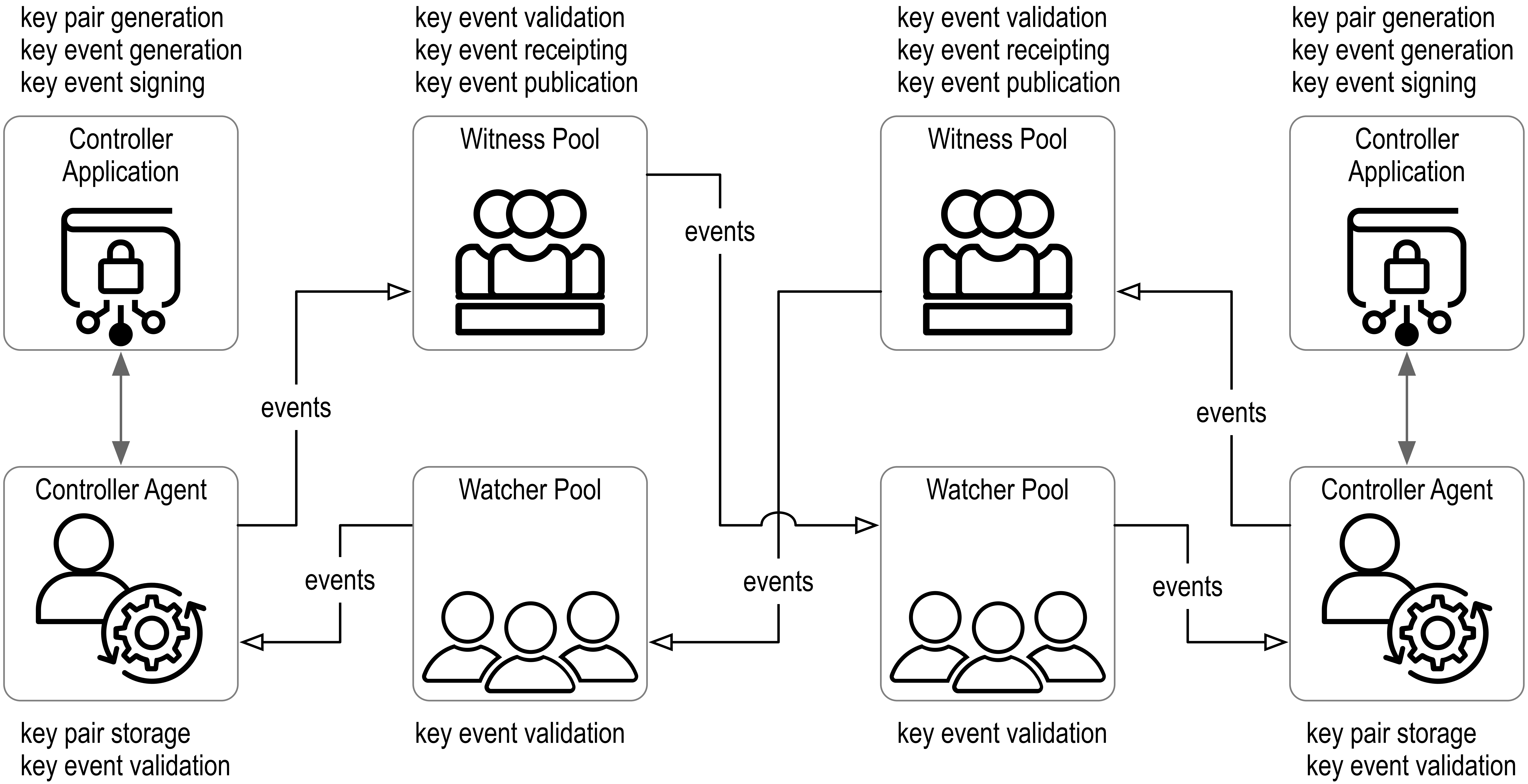


Modular, decentralized, web-based infrastructure without shared governance.

KERI Ecosystem Components: Peer-to-Peer Direct Mode



KERI Ecosystem Components: Witnesses and Watchers, Basic Indirect Mode



Modular decentralized web based infrastructure without shared governance

Dead Compromised Signing Keys

Post quantum surprise attack

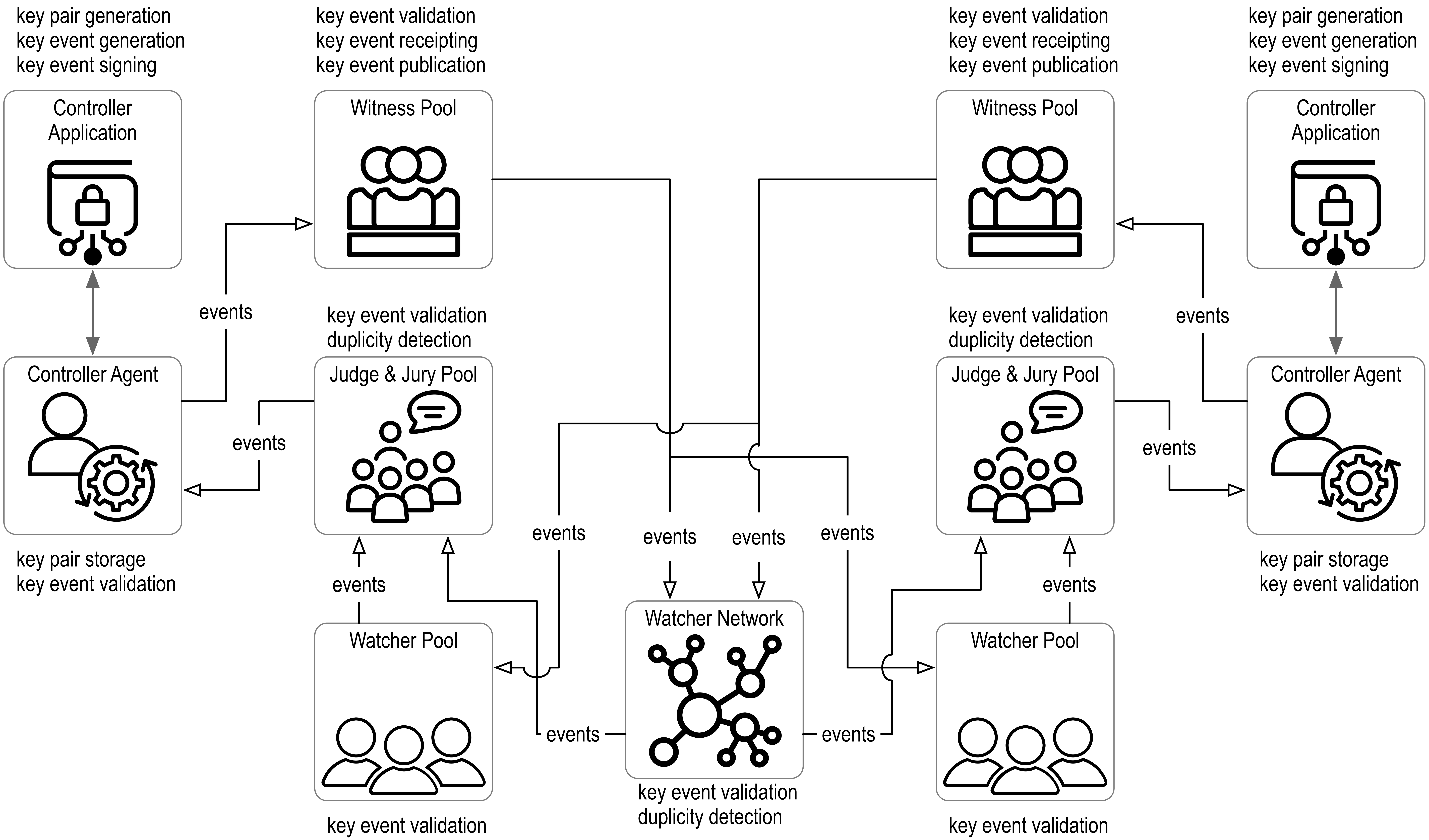
Only need regular watchers

Do not need Juror pool with Judge

But a Juror Pool with Judge may be useful as part of a dynamic appraisal.

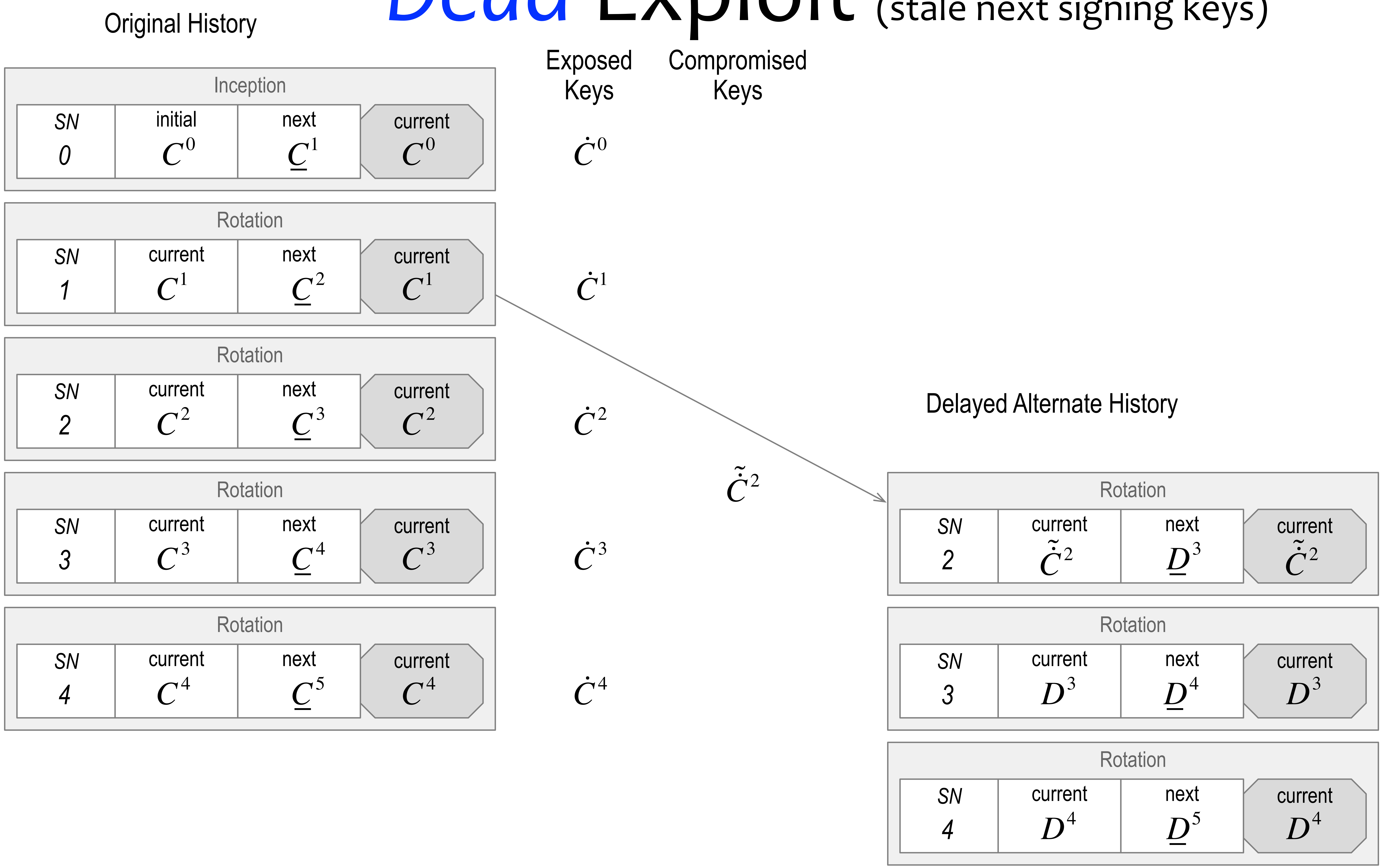
KERI Infrastructure is dynamically appraisable.

KERI Ecosystem Components: Witnesses and Watchers, Advanced Indirect Mode



Ambient Verifiability

Dead Exploit (stale next signing keys)



Any already-seen copy of the original KEL protects against later successful *dead* exploit:

First Seen Always Wins: *First Seen, Only Seen, Always Seen, Never Unseen,*

Watcher Mentoring

New watchers can inoculate themselves from dead exploit by syncing with old watchers before “seeing” any new KELs

Advanced Watchers: Risk Assessment vs. Appraisal

Risk Assessment: Internal process whereby a *Controller* assesses its own infrastructure to determine likely vulnerabilities to exploit and actual exploits and compromise.

Static Risk Assessment: Systemic Vulnerabilities Prior to Attack

Dynamic Risk Assessment: Pre and post (mostly) evaluation and evolution of suspected attacks.

- Largely forensic after the fact (lagging vs. leading indicator of compromise. "tracing the untraceable"

Appraisal: External process whereby a Validator evaluates some other Controller's infrastructure to determine likely vulnerabilities to exploit and actual exploits and compromise.

Static Appraisability: External Validator has visibility into the systemic vulnerabilities of some other Controller's infrastructure

Dynamic Appraisability: External Validator has visibility into the live state of exploit (compromise) of some other Controller's infrastructure

IETF-Remote ATestation ProcedureS (RATS) https://datatracker.ietf.org/group/rats/about/

In network protocol exchanges, it is often the case that one entity (a Relying Party) requires evidence about the remote peer (and system components [RFC4949] thereof), in order to assess the trustworthiness of the peer. Remote attestation procedures (RATS) determine whether relying parties can establish a level of confidence in the trustworthiness of remote peers, called Attesters. The objective is achieved by a two-stage appraisal procedure facilitated by a trusted third party, called Verifier, with trusted links to the supply chain.

The procedures for the two stages are:

- Evidence Appraisal: a Verifier applies policy and supply chain input, such as Endorsements and References Values, to create Attestation Results from Evidence.
- Attestation Results Appraisal: a Relying Party applies policy to Attestation Results associated with an Attester's Evidence that originates from a trusted Verifier. The results are trust decisions regarding the Attester.

To improve the confidence in a system component's trustworthiness, a relying party may require evidence about:

- system component identity,
- composition of system components, including nested components,
- roots of trust,
- an assertion/claim origination or provenance,
- manufacturing origin,
- system component integrity,
- system component configuration,
- operational state and measurements of steps which led to the operational state, or
- other factors that could influence trust decisions.

While domain-specific attestation mechanisms such as Trusted Computing Group (TCG) Trusted Platform Module (TPM)/TPM Software Stack (TSS), Fast Identity Online (FIDO) Alliance attestation, and Android Keystore attestation exist, there is no interoperable way to create and process attestation evidence to make determinations about system components among relying parties of different manufactures and origins.IETF-Remote ATestation ProcedureS (RATS) https://datatracker.ietf.org/group/rats/about/

KERI and Dynamic Appraisability

Duplicity Evident infrastructure dynamically (in near real-time) ensures that evidence of both live and dead compromise of key state is available to any validator.

Any validator can then perform a live appraisal of key state compromise before engaging in any trust task.

Live appraisal can be staged (graduated) to match the level of risk with the degree of evidence supporting the appraisal prior to committing to the trust determination.

Duplicity Evident Infrastructure removes the hard reliance on trusted third parties to perform live appraisals.

It is the simplest known approach to solving the dynamic appraisability problem without relying on trusted third parties or blockchain.

This is one of the primary security innovations that KERI as DKMI.

Duplicity Evident(KERI)

Duplicity Hiding (blockchain)

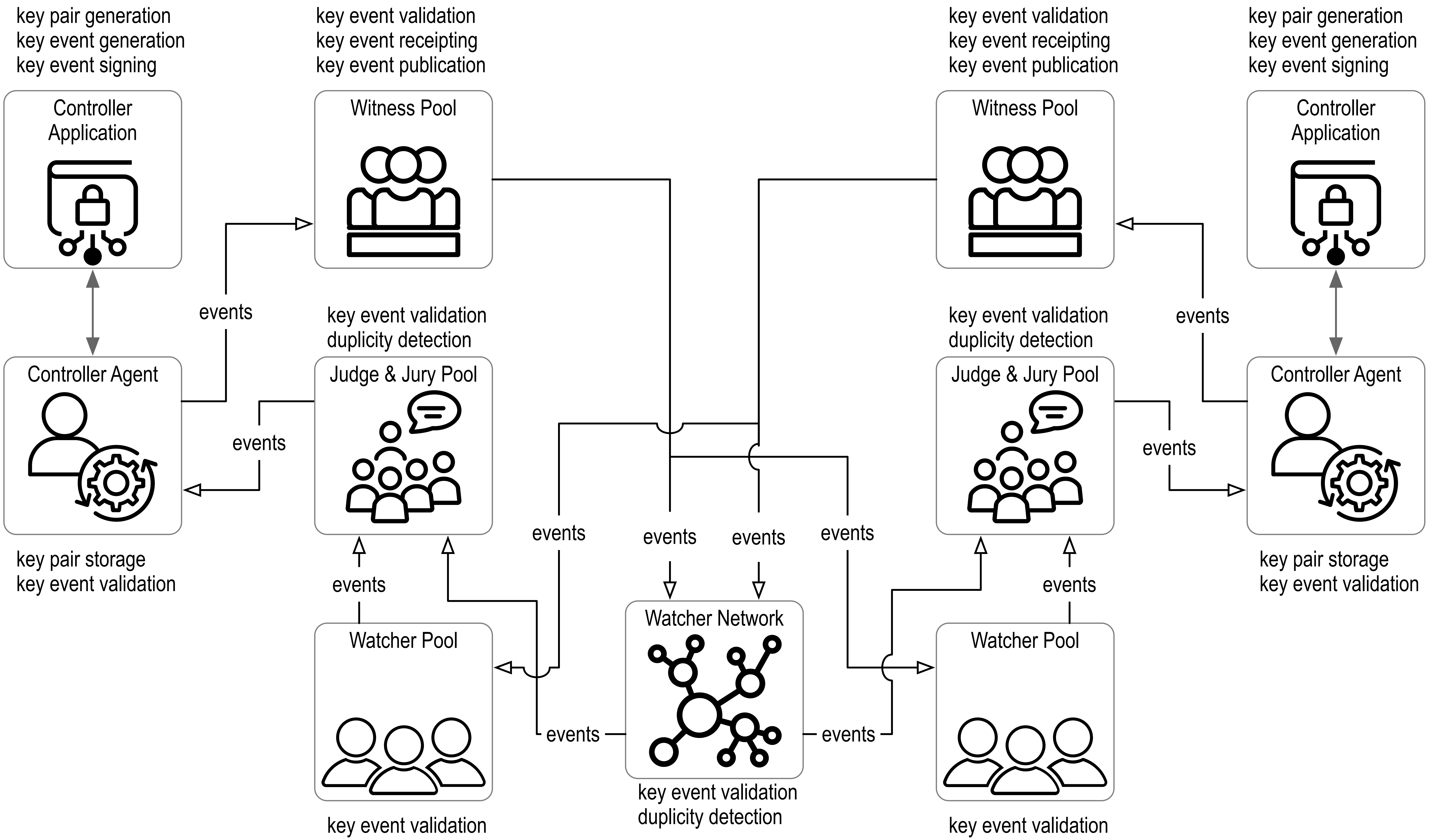
Duplicity Fostering(DNS/CA)

Live Compromised Signing Keys

Detectability

Recoverability

KERI Ecosystem Components: Witnesses and Watchers, Advanced Indirect Mode

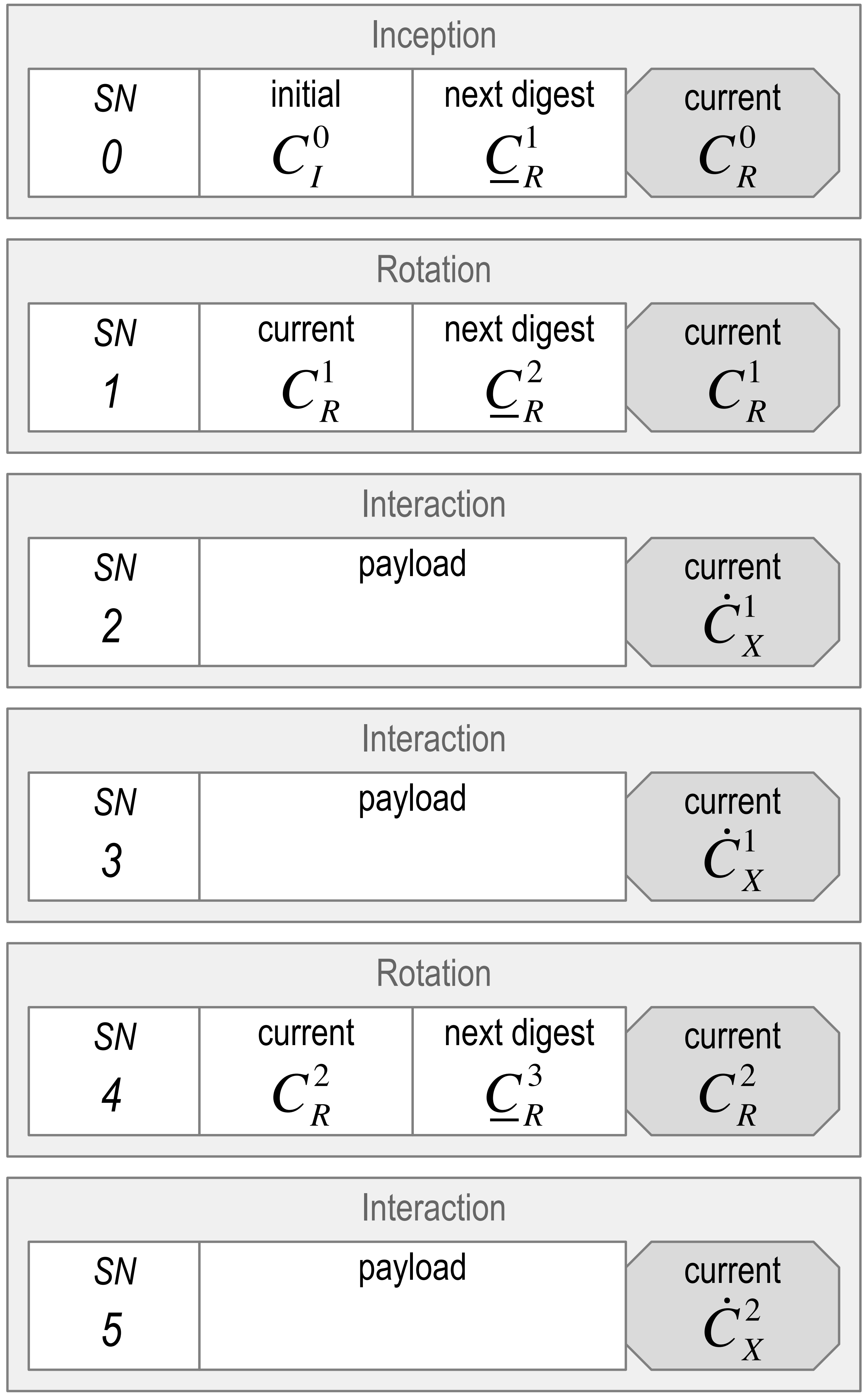


Ambient Verifiability

Live Exploit (current signing keys)

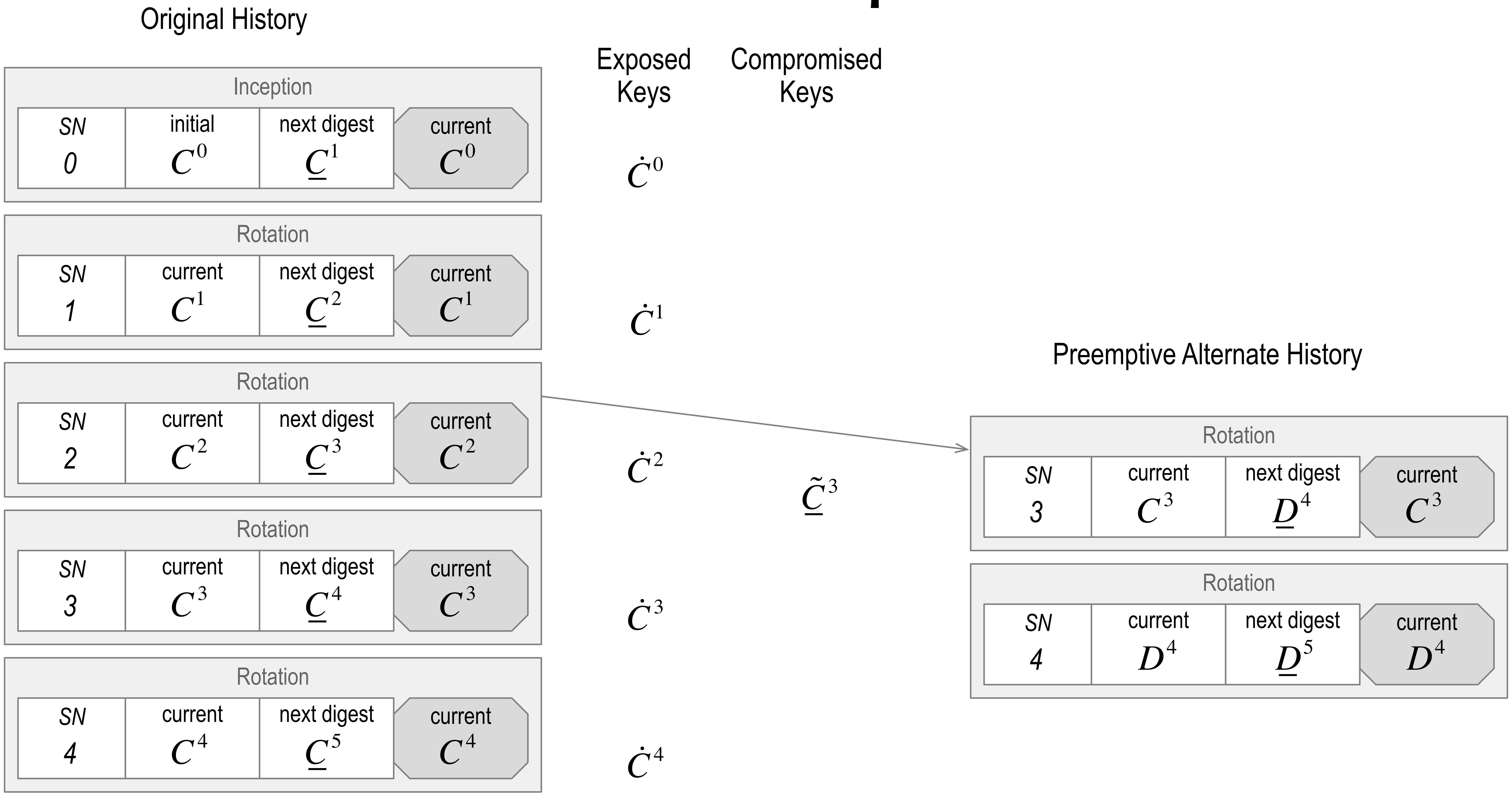
Hard Problem:

Recovery from Live Exploit of Current Signing Keys



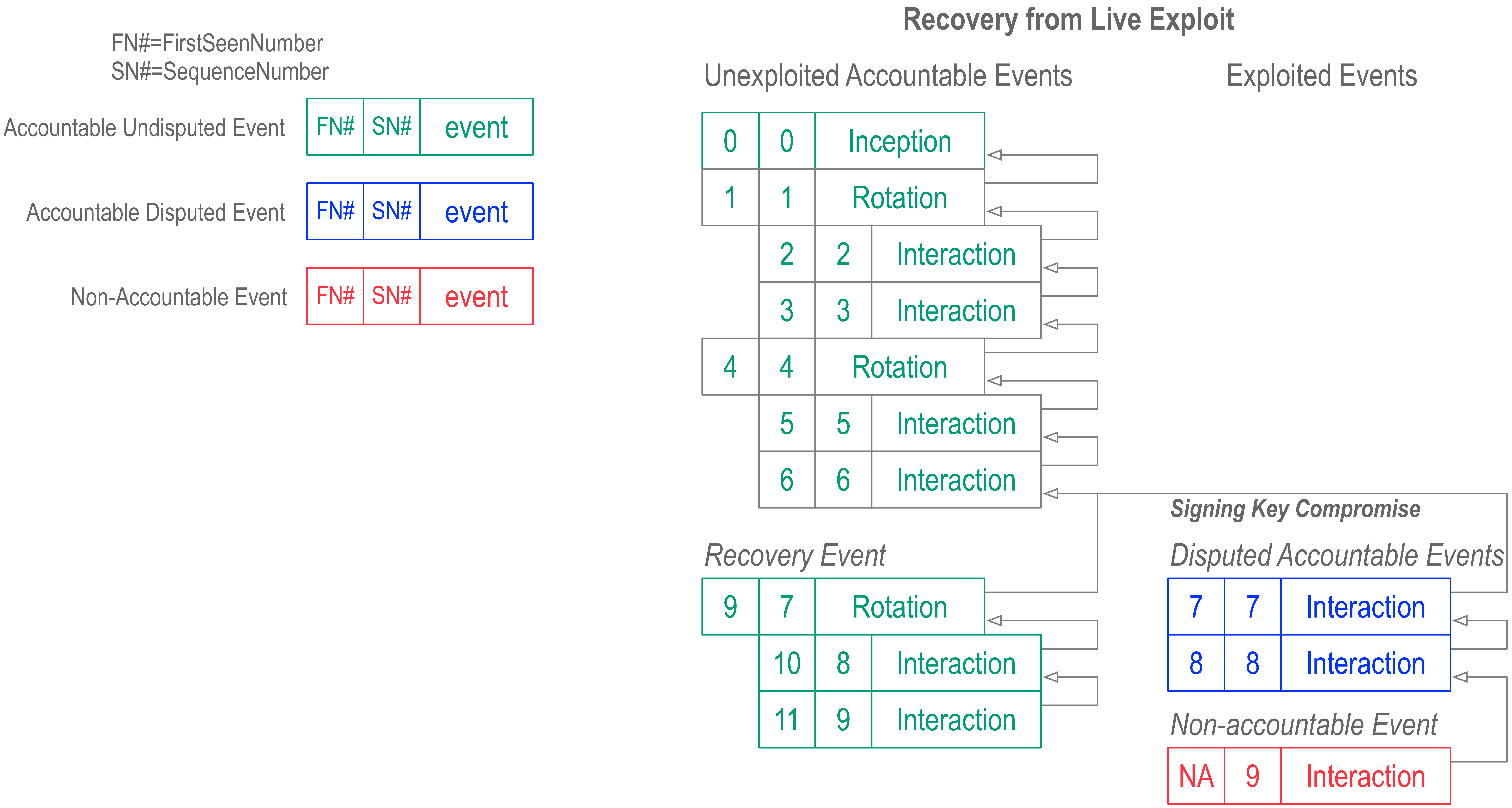
Pre-rotation provides protection from successful *live* exploit of current signing keys.

