Key Event Receipt Infrastructure

A Trust Spanning Layer for the Internet

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

Samuel M. Smith Ph.D.
sam@prosapien.com
IETF Blockchain WG 2021/11/23

Resources

Documentation:

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

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

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

https://github.com/WebOfTrust

https://github.com/WebOfTrust/keri

https://github.com/WebOfTrust/ietf-keri

https://github.com/WebOfTrust/ietf-cesr

https://github.com/WebOfTrust/ietf-said

https://github.com/WebOfTrust/ietf-ptel

ietf-kaace, ietf-ixp, ietf-pxp

ToIP ACDC (Authentic Chained Data Containers):

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

GLEIF:

https://www.gleif.org/en/lei-solutions/gleifs-digital-strategy-for-the-lei/introducing-the-verifiable-lei-vlei

Background References

Self-Certifying Identifiers:

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

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

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

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

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

Autonomic Identifiers:

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

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

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

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

Smith, S. M., “Key Event Receipt Infrastructure (KERI) Design”, 2019-2021
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI_WP_2.x.web.pdf

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

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

Certificate Transparency:

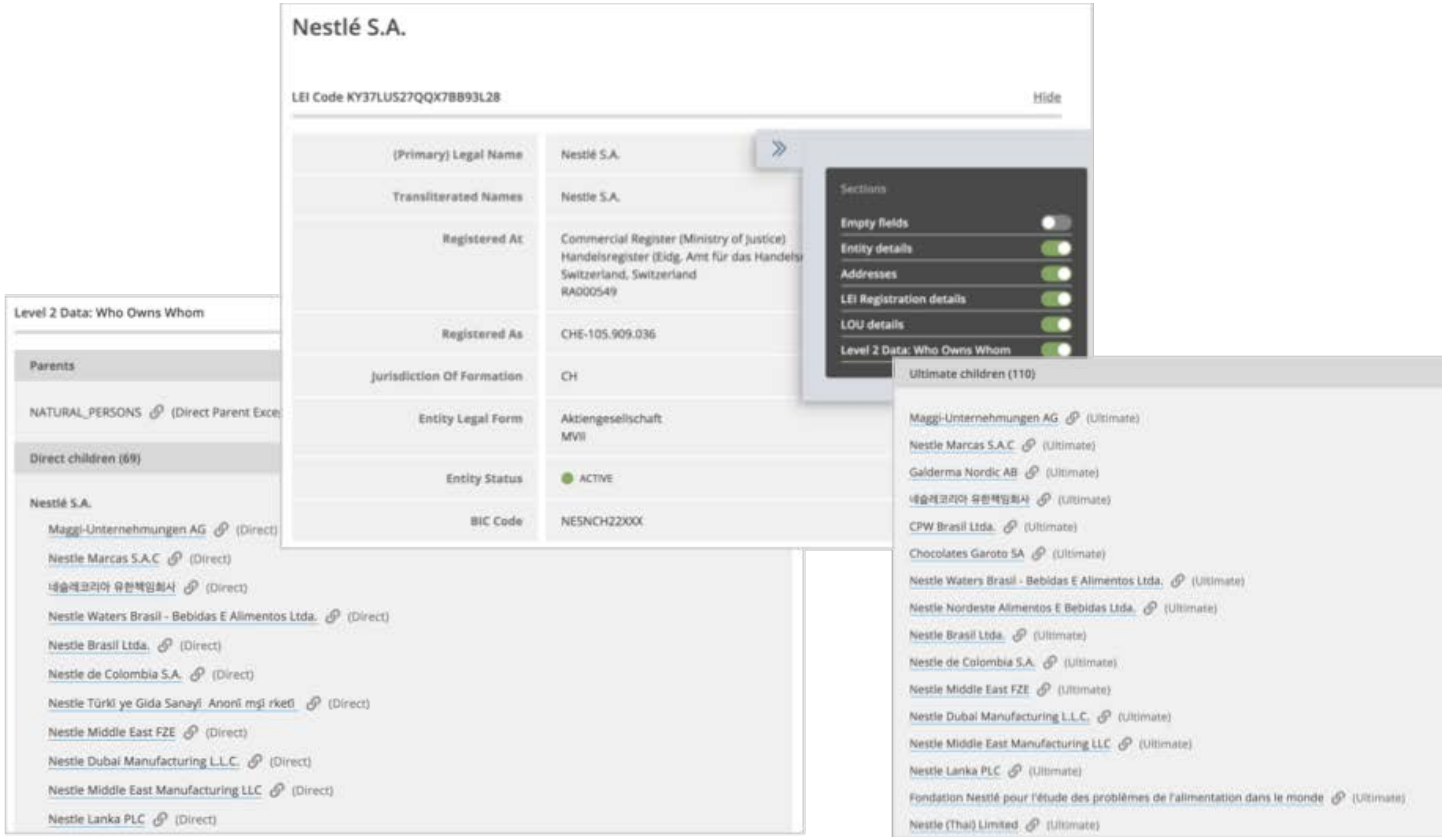
Laurie, B., “Certificate Transparency: Public, verifiable, append-only logs,” ACMQueue, vol. Vol 12, Issue 9, 2014/09/08
https://queue.acm.org/detail.cfm?id=2668154

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

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

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



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**

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

	OSI Model	IP Model	
	Application	Application	
	Presentation		
Authentication	Session		
	Transport	Transport	TCP, UDP
	Network	Network	IP
	Link	Link	
	Physical		

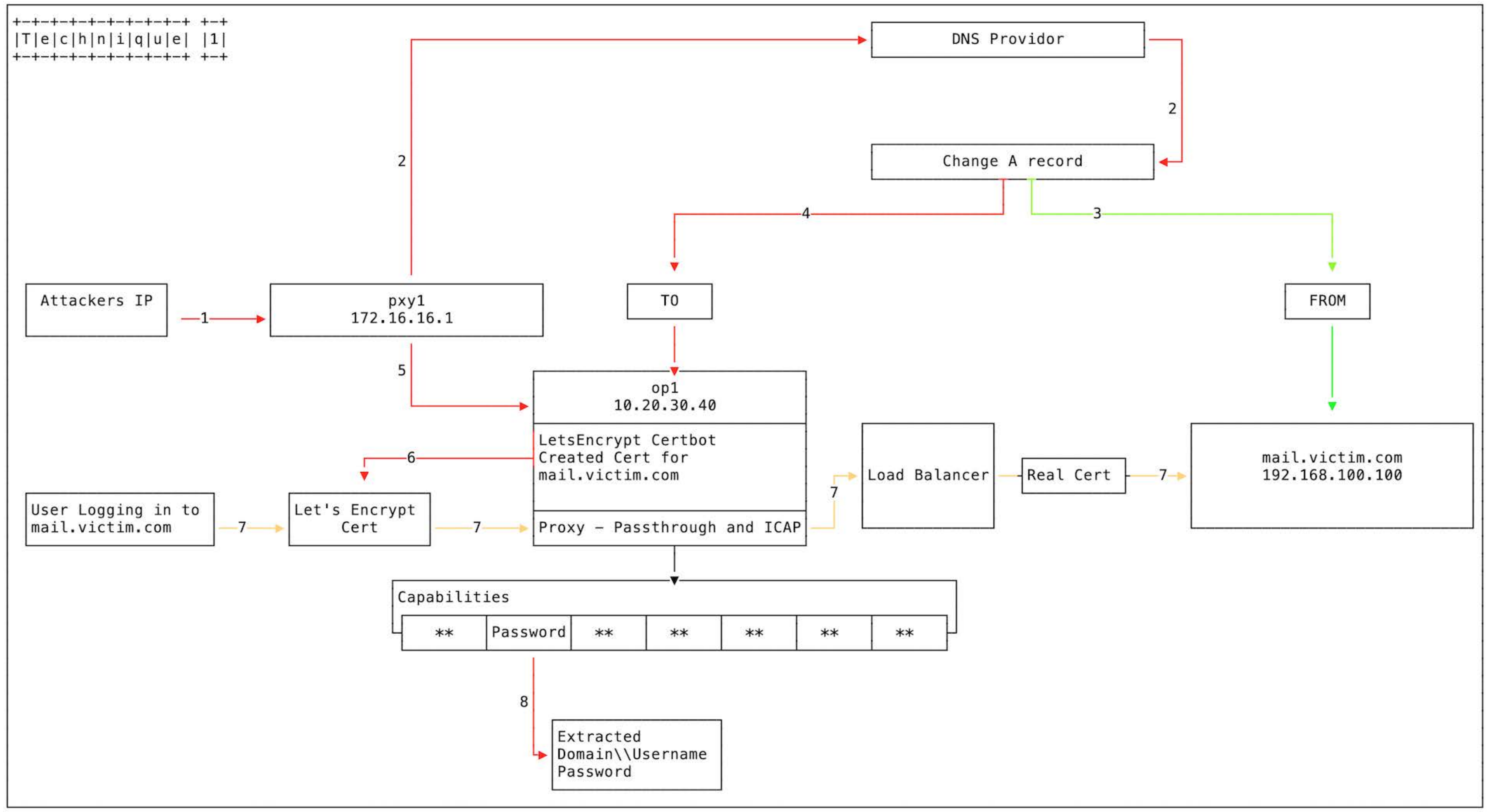
Instead …

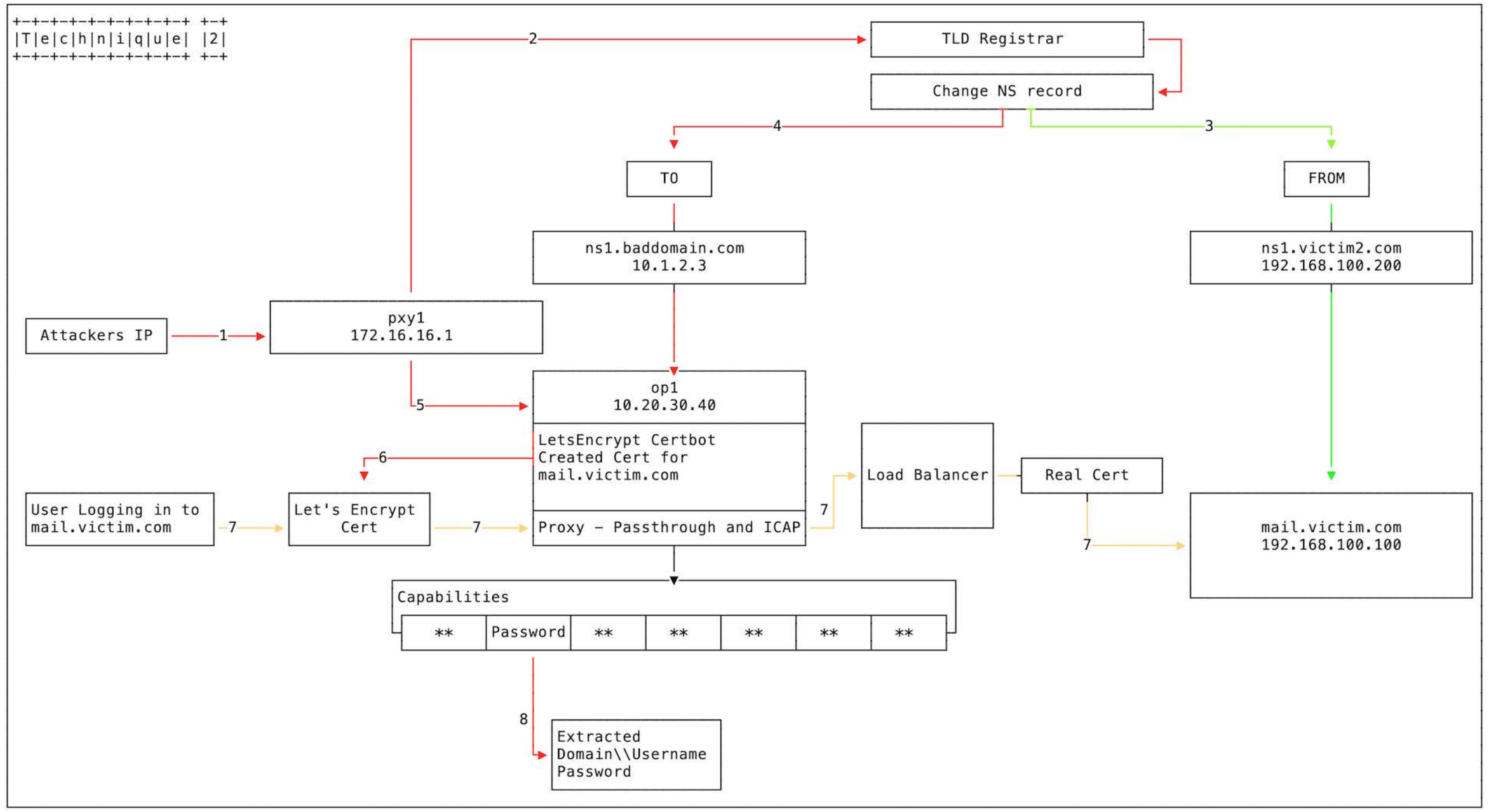
We use *bolt-on* identity system security overlays. (DNS-CA …)

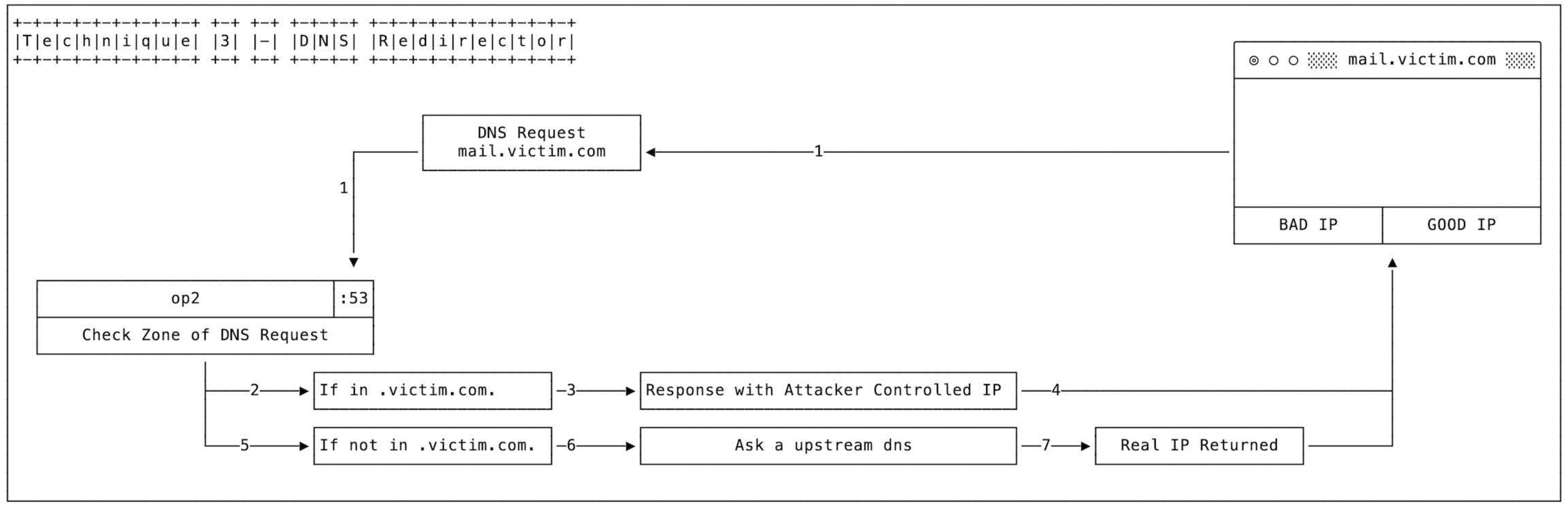
DNS Hijacking

DNS hijacking uses clever tricks that enable 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/





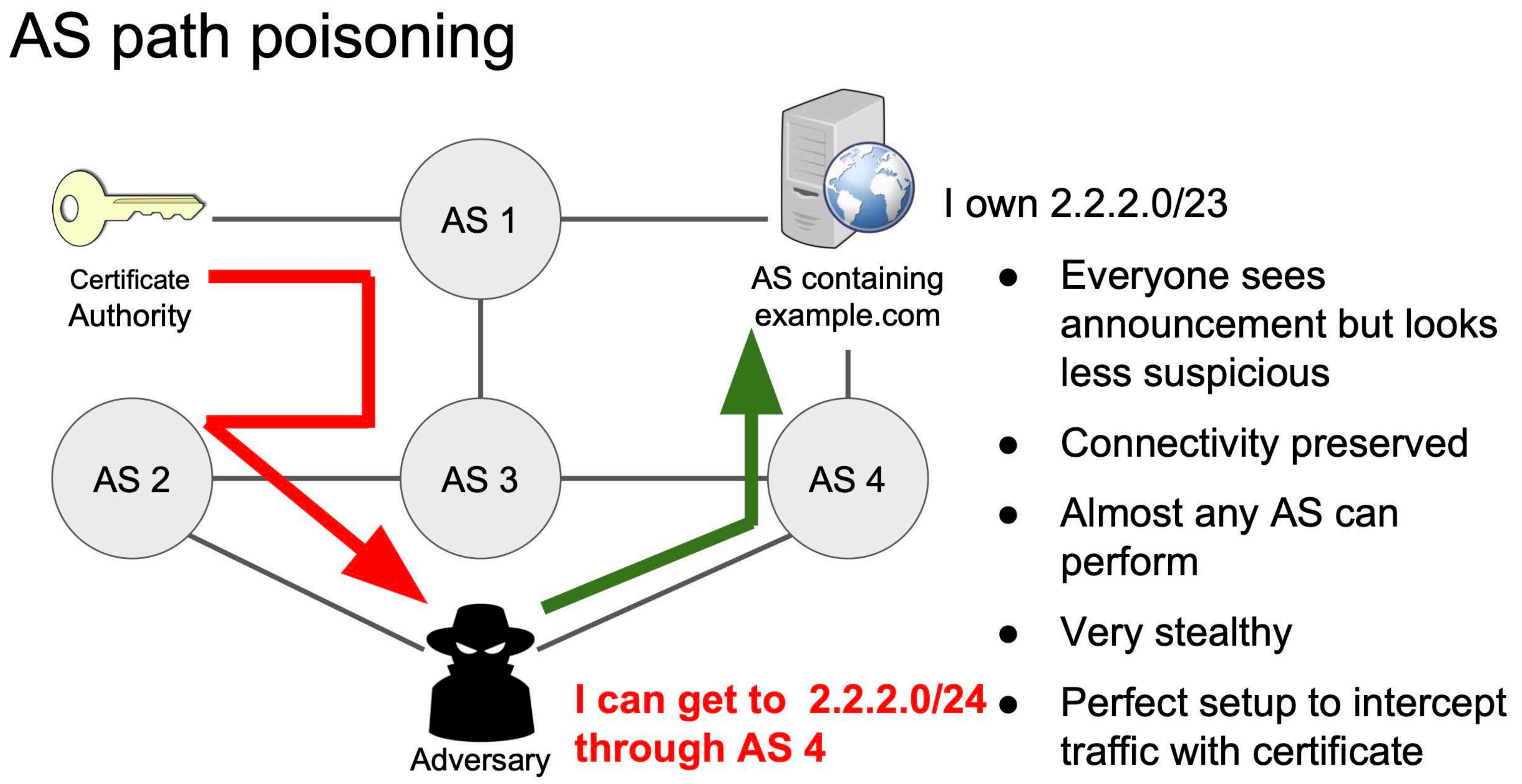


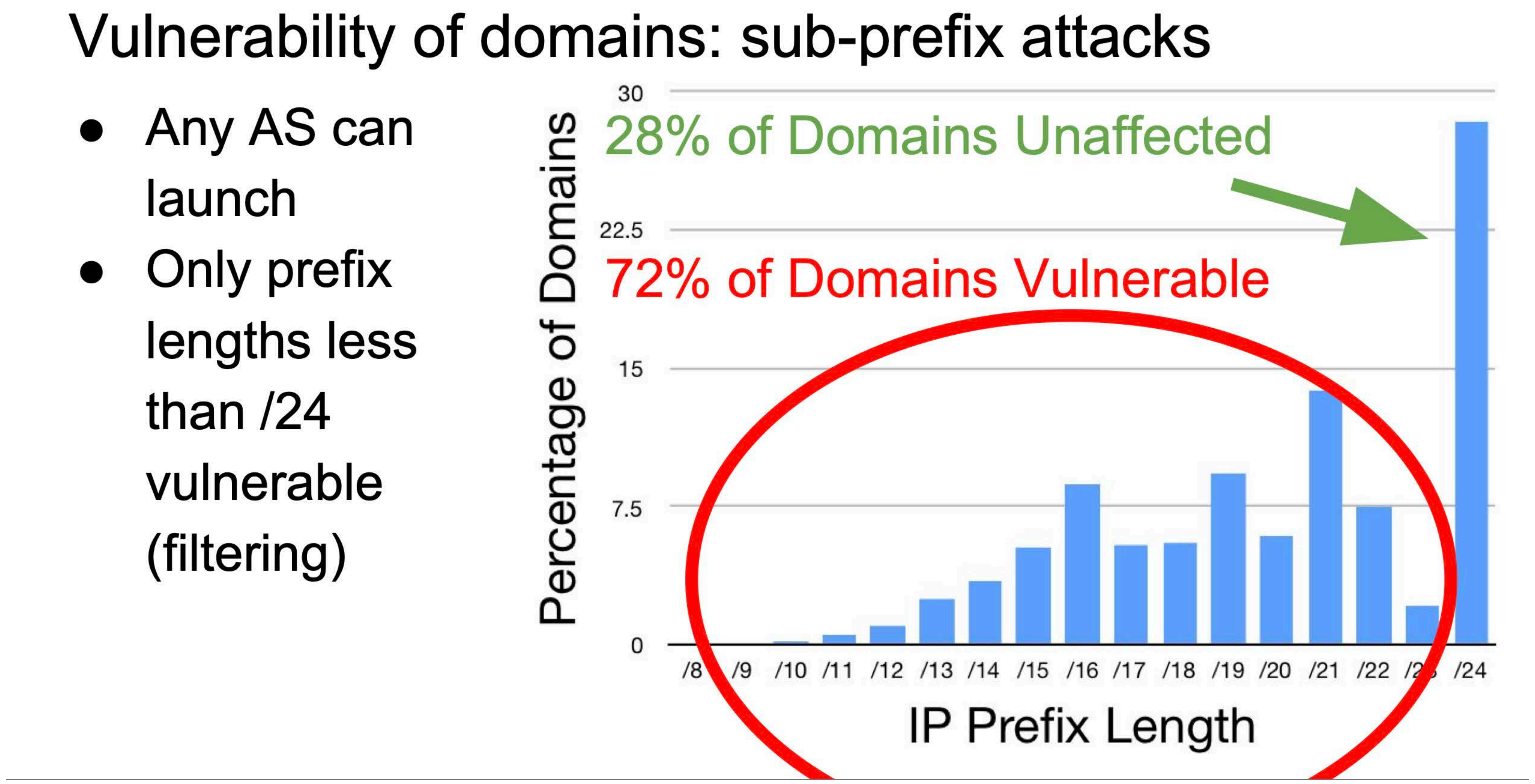
BGP Hijacking: AS Path Poisoning

Spoofing domain verification process from CA enables 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

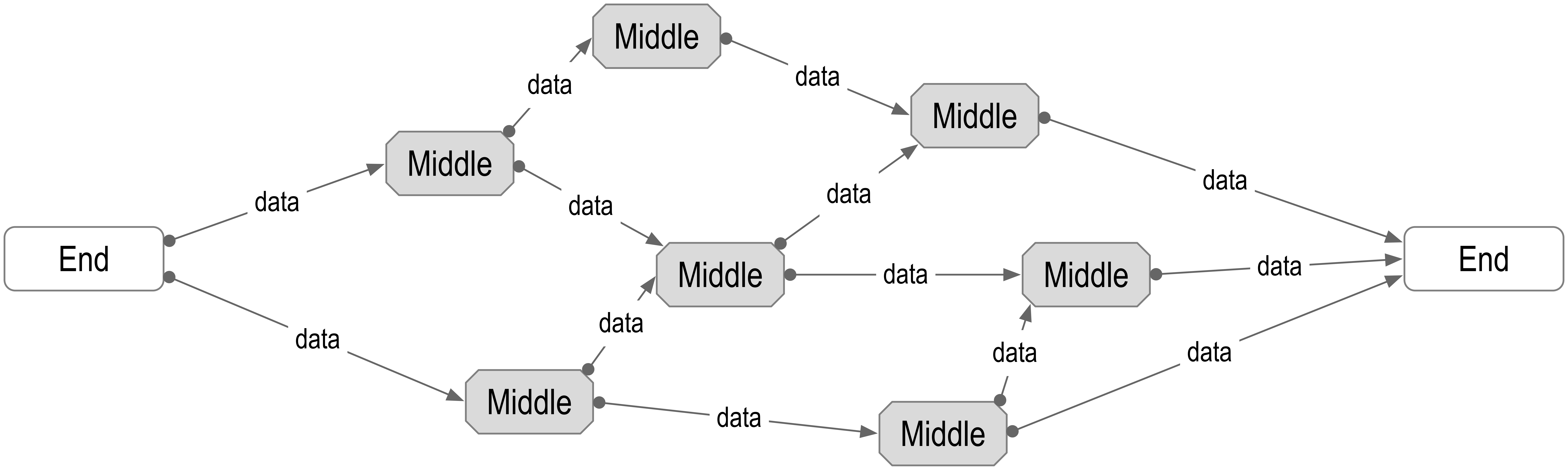
Gavrichenkov, A., "Breaking HTTPS with BGP Hijacking," BlackHat, 2015 https://www.blackhat.com/docs/us-15/materials/us-15-Gavrichenkov-Breaking-HTTPS-With-BGP-Hijacking-wp.pdf





End Verifiability

End-to-End Verifiability

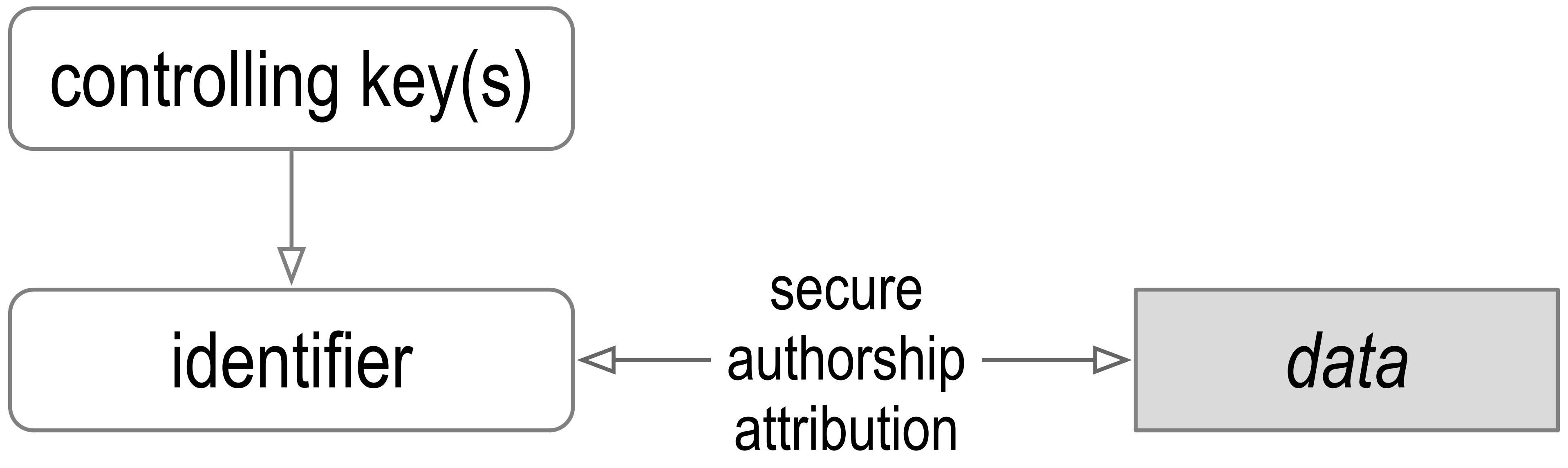


If the edges are secure, the security of the middle doesn't matter.

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

Zero-Trust-Computing

Secure Attribution Problem



Secure attribution of any communication to its source

Establish authorship of data, documents, credentials = authentic data provenance

Secure attribution via *non-repudiable* digital signatures using:

(public, private) key pairs (PKI) that control self-certifying identifiers

Duplicity evident appraisal of key state

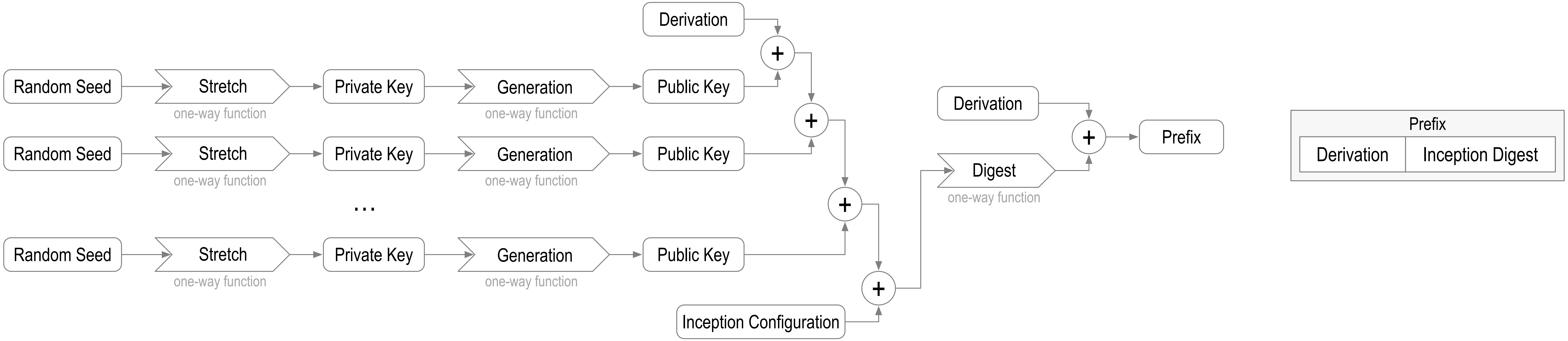
Key state proofs are portable verifiable data structures

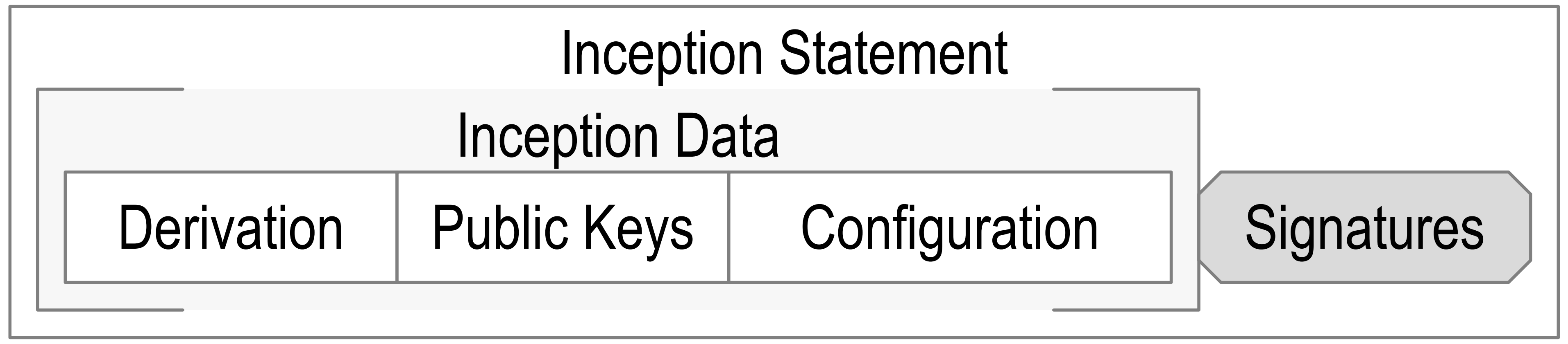
Dumb crypto is adoptable crypto (*minimally sufficient means*)

Share duplicity evident verifiable public key state

Keep private keys (secrets) private.

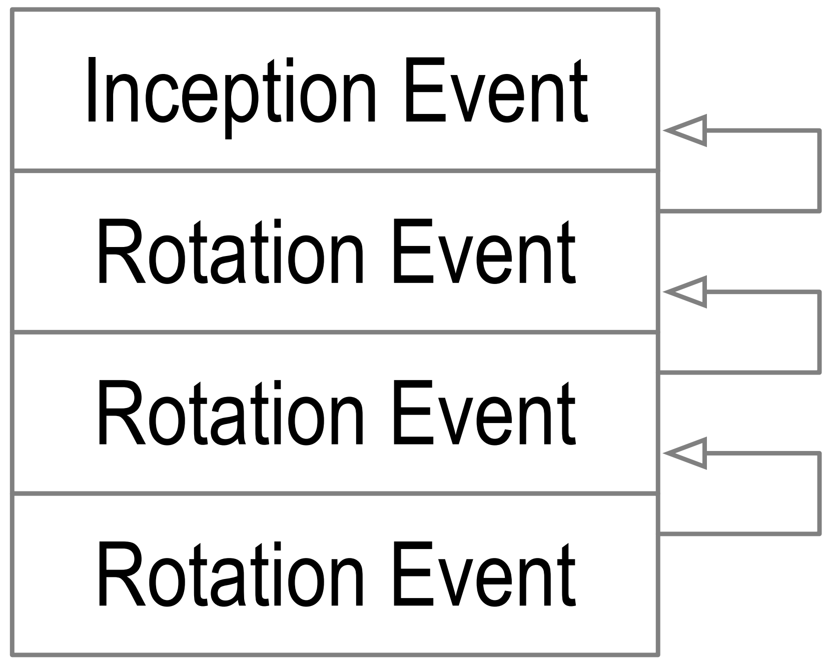
Cryptographic Root-of-Trust: Self-Certifying Identifier & Key Event Log





Key Event Log

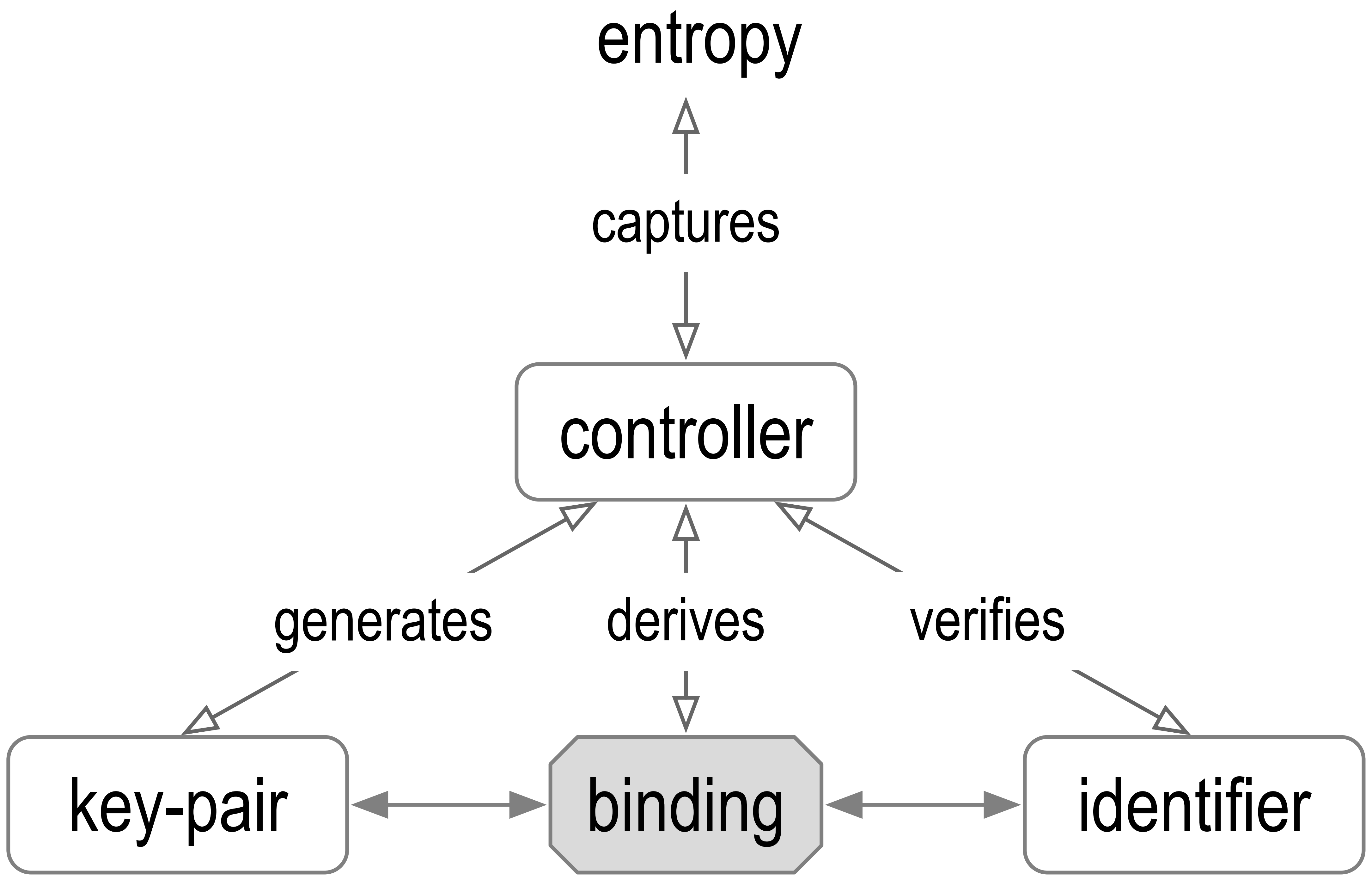
Establishment
Subsequence



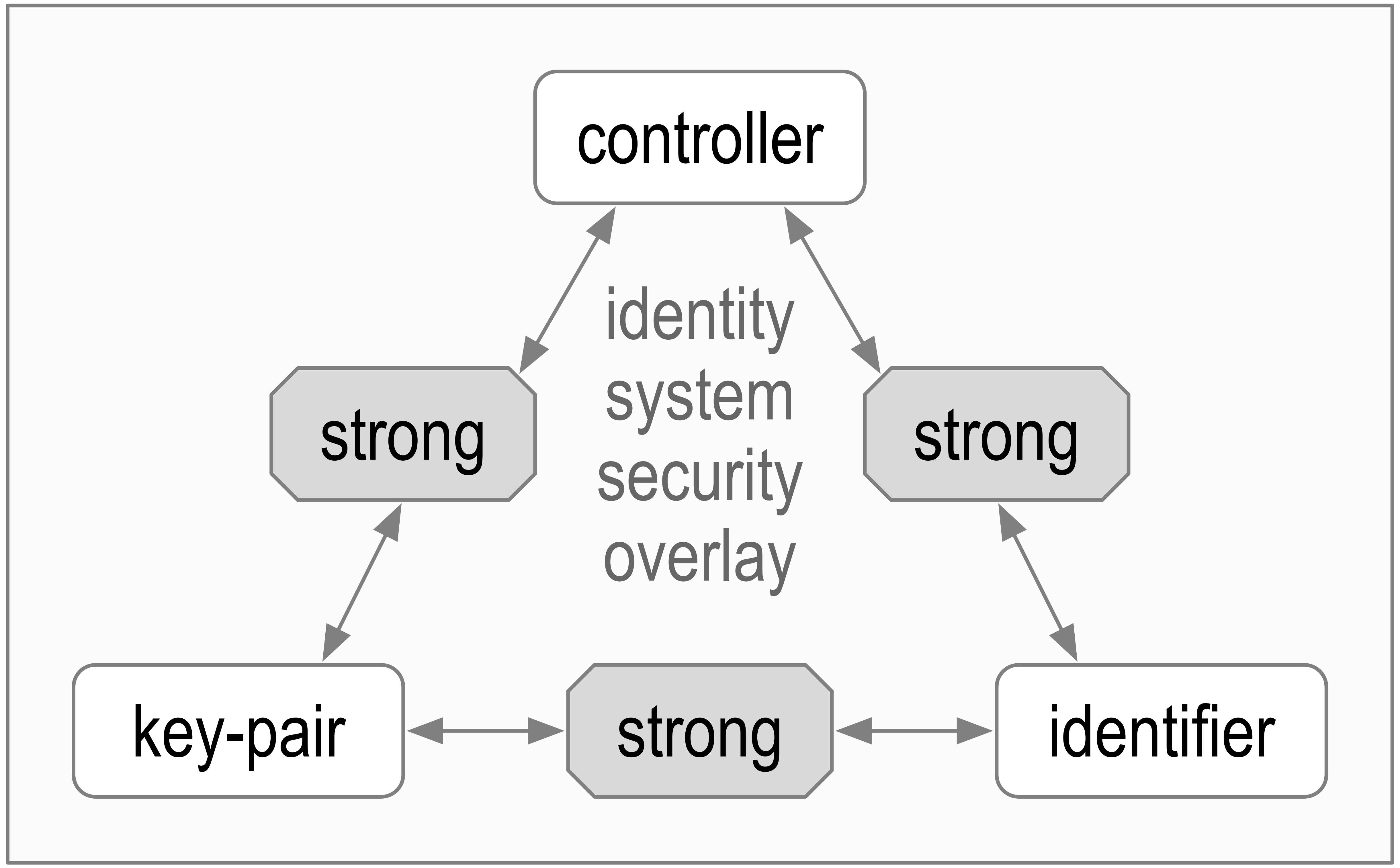
EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148

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

Self-Certifying Identifier (SCID): Issuance and Binding



self-certifying

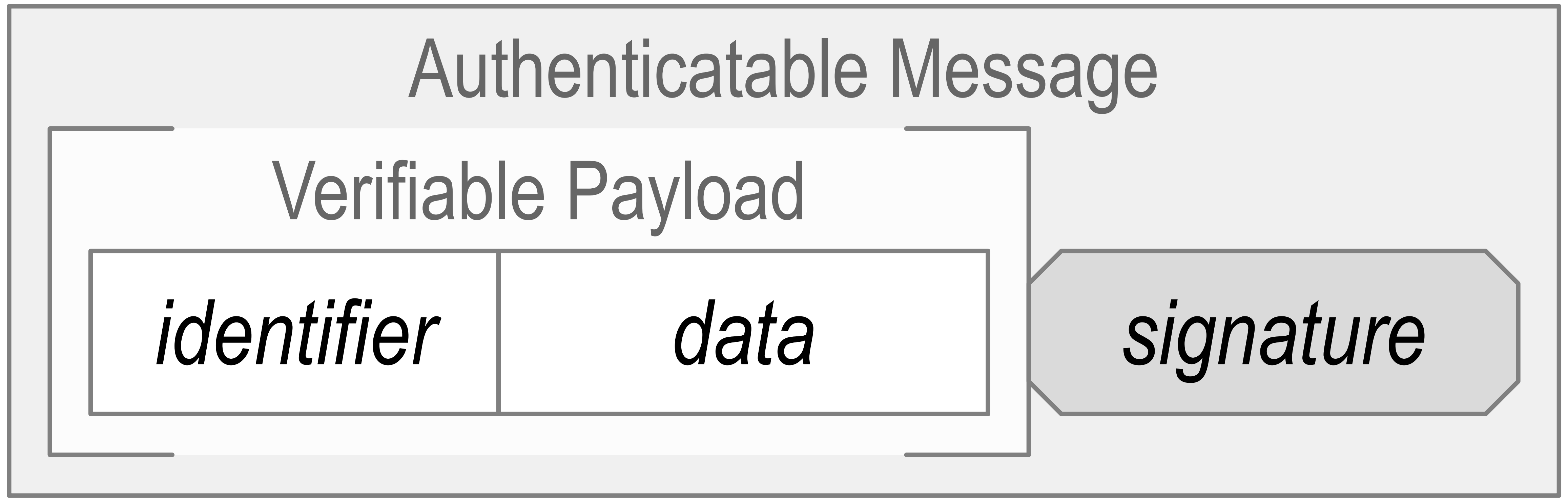


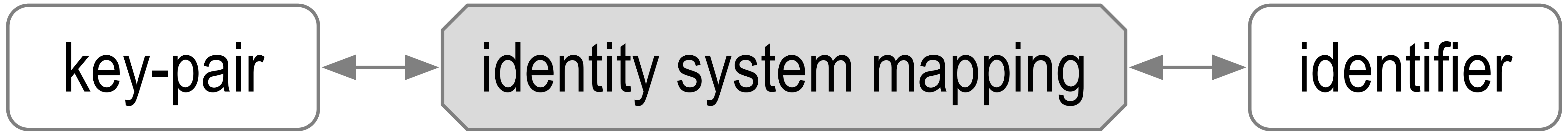
Self-Certifying Identifier Issuance

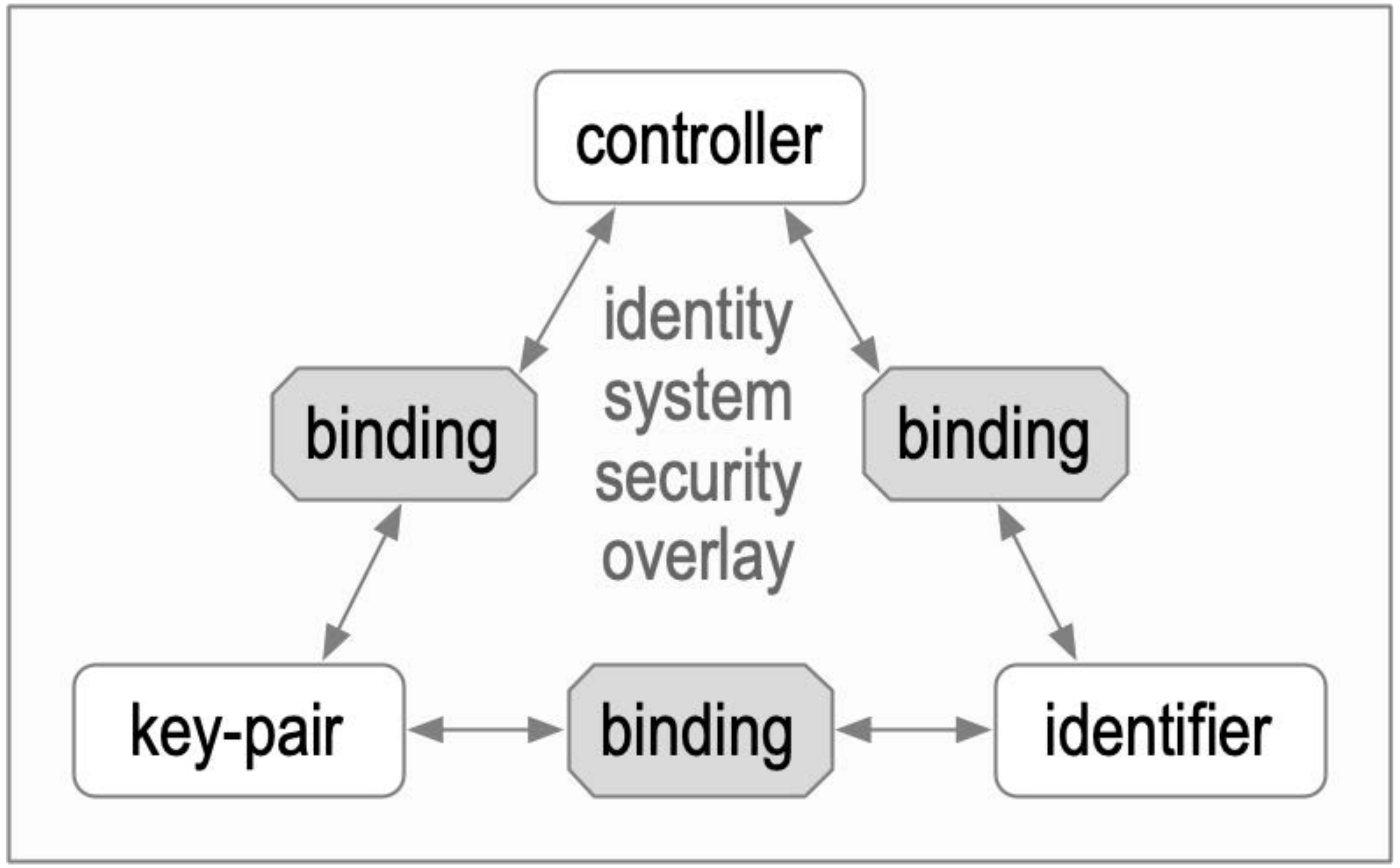
cryptographic root-of-trust

Identity System Security Overlay

Establish authenticity of IP packet's message payload.