Live Exploit (next signing keys)



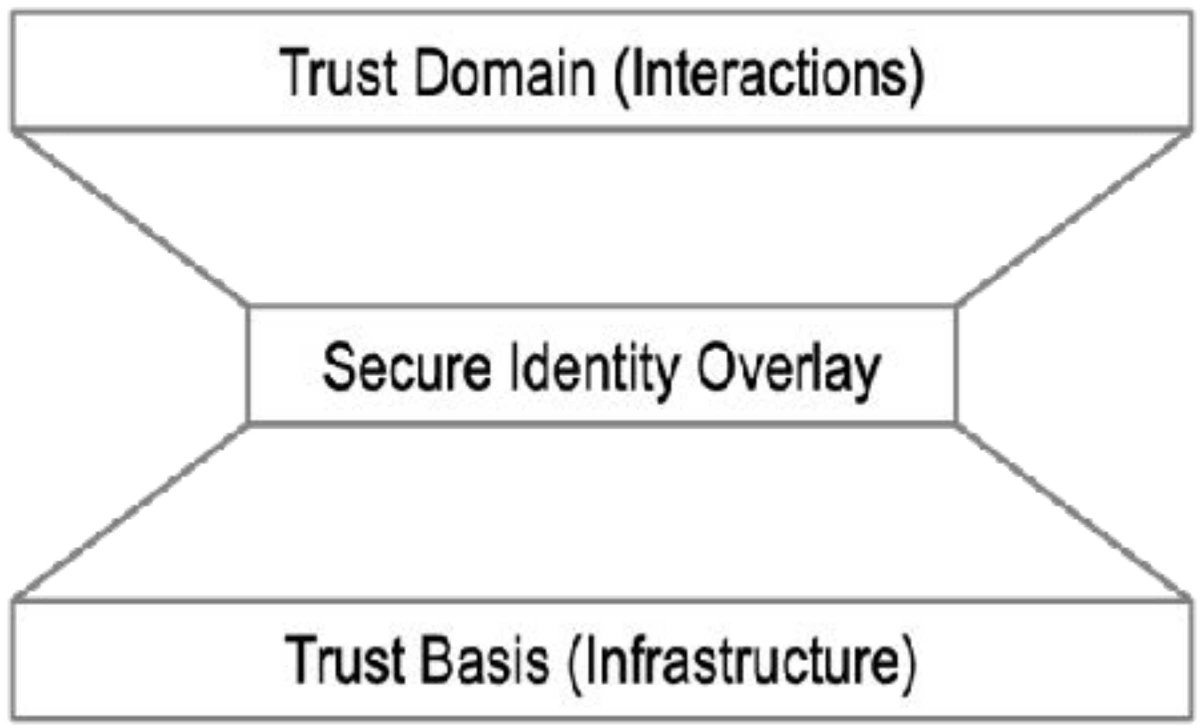
Difficulty of inverting *next* key(s) protects against successful *live* exploit.

Recovery from Live Exploit Of Current Signing Keys



Trust Domain

In security systems design, a *root-of-trust* is some component or process of the system upon which other components or processes are reliant. The *root-of-trust* has trustworthy security properties that provide a foundation of trust that other components or processes in the system may rely on.



A *primary* root-of-trust is *irreplaceable*.

A *secondary* root-of-trust is *replaceable*.

A *trust basis* binds controllers, identifiers, and key-pairs.

A *trust domain* is the ecosystem of interactions (functions) that rely on a trust basis.

The hard problem is cross-domain value transfer.The solution is transitive trust.

A *secure identity overlay* maps the *trust basis* to the *trust domain.*

Roots-of-trust, sources-of-truth, and loci-of-control.

Trust?

Trust must vs. may.

What must be trusted as a vs. what may be trusted as *root-of-trust*.

The validator **relies** on a *root-of-trust* as a *source-of-truth*.

How may one "ensure" a confidence level in the *roots-of-trust* as *sources-of-truth* in its *loci-of-control*.

WRT a relying party, a trusted third-party is a source-of-truth (root-of-trust) that is outside the loci-of-control of the relying party. The relying party has no ability to ensure a desired confidence level in the "truth" provided by the trusted third-party. It must "trust".

Private watchers and eclipse attacks:

Guy with two exclusive girlfriends at the same time.

Indefinitely Verifiable Issuances Despite Key Compromise

No Shared Secrets

Phish Resistant (both technically and psychologically)

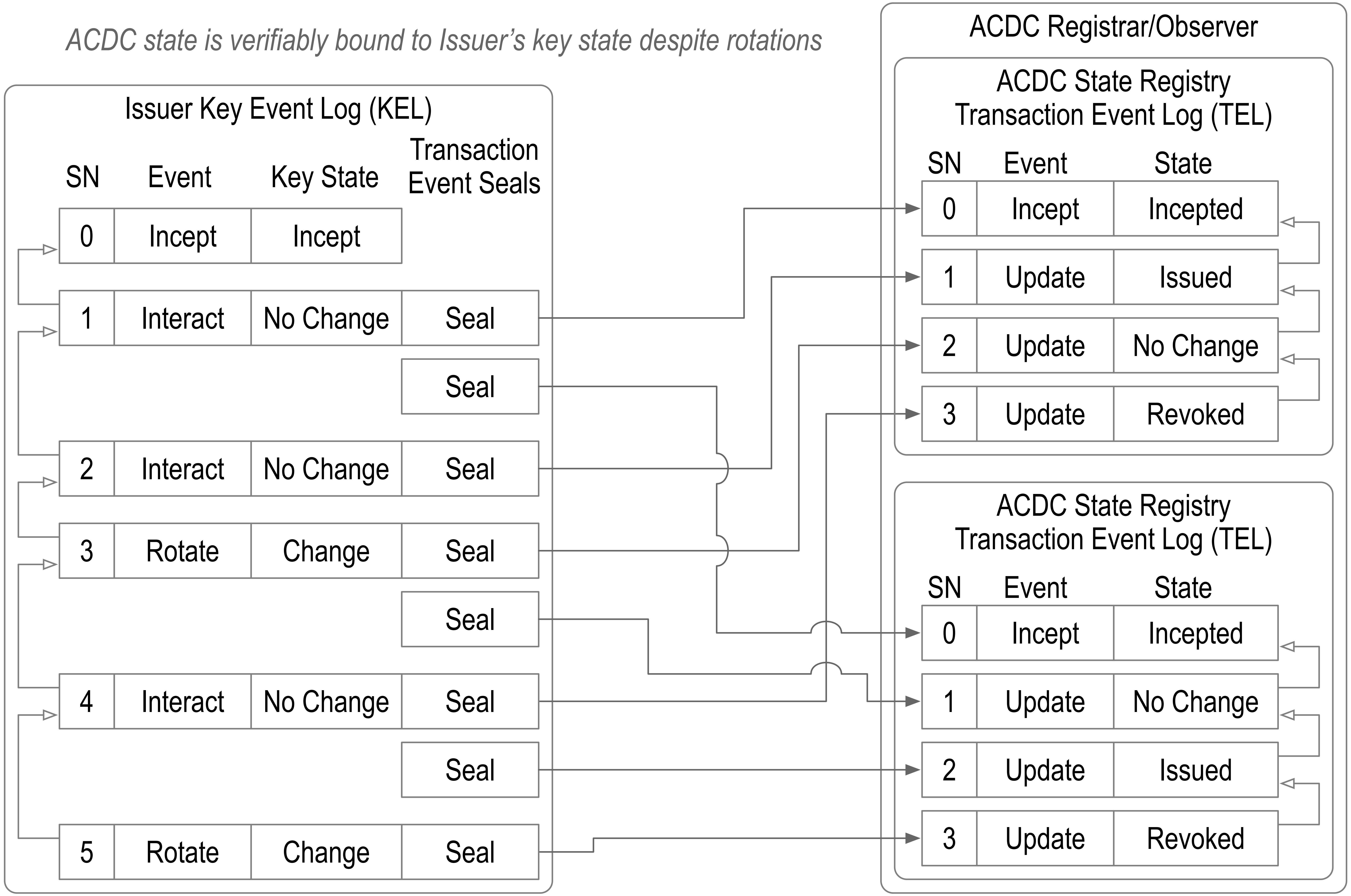
No trusted third-party registries (VDRs not TRs)

No trusted device lock-in

Signed (**Sealed**) Everything

ACDC State Registry using Transaction Event Log (TEL)

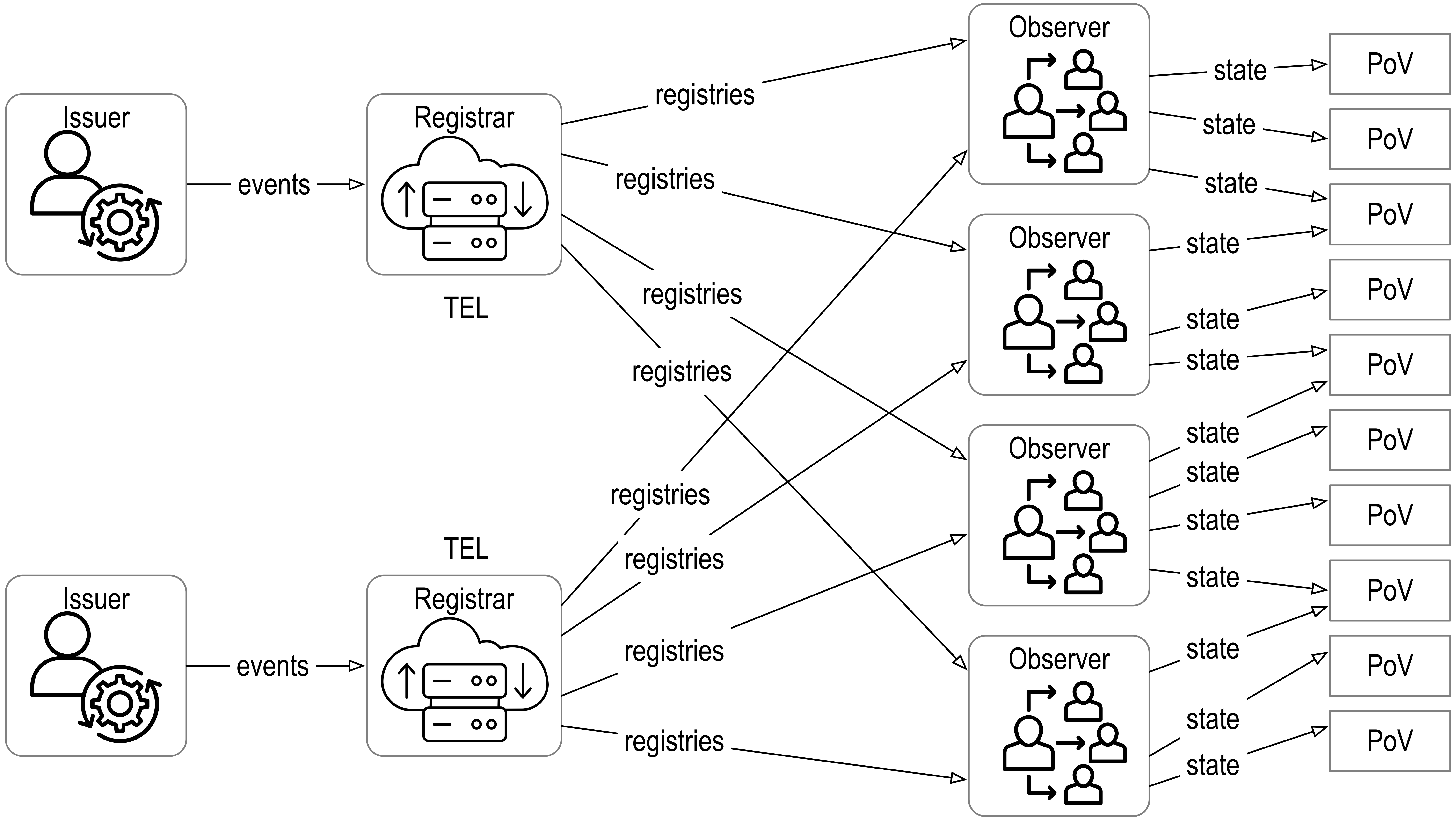
Each Transaction Event is bound to Issuer key stated via anchoring seal in Issuers KEL



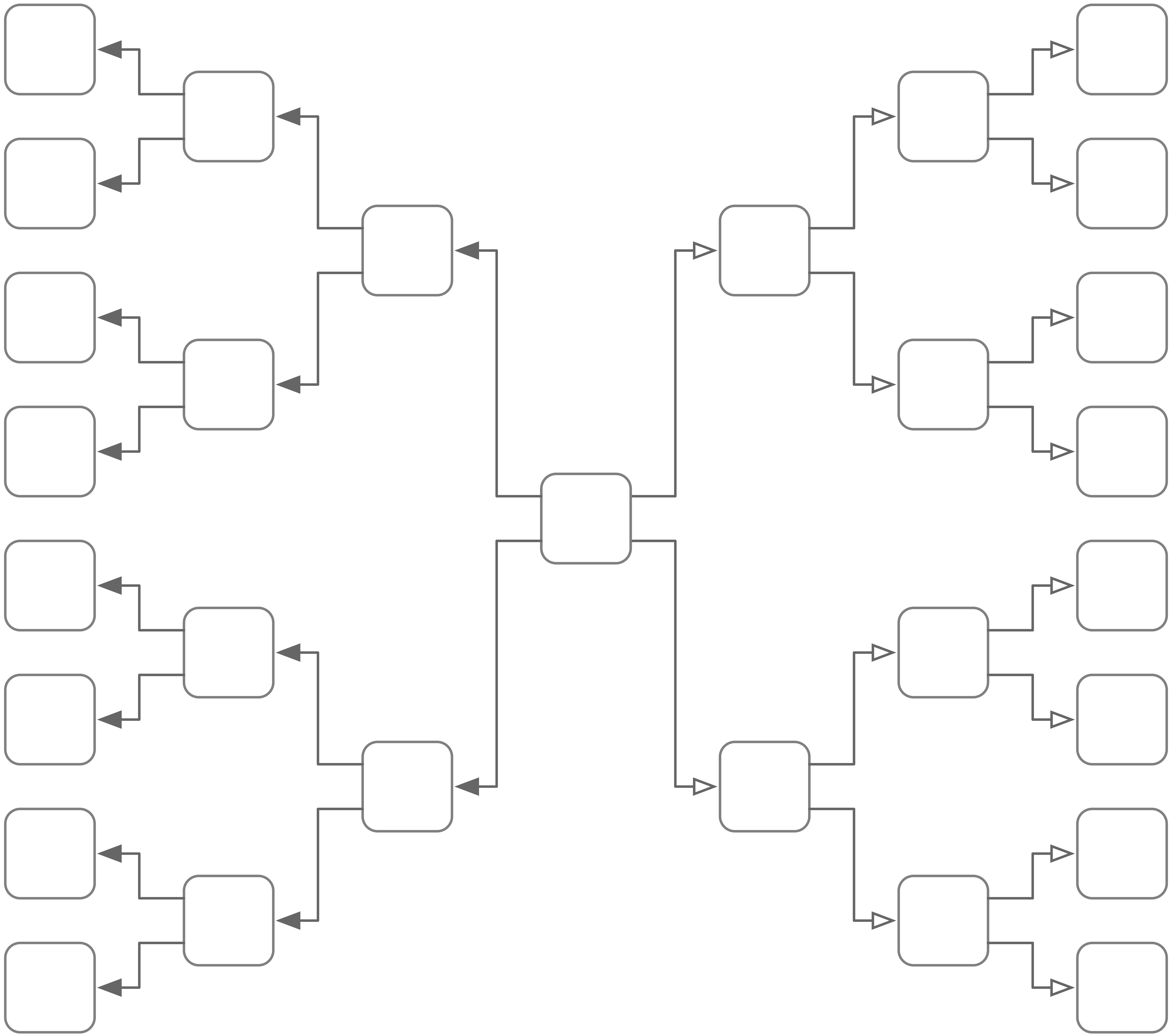
Registars and Observers

TEL Registrar operates under the auspices of the ACDC Issuer to maintain and publish a Registry of the ACDC state via a TEL.

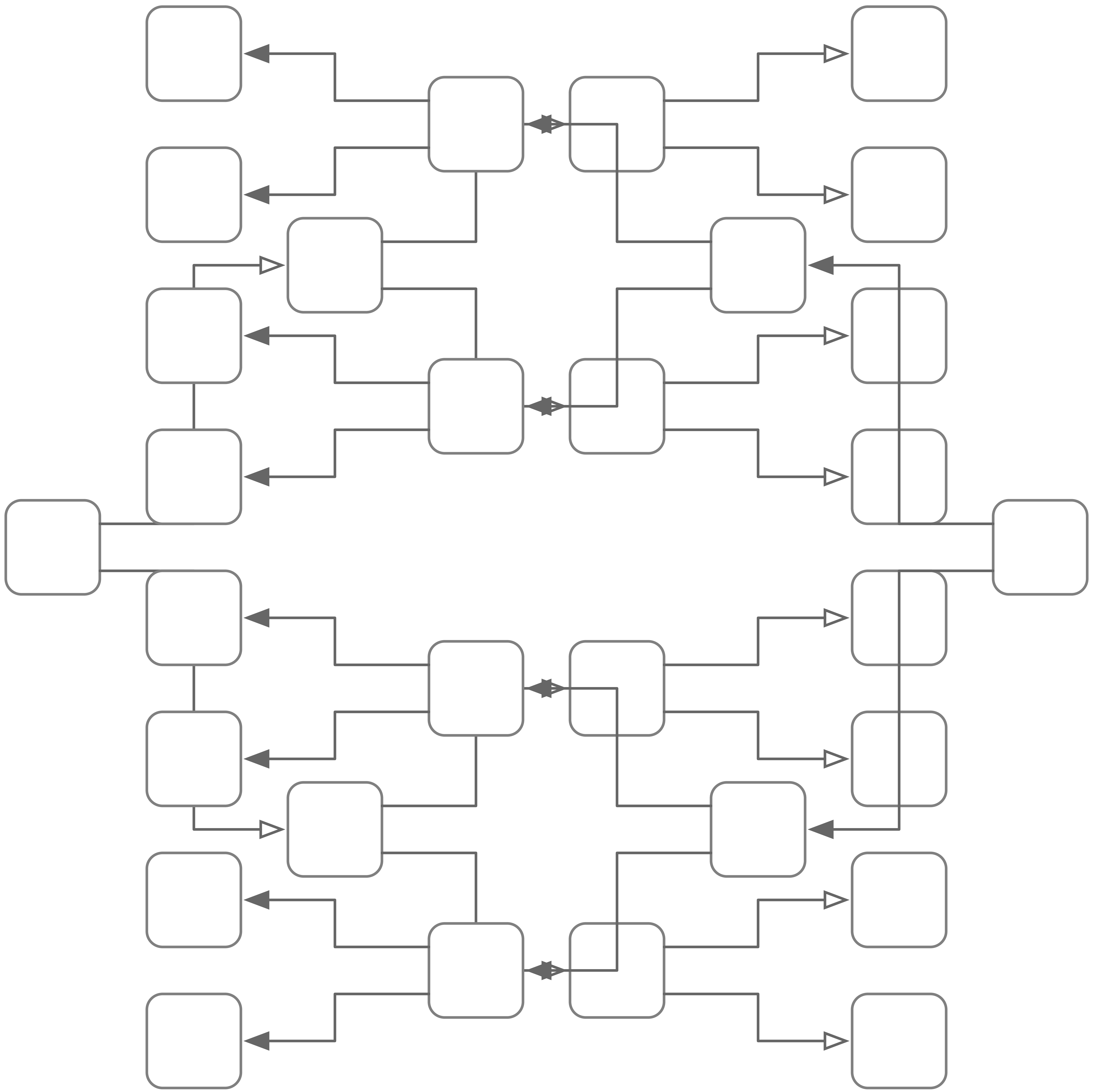
TEL Observer, on the other hand, is a computing component that operates under the auspices of one or more Validators to cache the Registry, allowing Validators to validate the state of a given ACDC without exposing a point of validation (PoV). To clarify, an important feature of an Observer is that it can mask the usage of a given ACDC from the Issuer. The Observer maintains an updated cache of the Registry, reflecting state updates provided by the Registry. A validator queries its Observer, not the Registrar, at a point of validation (PoV) for an ACDC. A point of validation (PoV) occurs when an ACDC is presented to a Validator for validation. Whereas the interactions between the Observer and Registrar occur when there are ACDC state changes, not when there is a PoV of a given ACDC state. This protects against forced validator-to-issuer correlation of ACDC usage, i.e., no forced phone home validation.



BowTie Model of Verifiable Data



Graph Model of Verifiable Data (interconnected bowties)



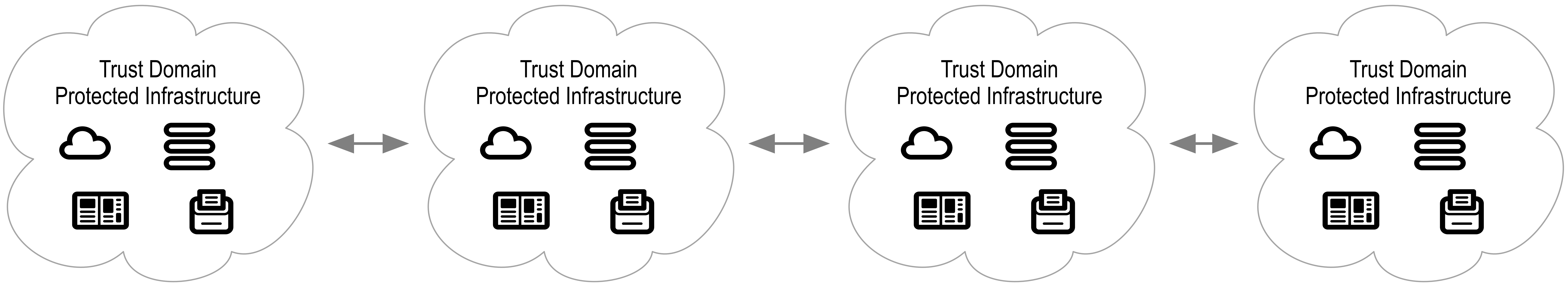
Cross Domain Trust Transfer Problem

The *hard problem* of identity.

All popular identity architectures are ill-suited to solve this problem securely!

Closed-loop architectures: OpenID, Kerberos

Open-loop architectures: DNS/CA



Trust Domain

A *primary* root-of-trust is *irreplaceable*.

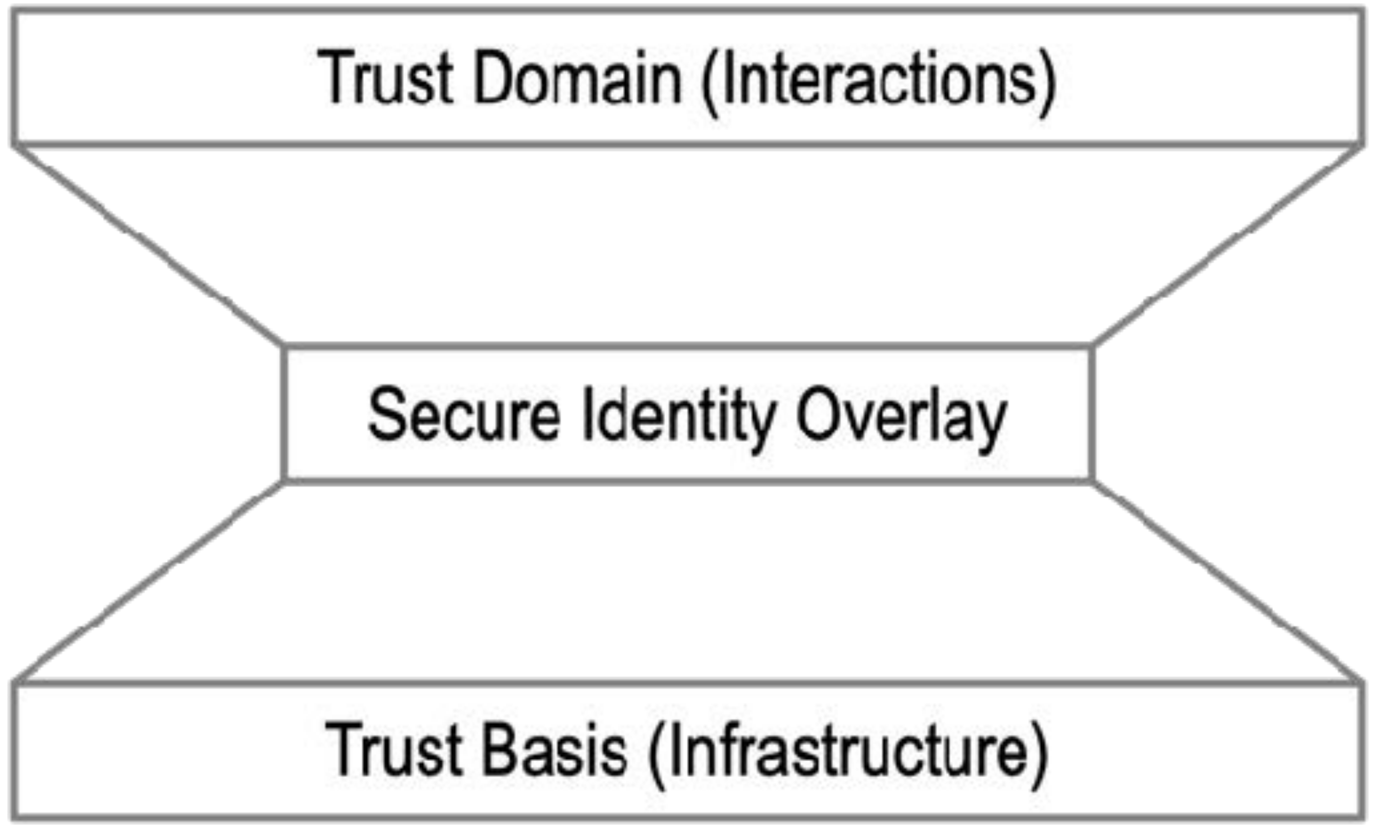
A *secondary* root-of-trust is *replaceable*.

A *trust basis* binds controllers, identifiers, and key-pairs.

A *trust domain* is the ecosystem of interactions (functions) that rely on a trust basis.

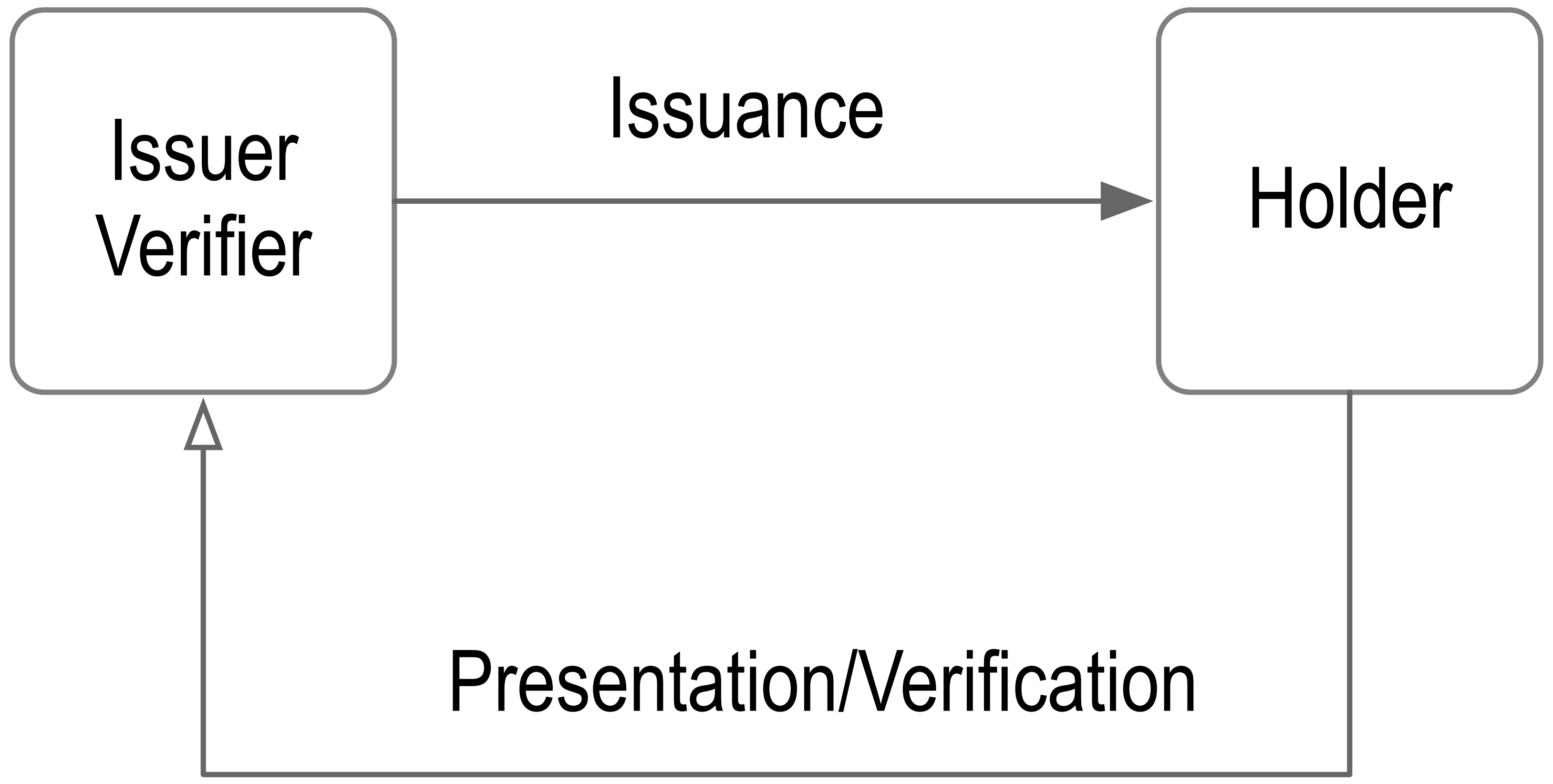
The hard problem is cross-domain value transfer.
The solution is transitive trust.

A *secure identity overlay* maps the *trust basis* to the *trust domain.*

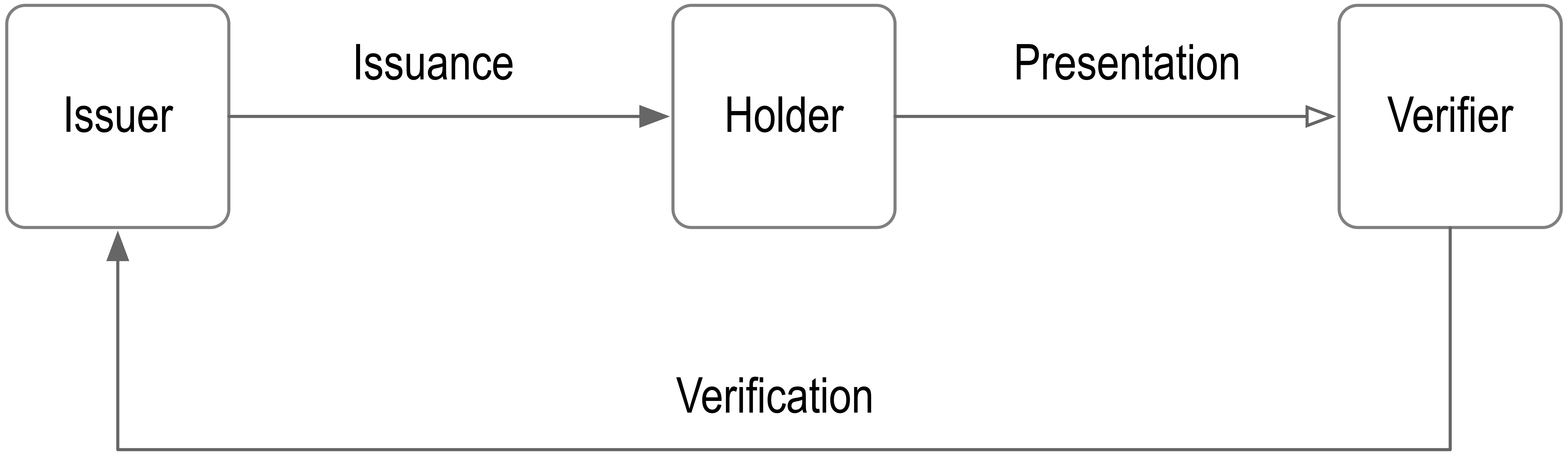


Closed Loop Models

Closed Loop Issuer-Holder Model

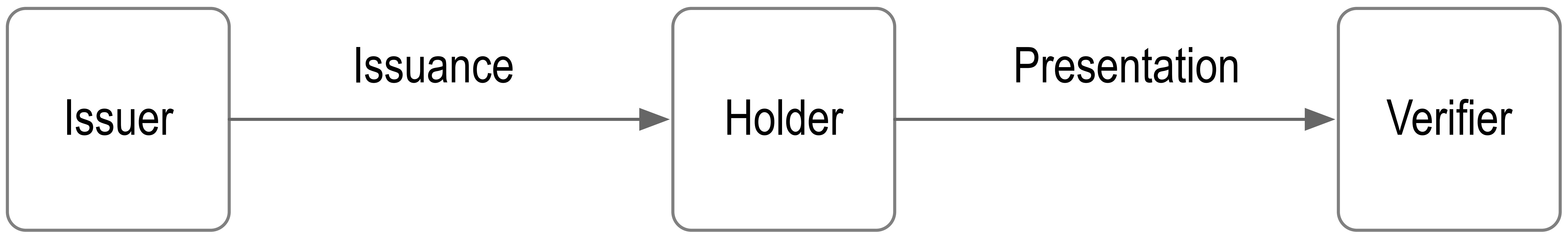


Closed Loop Issuer-Holder-Verifier Model

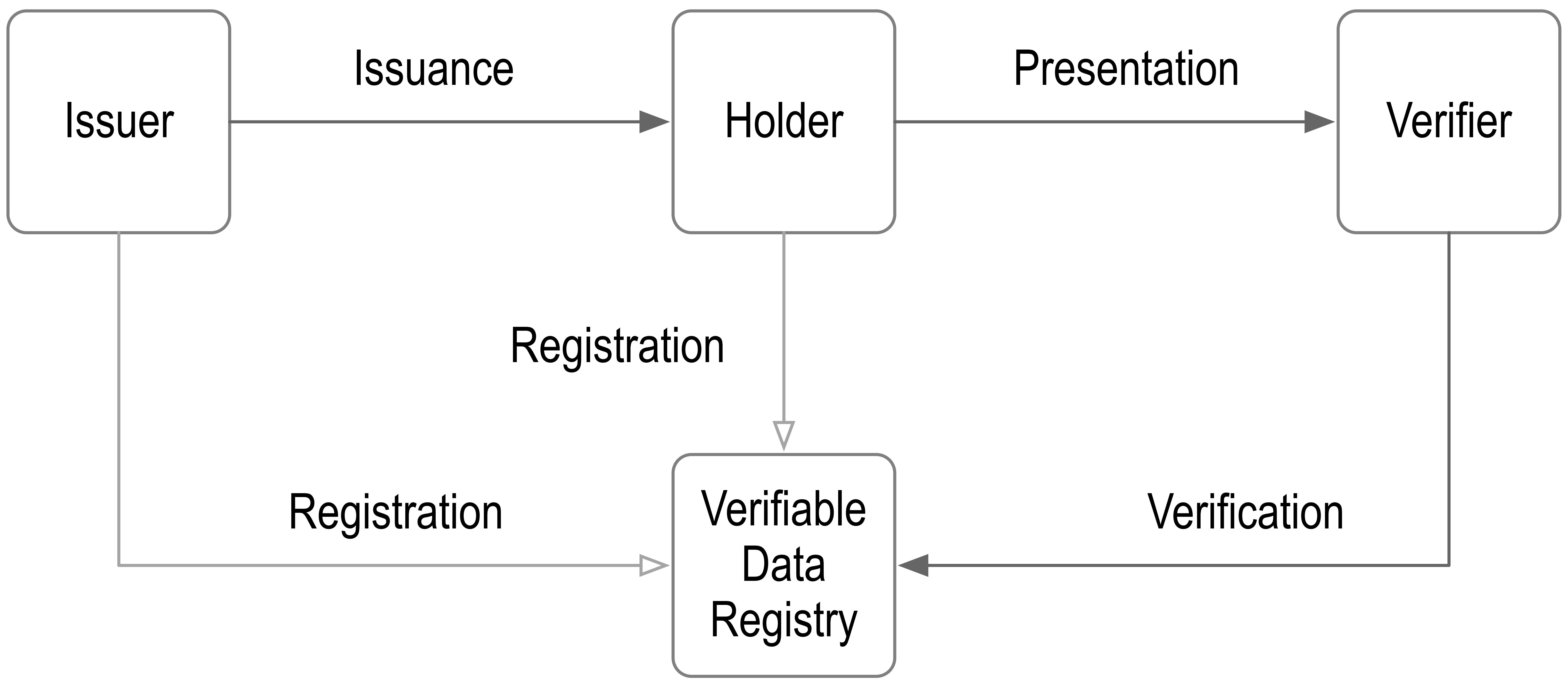


Open Loop Models

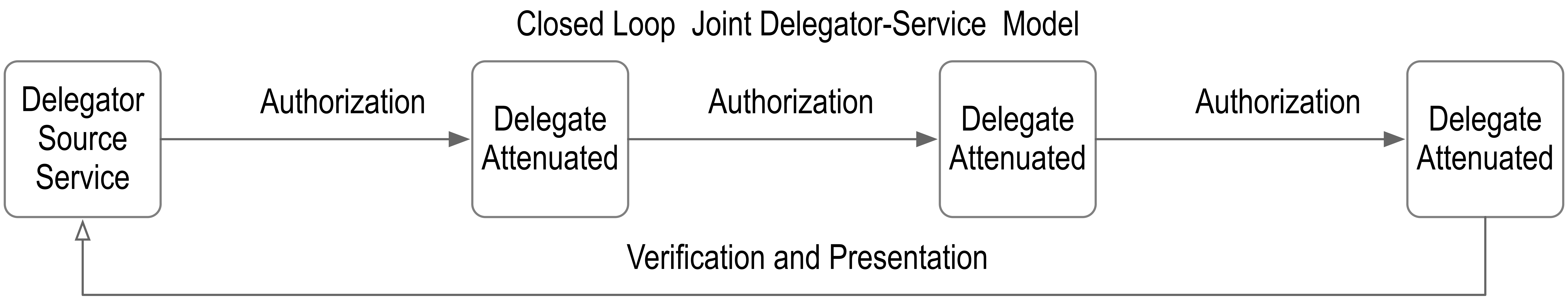
Open Loop Issuer-Holder-Verifier Model

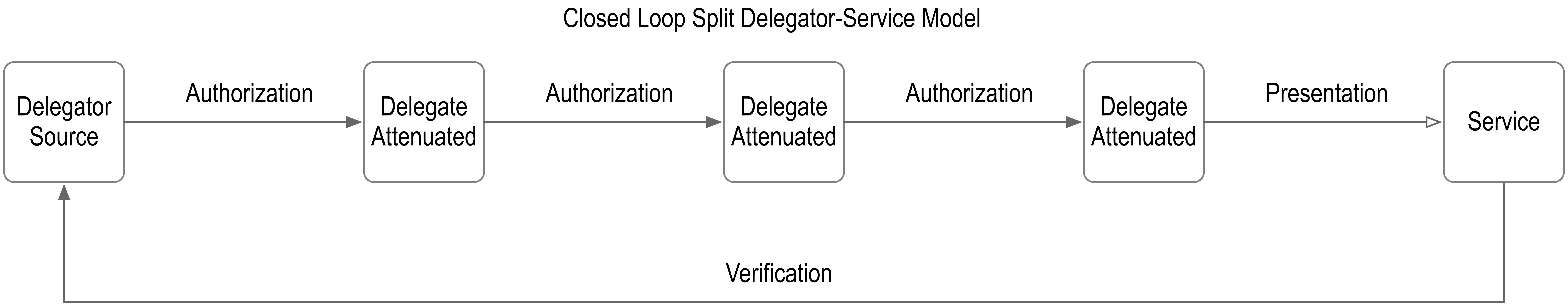


Open LoopIssuer-Holder-Verifier VDR Model



Closed Loop Delegation Models

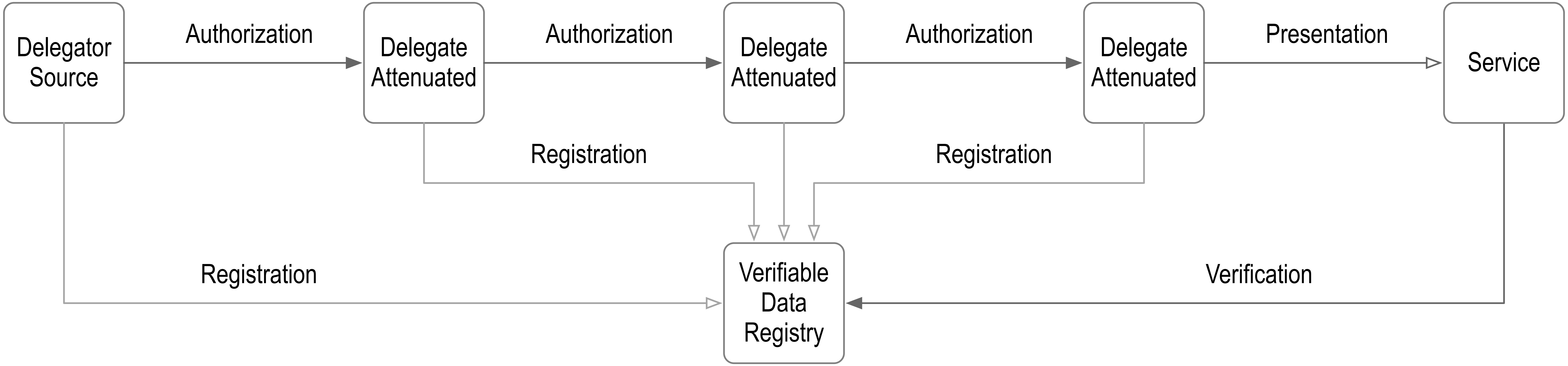




Open Loop Delegation Models

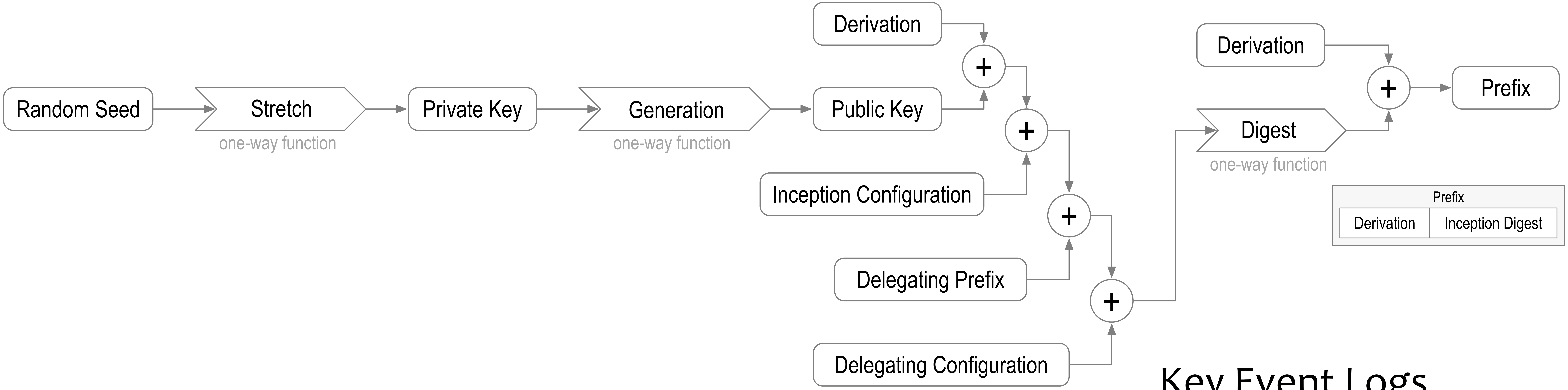
Open Loop Split Delegator-Service Model

Open Loop Split Delegator Service VDR Model

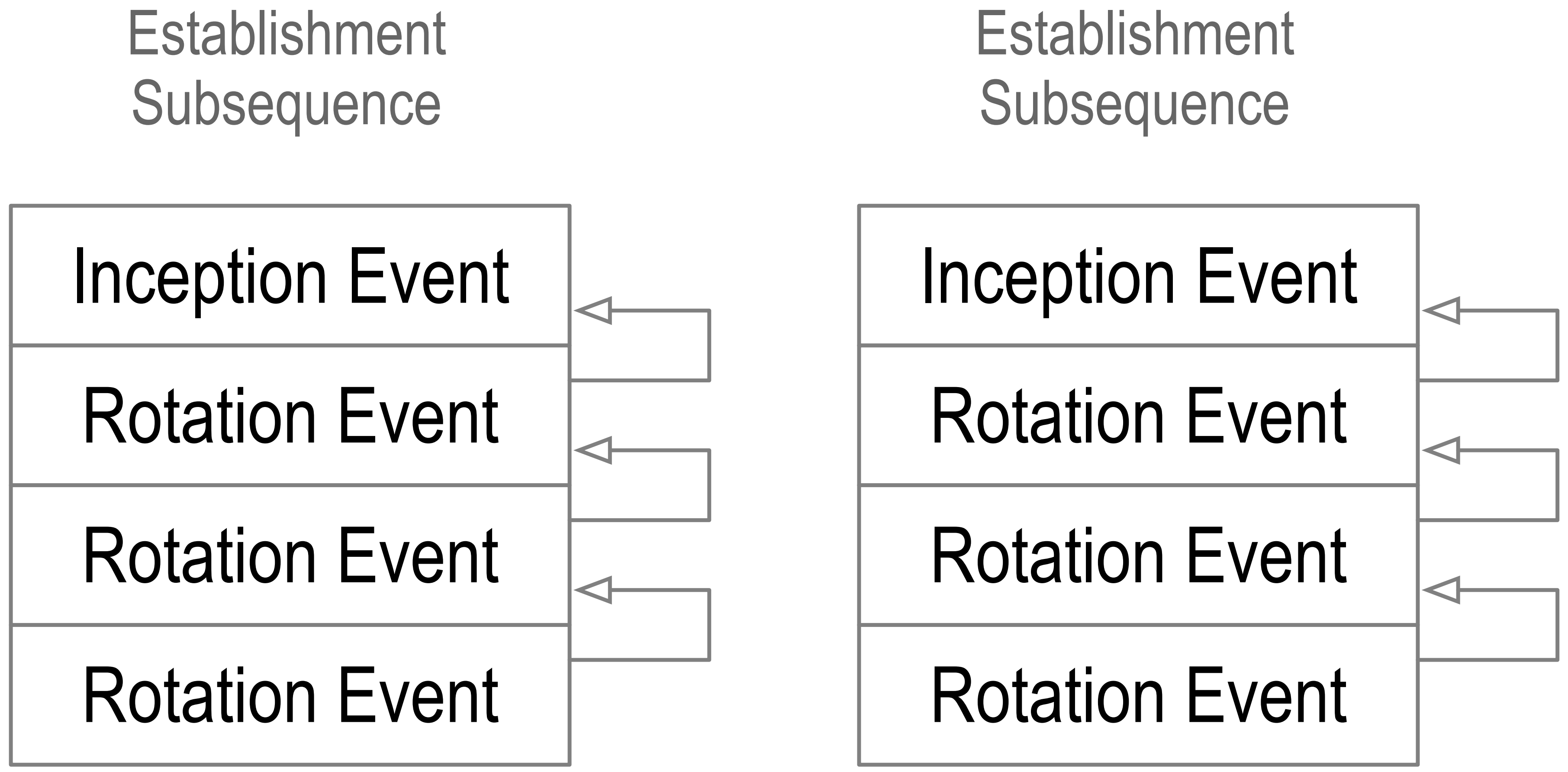


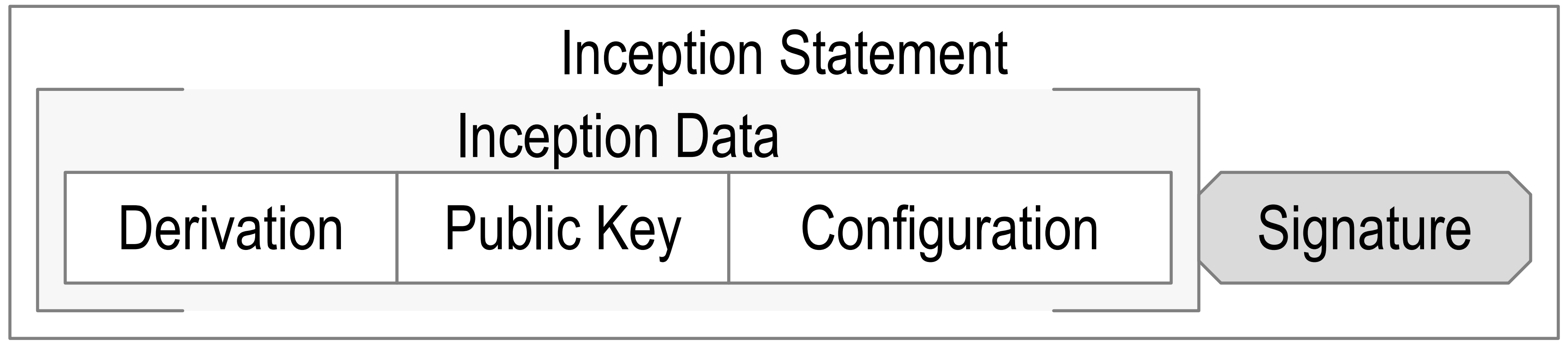
Unlimited Scalability of Secure Signing Infrastructure