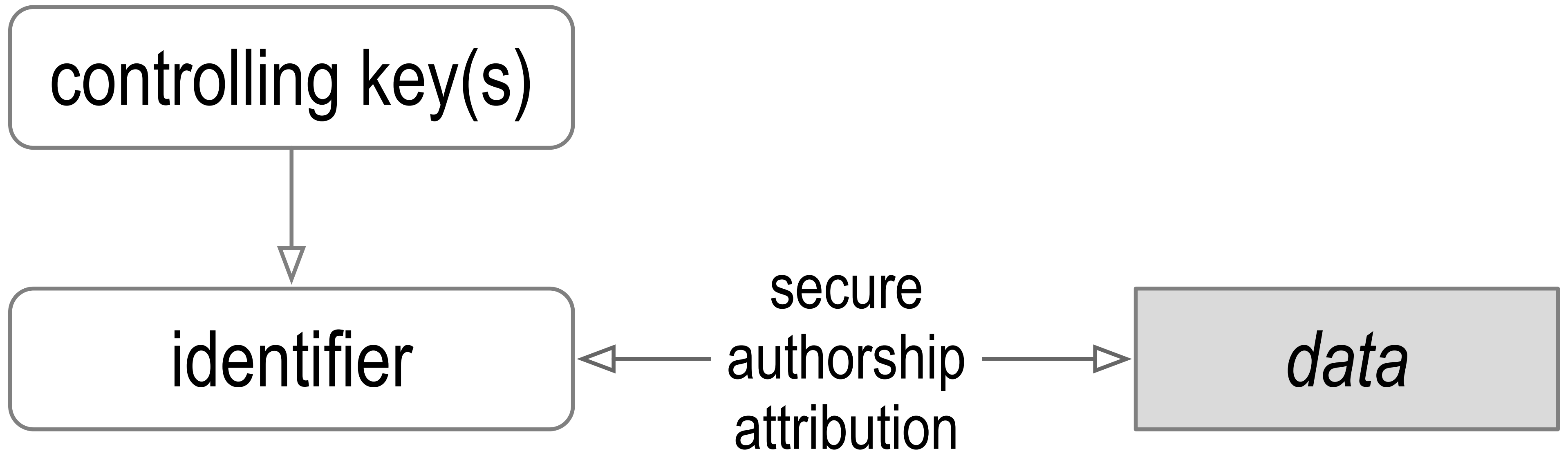




Identifier Issuance

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

Flaw of PKI (DNS/CA)



Use of private keys exposes them to side-channel attack.

Over-time, exposure makes private keys weak.

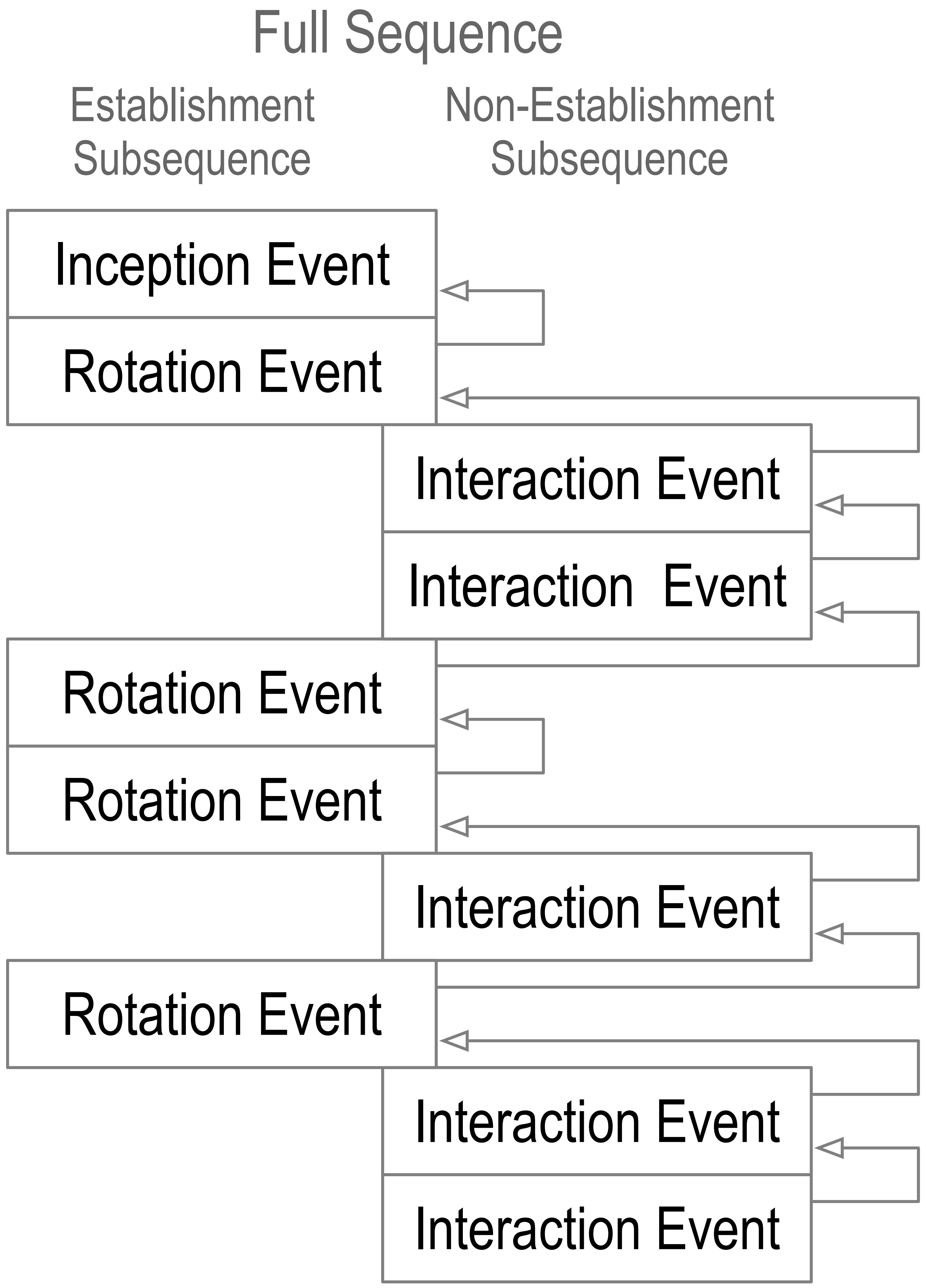
Thus, from time-to-time one must revoke and replace the controlling private keys for a given identifier

Hence key rotation

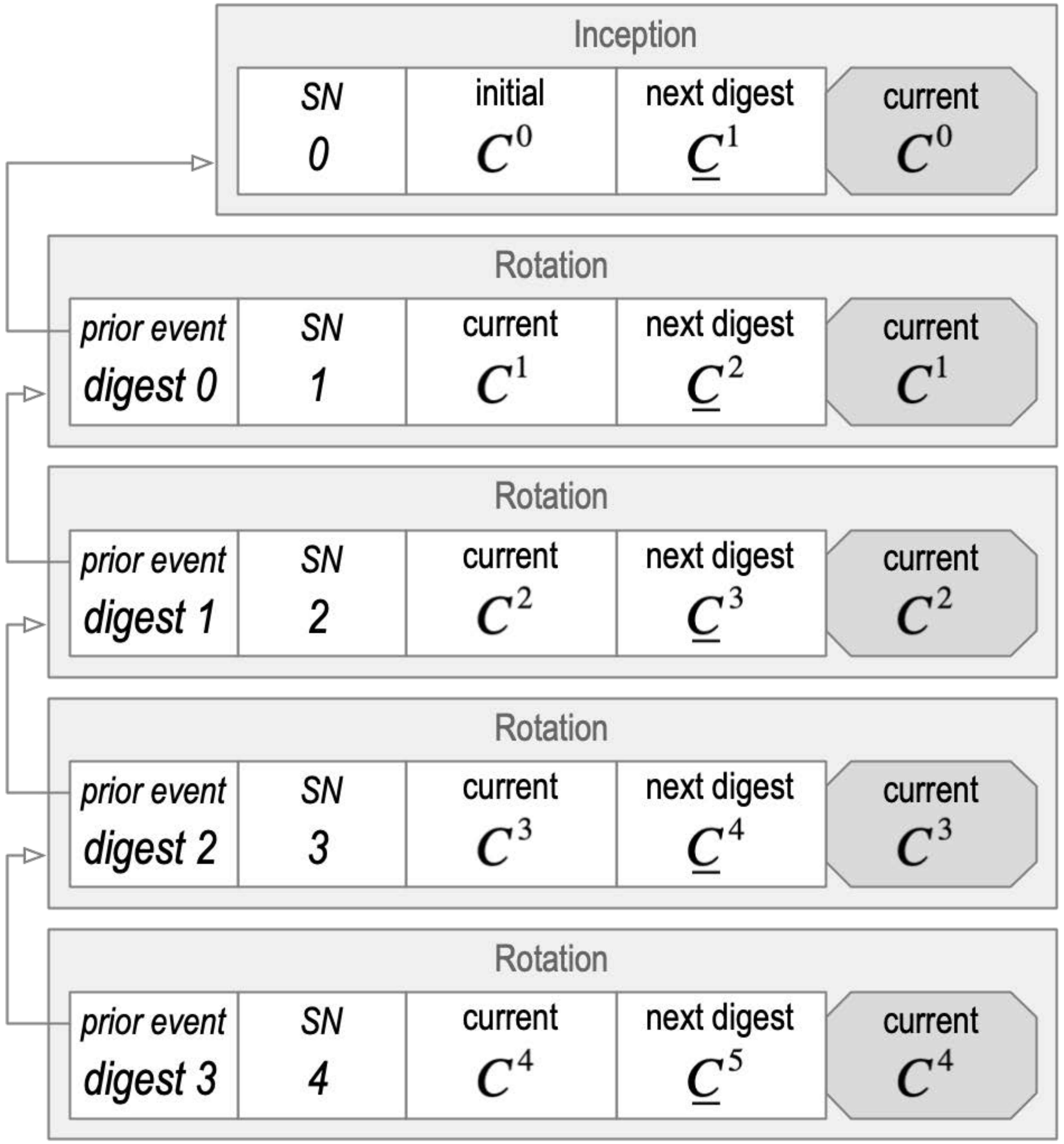
Existing PKI must re-establish the root-of-trust with each rotation thereby making it vulnerable to attack

Breaks the chain-of-trust-of-control over the identifier

Solution: Key Pre-Rotation



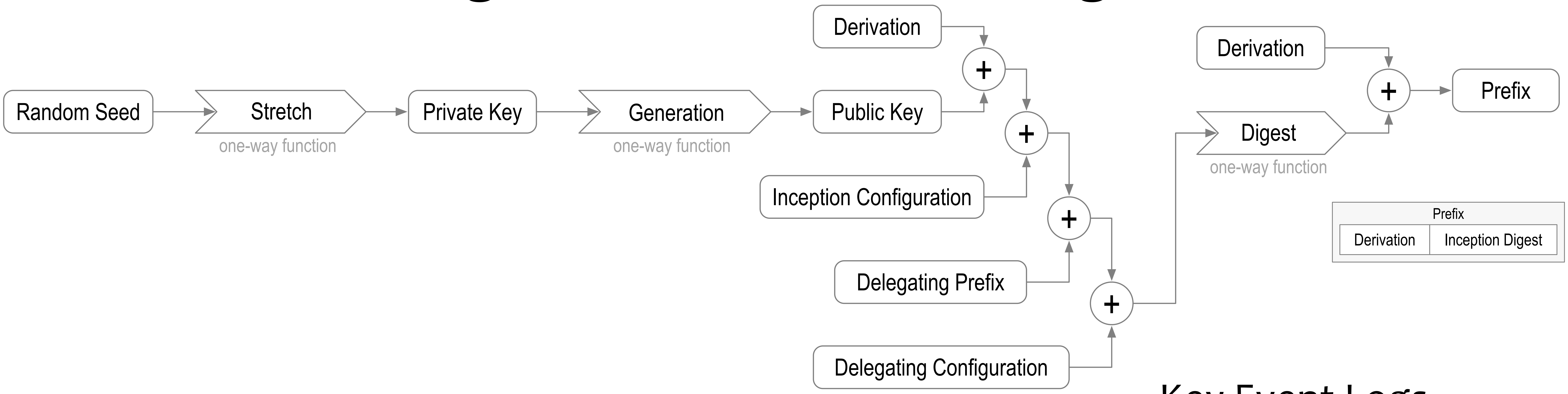
duplicity evident
verifiable data structure



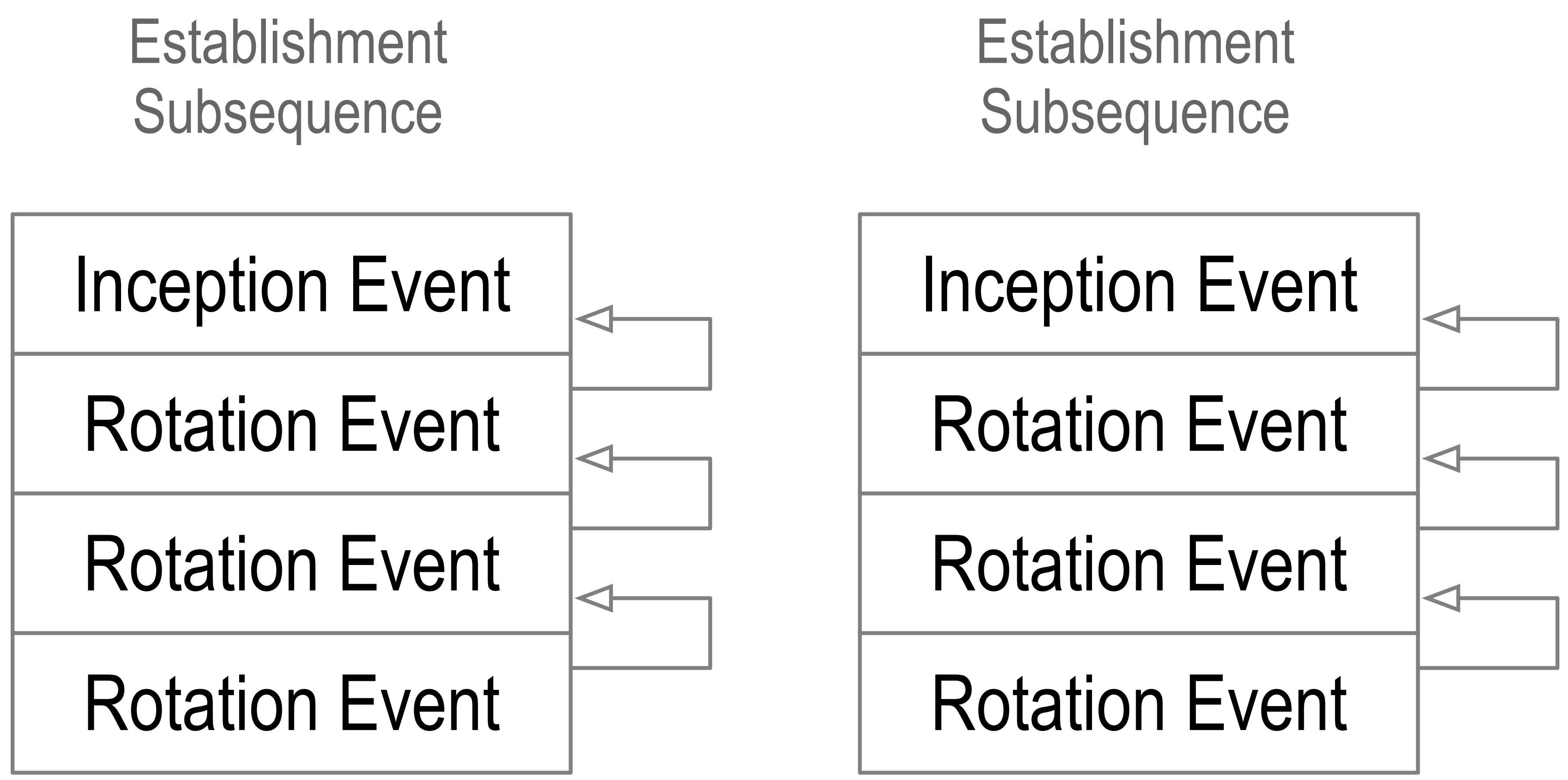
Digest of *next* key(s) makes pre-rotation post-quantum secure

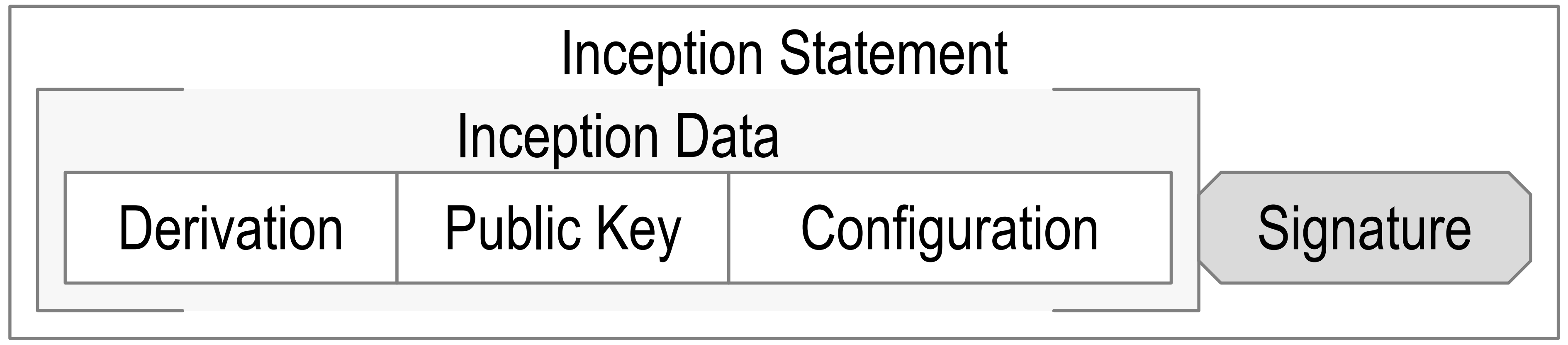
D.J. Bernstein: https://cr.yp.to/hash/collisioncost-20090517.pdf

Delegated Self-Addressing SCID



Key Event Logs

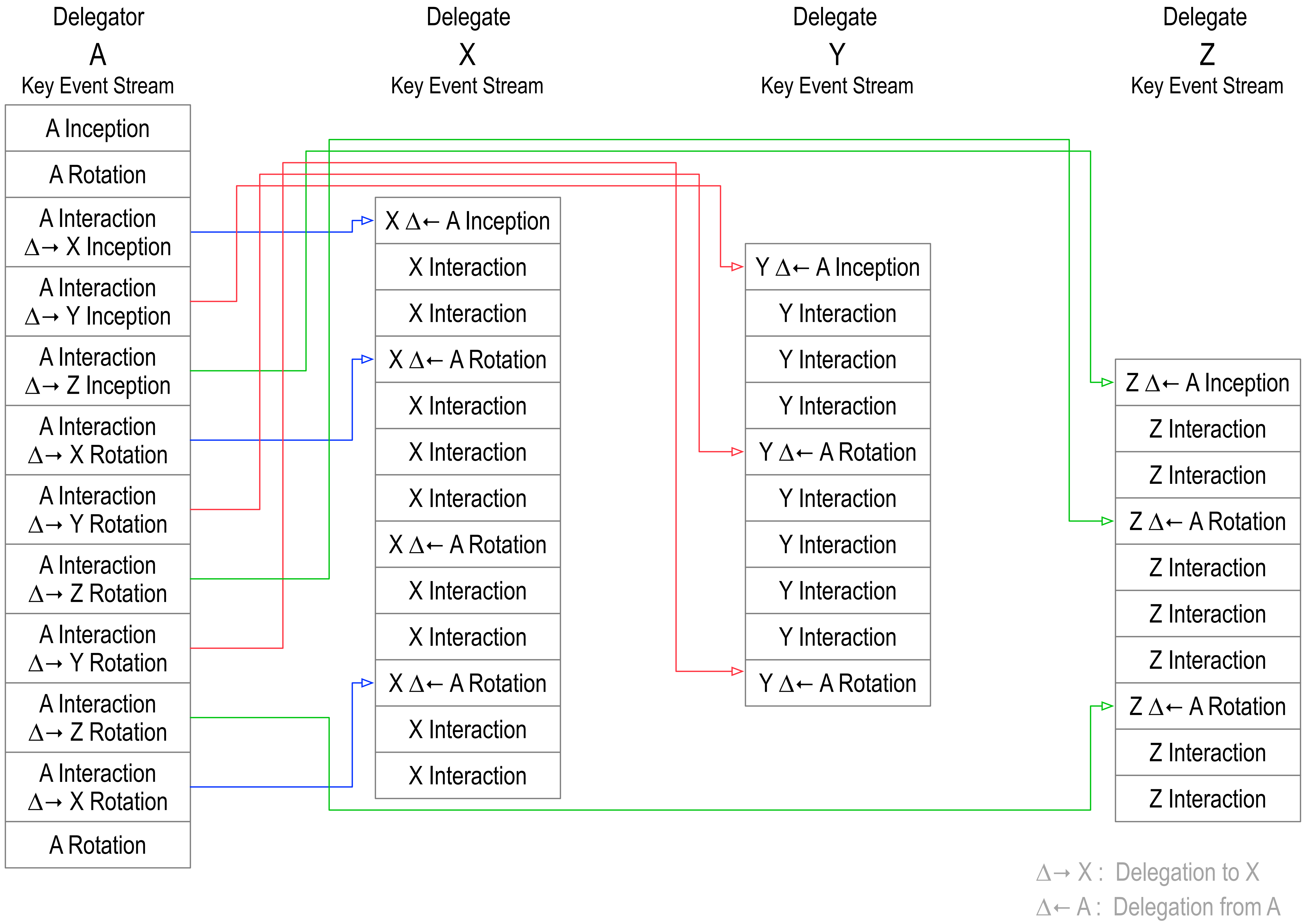


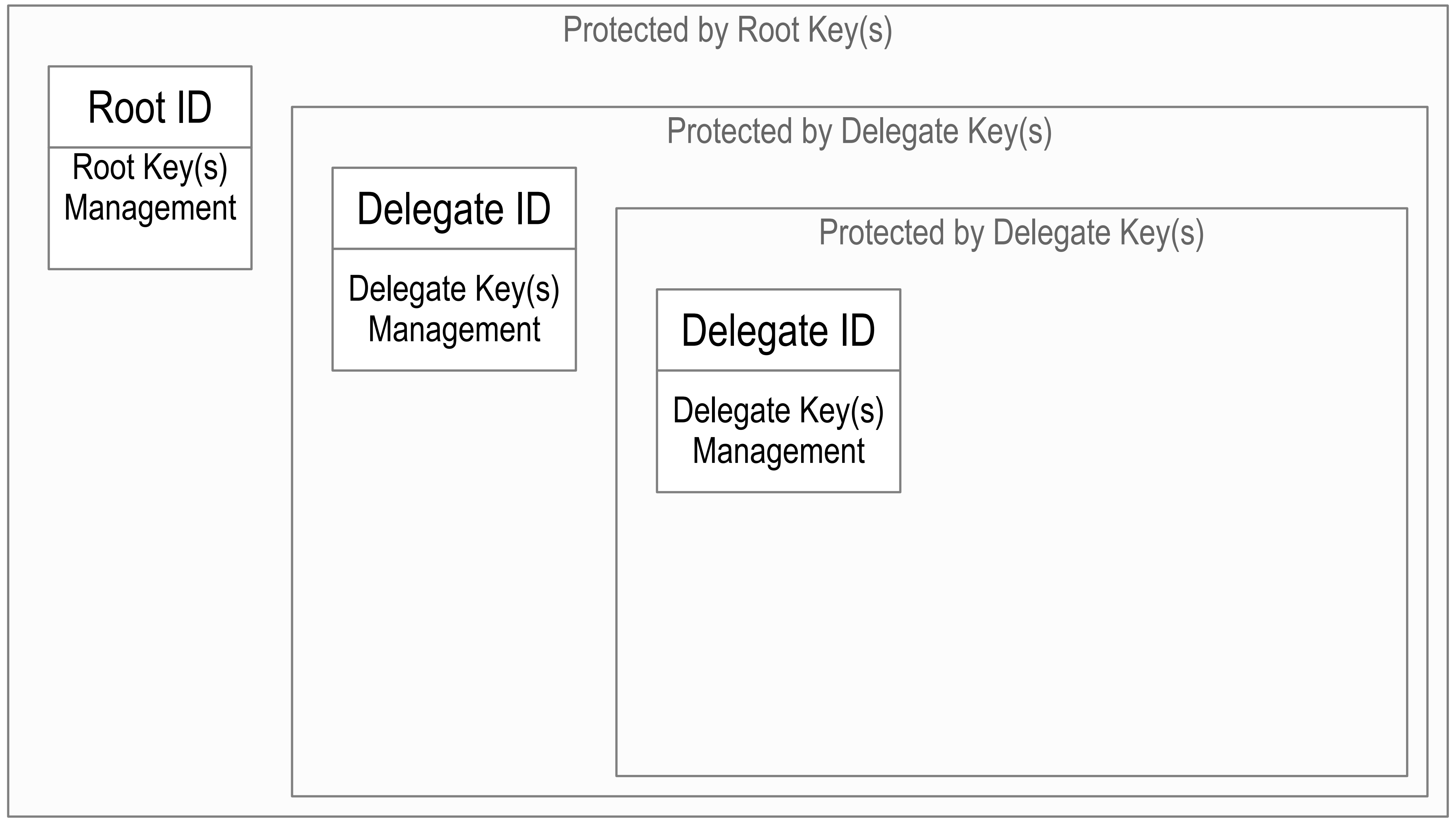


EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148

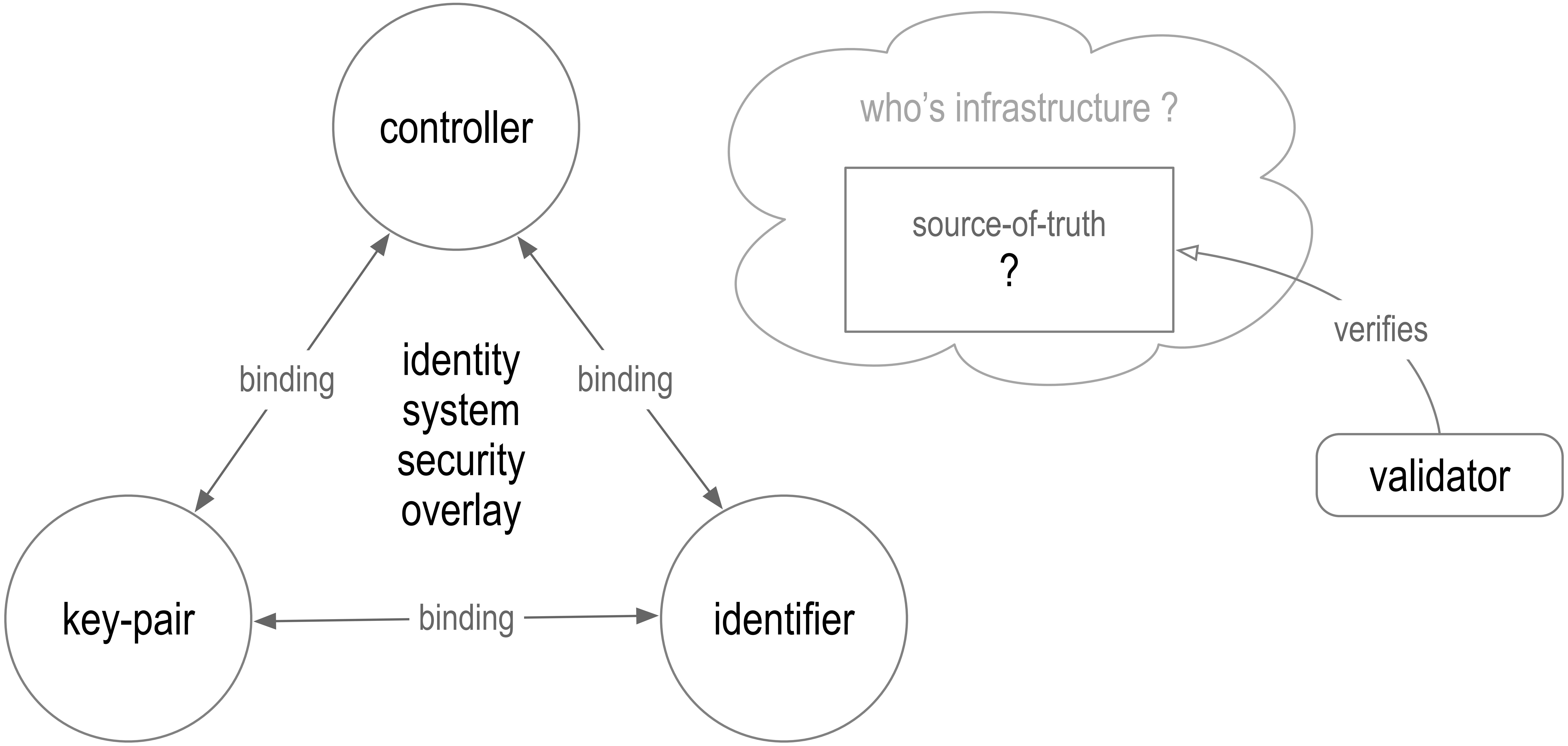
did:un:EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148/path/to/resource?name=secure#really

Identifier Delegation: Scaling & Protection



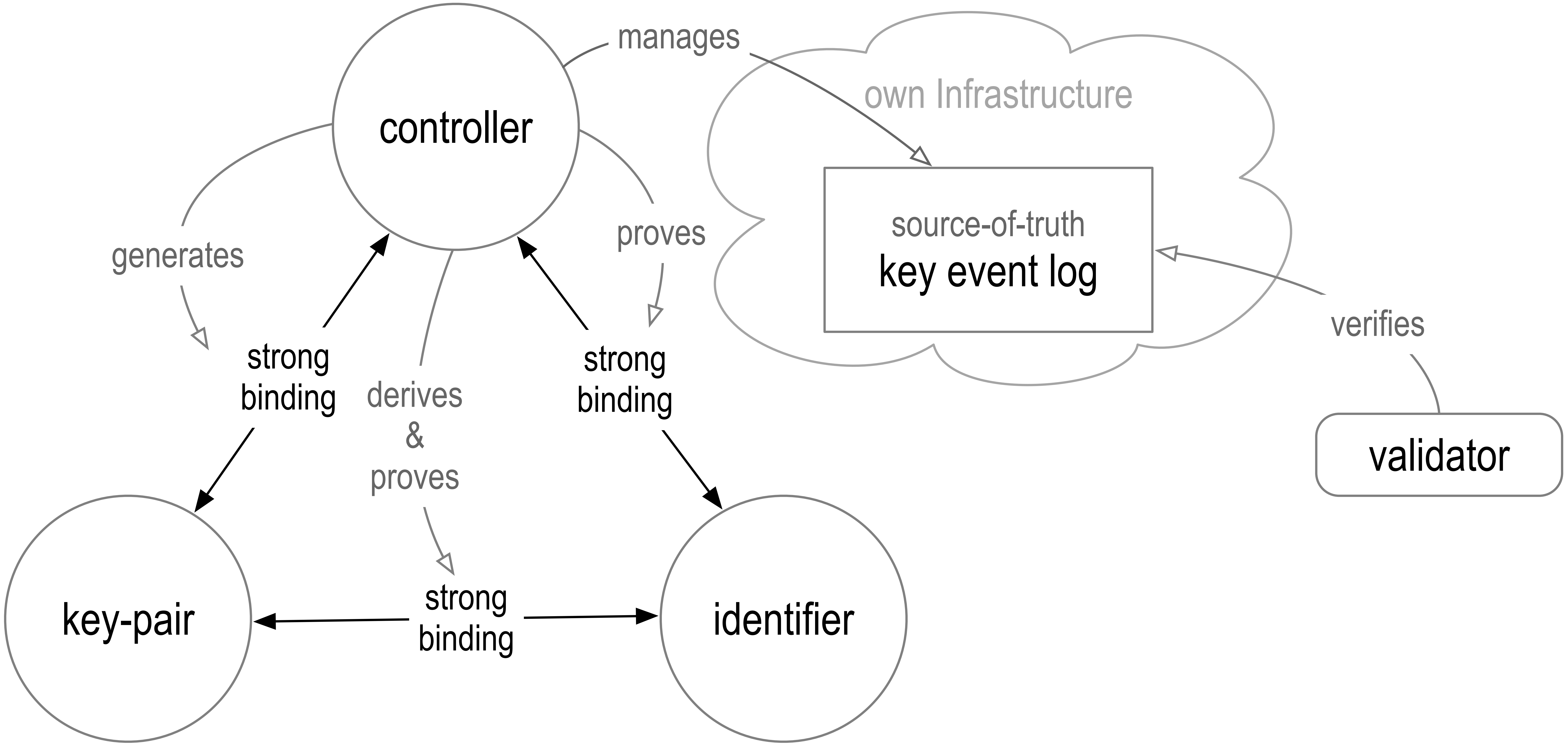


Trust Basis


controller
binding
identity
system
security
overlay
binding
key-pair
binding
identifier
who's infrastructure ?
source-of-truth
?
verifies
validator

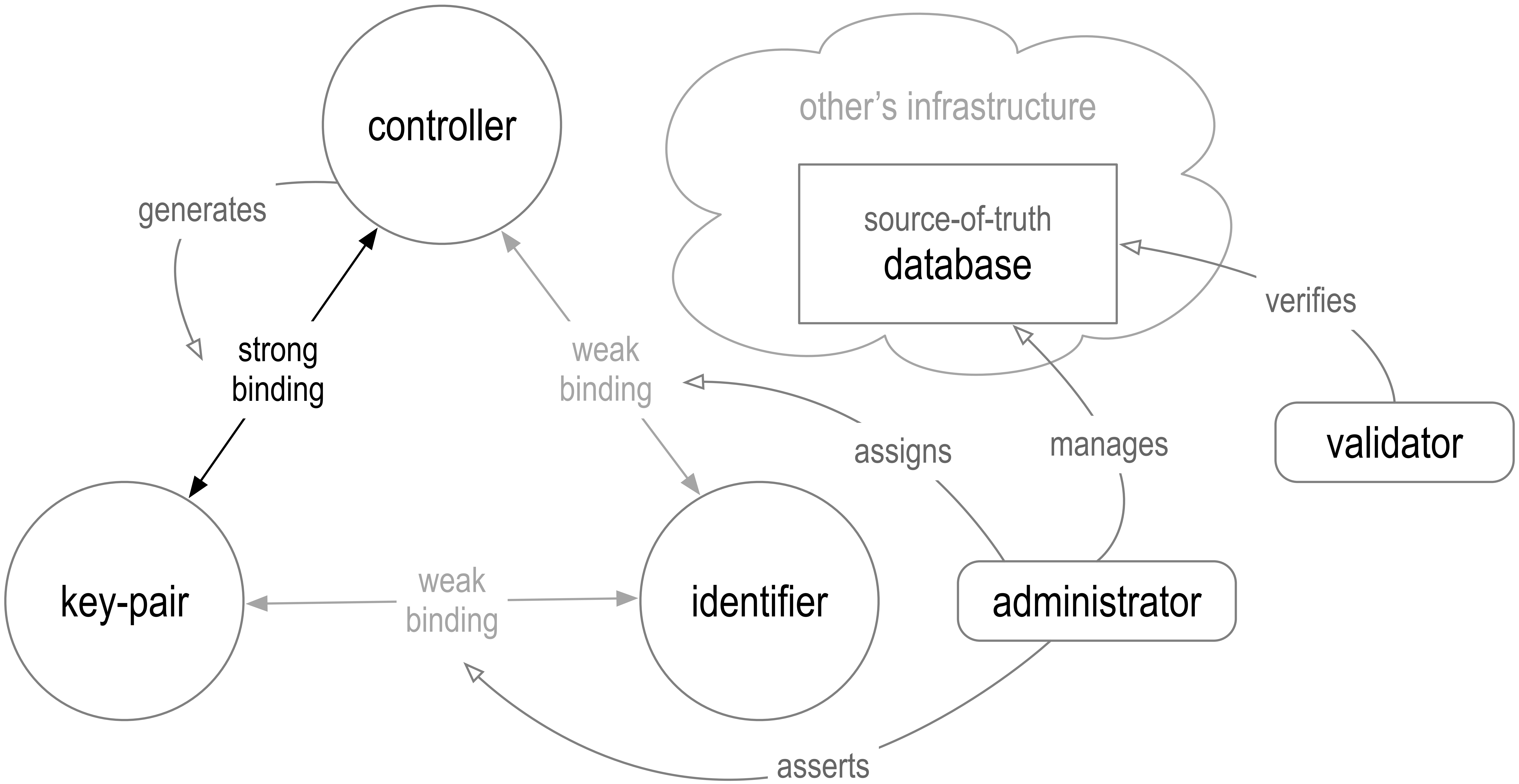
Autonomic Trust Basis

Cryptographic Proofs



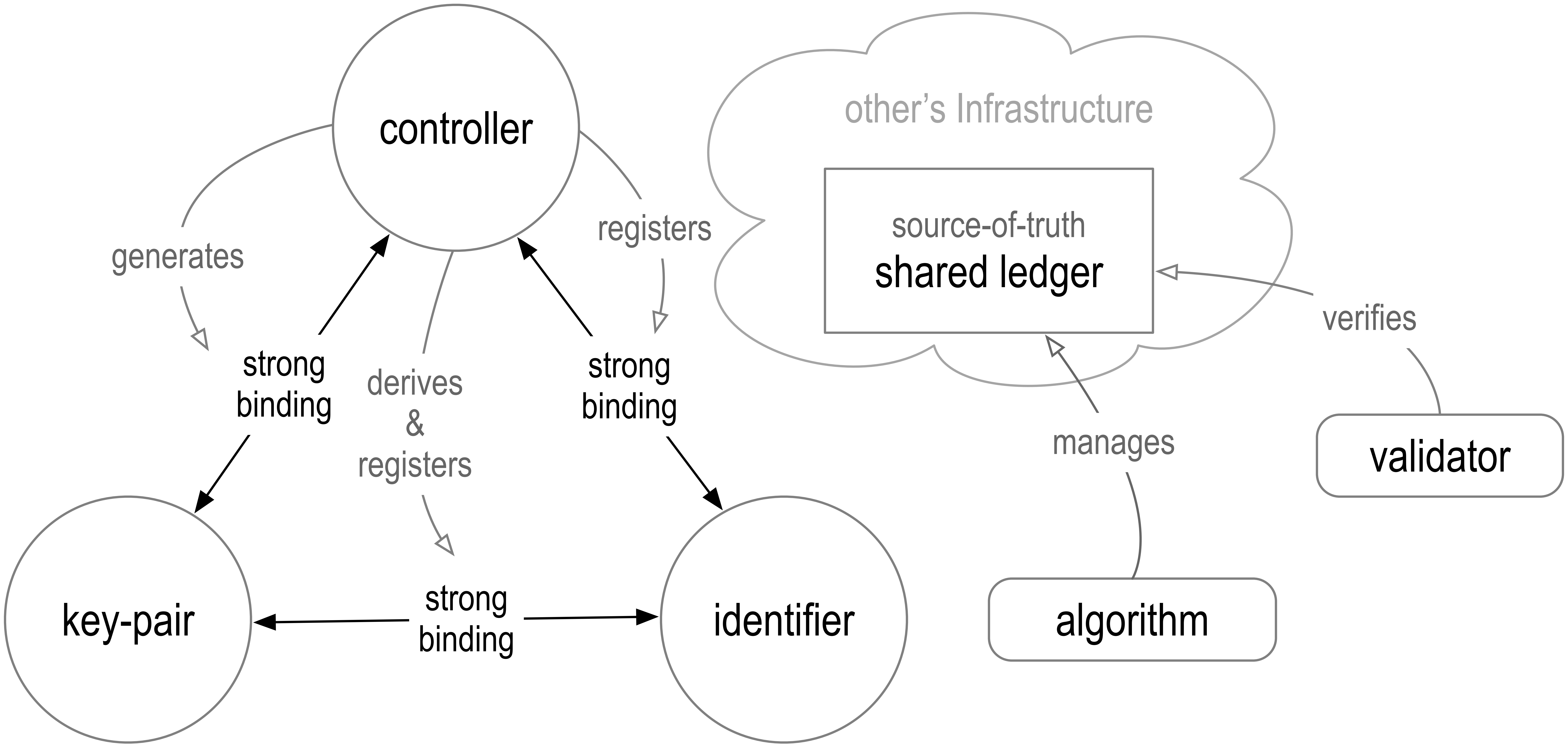
Administrative Trust Basis

DNS/Certificate Authorities



Algorithmic Trust Basis

Shared Distributed Ledgers



KERI is not Identity Proofing?

KERI Identifiers are pseudonymous = high entropy pseudo random strings of characters

`EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148`

A given KERI Identifier may be associated with a natural person or legal entity via identity proofing

The advantage of KERI is that this association need only be made once at inception.

The association persists in spite of change of control of the identifier via rotation of its keys.

KERI provides persistent control of its pseudonymous identifiers in spite of key rotations.

KERI uses pre-rotation, a forward blinded commitment to a rotation key to replace signing keys.

Rotation keys are one-time only.

KERI provides recovery of control of an identifier in spite of signing key compromise.

What is KERI?

Key Event Receipt Infrastructure: Decentralized Key Management Infrastructure

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

The flaw in PKI is key rotation.

Authorship is established in PKI with asymmetric (public, private) signing key pairs.

KERI solves the key rotation problem for control over an identifier

KERI uses portable verifiable data structures called *key event logs* (KELs) to provide duplicity evident proof of the controlling key state for pseudonymous cryptographic self-certifying identifiers (SCIDs).

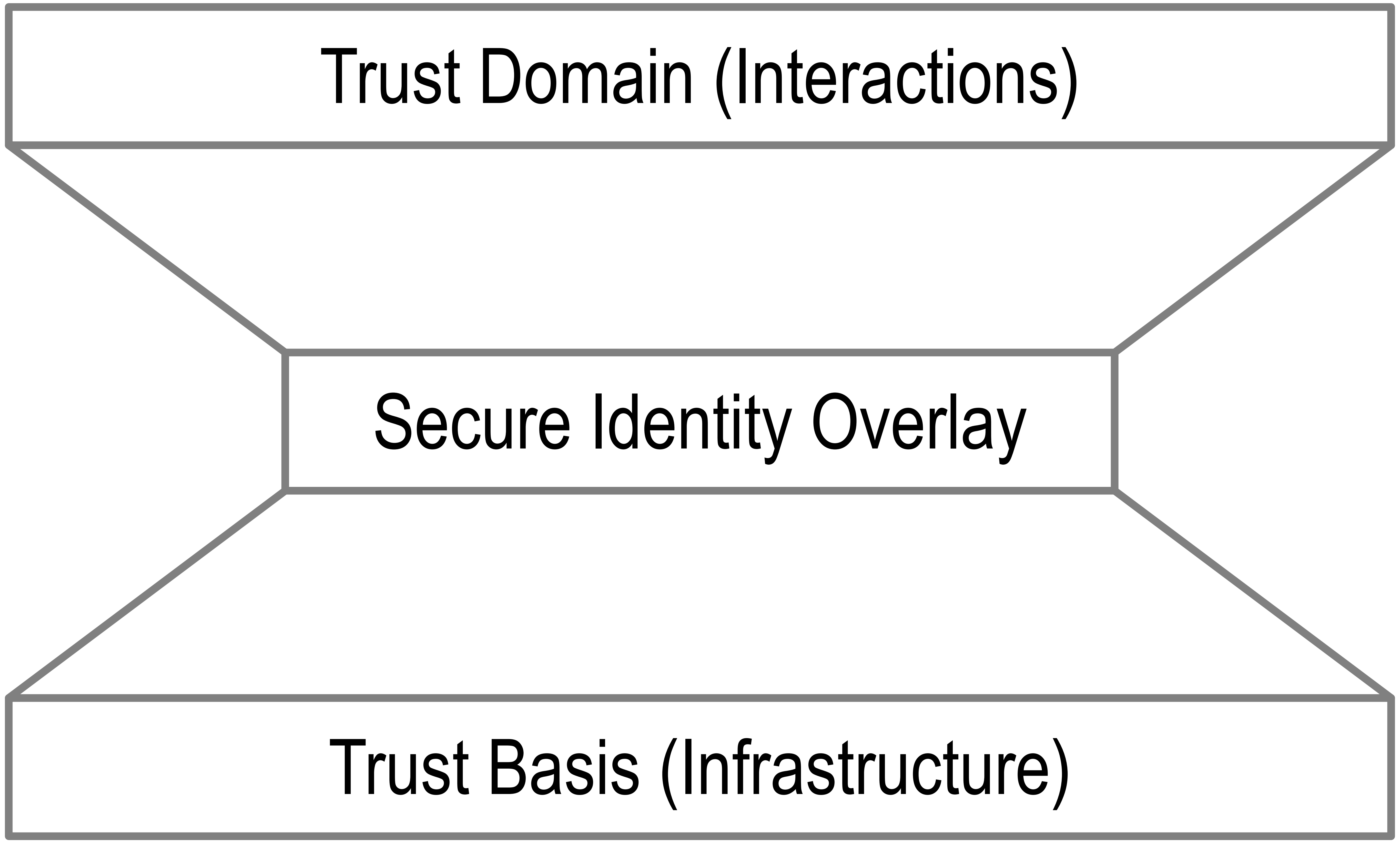
With KERI, key state is cryptographically verifiably bound to self-certifying identifiers

In contrast conventional PKI uses assertions made by trusted entities to bind key state to identifiers

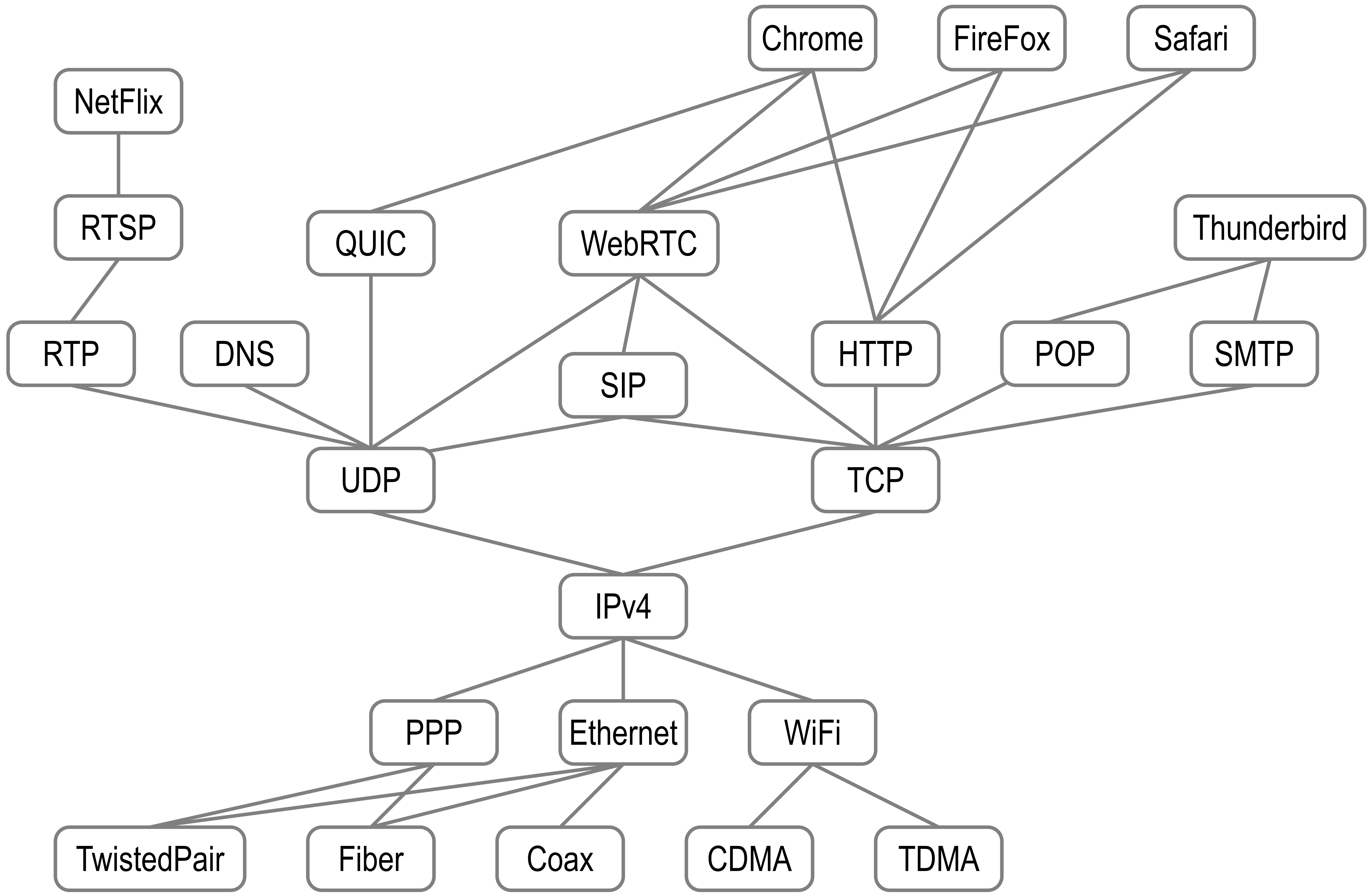
KERI solves the *secure attribution* problem with zero trust.