Delegated Identifiers



Key Event Logs

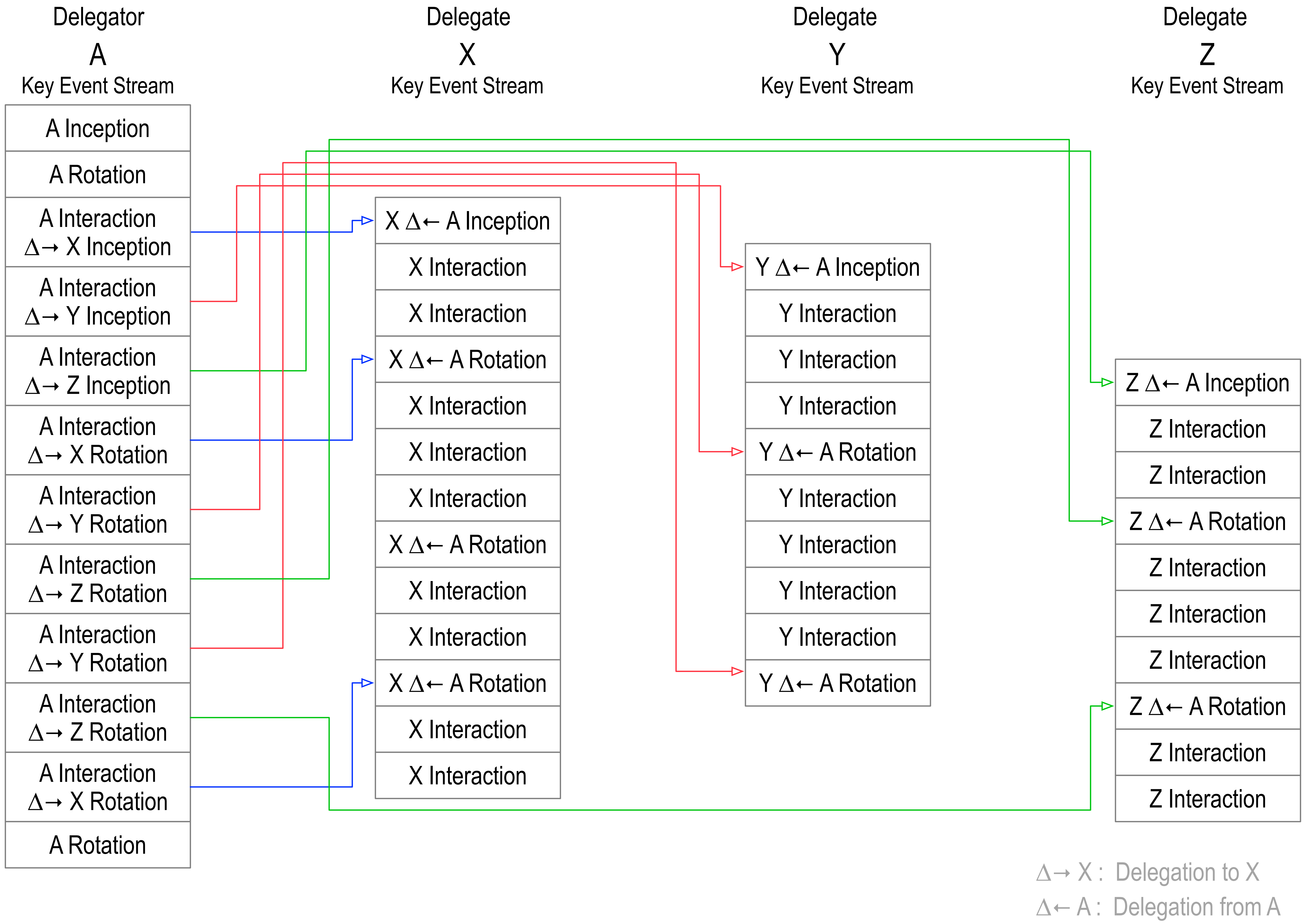


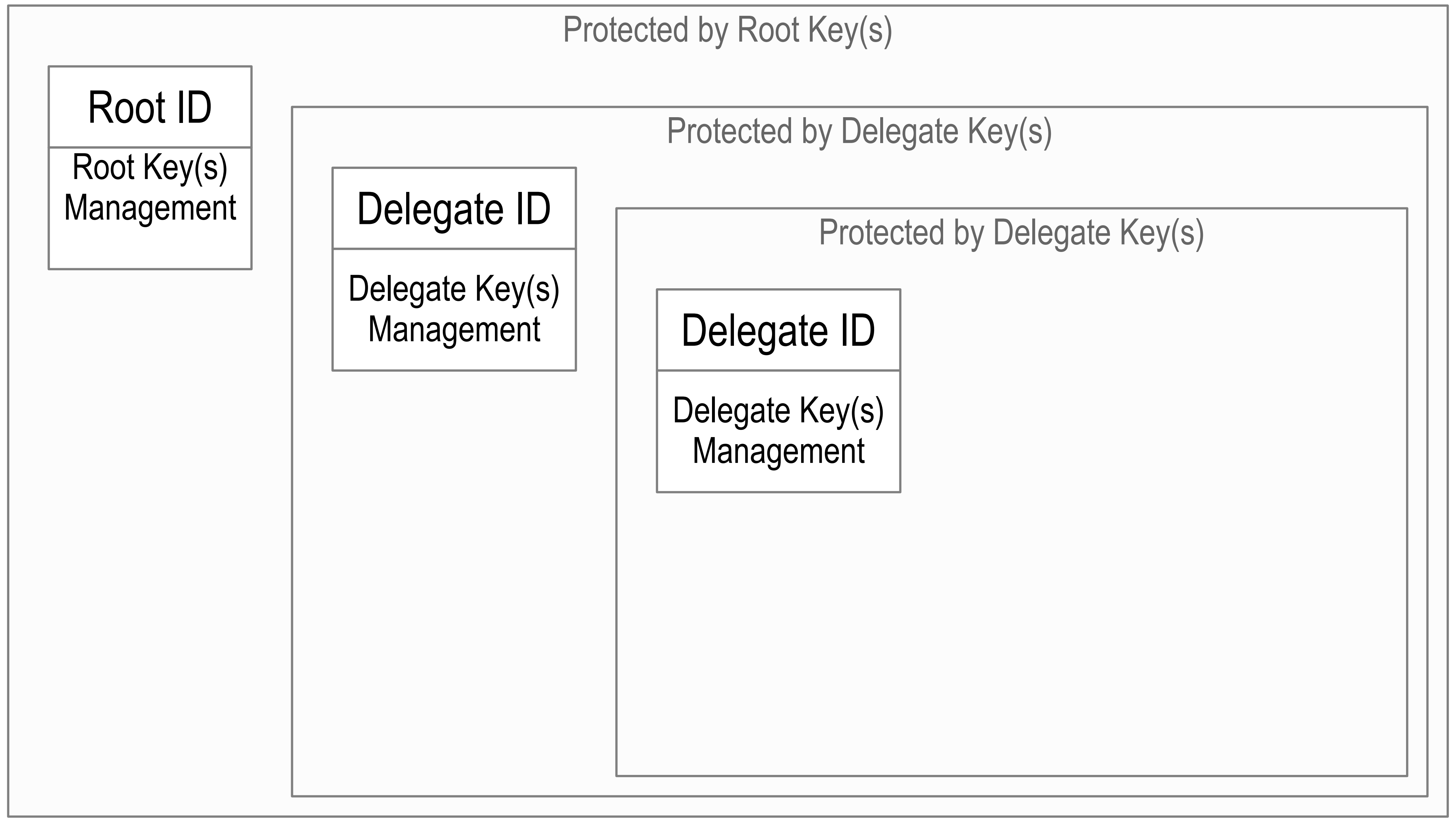


`EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148`

`did:keri:EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148/path/to/resource?name=sec#yes`

Identifier Delegation: Scaling & Protection





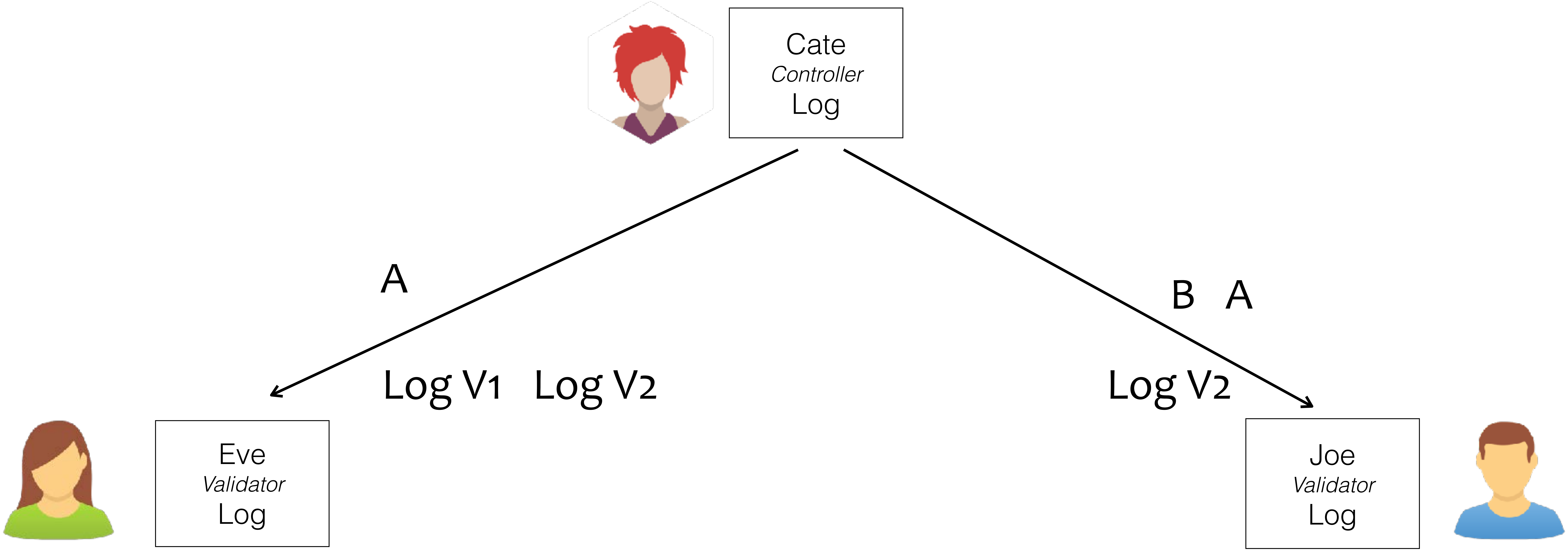
Backup

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?

Cate
Controller
Log

Truncate Log

Delete Log

Service/Agent
Controlled by Cate
Log

A

B A

Log V1 Log V2

Log V2

Eve
Validator
Log

Joe
Validator
Log

highly available, private (one-to-one) interactions

Duplicity Game

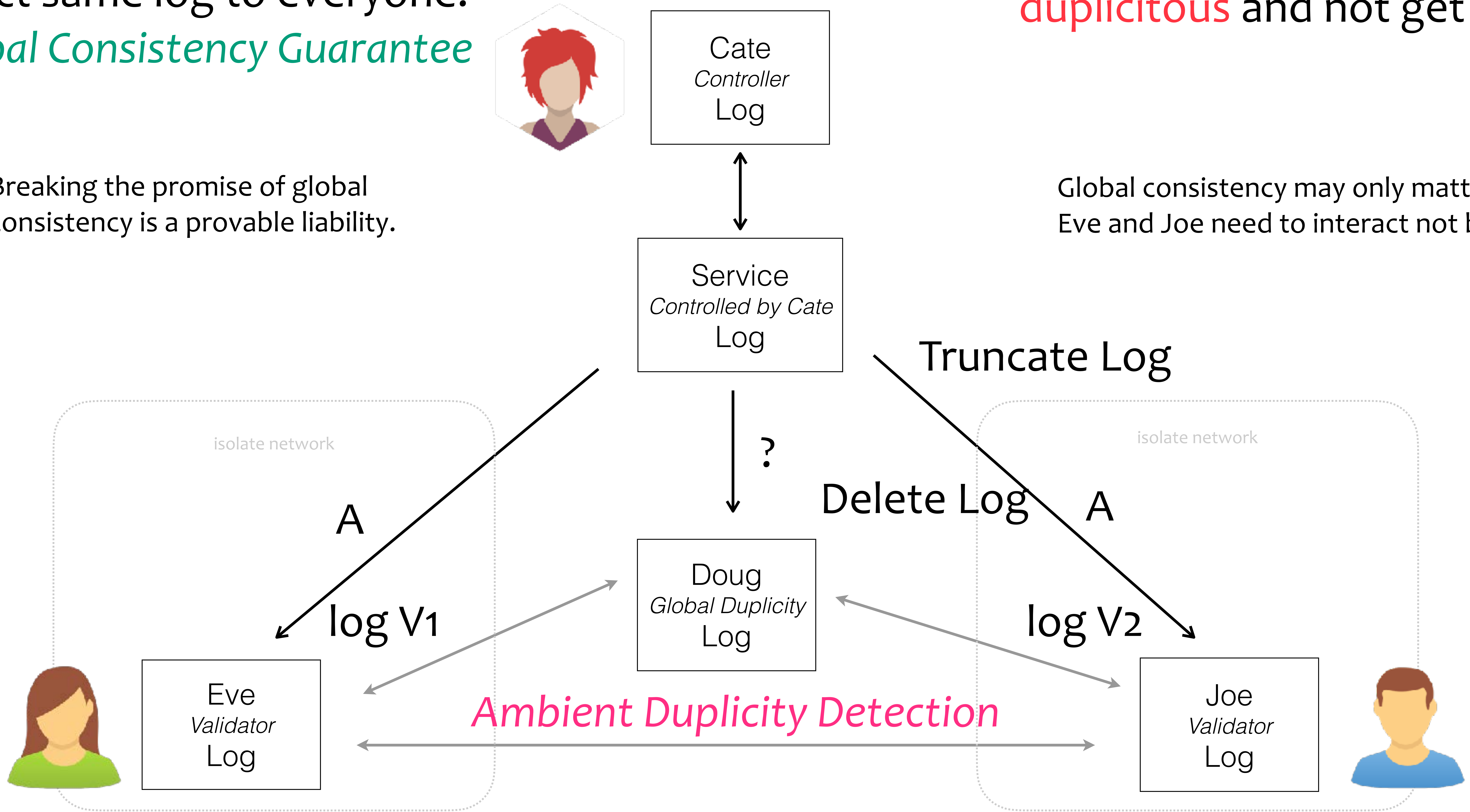
Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.

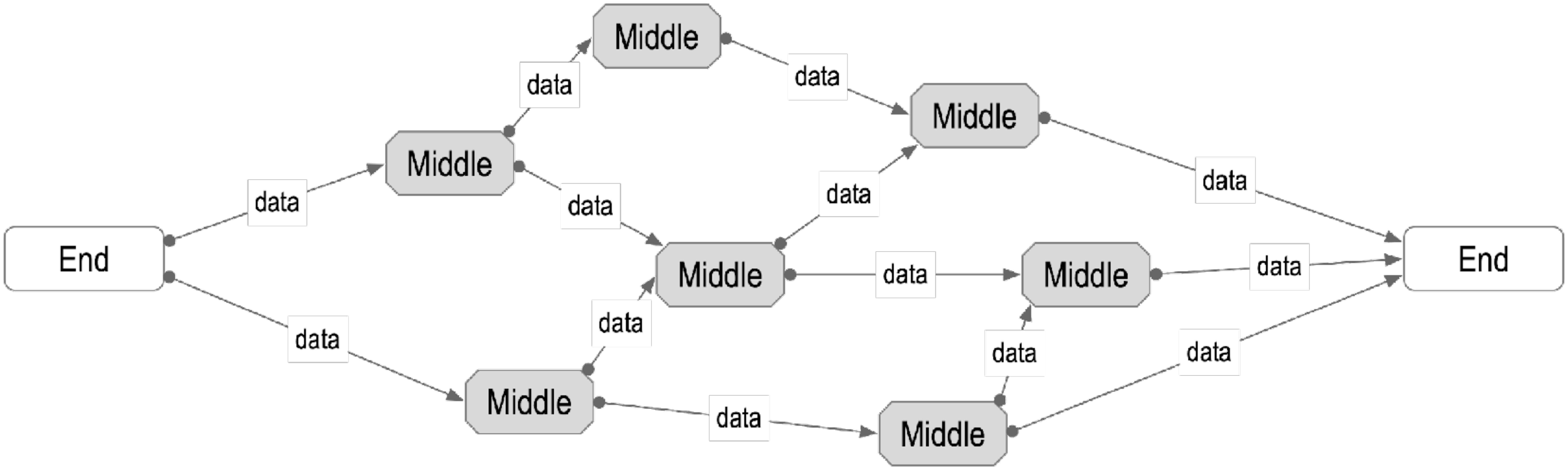
Global consistency may only matter after Eve and Joe need to interact not before.



global consistent, highly available, and public (one-to-any) interactions

End Verifiability

End-to-End Verifiability



If the edges are secure, the security of the middle doesn't matter.

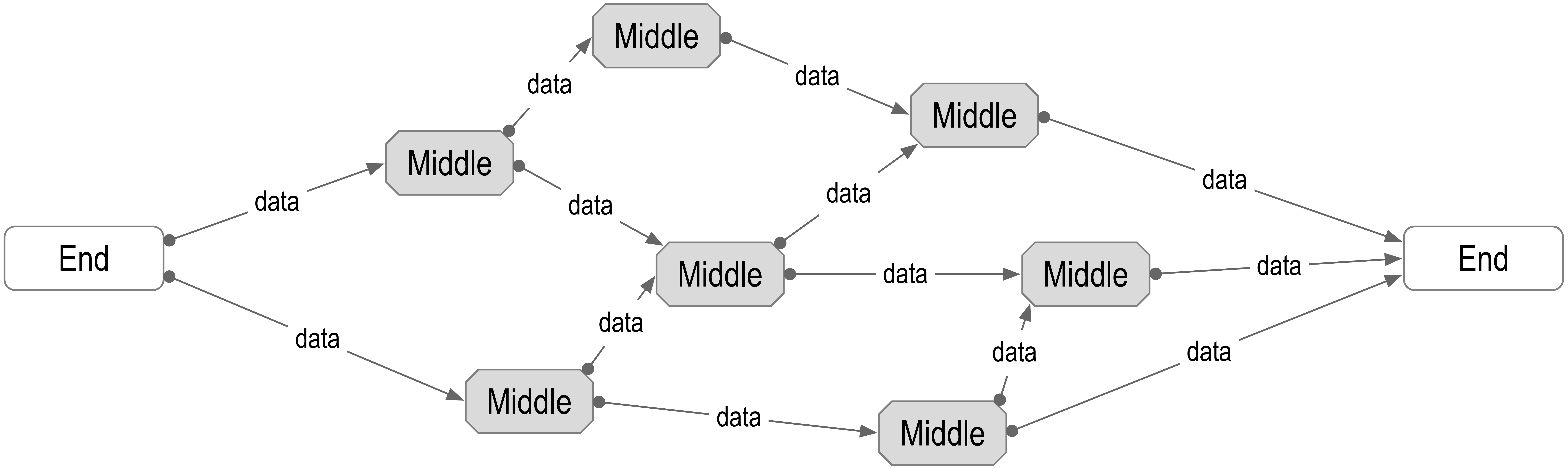
Ambient Verifiability: any-data, any-where, any-time by any-body

Zero-Trust-Computing

It's much easier to protect one's private keys than to protect everyone else's internet infrastructure

Dual: End Verifiability and End-Only Viewability

End-to-End Verifiability *of Authenticity*
Only-at-End Viewability via Confidentiality



Ambient Verifiability: any-data, any-where, any-time by any-body

End only Viewability: one-data, one-where, one-time by one-body

If the edges are secure, the security of the middle doesn't matter.

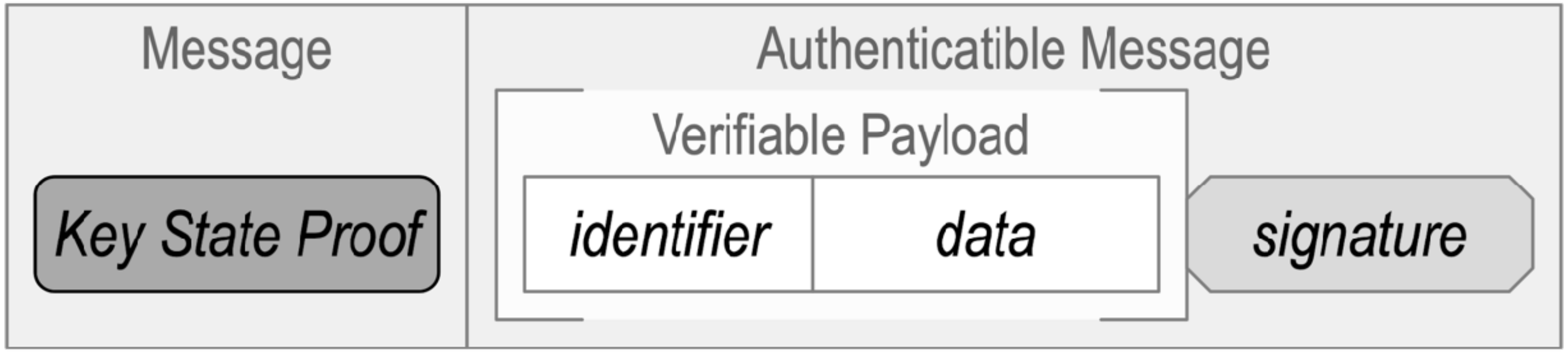
Zero-Trust-Computing

Its much easier to protect one's private keys than to protect all internet infrastructure

Identity (-ifier) System Security Overlay

persistent mapping via verifiable data structure of key state changes

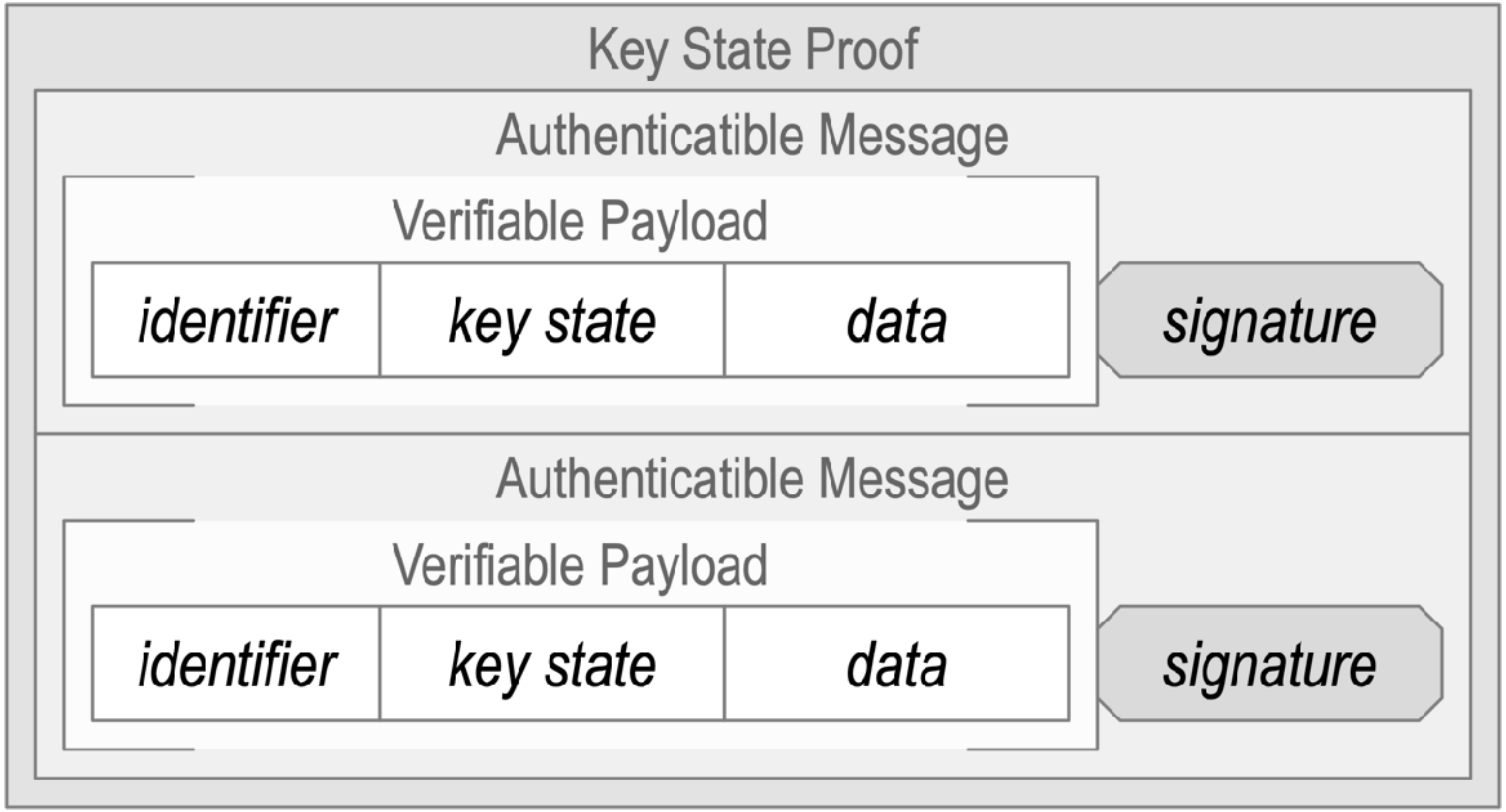
Establish authenticity of IP packet's message payload.



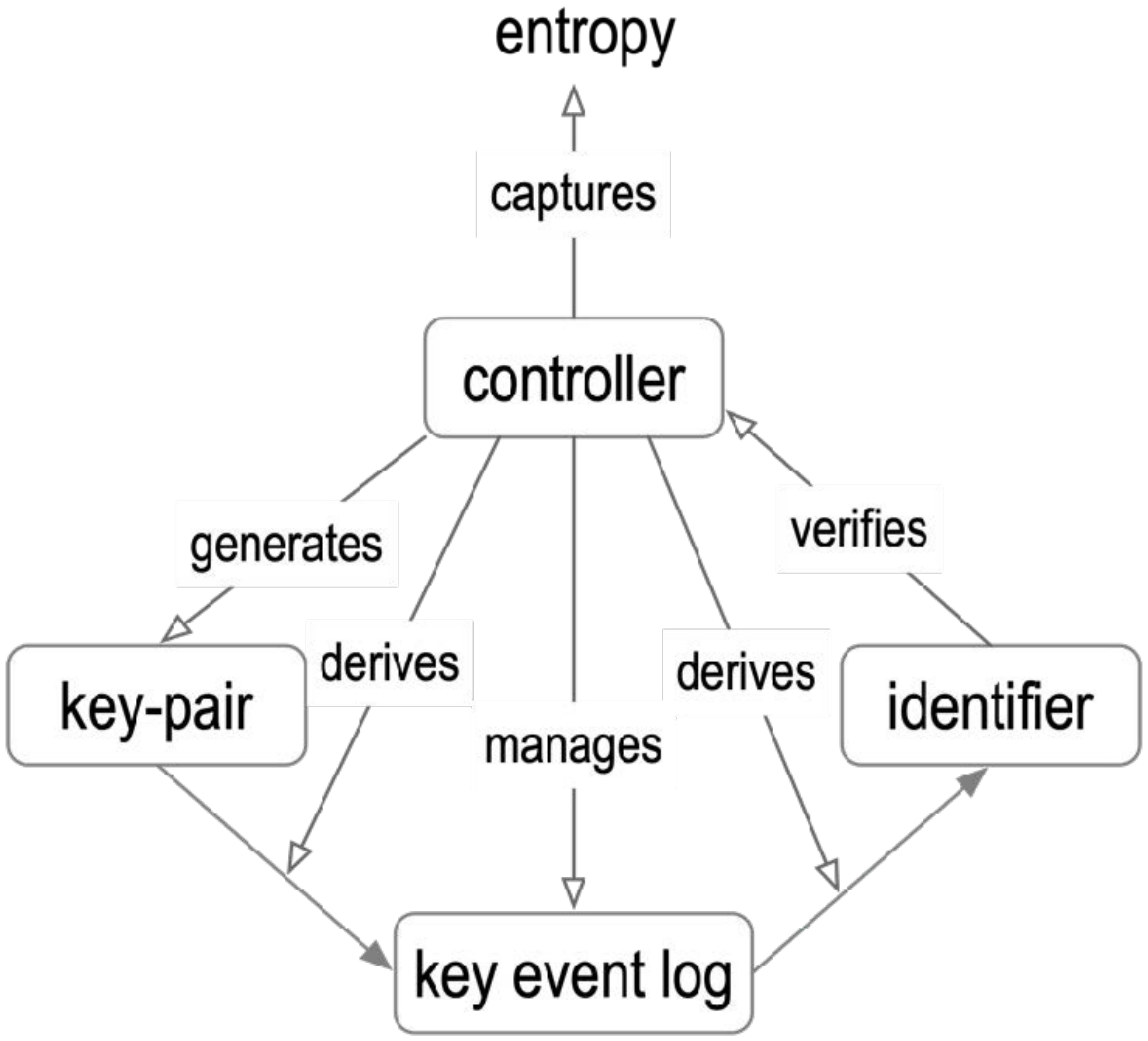
The overlay's security is contingent on the mapping's security.