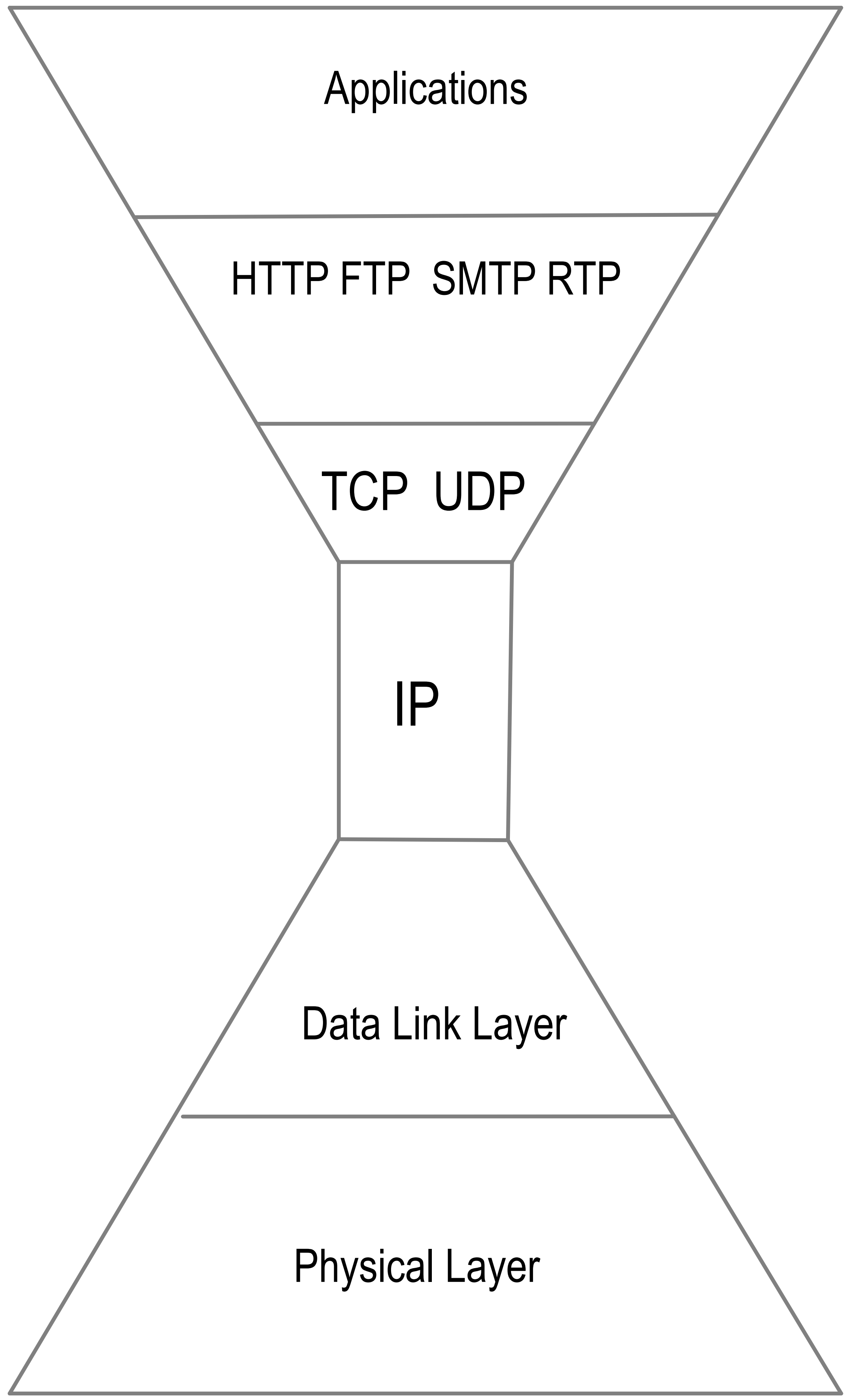
Every statement associated with an identifier may be non-repudiably and securely attributed to the controller of the identifier via a signature made with the keys determined by cryptographically verifiable key state.

Identity System Security Overlay



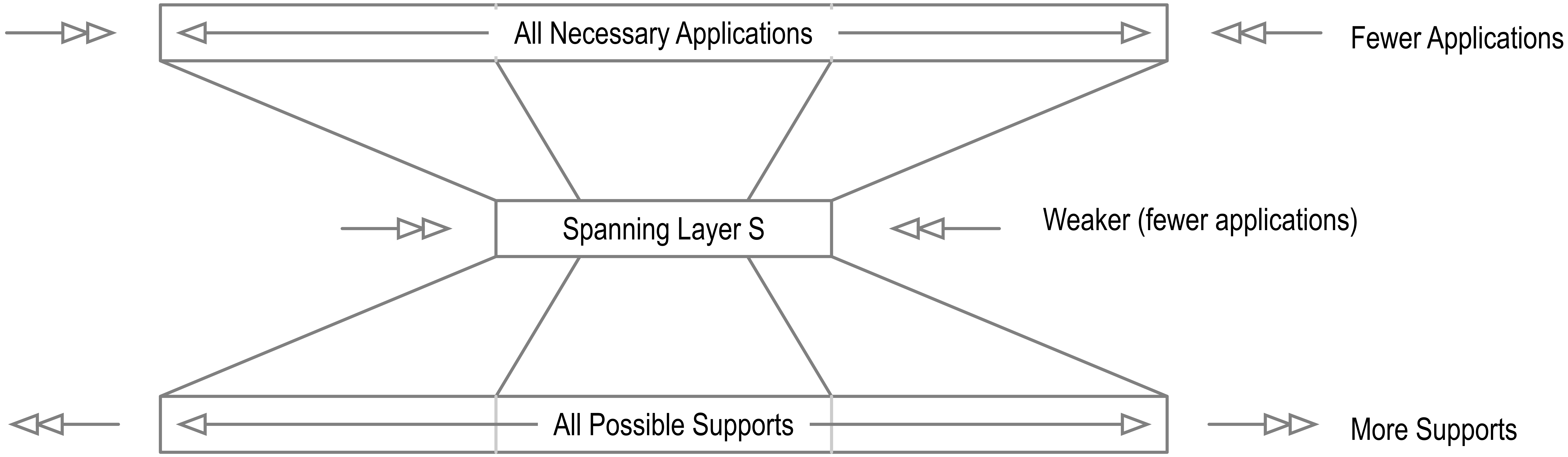
Spanning Layer





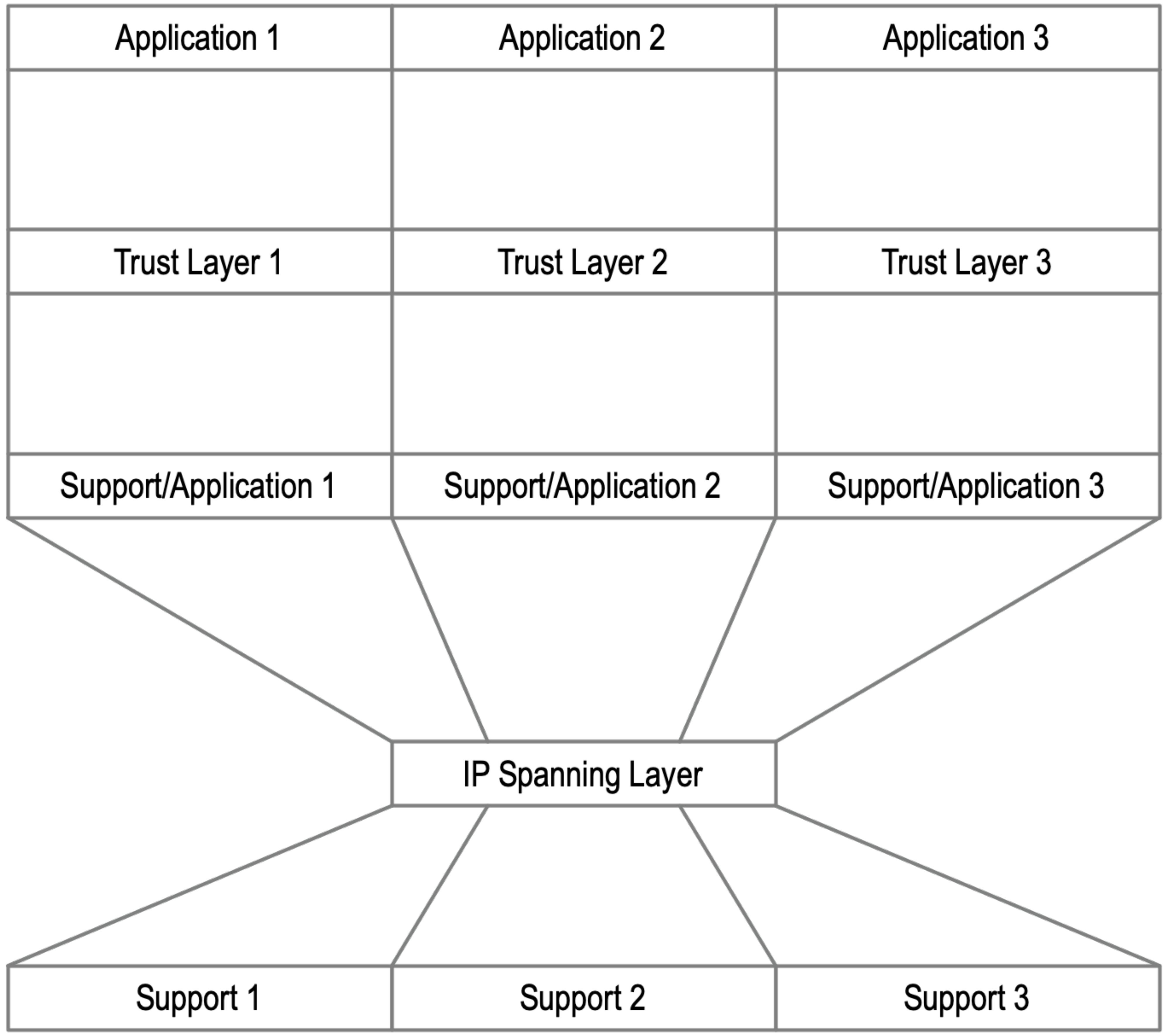
https://web.archive.org/web/20050415042854/http://www.csd.uch.gr/~hy490-05/lectures/Clark_interoperation.htm