Key State Proof is Recursive Application of Overlay

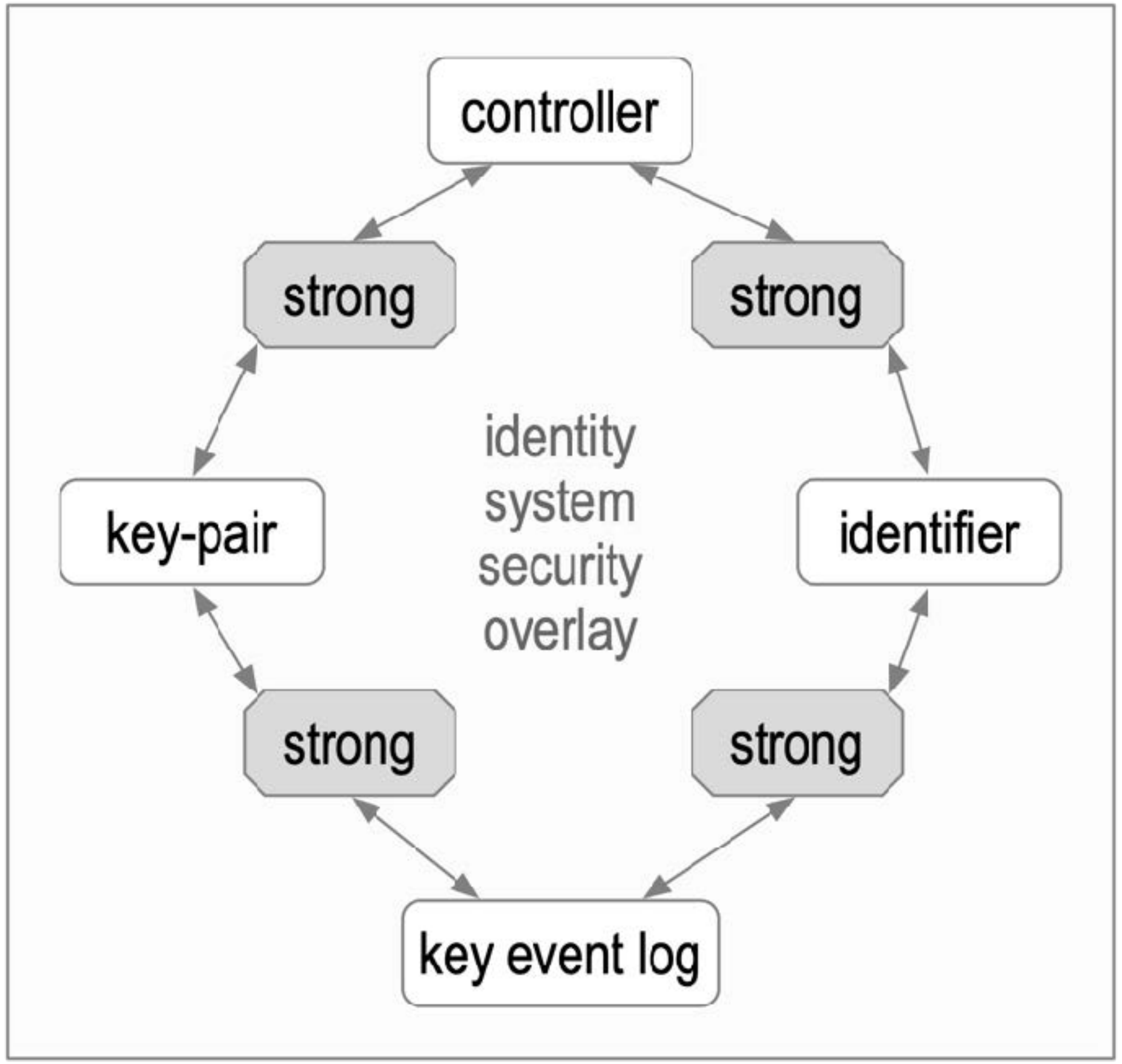
Persistent mapping via verifiable data structure of key state changes



Autonomic Identifiers (AIDs): (type of self-certifying identifier) Issuance and Binding



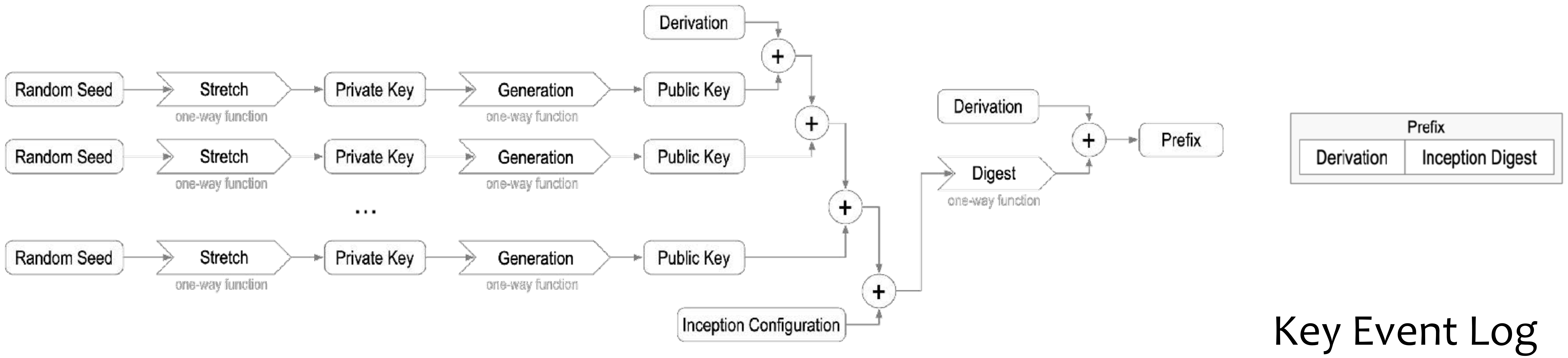
autonomic, self-managing



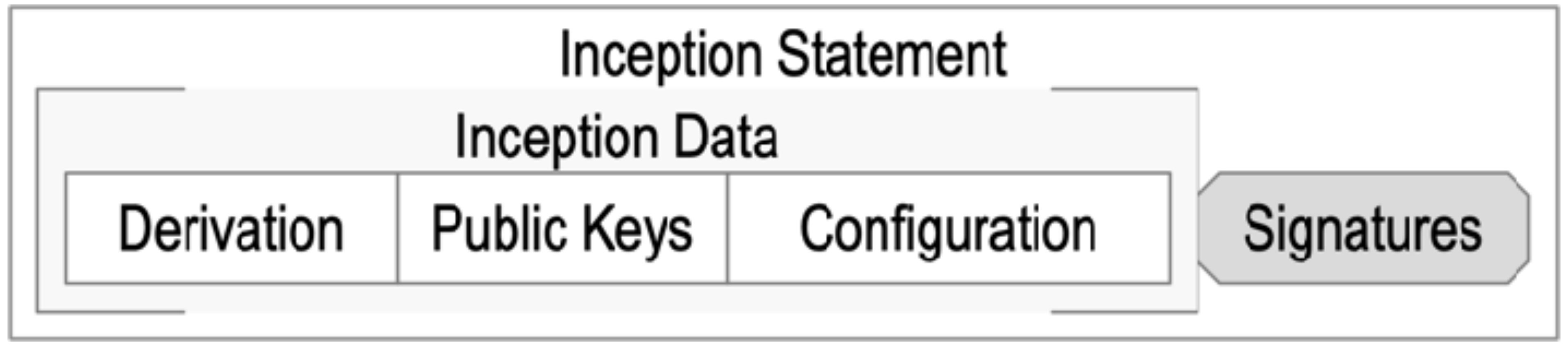
Autonomic Identifier Issuance Tetrad

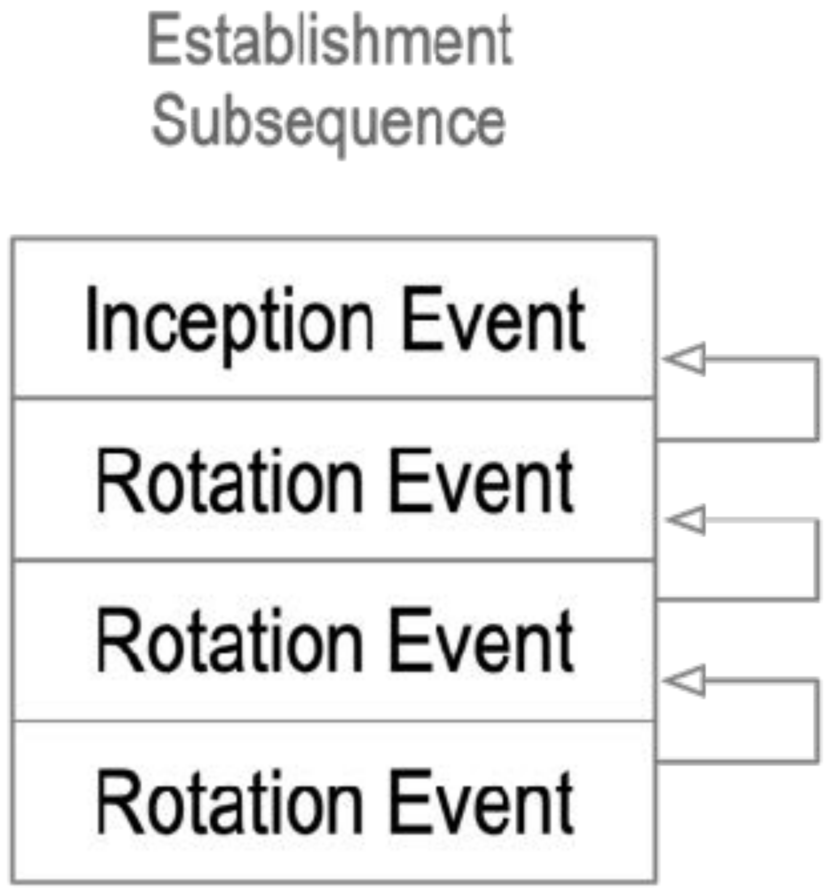
cryptographic root-of-trust with verifiable persistent control

Cryptographic Root-of-Trust:
Self-Certifying Identifier (SCID) + Key Event Log = Autonomic Identifier (AID)



Key Event Log



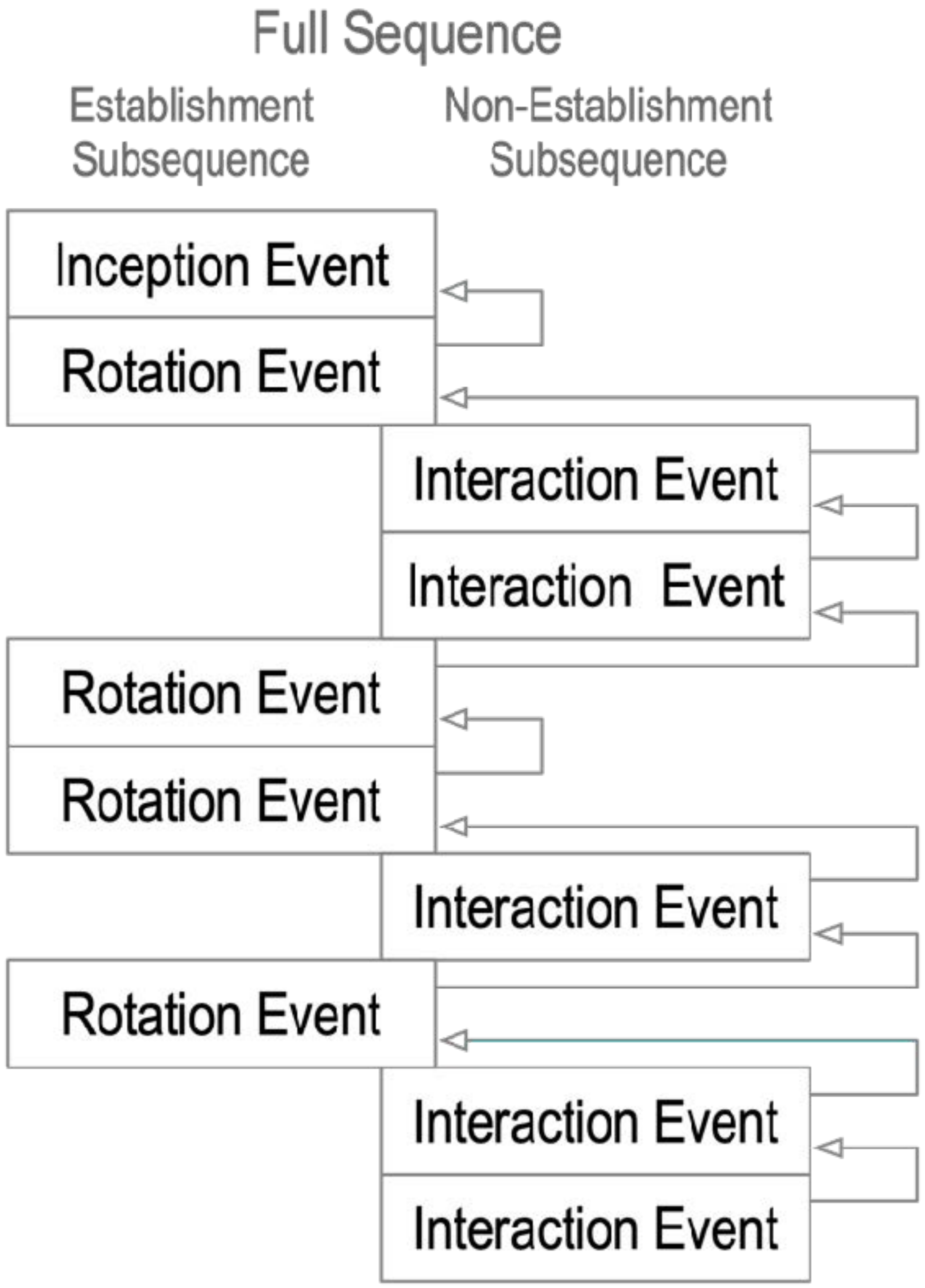


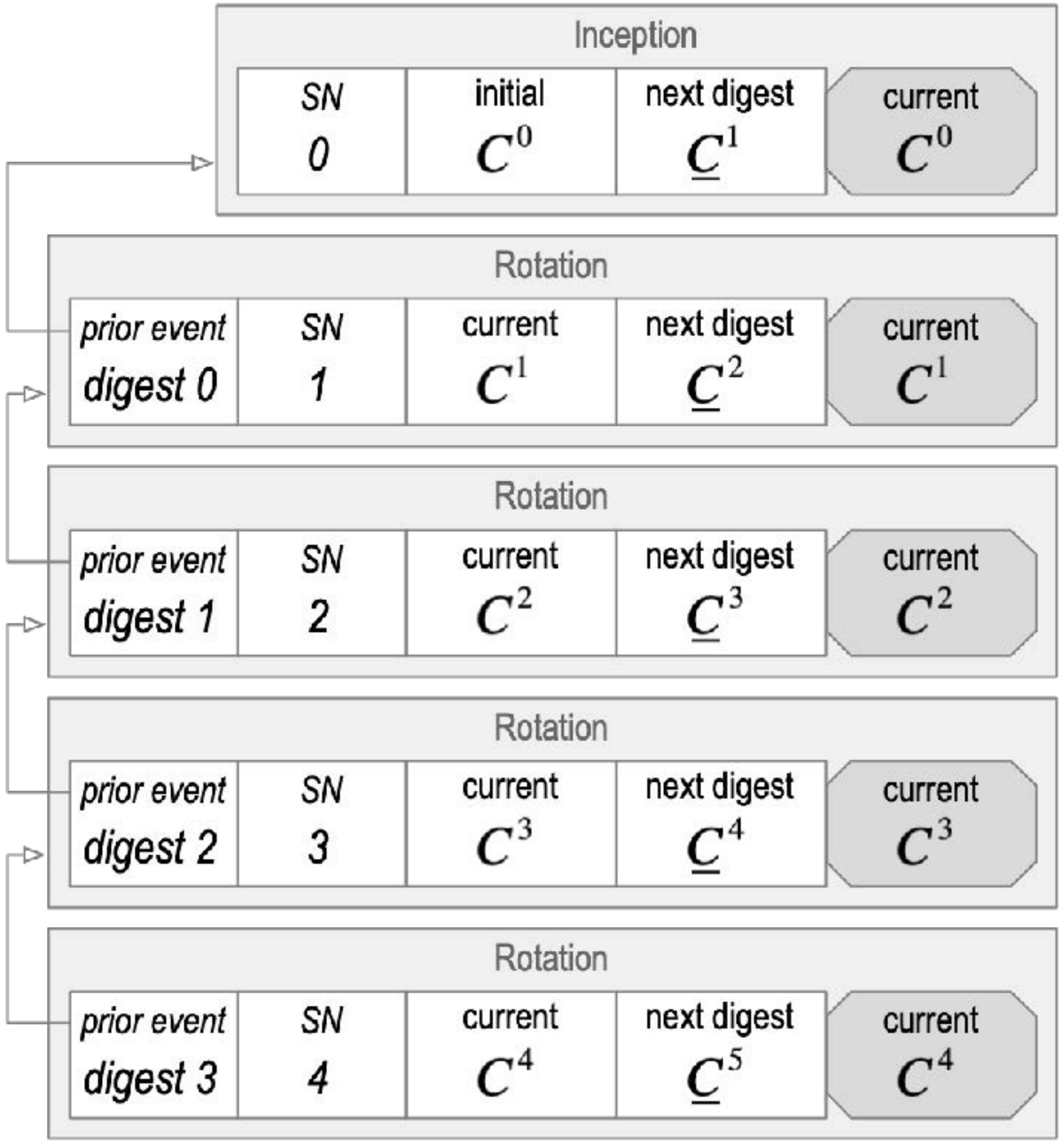
EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148

did:un:EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148/path/to/resource?name=secure#really

Solution: Key Pre-Rotation

duplicity evident
verifiable data
structure





Digest of *next* key(s) makes pre-rotation post-quantum secure

D.J. Bernstein: https://cr.yp.to/hash/collisioncost-20090517.pdf

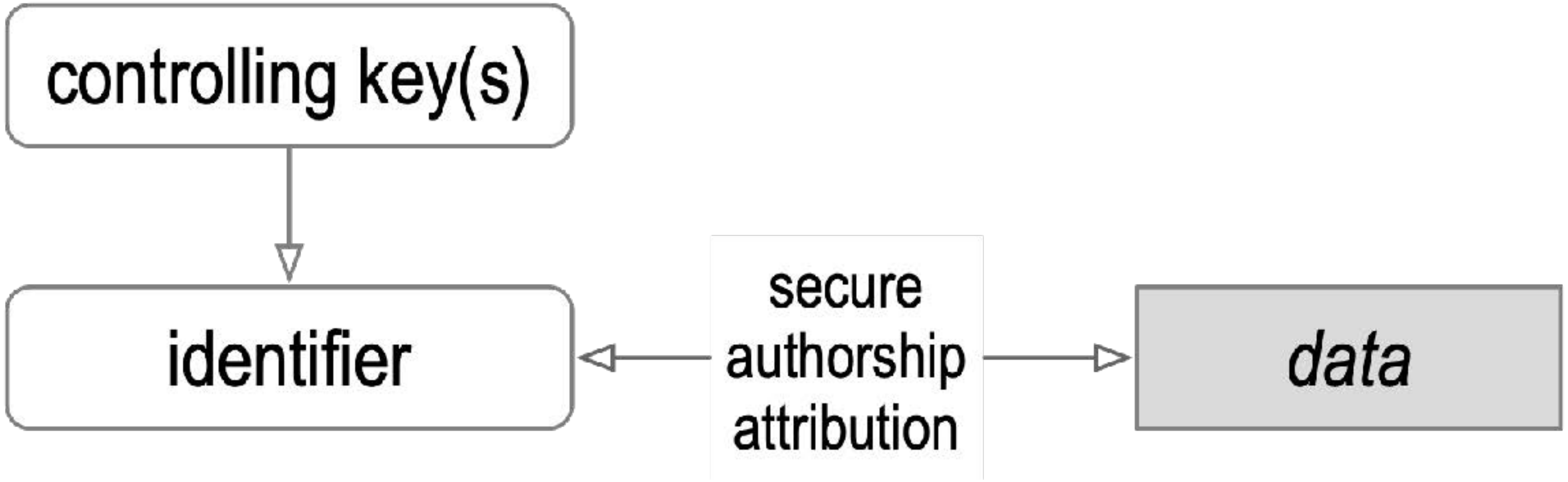
Universal Secure Attribution Problem

Establish authorship of data, documents, credentials, entitlements, …

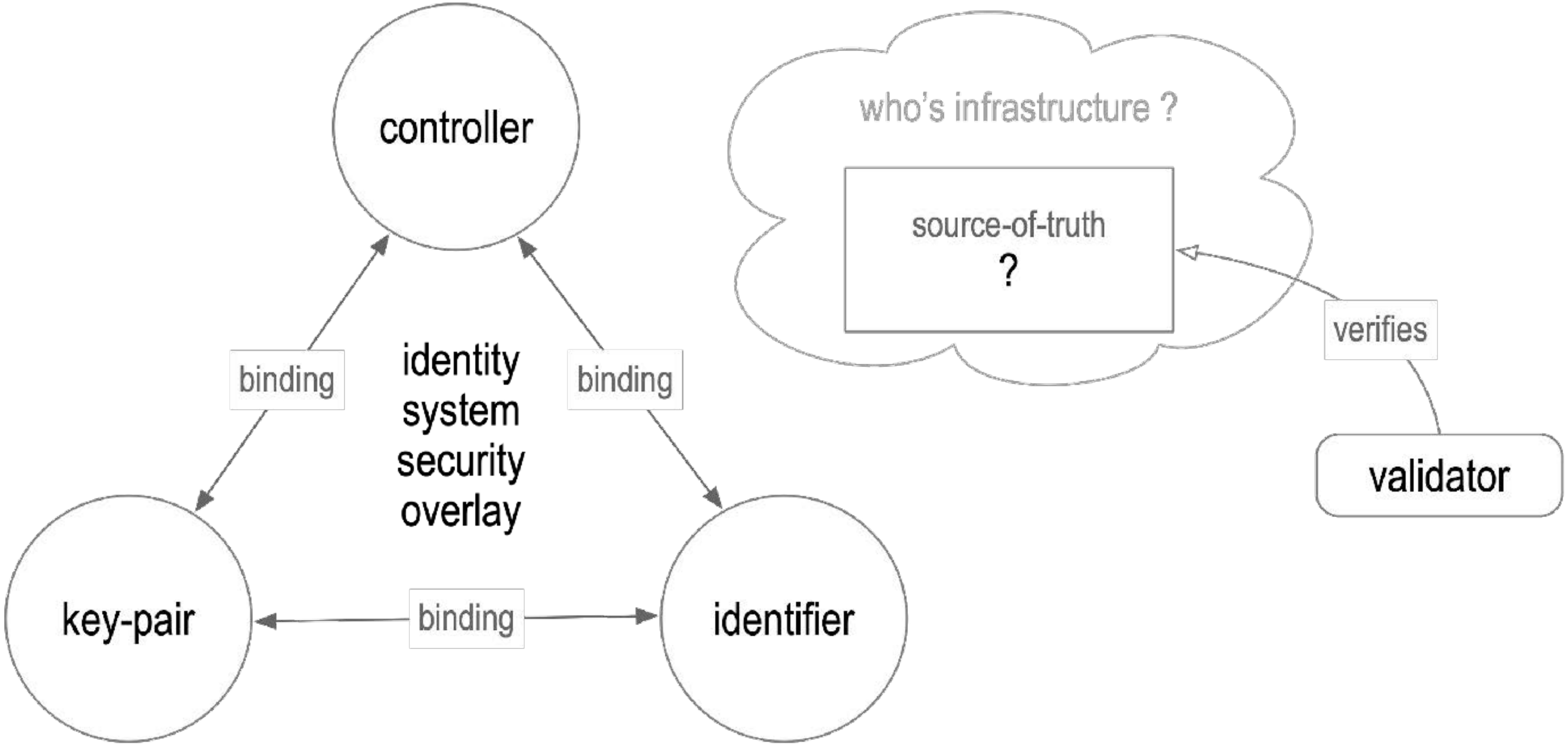
= Verifiable secure attribution of any communication to its source

= Authentic data provenance by anyone to anyone from anyone

Solve data provenance to solve security

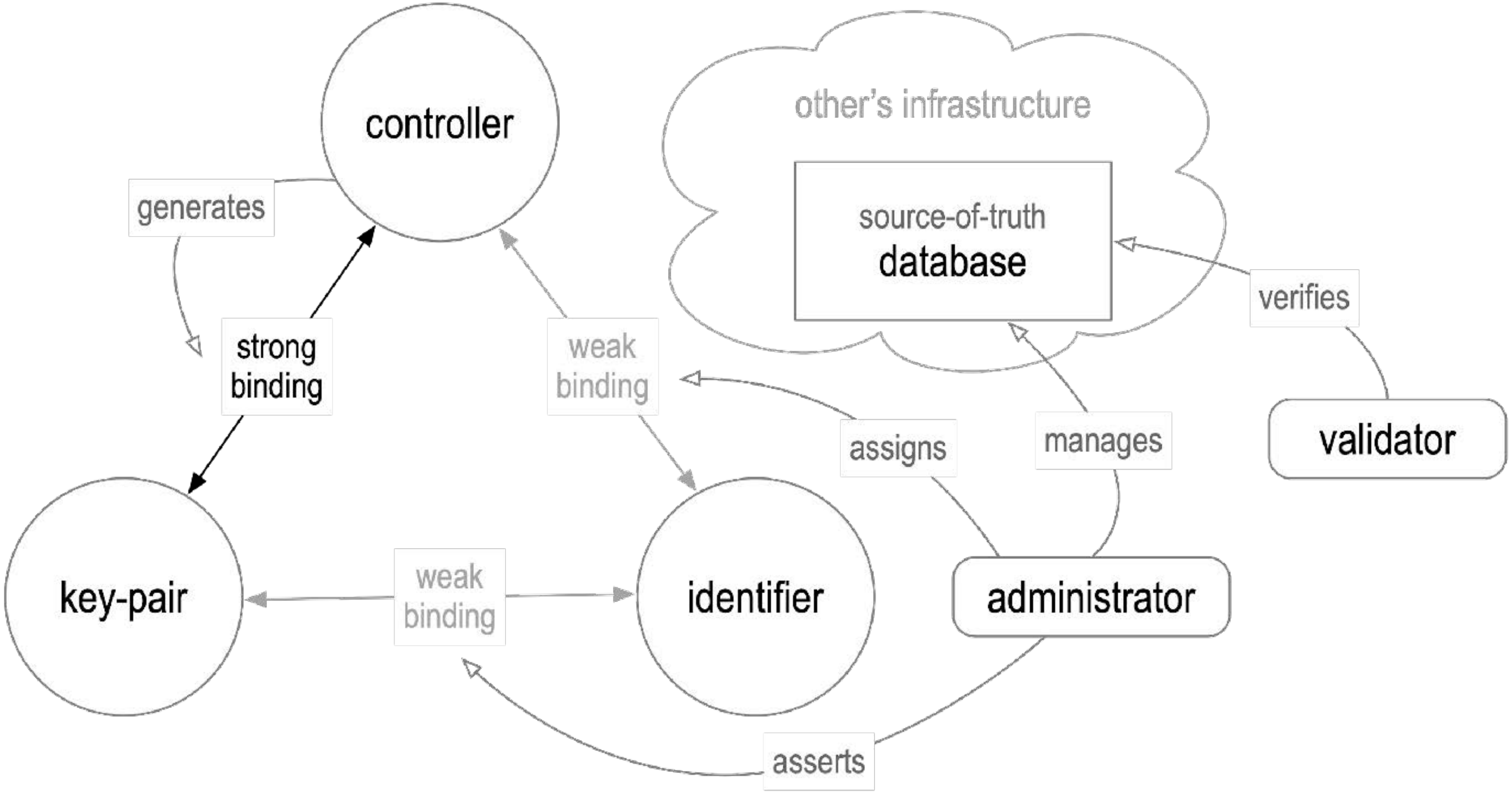


Trust Basis of a Trust Domain



Administrative Trust Basis

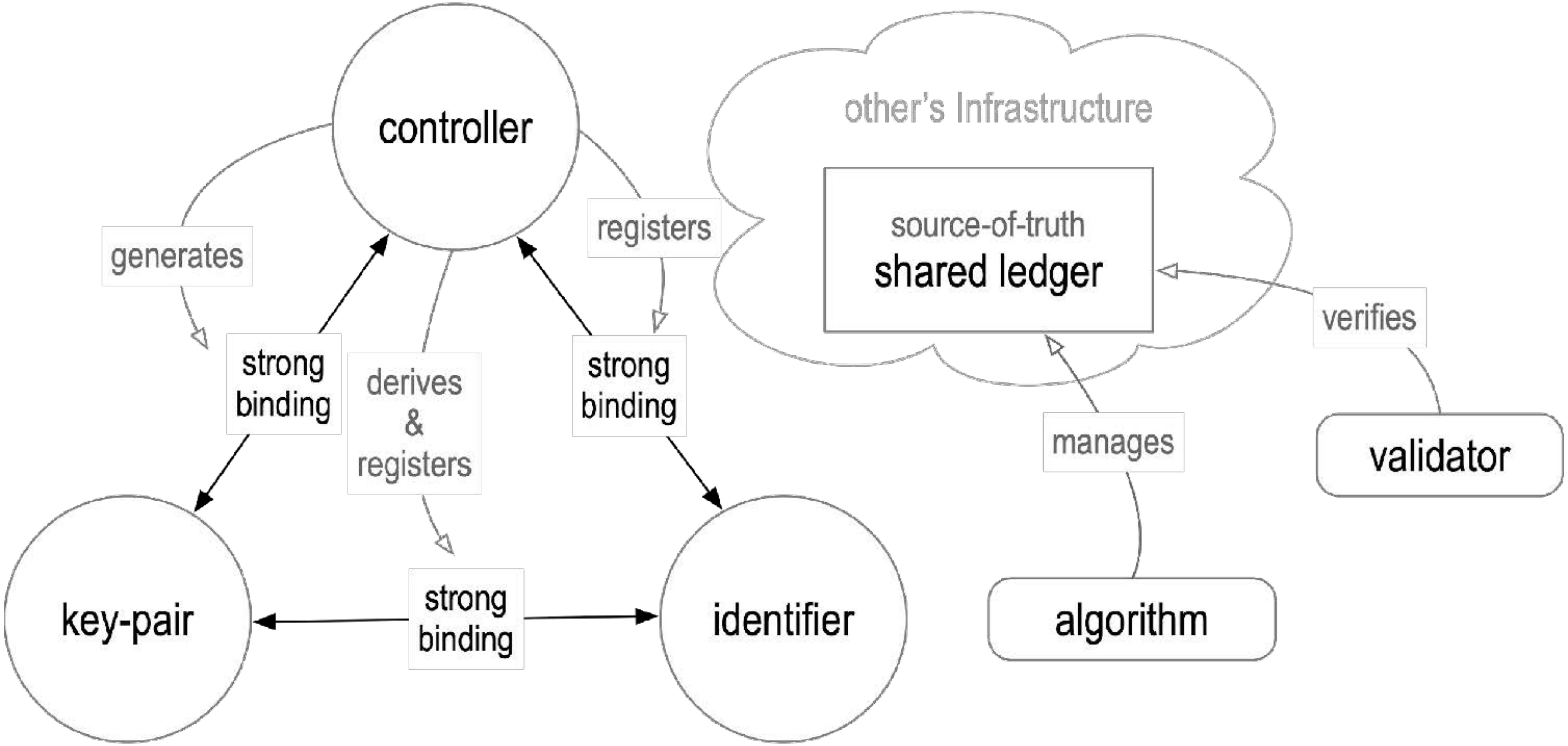
DNS/CA, OIDC IP



root-of-trust in non-verifiable operational infrastructure with opaque governance

Algorithmic Trust Basis

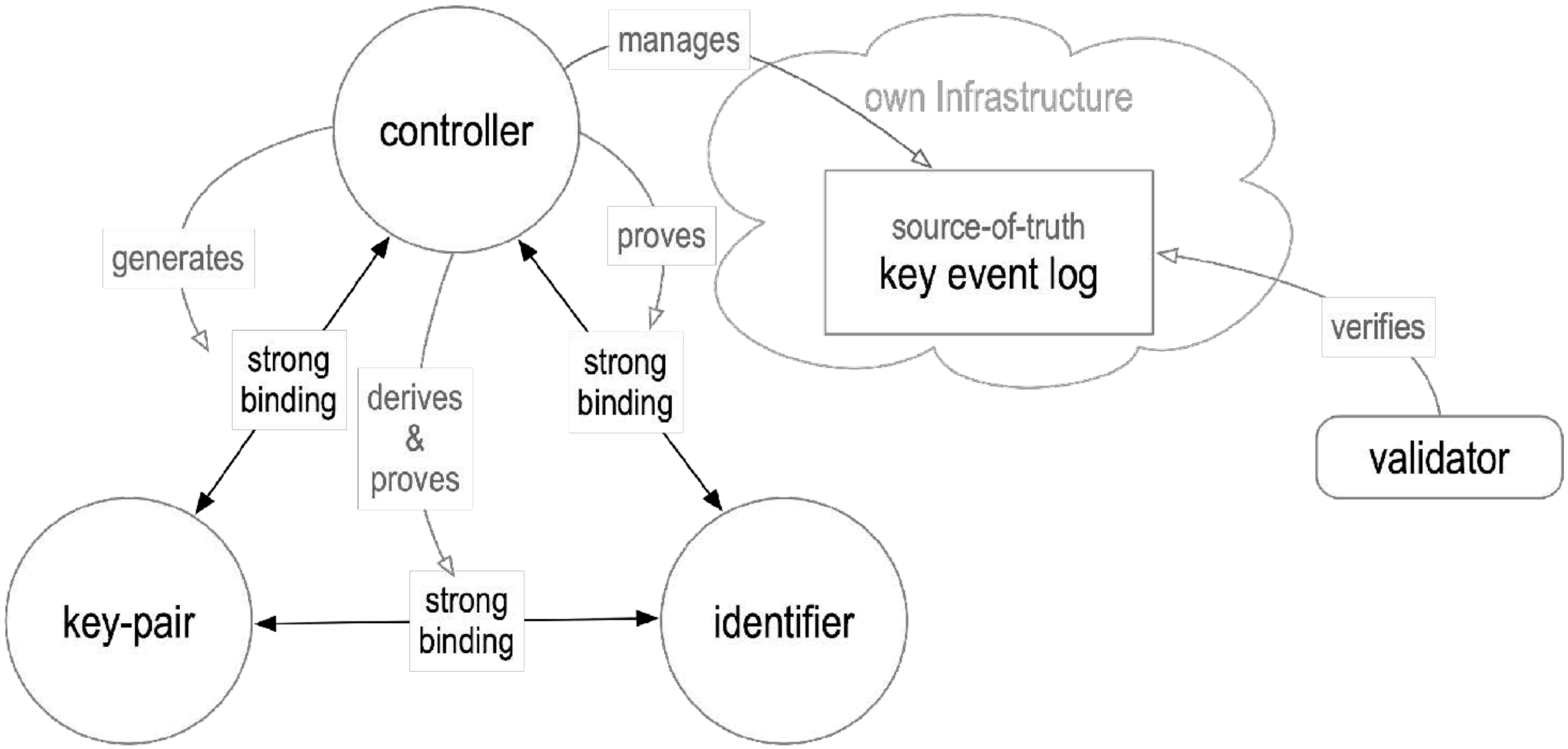
Shared distributed ledgers



root-of-trust in verifiable operational infrastructure with shared governance

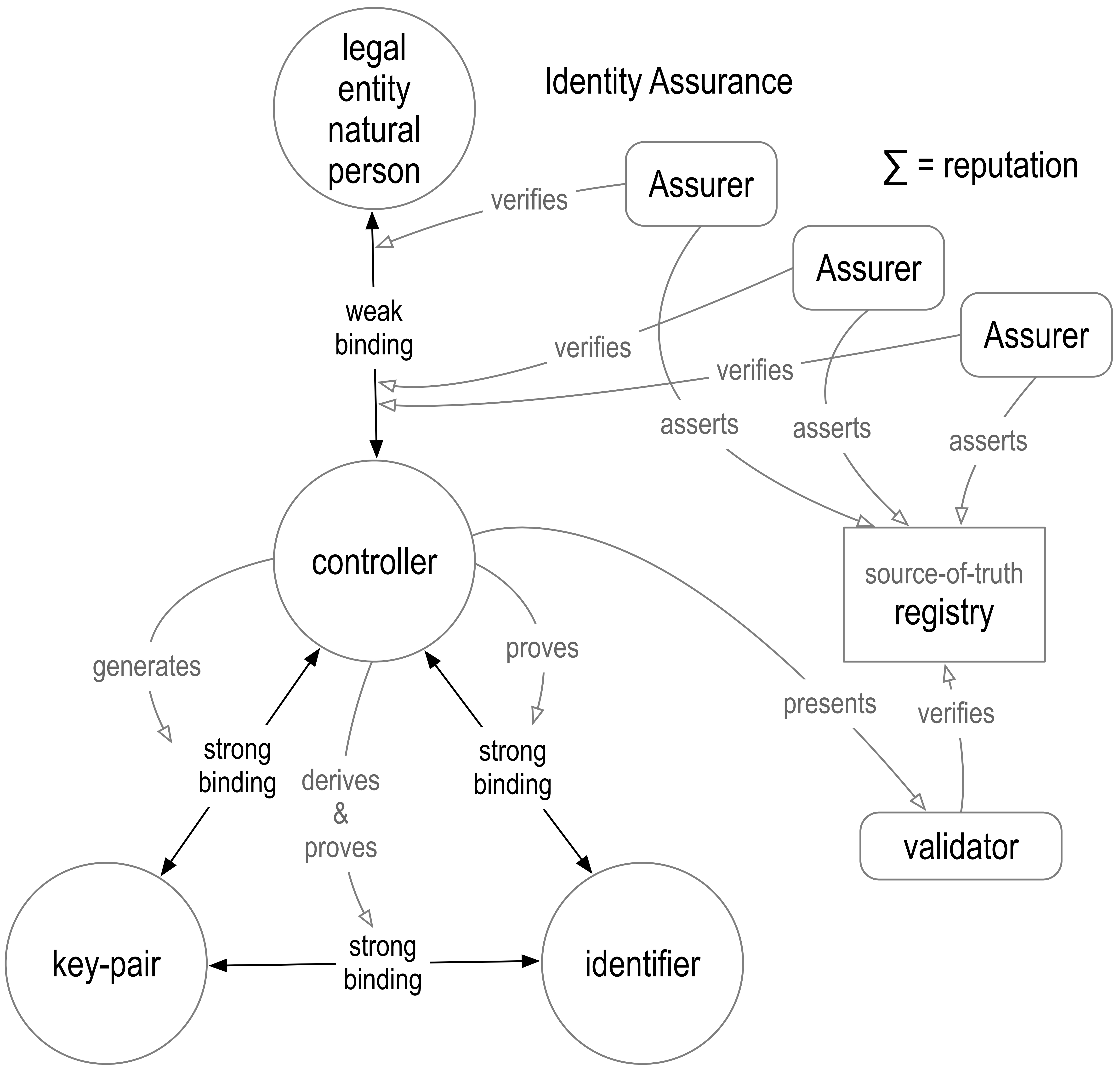
Autonomic Trust Basis

Cryptographic proofs via verifiable data structures



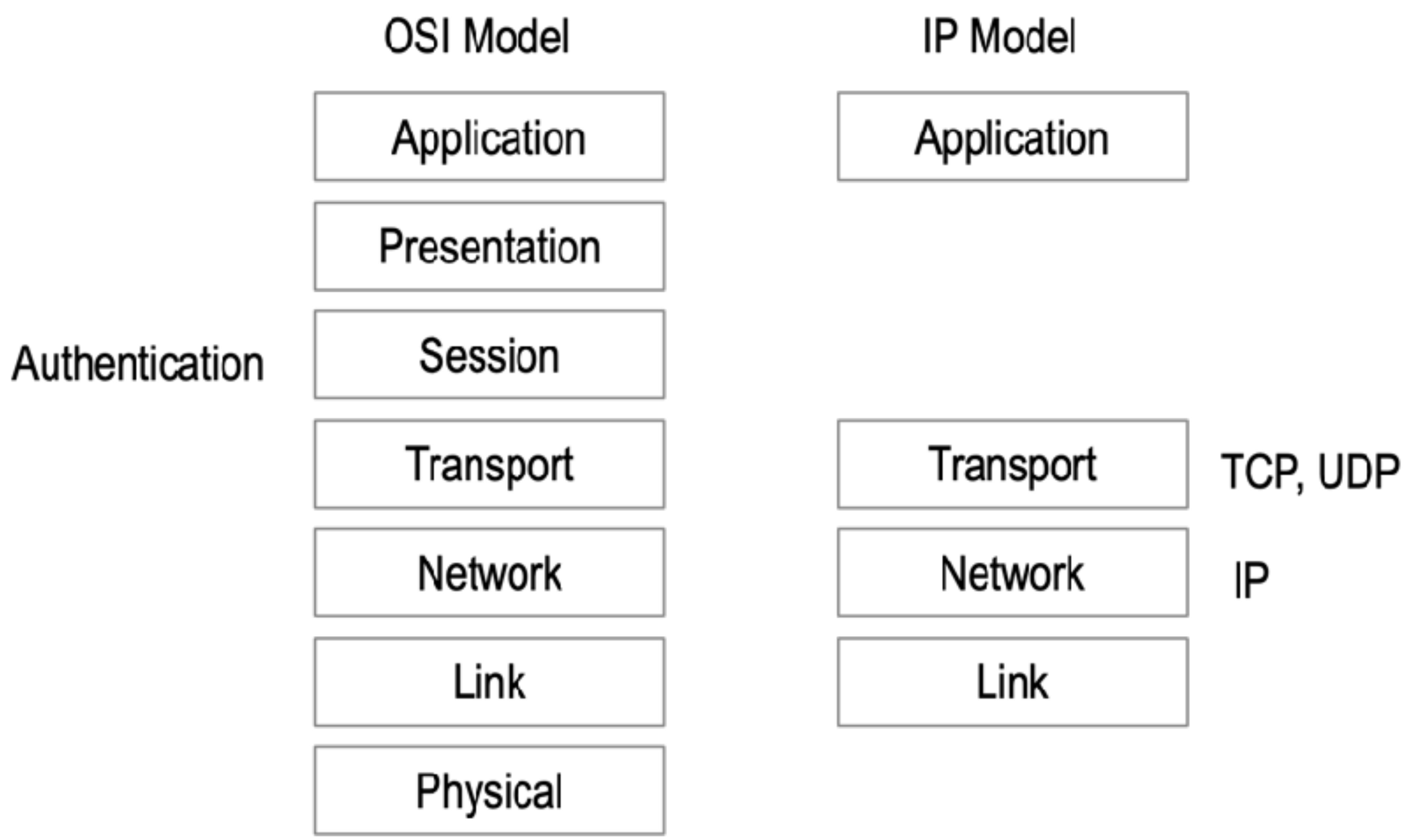
root-of-trust in verifiable cryptographic proofs of infrastructure with no shared governance

Identity Assurance and Reputation



Relatively weak binding reinforced by multiple bindings = reputation

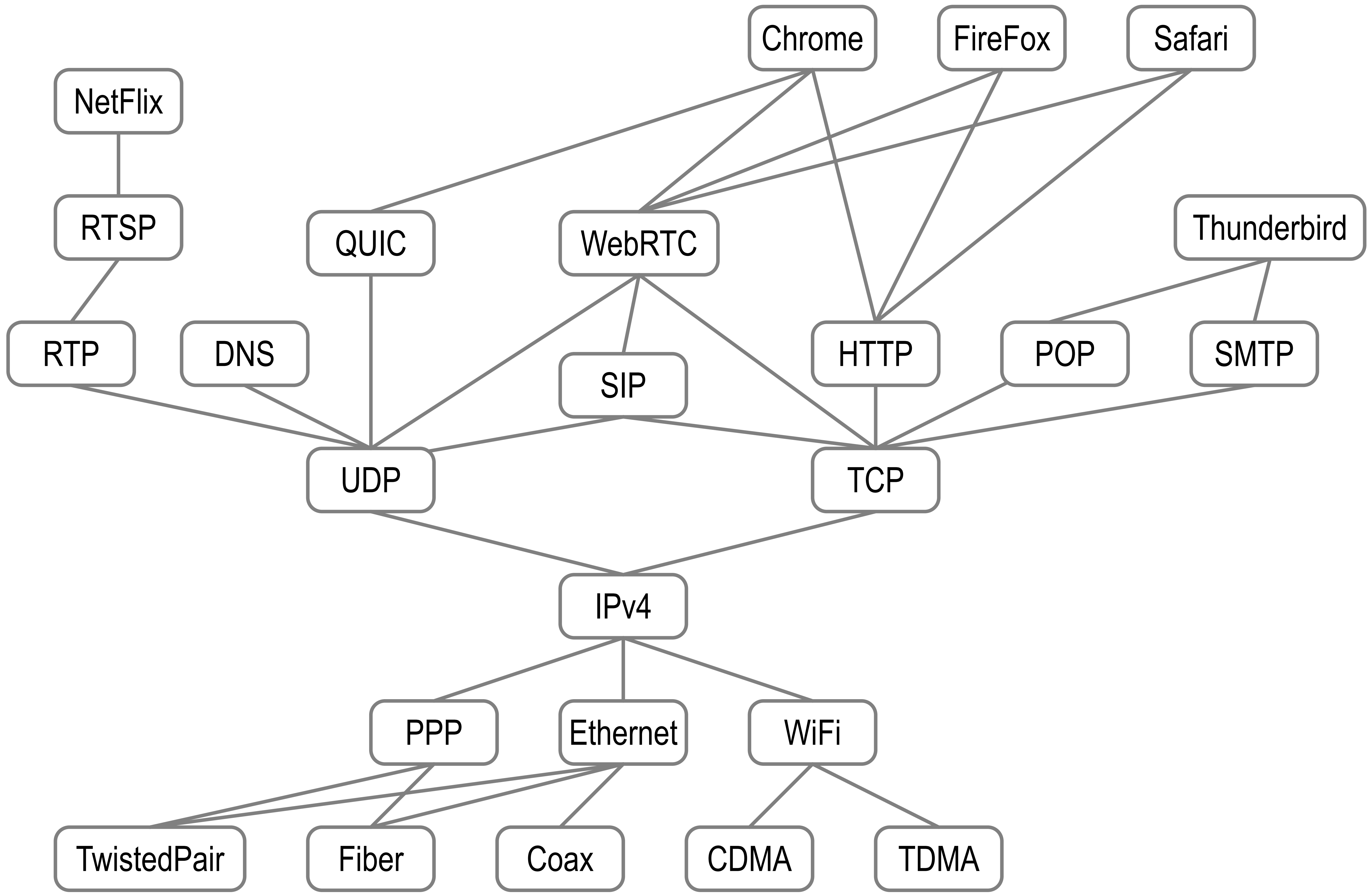
The Internet Protocol (IP) is *bro-ken* because it has no *security* (*trust*) layer.

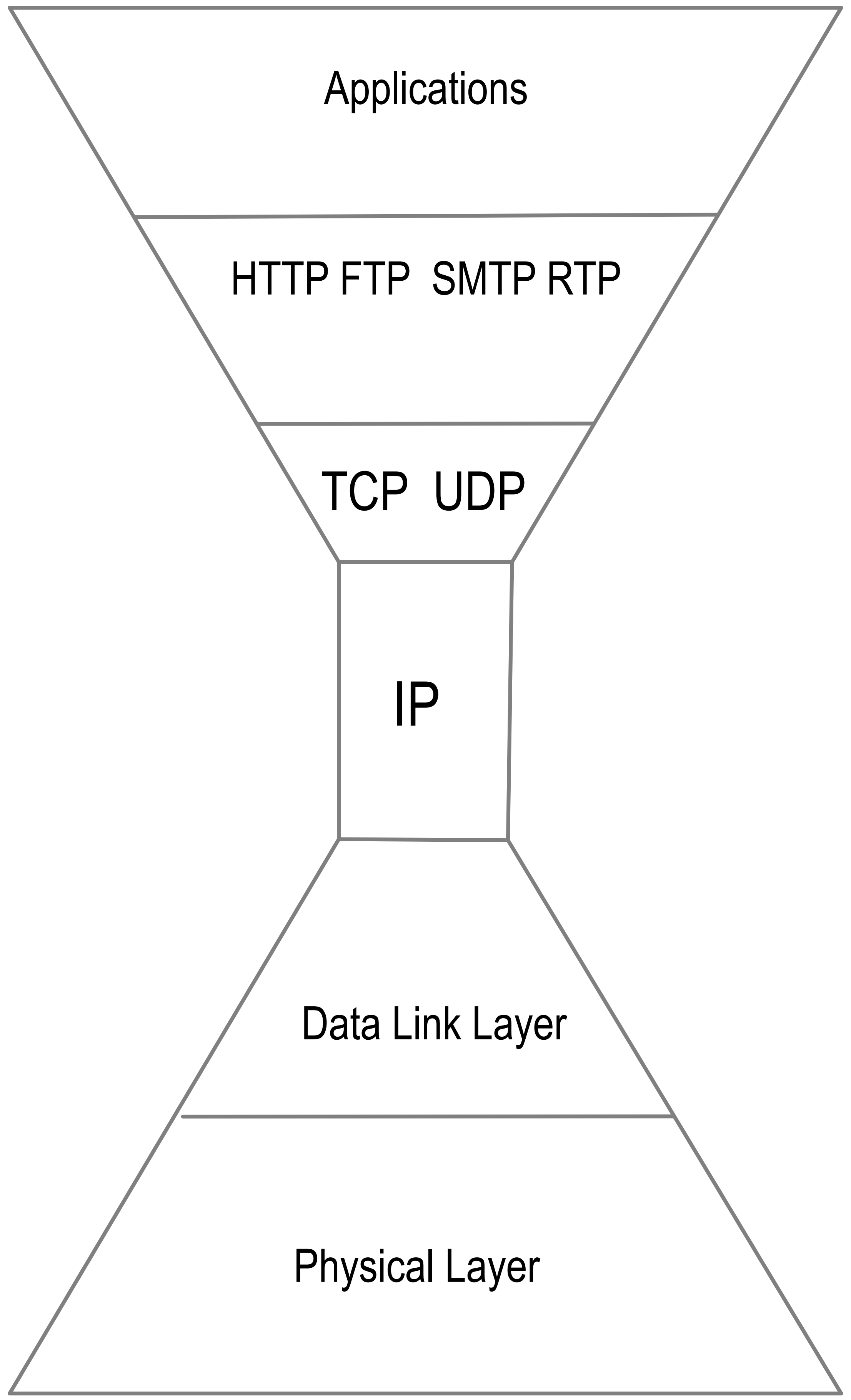


Instead …

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

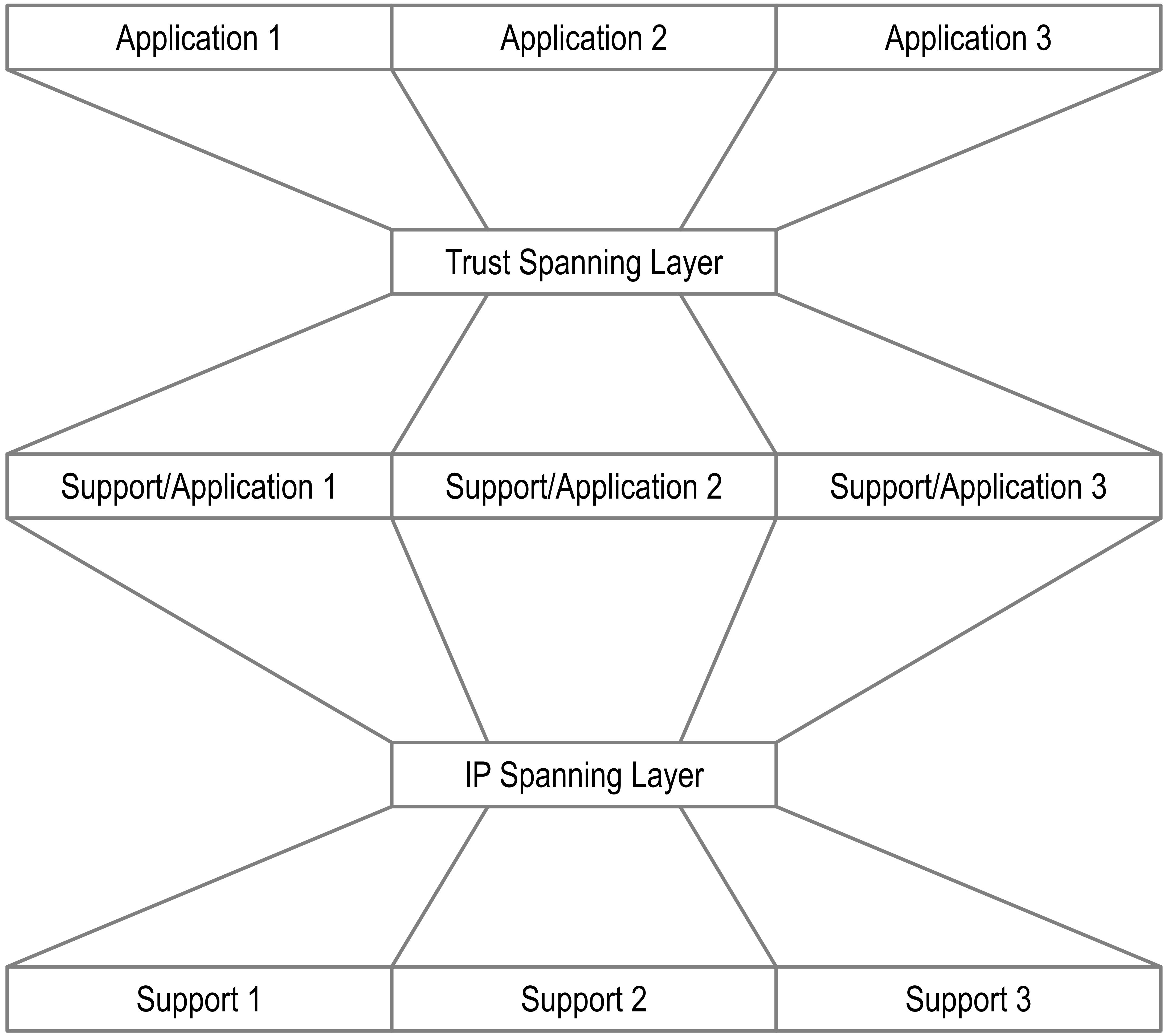
Spanning Layer

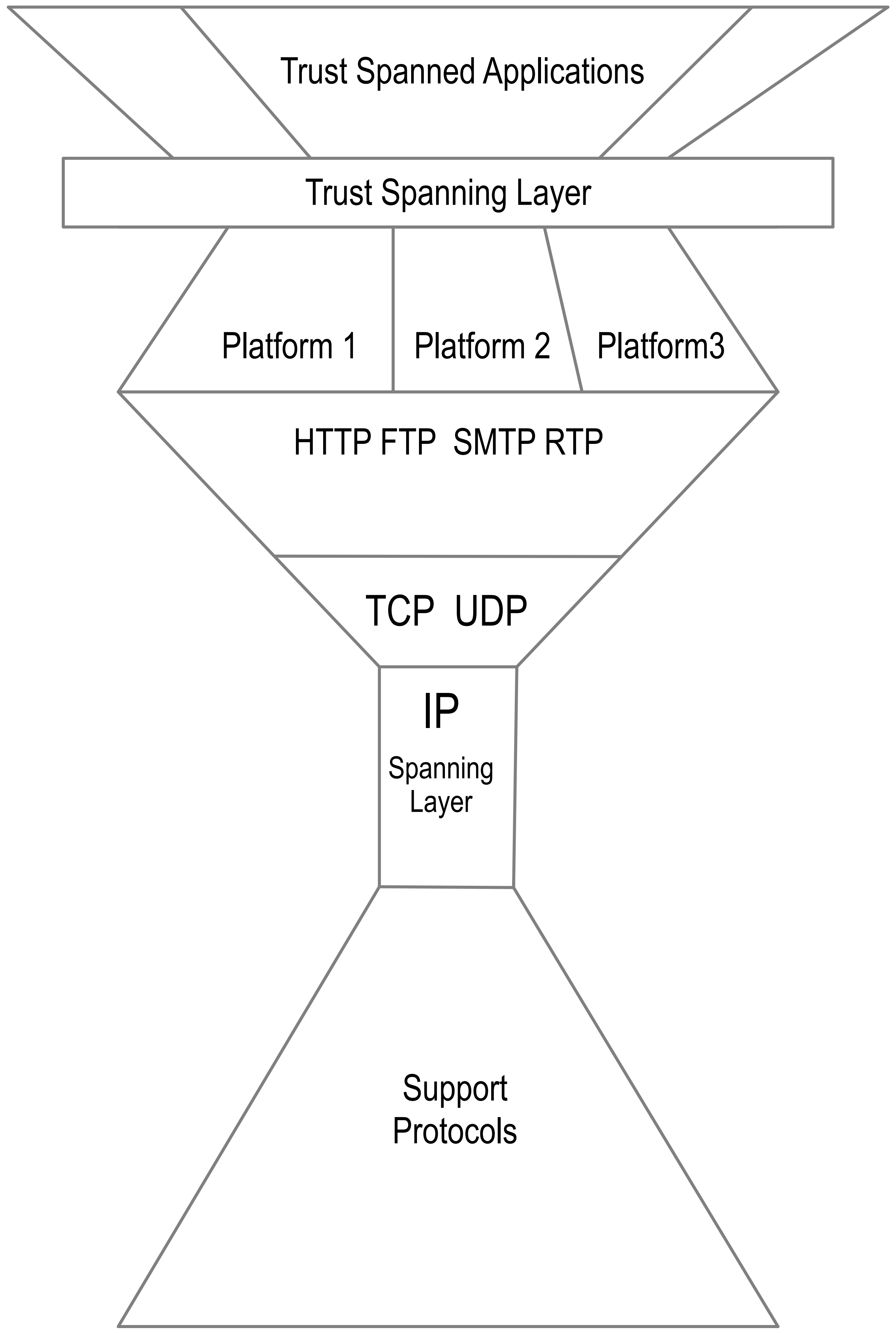




https://web.archive.org/web/20050415042854/http://www.csd.uch.gr/~hy490-05/lectures/Clark_interoperation.htm