Hourglass

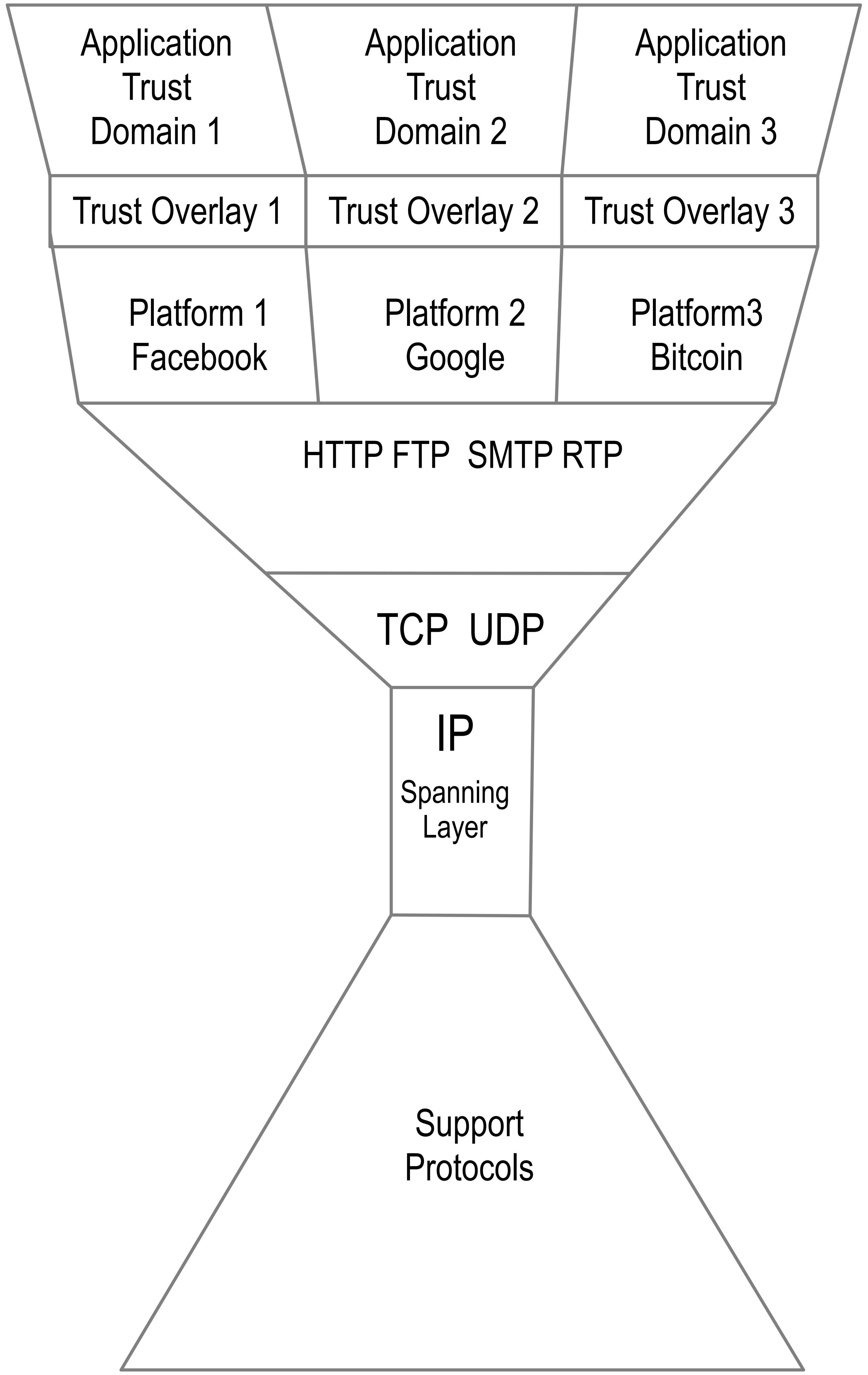


https://cacm.acm.org/magazines/2019/7/237714-on-the-hourglass-model/fulltext

Platform Locked Trust

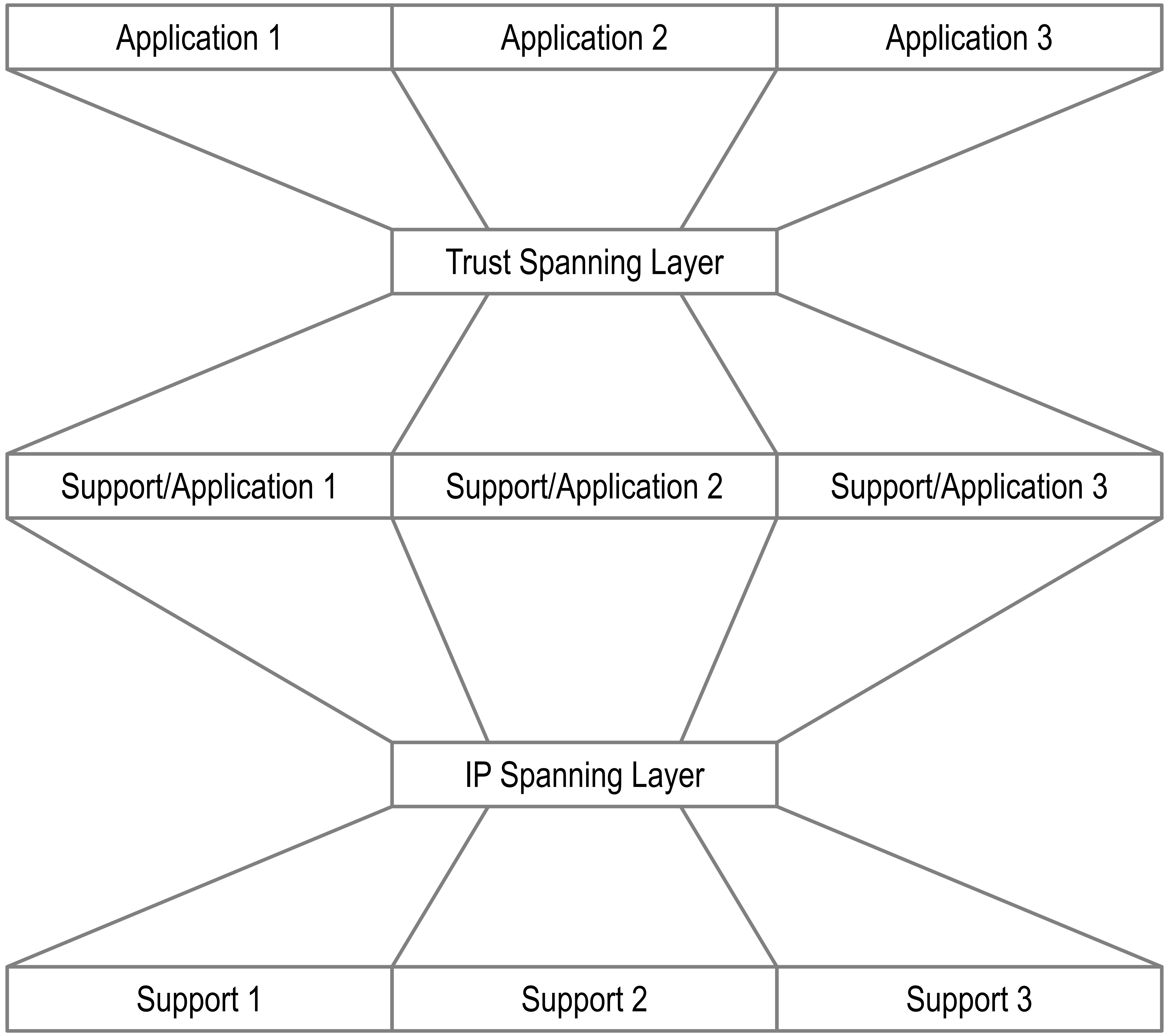


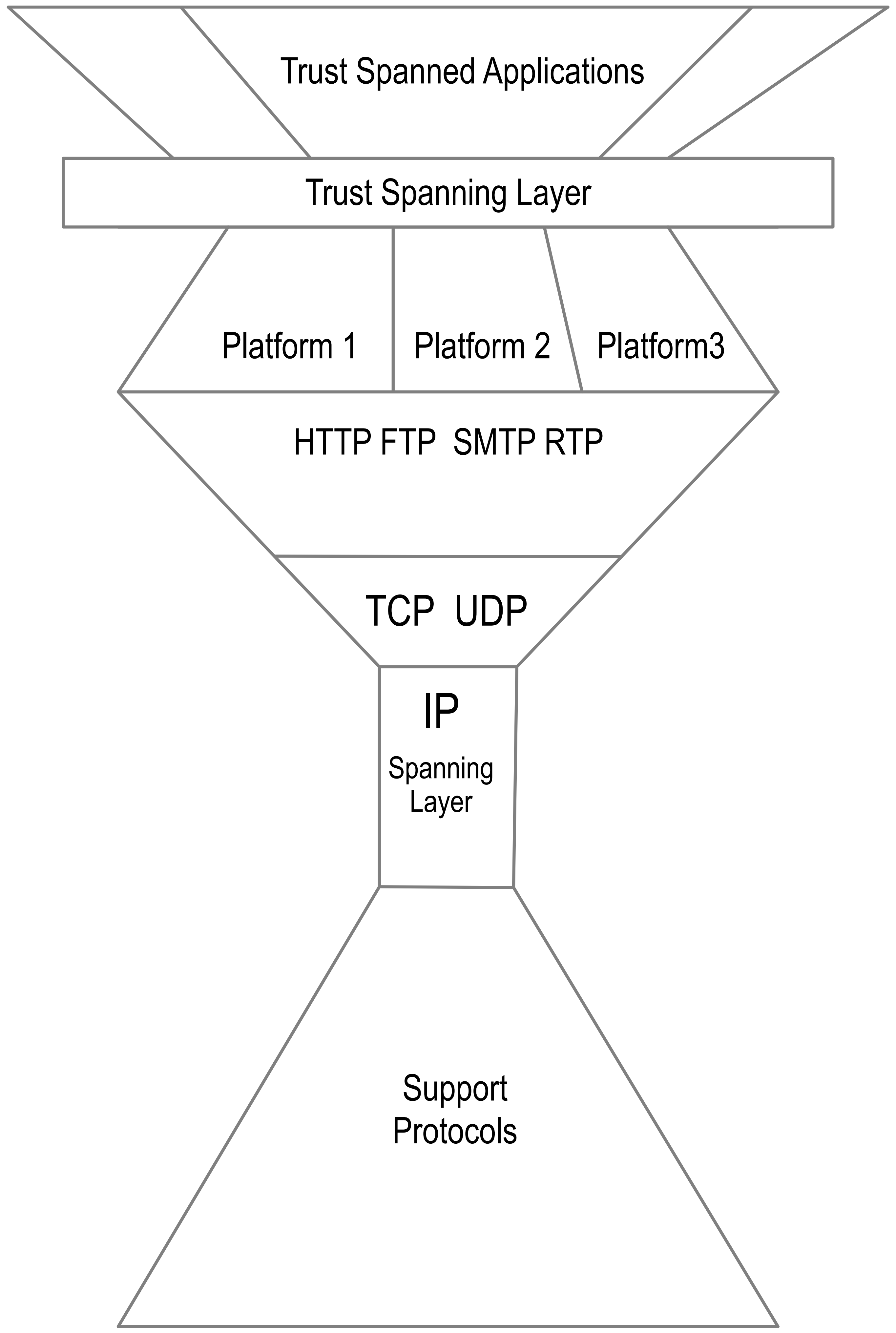
Trust Domain Based Segmentation



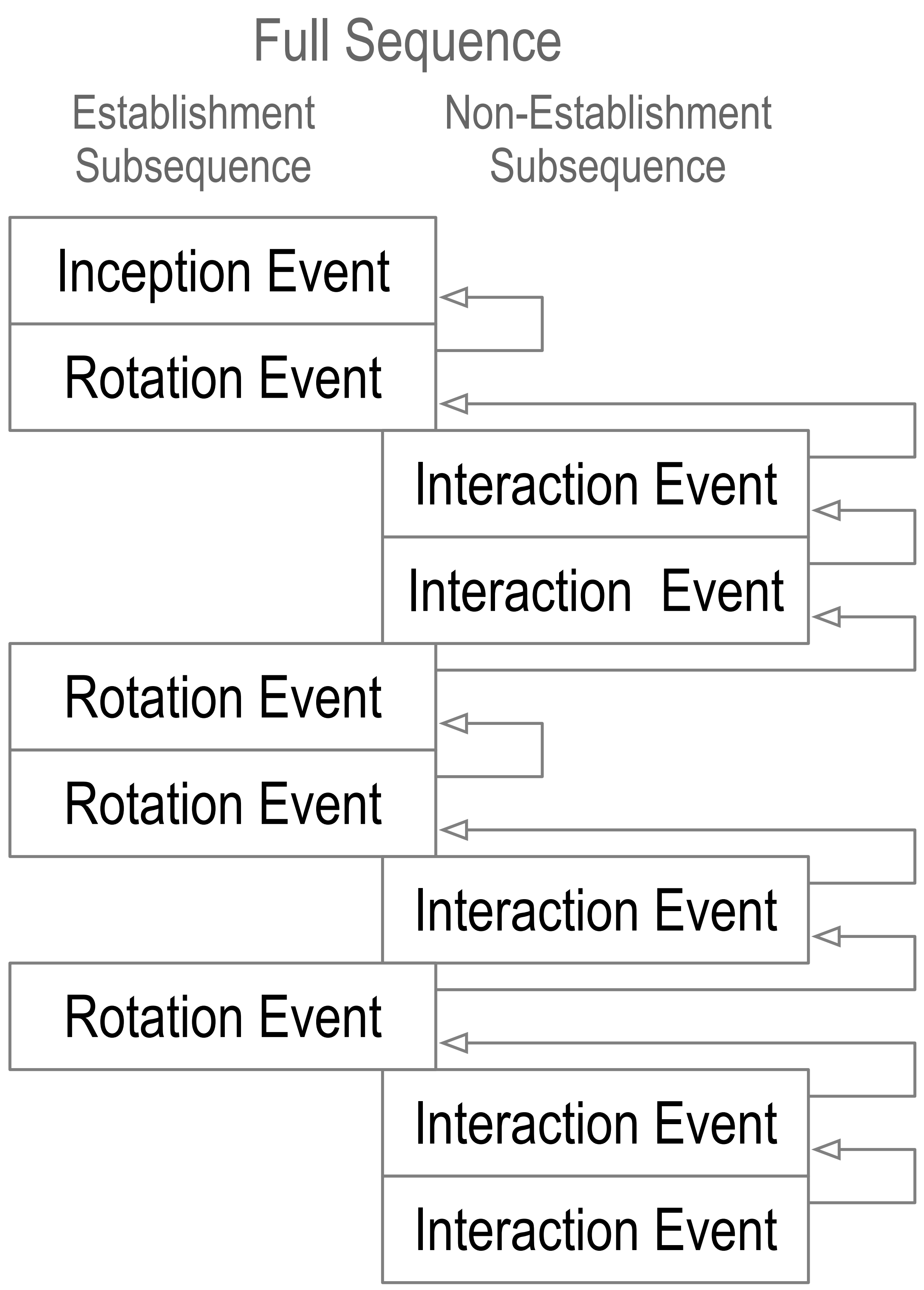
Each trust layer only spans platform specific applications
Bifurcated internet trust map
No *spanning* trust layer

Solution: Waist and Neck





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 self-certifying root-of-trust protects against internal inconsistency.

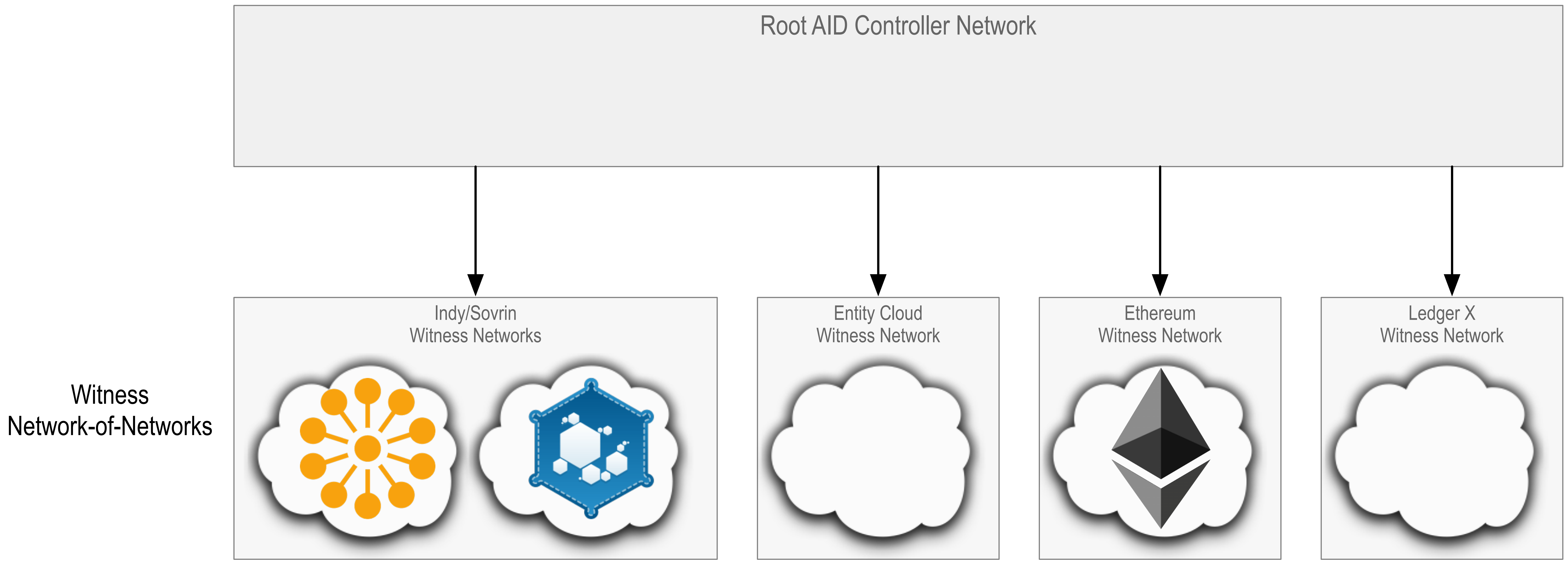
Externally inconsistent log with a purported copy of log but both verifiable = duplicitous.

Duplicity detection protects against external inconsistency.

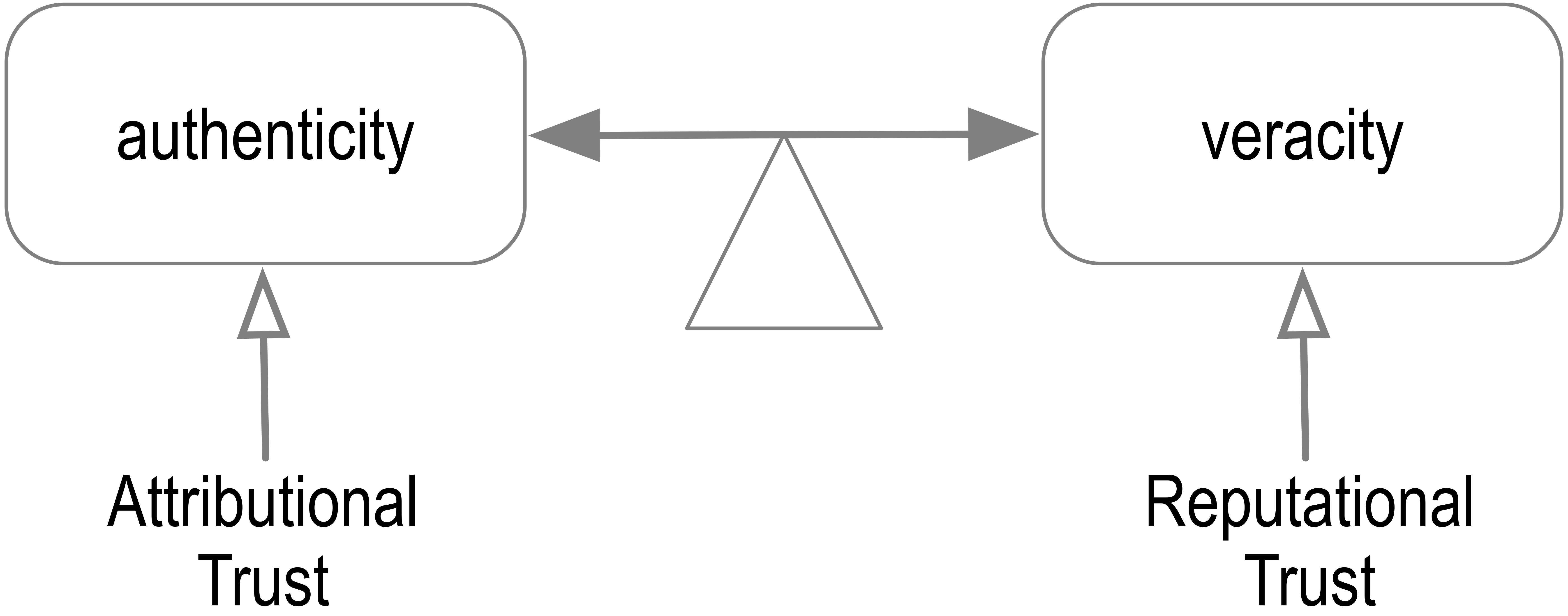
KERI provides duplicity evident DKMI

To Learn More About KERI.

https://keri.one


Root AID Controller Network
Indy/Sovrin
Witness Networks
Entity Cloud
Witness Network
Ethereum
Witness Network
Ledger X
Witness Network
Witness
Network-of-Networks

Trust Balance



Cryptographic

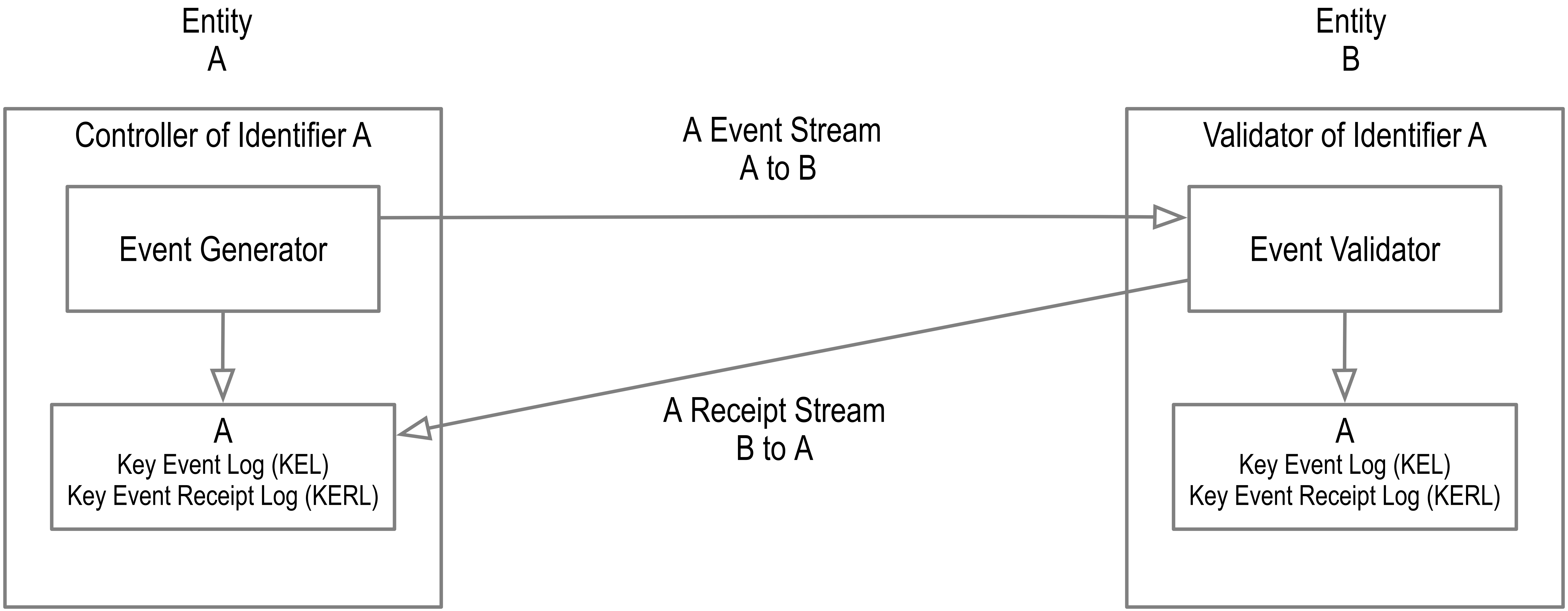
Behavioral

Protocol Operational Modes

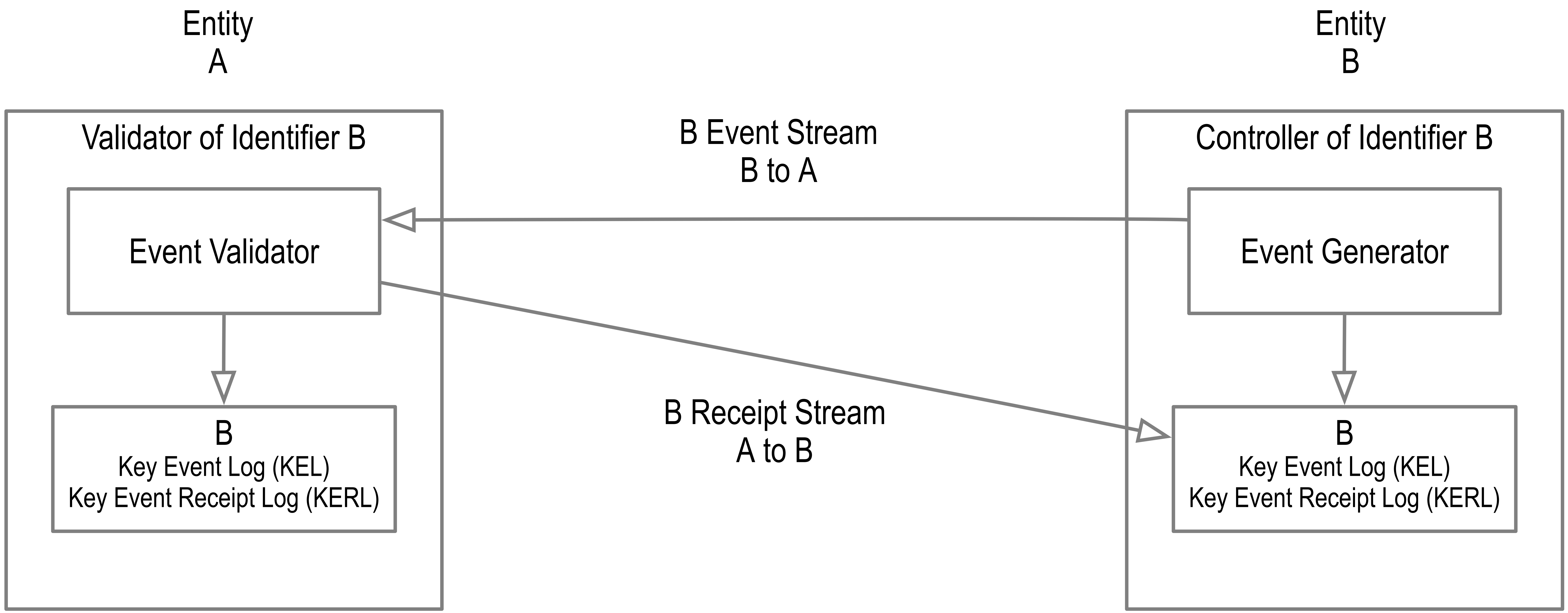
Direct Event Replay Mode (one-to-one)

Indirect Event Replay Mode (one-to-any)