Solution: Waist and Neck





Organizational Identity

Zero-trust architecture

Autonomic (cryptographic) decentralized root-of-trust (per organization)

Protocol not Platform

Delegable Authority

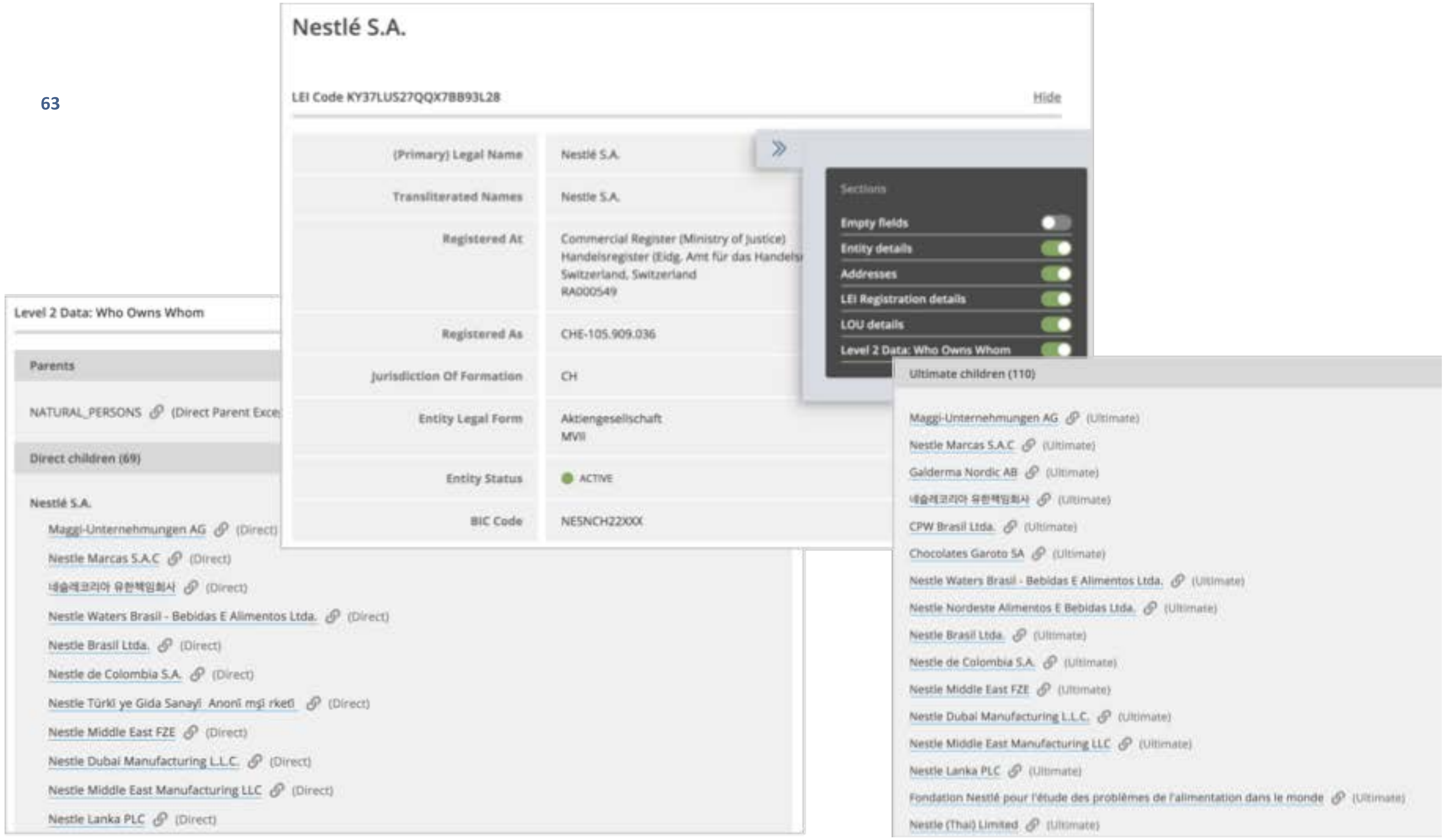
Multi-sig DPKI

Authentic Chained Data Containers

The Legal Entity Identifier – the LEI

- The LEI is a life-long code **owned** by the respective legal entity.
- It points to the associated reference data.
- The LEI is an ISO standard ISO 17442

63



The LEI as a Verifiable Credential – the vLEI Trust Chain

- Every verifiable LEI (vLEI) is created by an **issuer**
- The issuer **cryptographically** signs the credential with its private key
- An issuer is the organization or entity that asserts information about a **subject** to which a credential is issued
- The vLEI Issuer is an organization **qualified** by GLEIF as part of a trusted network of partners
- GLEIF issues vLEIs to Qualified vLEI Issuers as attestation of trust.
- GLEIF is the Root of Trust

GLEIF
- **Qualified vLEI Issuers**
  - **Legal Entities**
    - **Persons Representing Legal Entities**

PKI Then and Now

Who uses a password manager?

Who uses an authenticator app?

Who uses password-less login?

Then: Managing private keys impossible for users, federated identity.

Now: Mobile Devices with MFA & secure boot, password-less login.

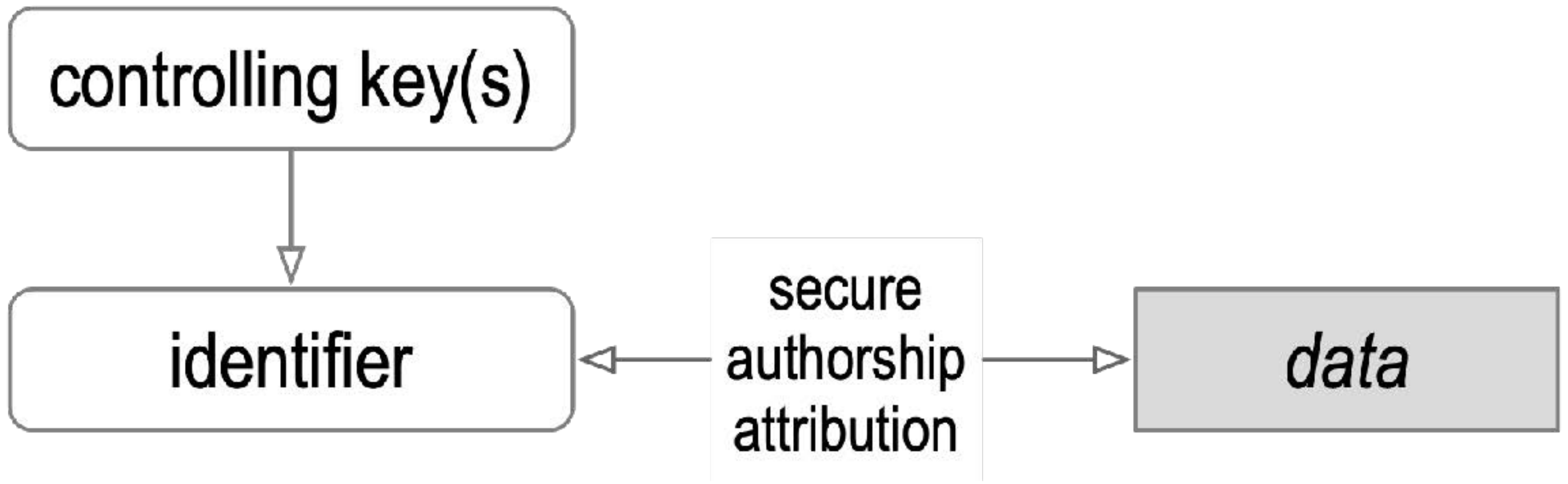
Then: Weak Crypto

Now: Strong crypto: ECC signing & ECC asymmetric encryption.

Then: Perimeter security, no persistence of control over identifiers.

Now: decentralized zero-trust architecture for identity (KERI).

Flaw of PKI (DNS/CA)



Conventional PKI uses signed assertions (x509 certs) made by trusted entities to bind key state (public, private) key pairs to identifiers.

Use of private keys for either signing or decryption exposes them to side-channel attack.

Over-time, exposure makes private keys weak.

Thus, from time-to-time one must therefore revoke and replace, i.e. rotate the controlling private keys for a given identifier

Conventional PKI must re-establish the root-of-trust with each rotation thereby making it vulnerable to attack

This breaks the chain-of-trust-of-control over the identifier

What is KERI? (Key Event Receipt Infrastructure)
Decentralized Key Management Infrastructure (DKMI)
Decentralized Public Key Infrastructure (DPKI)

KERI fixes the security flaw (authenticity) in PKI (Public Key Infrastructure):

That flaw is key rotation.

In conventional PKI there is no cryptographic binding between one set of keys and the next.

KERI solves the key rotation problem for control over an identifier via pre-rotation which binds the next key-state to the prior key-state.

With KERI, key state is cryptographically verifiably bound to a class of self-certifying identifiers that use portable verifiable data structures called *key event logs* (KELs) to provide duplicity evident proof of the controlling key state.

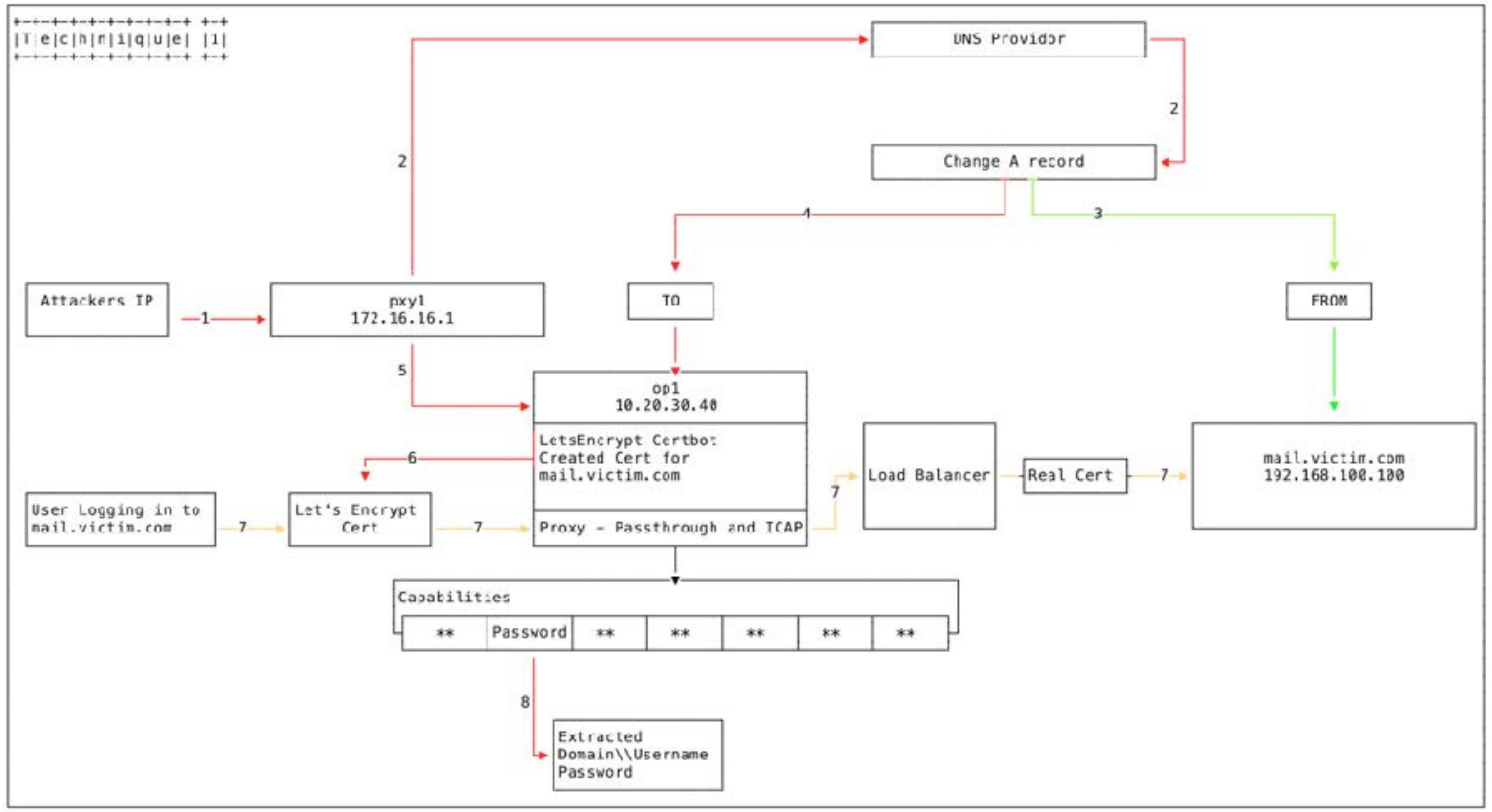
With KERI every statement associated with a KERI identifier may be non-repudiably and securely attributed to the controller of the identifier via a signature made with keys given by cryptographically verifiable key state.

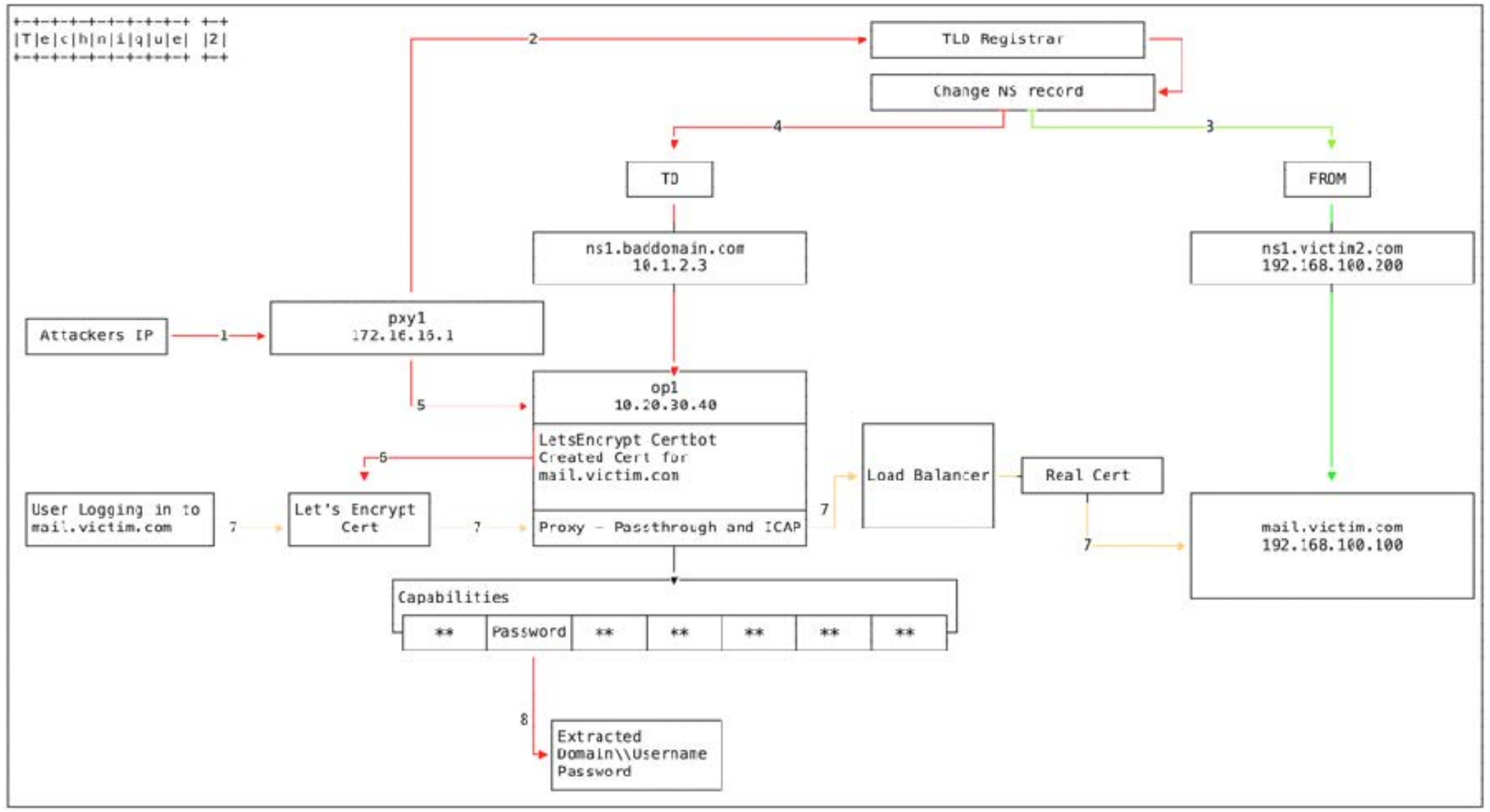
KERI solves the *secure attribution* problem with zero trust.

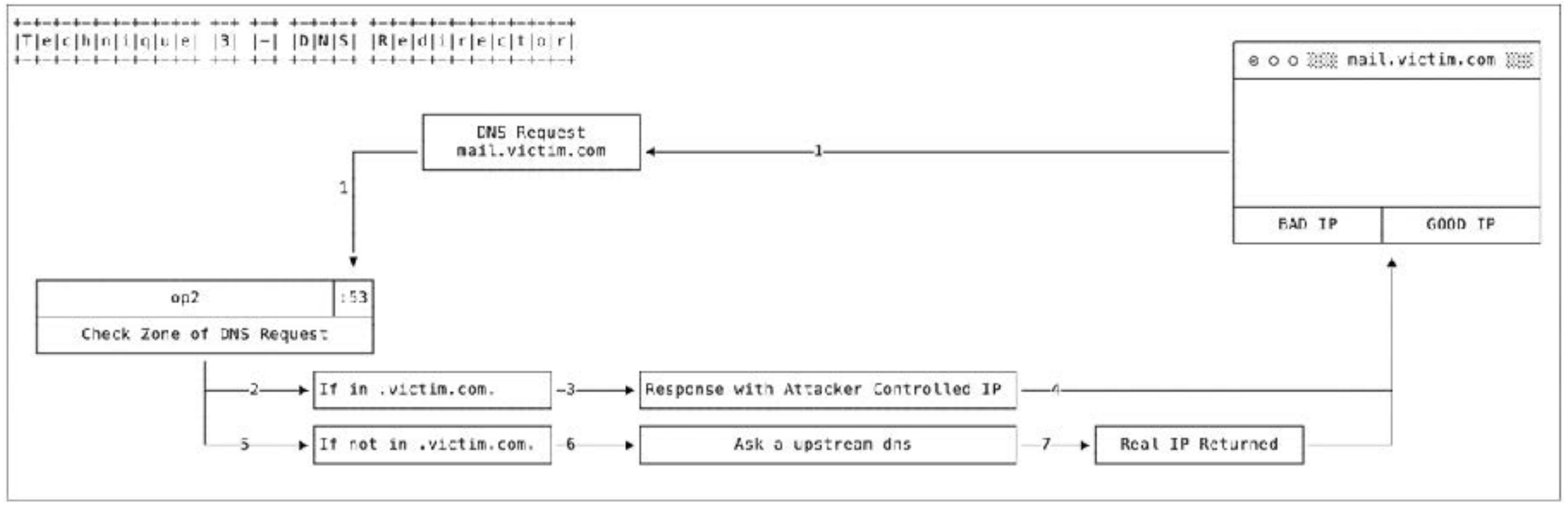
DNS Hijacking

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/



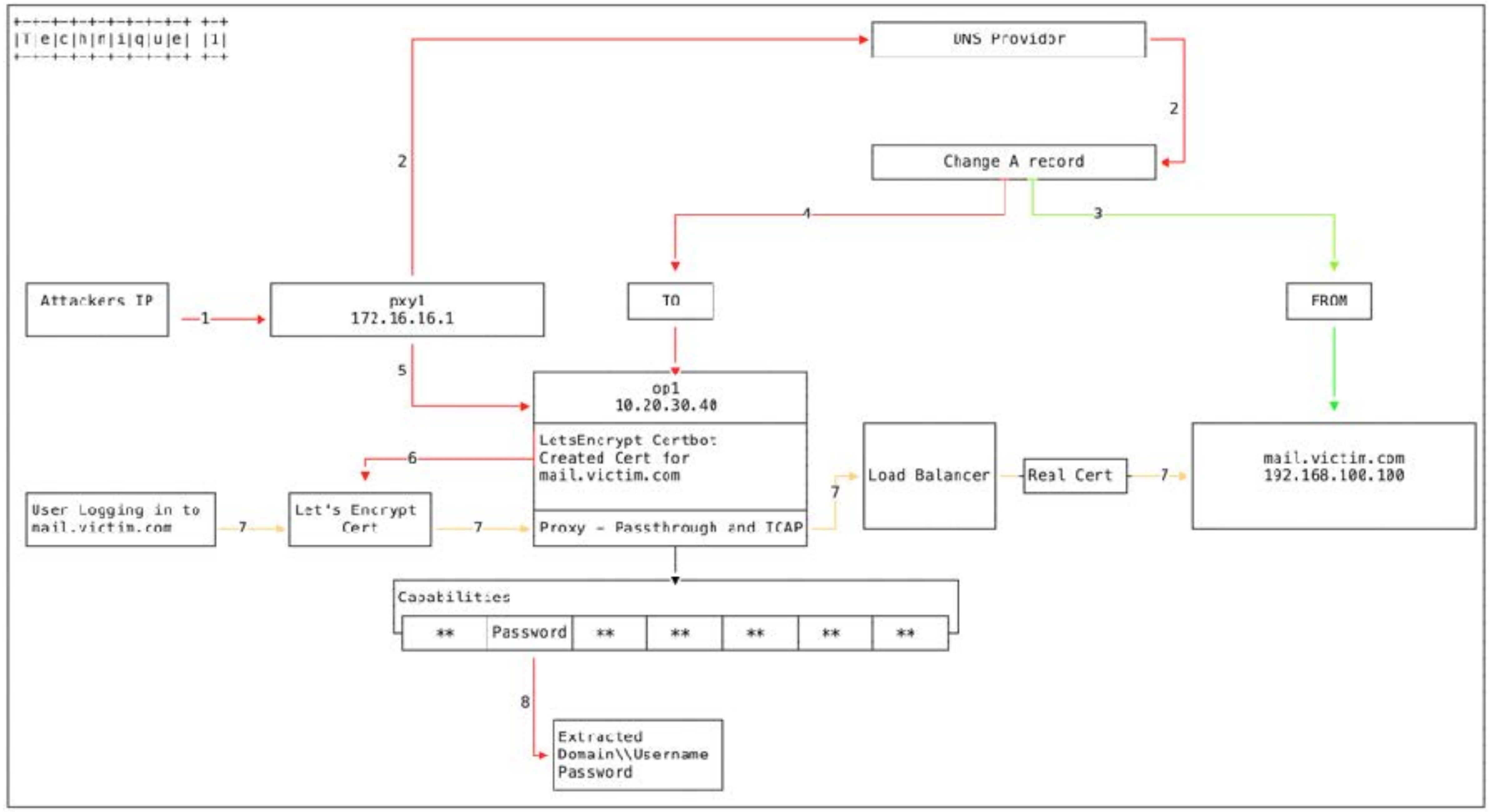


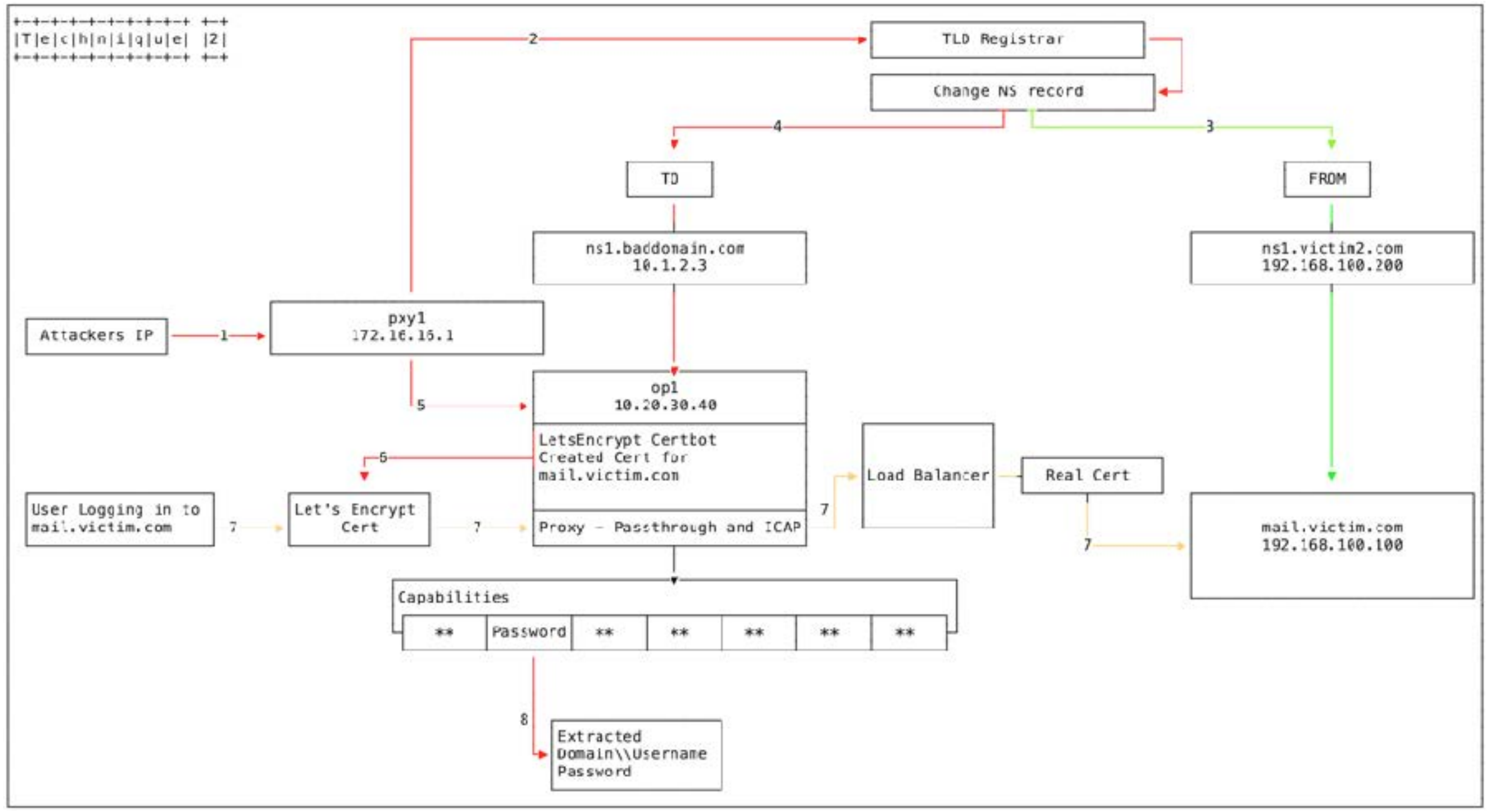


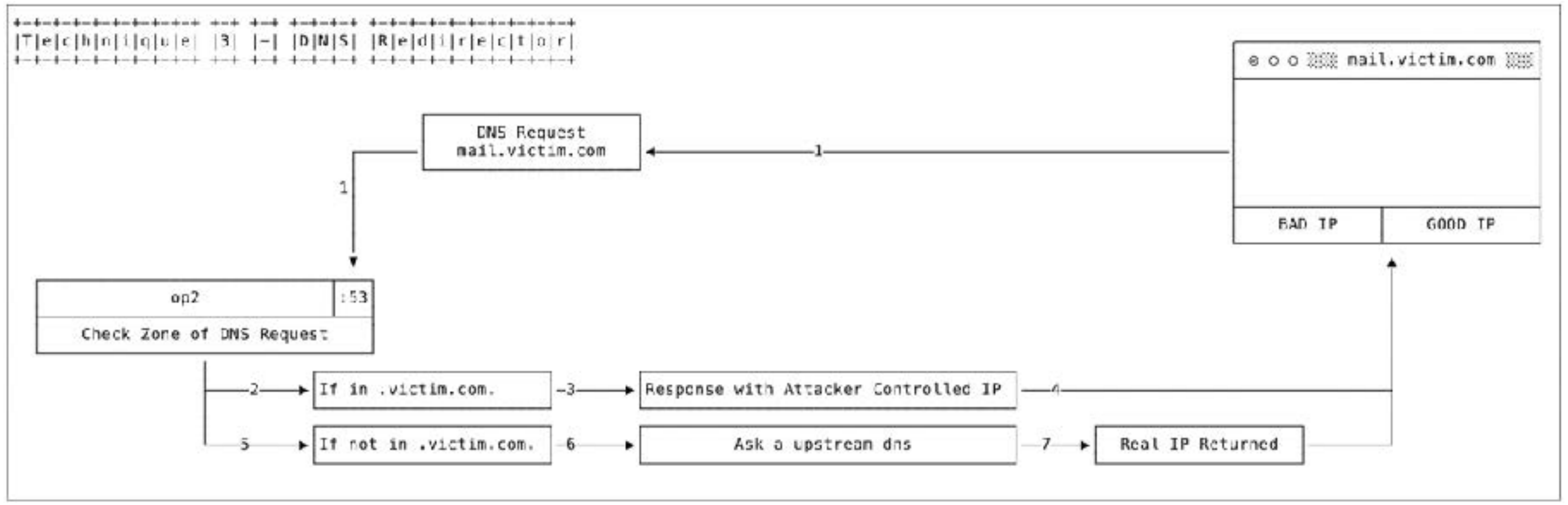
DNS Hijacking

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/





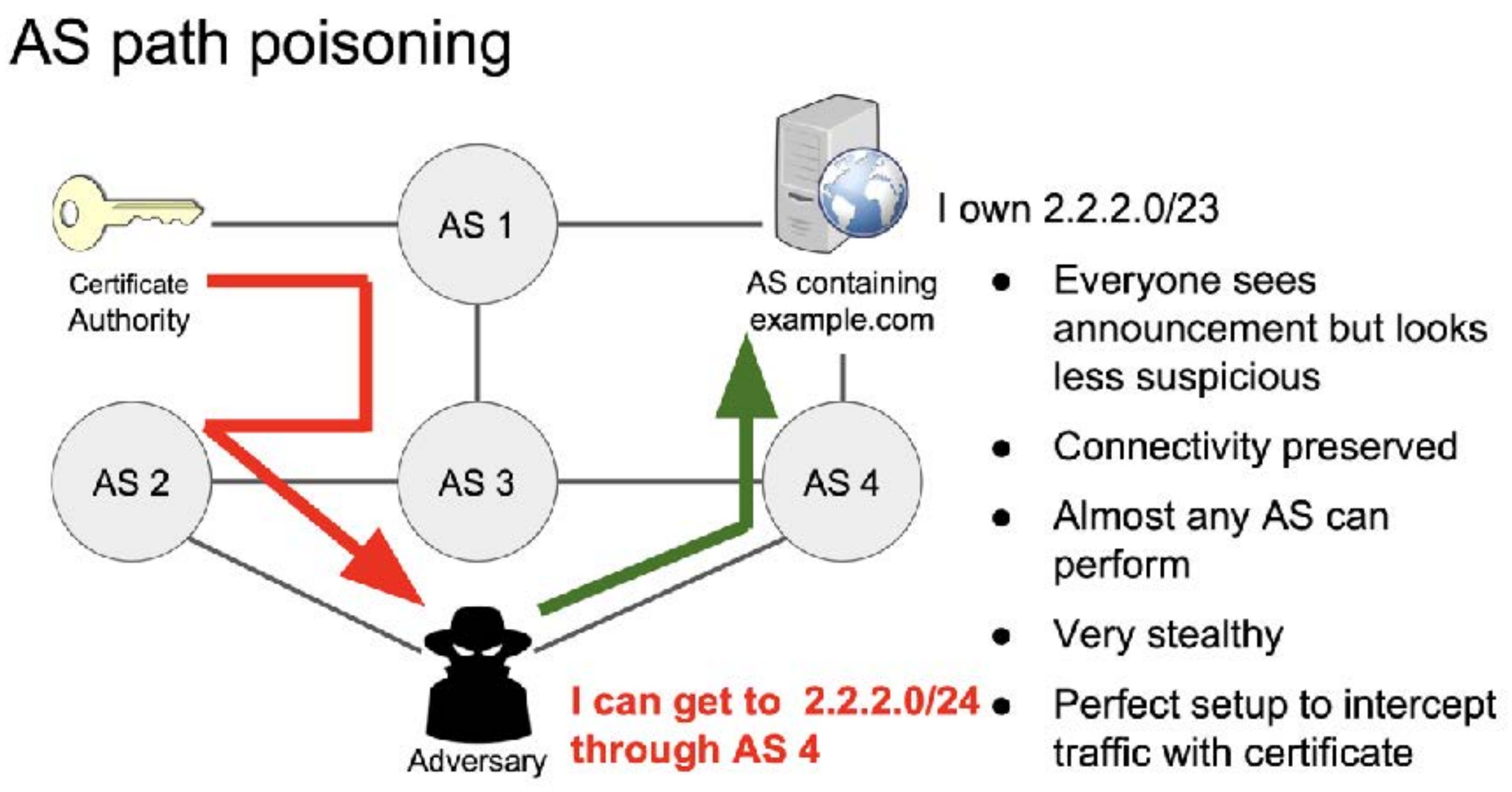


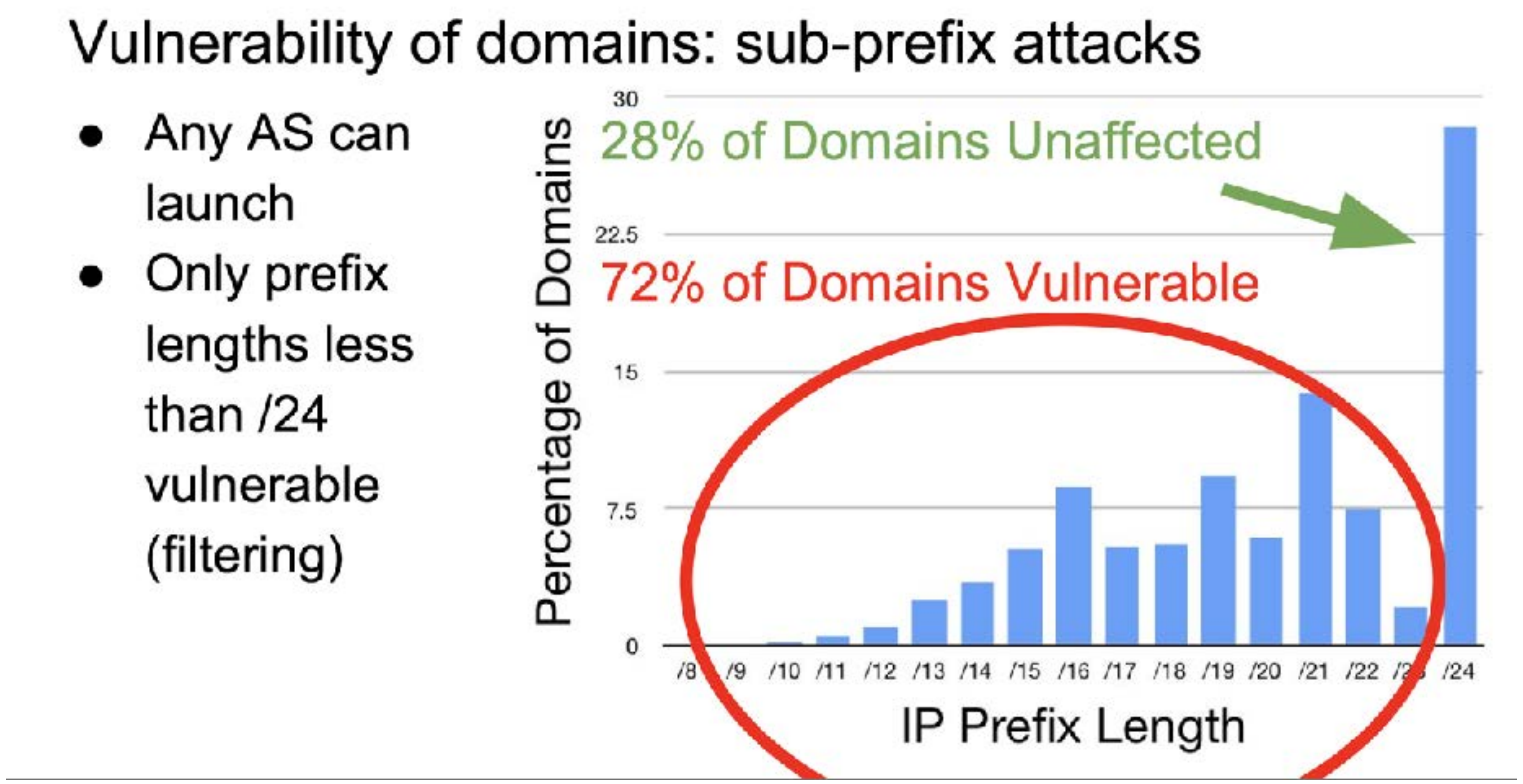
BGP Hijacking: AS Path Poisoning

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

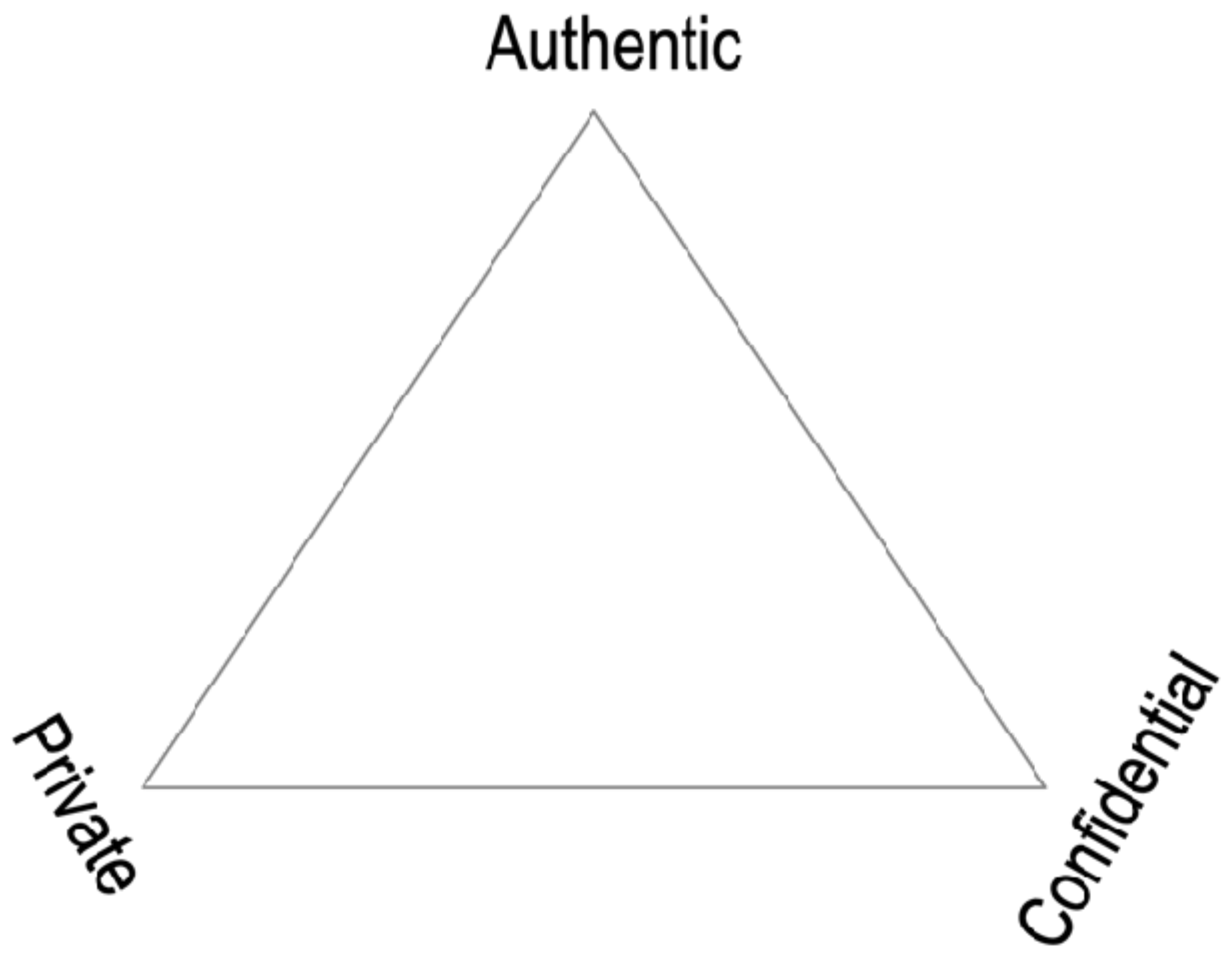
Gavrichenkov, A., "Breaking HTTPS with BGP Hijacking," BlackHat, 2015 https://www.blackhat.com/docs/us-15/materials/us-15-Gavrichenkov-Breaking-HTTPS-With-BGP-Hijacking-wp.pdf





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!

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

Flaws of DNS/CA as Trust Spanning Layer

Insecure Key Rotation

Binding between the controlling keys and the controlled identifier is asserted by one or more CAs.

Security strength or weakness derived not cryptography but from the operational processes of CAs.

DNS provides rented identifiers under centralized control. DNS protocols are insecure due to certain structural security limitations. Domain validation weakness problem: DNS is always vulnerable to attacks that allow an adversary to observe the domain validation probes that CAs send. These can include attacks against the DNS, TCP, or BGP protocols (which lack the cryptographic protections of TLS/SSL), or the compromise of routers. Such attacks are possible either on the network near a CA, or near the victim domain itself.

It is difficult to assure the correctness of the match between data and entity when the data are presented to the CA (perhaps over an electronic network), and when the credentials of the person/company/program asking for a certificate are likewise presented.

Aggregation problem: Identity claims (authenticate with an identifier), attribute claims (submit a bag of vetted attributes), and policy claims are combined in a single container. This raises privacy, policy mapping, and maintenance issues.

Delegation problem: CAs cannot technically restrict subordinate CAs from issuing certificates outside a limited namespaces or attribute set; this feature of X.509 is not in use. Therefore, a large number of CAs exist on the Internet, and classifying them and their policies is an insurmountable task. Delegation of authority within an organization cannot be handled at all, as in common business practice.

Federation problem: Certificate chains that are the result of subordinate CAs, bridge CAs, and cross-signing make validation complex and expensive in terms of processing time. Path validation semantics may be ambiguous. The hierarchy with a third-party trusted party is the only model. This is inconvenient when a bilateral trust relationship is already in place.

DNS/CA is badly broken.

Attempts to secure it without changing its fundamental design is like putting a bandage on a compound fracture.

https://en.wikipedia.org/wiki/X.509

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

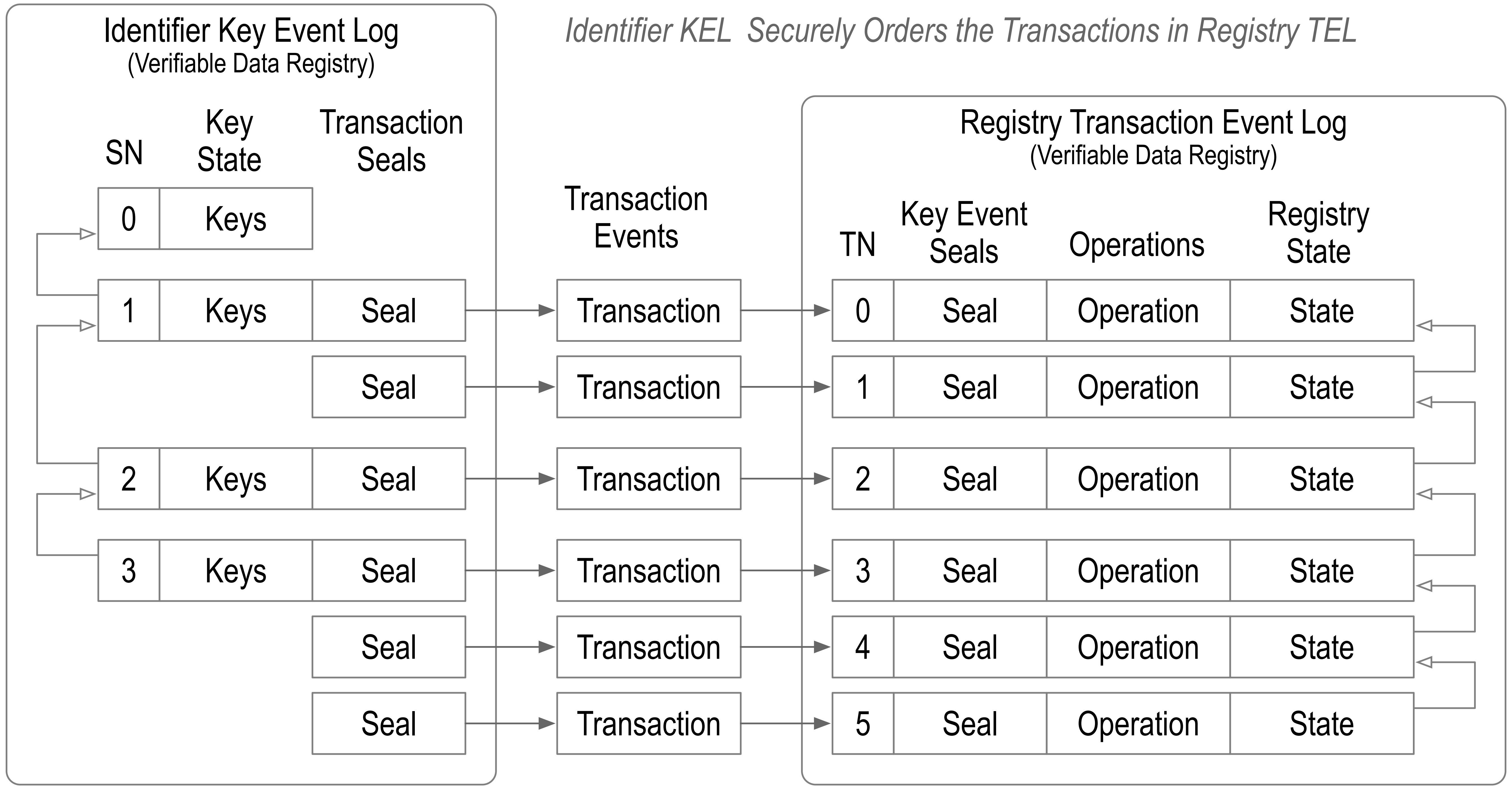
Flaws of original PGP Web-of-Trust as Trust Spanning Layer

No in-band Key Rotation mechanism

Limited supporting protocols (non minimally sufficient support)

Limited supported protocols (all essential applications not supported)

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



seal = proof of authenticity

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 that 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).

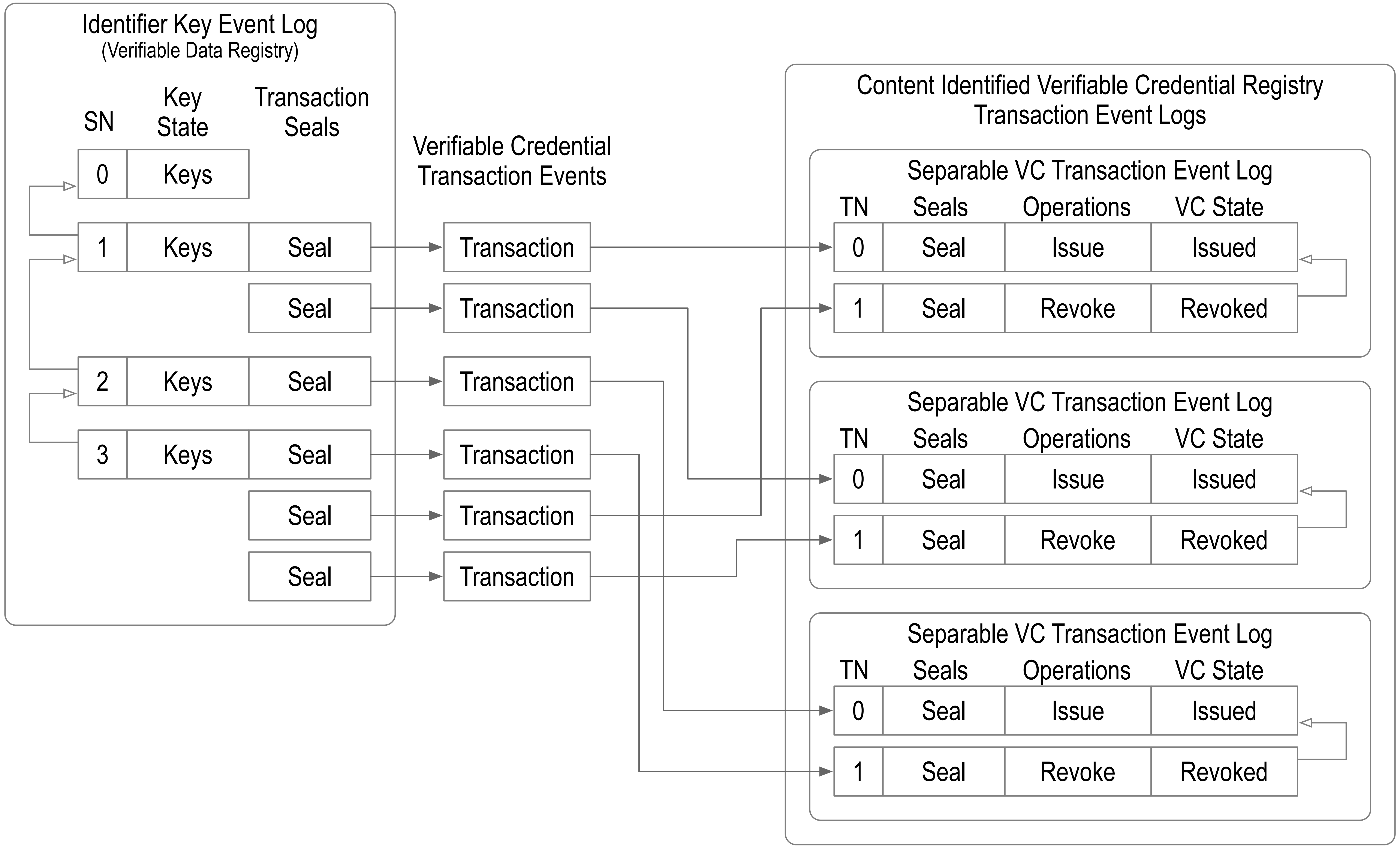
The transactions likewise contain a reference seal back to the key event authorizing the transaction.

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.

KEL Anchored Issuance-Revocation Registry with Separable VC TELs



Each VC has a uniquely self-addressing identifier (SAID)
Each VC has a uniquely identified issuer (AID)
Each VC may have a uniquely identified issuee (AID).
All VC Schema are immutable