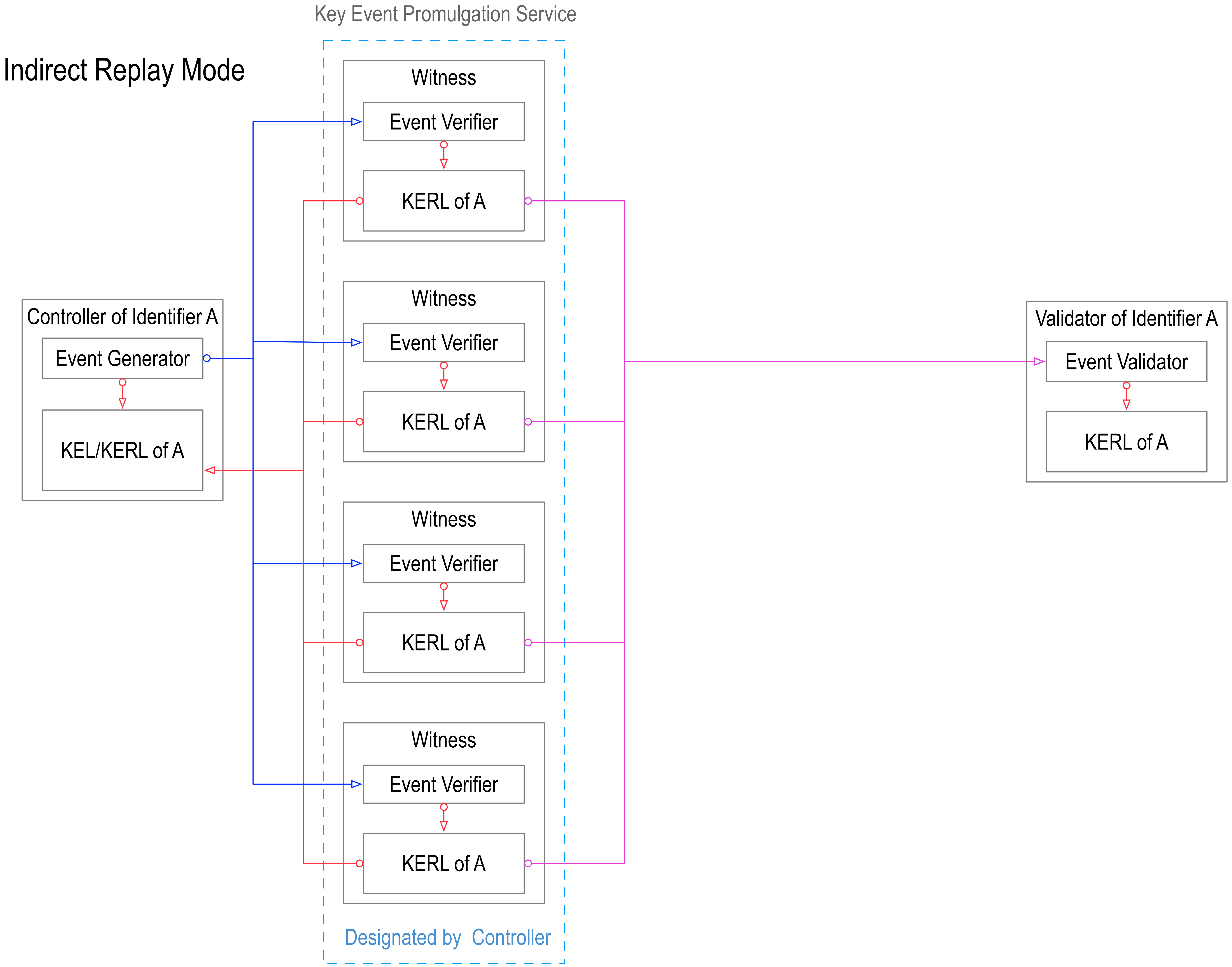
Direct Mode: A to B



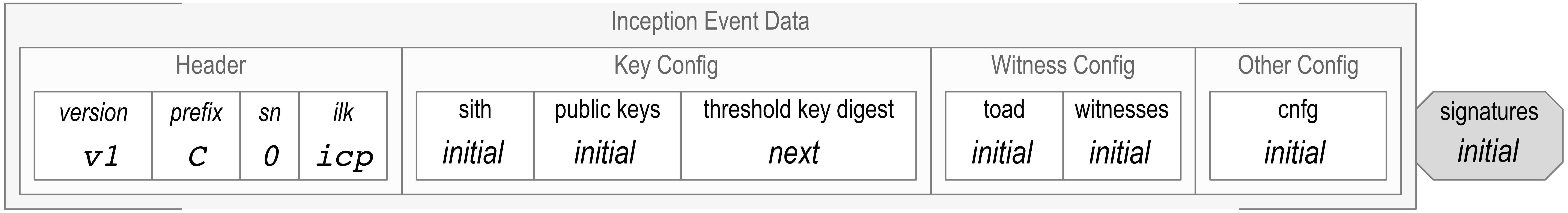
Direct Mode: B to A



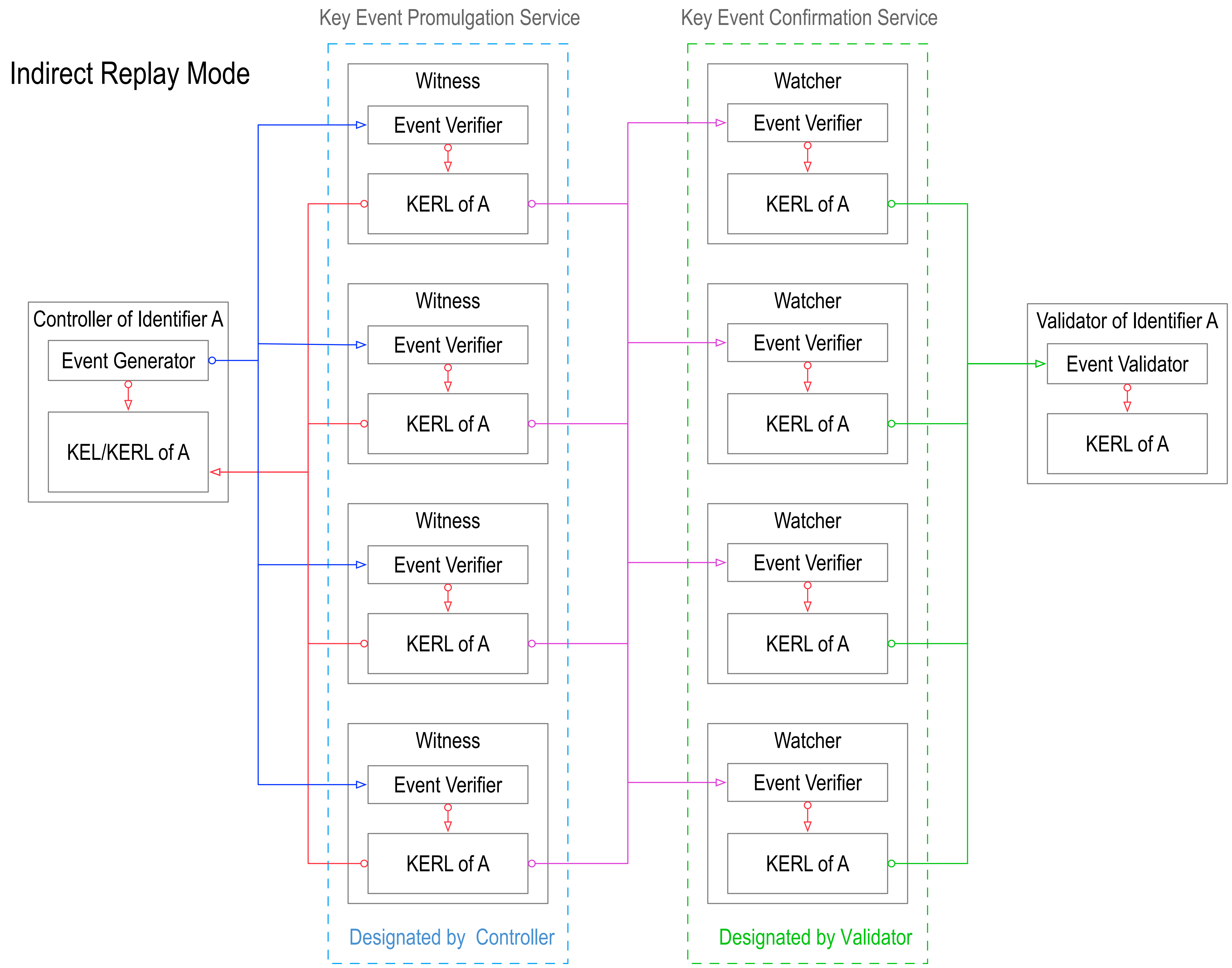
Indirect Mode Promulgation Service



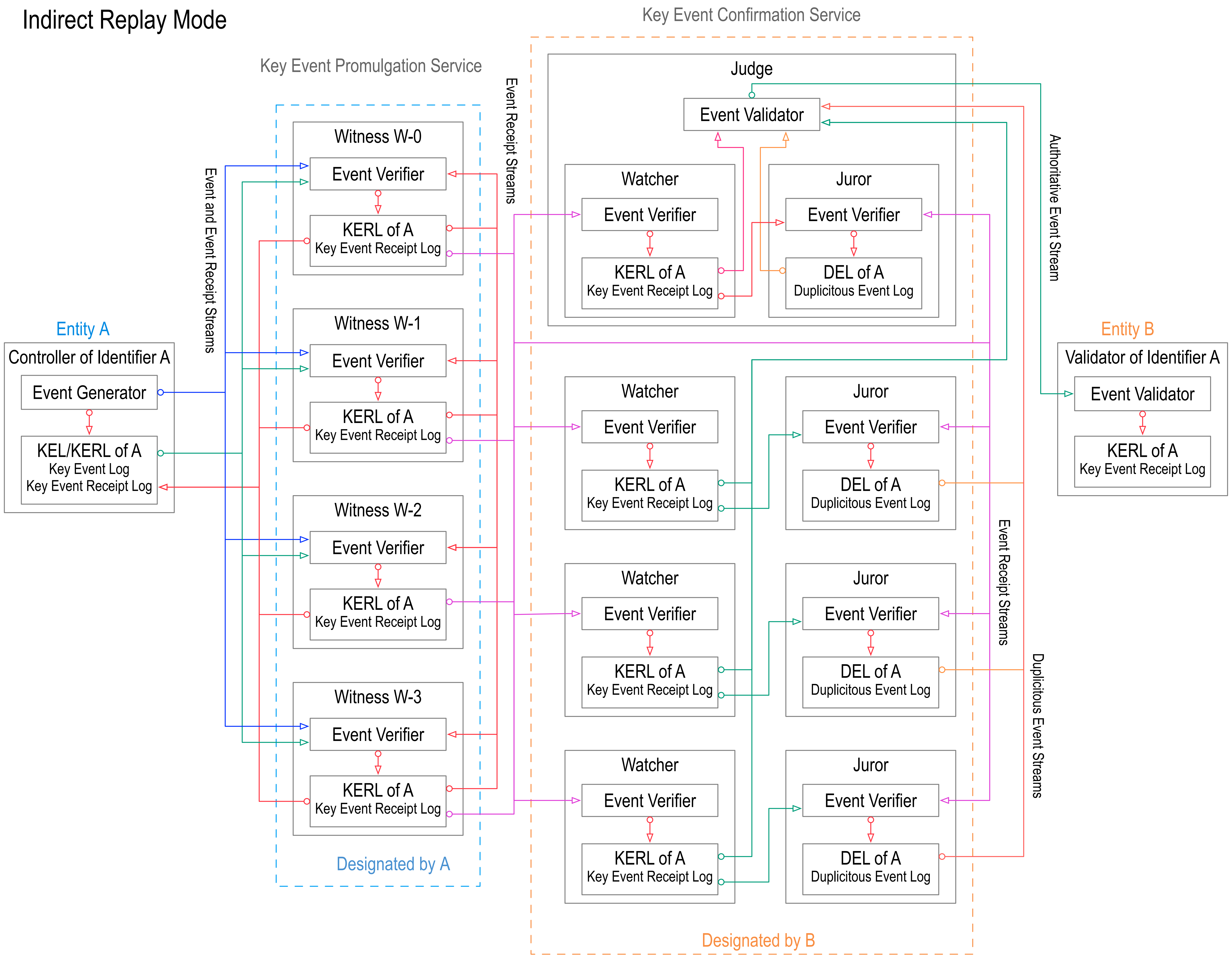
Establishment Events



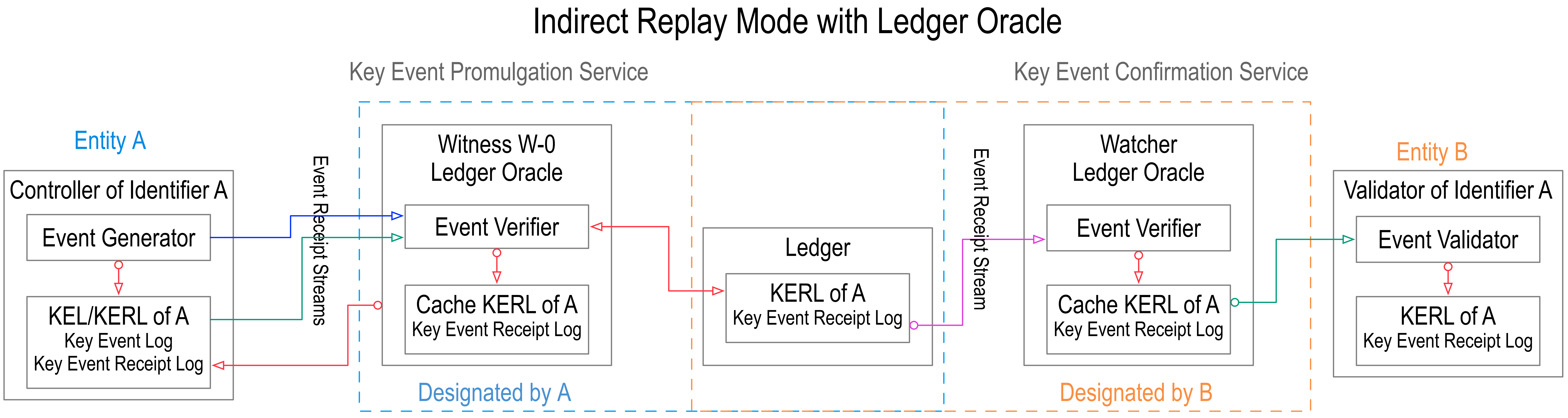
Indirect Mode
Promulgation and Confirmation Services



Indirect Mode Full



Indirect Mode with Ledger Oracles



Separation of Control

Shared ledger = *shared control* over *shared data*.

Shared *data* = *good*, shared *control* = *bad*.

Shared control between controllers and validators may be problematic for governance, scalability, and performance.

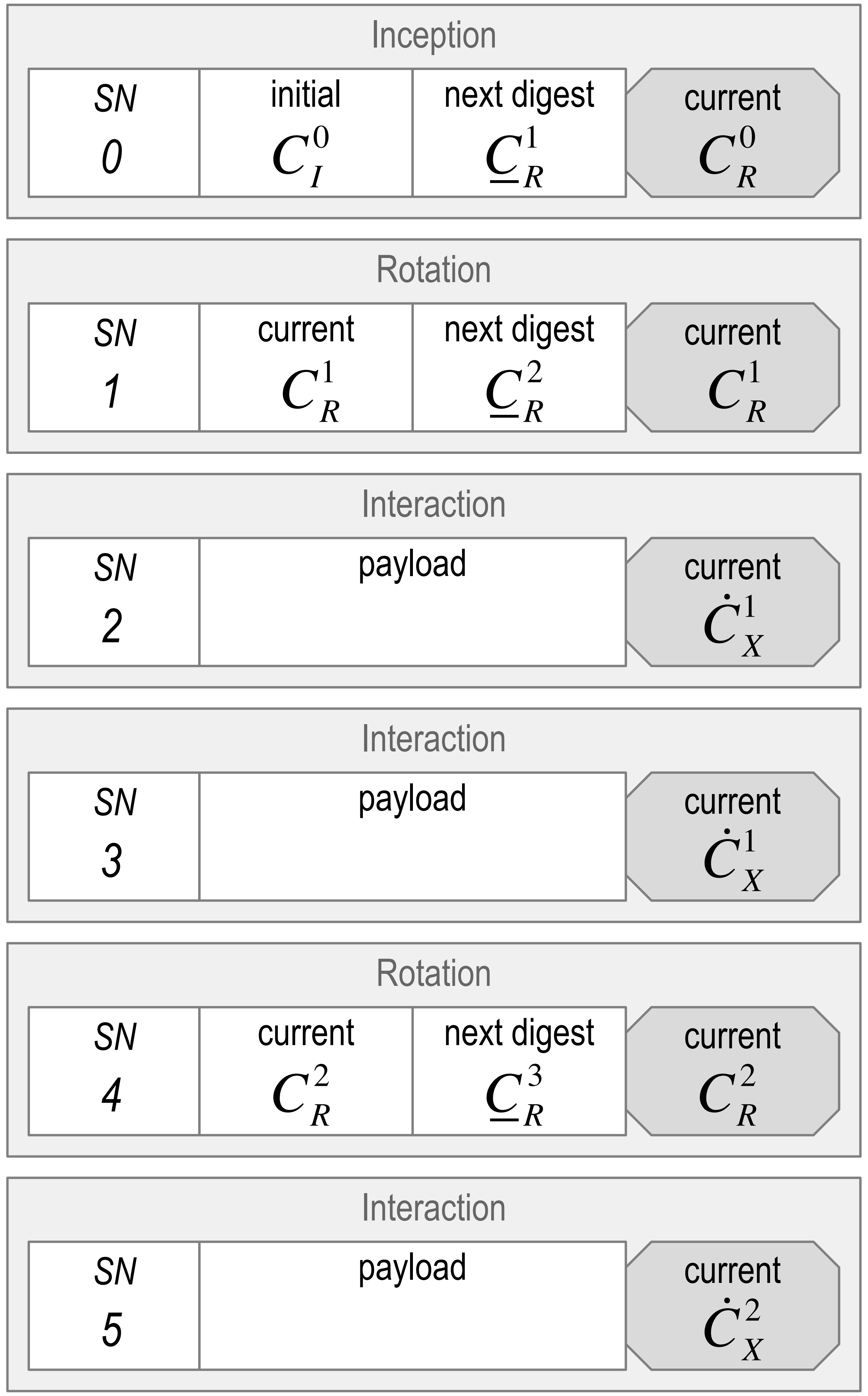
KERI = *separated control* over *shared data*.

Separated control between controllers and validators may provide better decentralization, more flexibility, better scalability, lower cost, higher performance, and more privacy at comparable security.

Live Exploit (current signing keys)

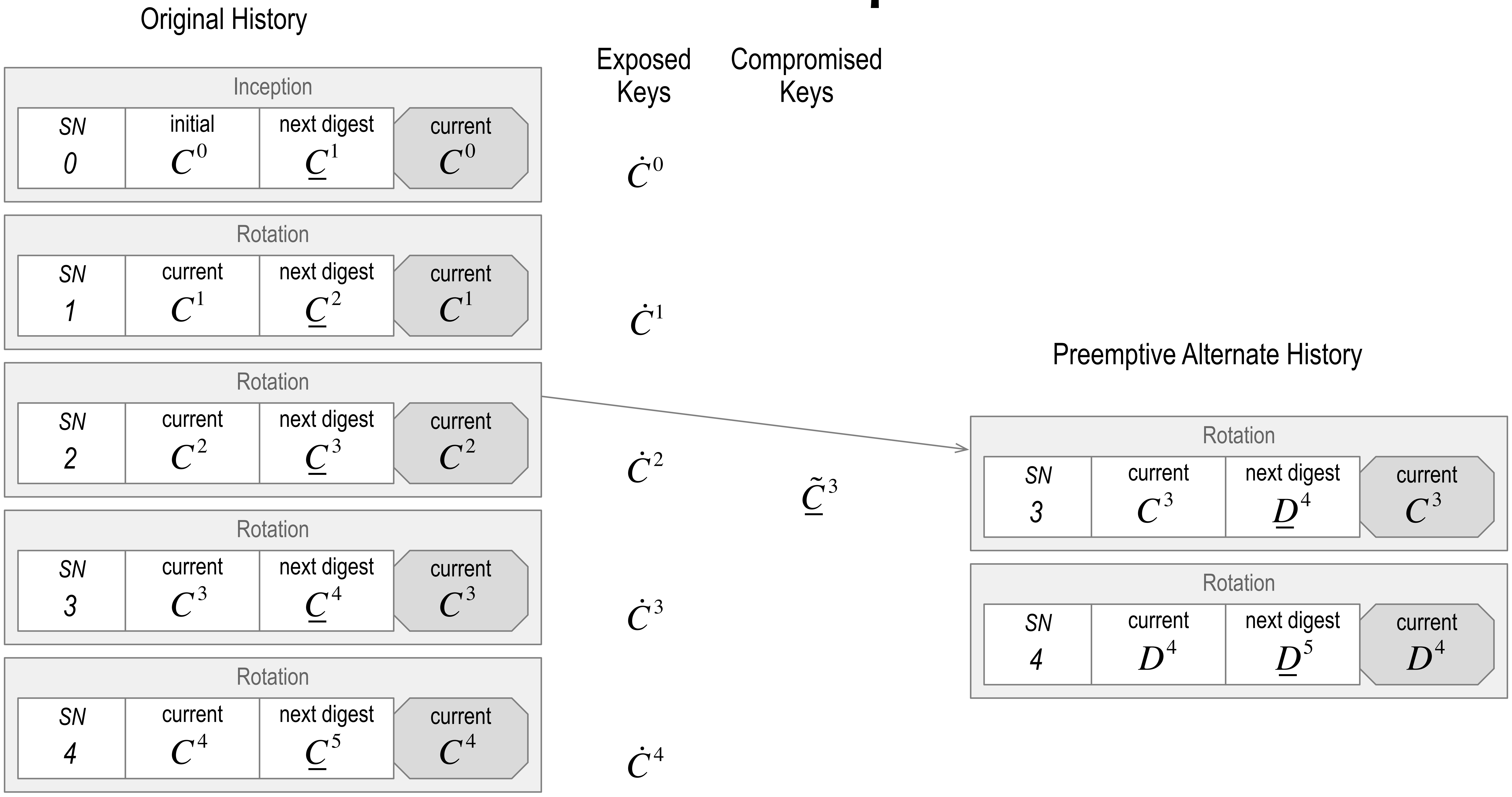
Hard Problem:

Recovery from Live Exploit of Current Signing Keys



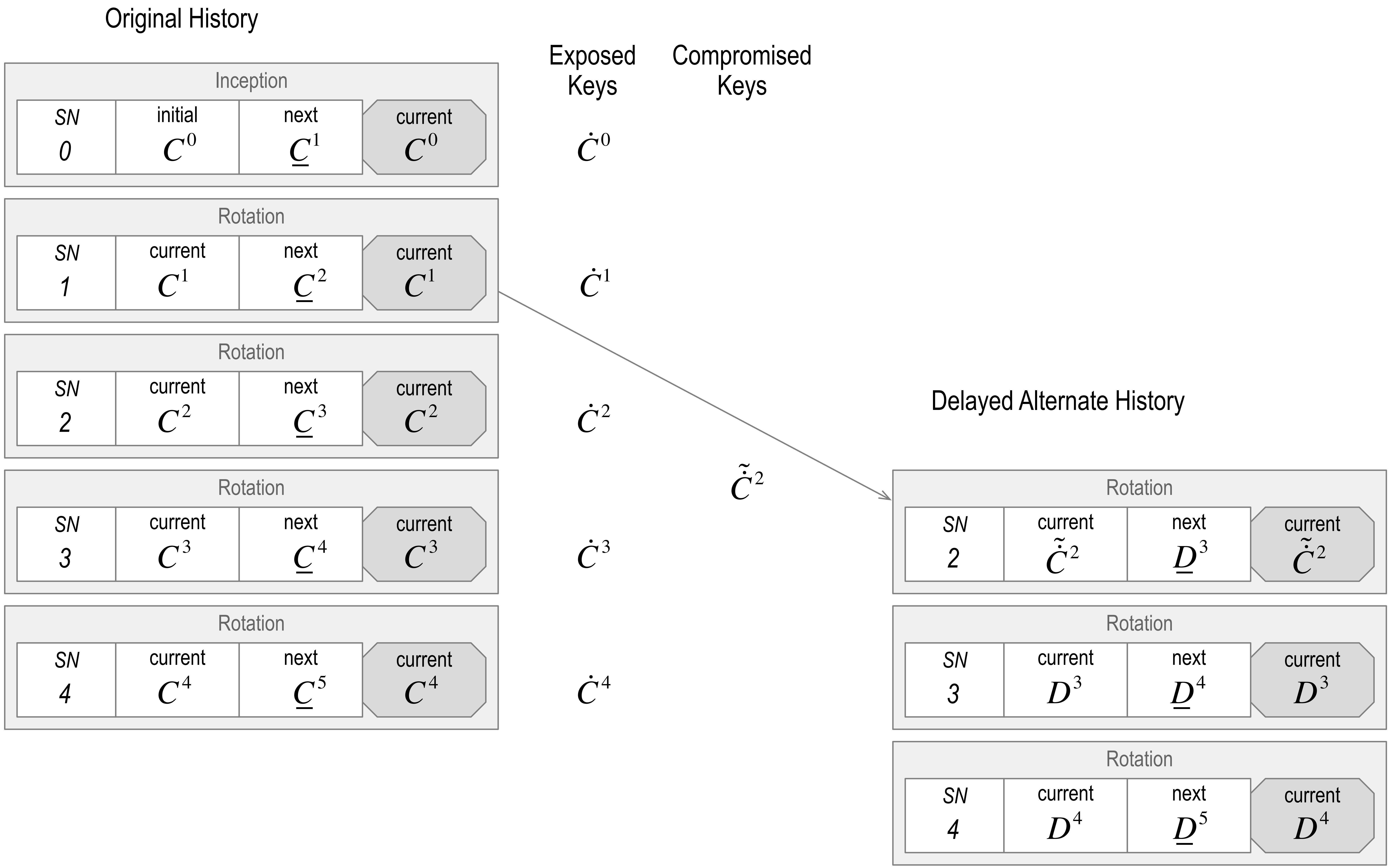
Pre-rotation provides protection from successful *live* exploit of current signing keys.

Live Exploit (next signing keys)



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

Dead Exploit (stale next signing keys)

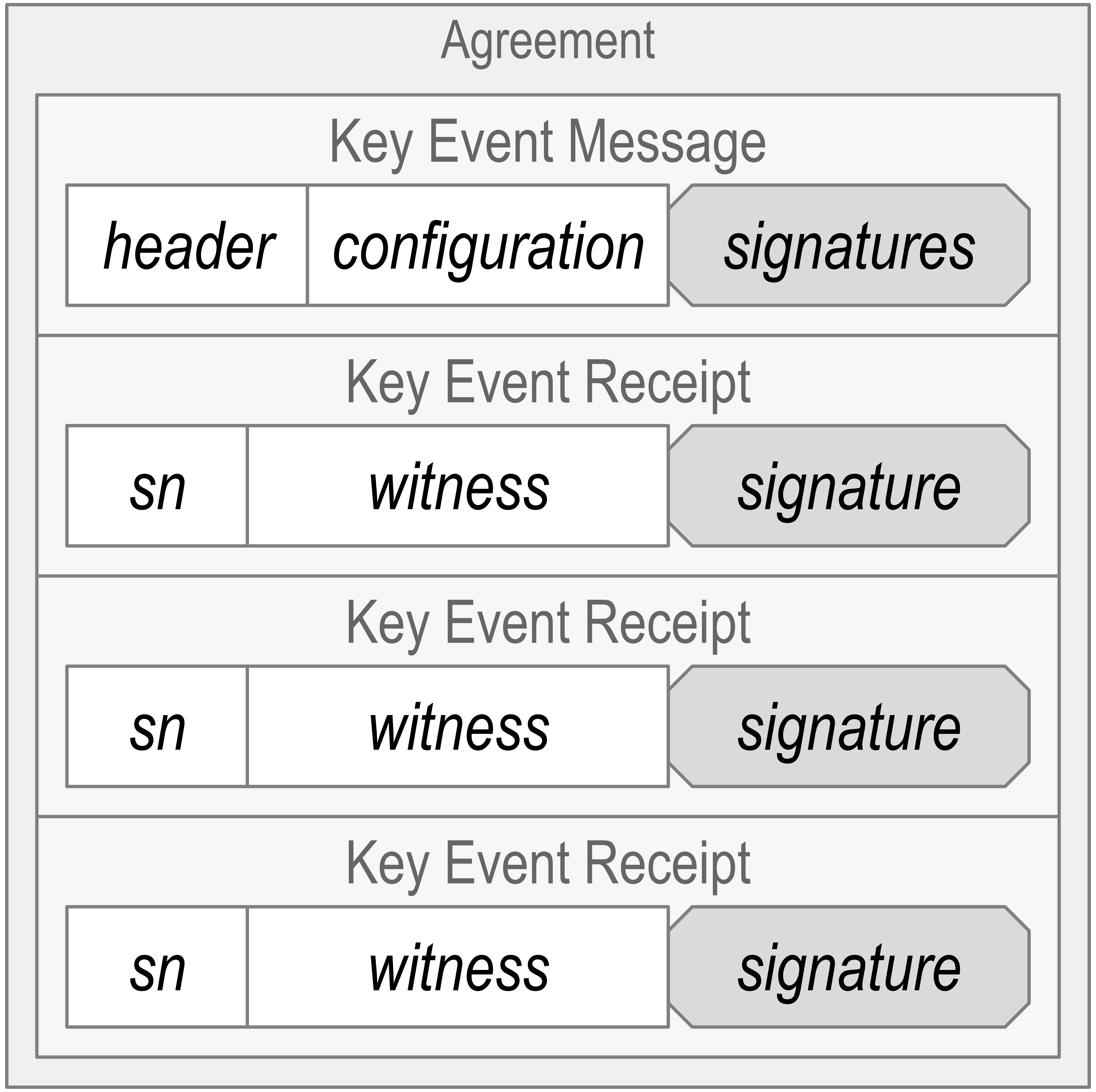


Any copy of original history protects against successful *dead* exploit: First Seen Wins

(KA²CE)

Keri’s Agreement Algorithm for Control Establishment

Produce Witnessed
Agreements
with Guarantees



Witnessing Rules

An honest witness will only *witness*, (i.e. create, store, and promulgate a receipt for), at most *one and only one version* of any event.
That event version must first be verified.
A verified event version must be signed by the controller's authoritative keys as determined by prior events.
A verified event version must be consistent with all prior events.

Agreement

A *state of agreement* about a version of an event is defined with respect to *set* of witnesses in agreement:
Each witness in that *set* has witnessed the same version of that event and each receipt in that set has been promulgated to every other witness in that *set*.
The size of an agreement is the number of contributing witnesses in the *set*.
The associated *agreement* include a receipt from each witness in the *set*.
This state of agreement is provable to any validator, watcher, juror, or judge via a verifiable fully receipted copy of the event i.e the *agreement*.
This copy provides *proof of agreement*.
Such a proof may be obtained via any verifiable KERL that includes that version of that event.

Definitions

$N =$ number of witness

$M =$ size of agreement

$F =$ faulty witnesses

V *Validator*

C *Controller*

Threshold of Accountable Duplicity

TOAD

M_C

M_V

Sufficient Agreement

Controller's Guarantee

$$M \geq M_C$$

Validator's Choice

$$M_V \geq M_C$$

Algorithm Objectives

Any pre-existing copy or digest of original KERL available to Validator protects Validator from future dead exploits.

KAACE provides fault tolerance from live exploit.

A successful but recoverable live exploit is a compromise of the controller's current signing keys and/or a compromise of the witnesses' signing keys.

A) WRT Controller, a successful live exploit of the witnesses' would prevent sufficient agreement thereby inducing a recovery operation.

B) WRT Validator, a successful live exploit would produce undetectably duplicitous but sufficient agreement about current events.

(KAACE immune agreement prevents this, i.e. Validator is immune)

Detectable vs Undetectable Duplicity

Witness Duplicity

Witness Duplicity is Detectable.

Controller Duplicity

Controller Duplicity wrt witnesses is undetectable if a sufficient number of witnesses are not duplicitous and sufficient agreement is small enough.

Agreement Constraints

$N =$ number of witness

$M =$ size of agreement

$F =$ faulty witnesses

V *Validator*

C *Controller*

Proper Agreement

$$F + 1$$

Bounds on Sufficient Agreement

$$M > F$$

$$M \leq N - F$$

$$F < M \leq N - F$$

Intact Agreement

$$N \geq 2F + 1$$

One Agreement or None at All (*Immune*)

first seen rule limits liveness induces recovery rotation

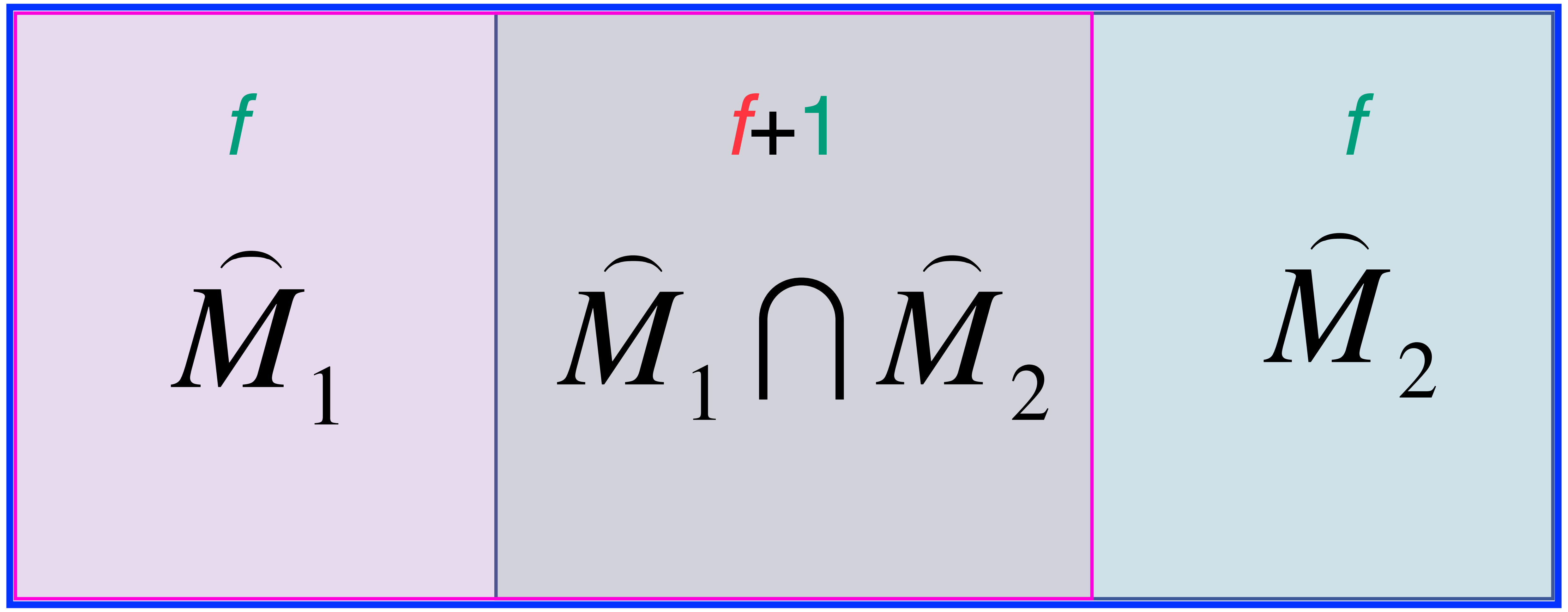
$$\left|\widehat{N}\right| = N \qquad \left|\widehat{M}_1\right| = \left|\widehat{M}_2\right| = M$$

$$\left|\widehat{M}_1 \cup \widehat{M}_2\right| = \left|\widehat{N}\right| = N$$

$$\left|\widehat{M}_1\right| + \left|\widehat{M}_2\right| = \left|\widehat{M}_1 \cup \widehat{M}_2\right| + \left|\widehat{M}_1 \cap \widehat{M}_2\right|$$

Overlapping Sets

$$\widehat{M}_1 \cup \widehat{M}_2 = \widehat{N}$$



$$2M = N + F + 1$$

$$M \geq \left\lceil \frac{N + F + 1}{2} \right\rceil$$

$$M \leq N - F$$

One honest witness if:

$$\left|\widehat{M}_1 \cap \widehat{M}_2\right| \geq F + 1$$

Immune Agreement

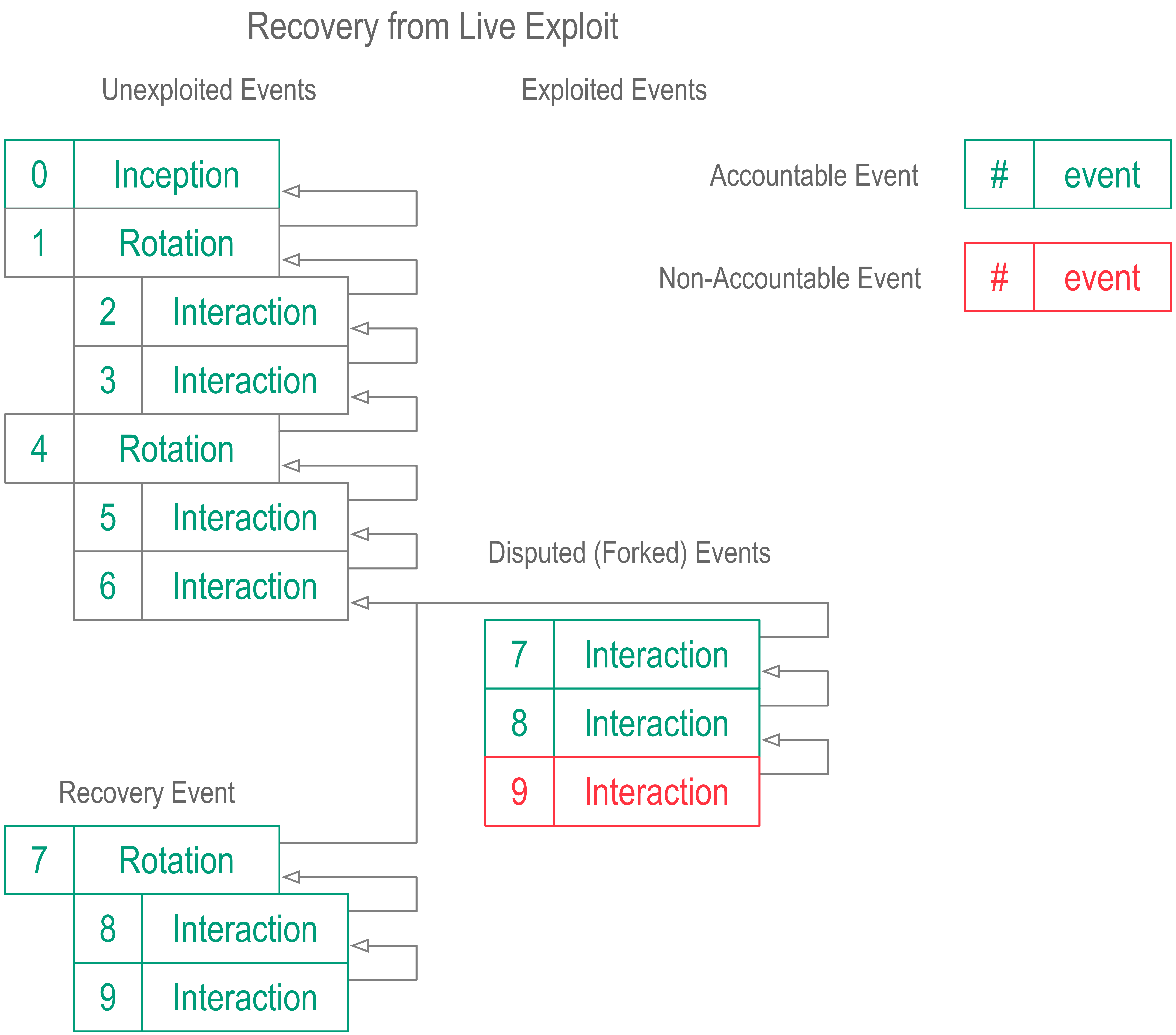
$$\frac{N + F + 1}{2} \leq M \leq N - F$$

Example Values

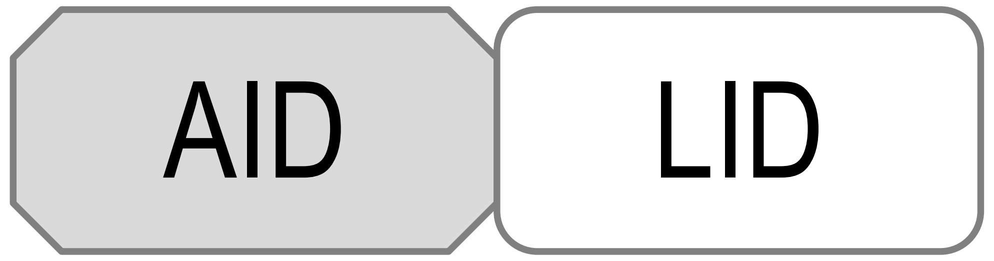
Immunity

F	N	3F+1	$\left\lceil \frac{N+F+1}{2} \right\rceil$	N-F	M
1	4	4	3	3	3
1	5	4	4	4	4
1	6	4	4	5	4, 5
1	7	4	5	6	5, 6
1	8	4	5	7	5, 6, 7
1	9	4	6	8	6, 7, 8
2	7	7	5	5	5
2	8	7	6	6	6
2	9	7	6	7	6, 7
2	10	7	7	8	7, 8
2	11	7	7	9	7, 8, 9
2	12	7	8	10	8, 9, 10
3	10	10	7	7	7
3	11	10	8	8	8
3	12	10	8	9	8, 9
3	13	10	9	10	9, 10
3	14	10	9	11	9, 10, 11
3	15	10	10	12	10, 11, 12

Recovery from Live Exploit Of Current Signing Keys



Unified Identifier Model



AID: Autonomic Identifier (primary)

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

secure, decentralized, portable, universally unique

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

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

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

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

AID|LID Couplet

625.127C125r

EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148|625.127C125r

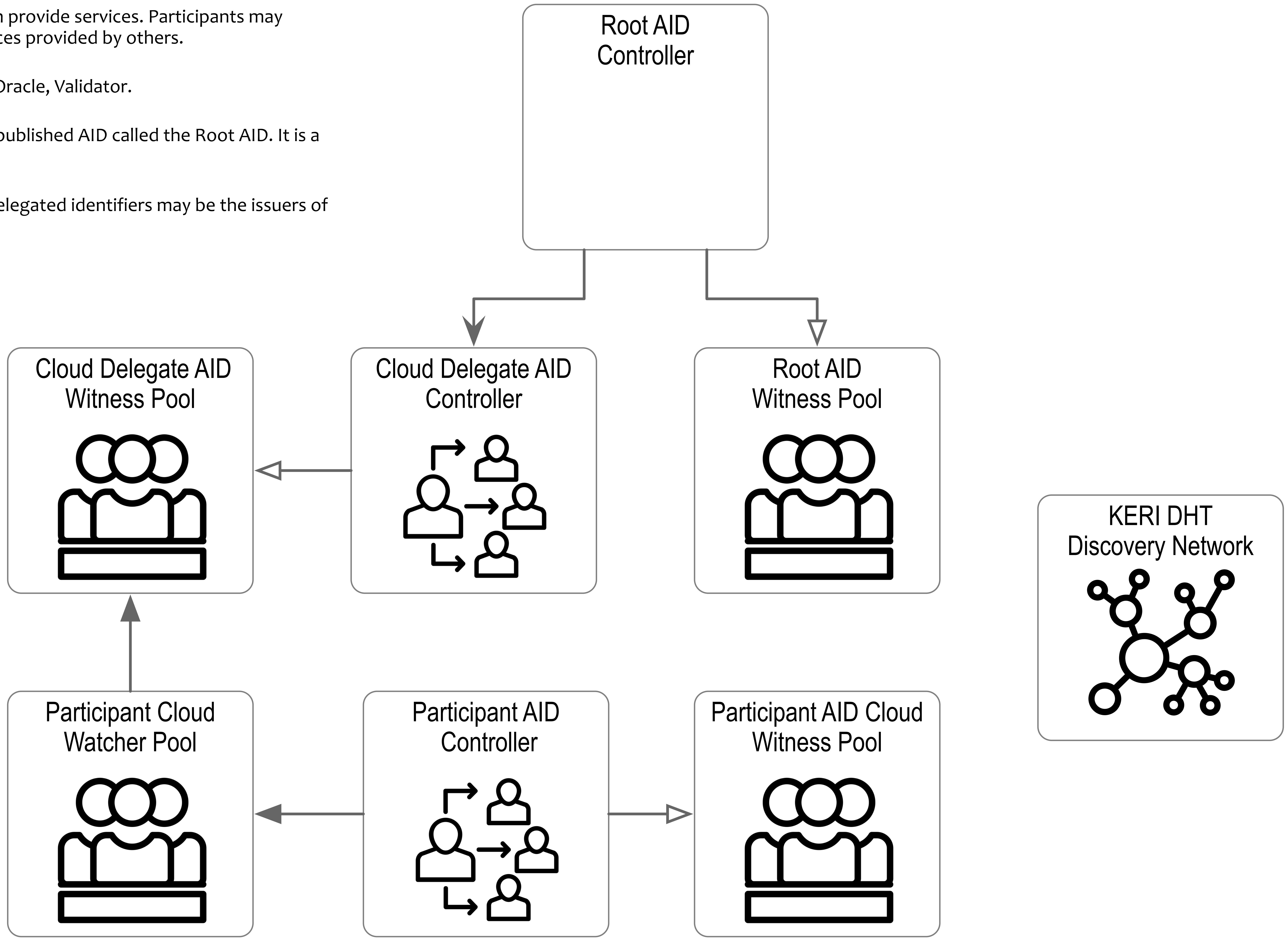
Basic KERI Stack

KERI employs a modular architecture with modular components that each provide services. Participants may configure their stacks to provide some of all of the services or share services provided by others.

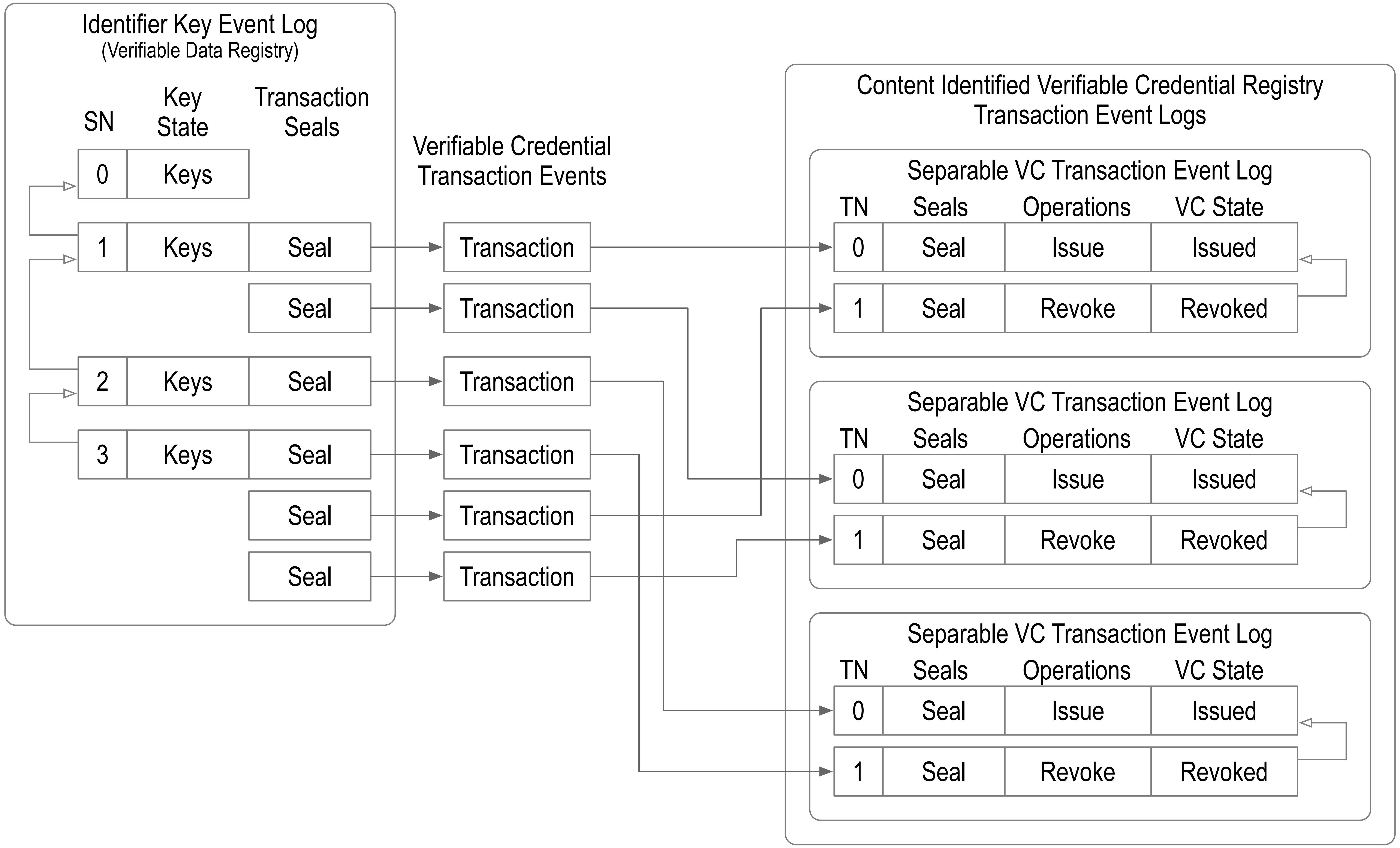
The component services include Controller, Witness, Watcher, Delegate, Oracle, Validator.

The root-of-trust for the GLEIF ecosystem is provided by a single globally published AID called the Root AID. It is a KERI DID.

This Root AID is the issuer of delegations to other KERI AID DIDs. These delegated identifiers may be the issuers of VCs.



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

Qualification testing of the vLEI Beta software

Participating in the sandbox

- Organizations confirmed for the review
  - 8 LEI Issuers
  - 4 external organizations

(additional participation is expected)

- Functionality covered
  - vLEI Credential issuance scenarios (creating vLEIs)
  - vLEI Credential presentation scenarios (using vLEIs)
  - Identifier and Key Management scenarios (ensuring a secure vLEI infrastructure)
  - vLEI Credential revocation scenarios ('retiring' vLEIs)

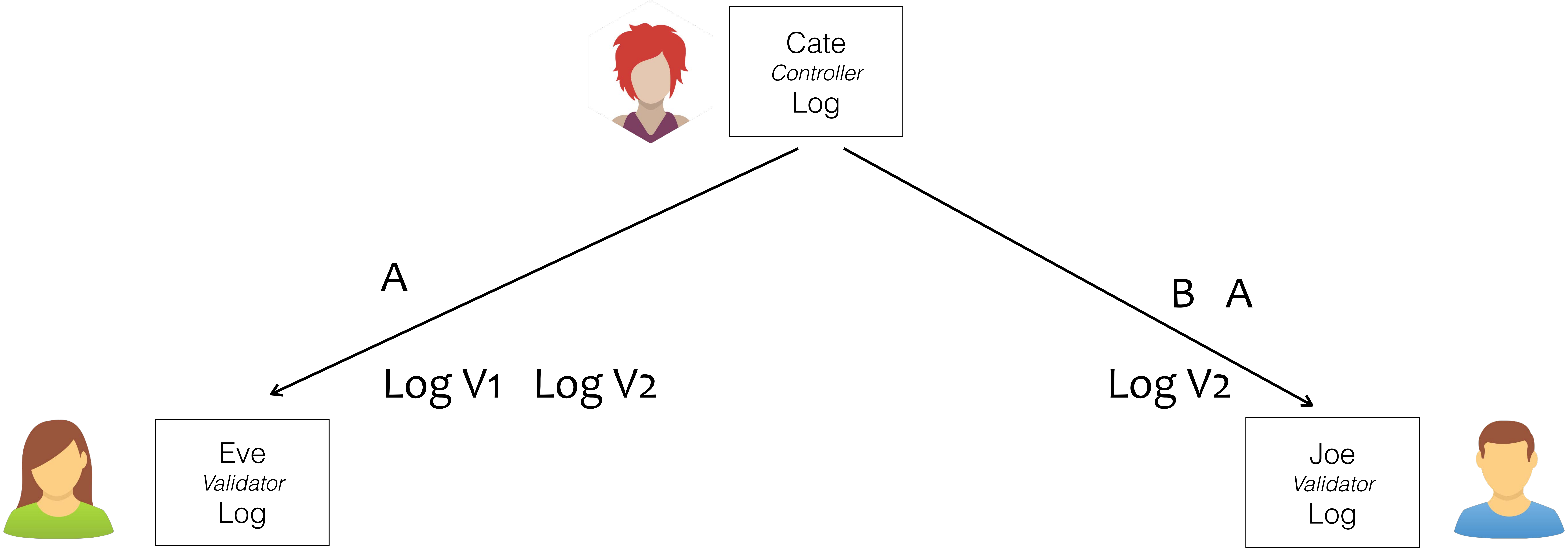
- GLEIF looks forward to the feedback received for GLEIF to consider for incorporation into the version to be used for the vLEI pilots
  - Feedback encouraged until mid-November
  - Sandbox will be in place until year-end 2021

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



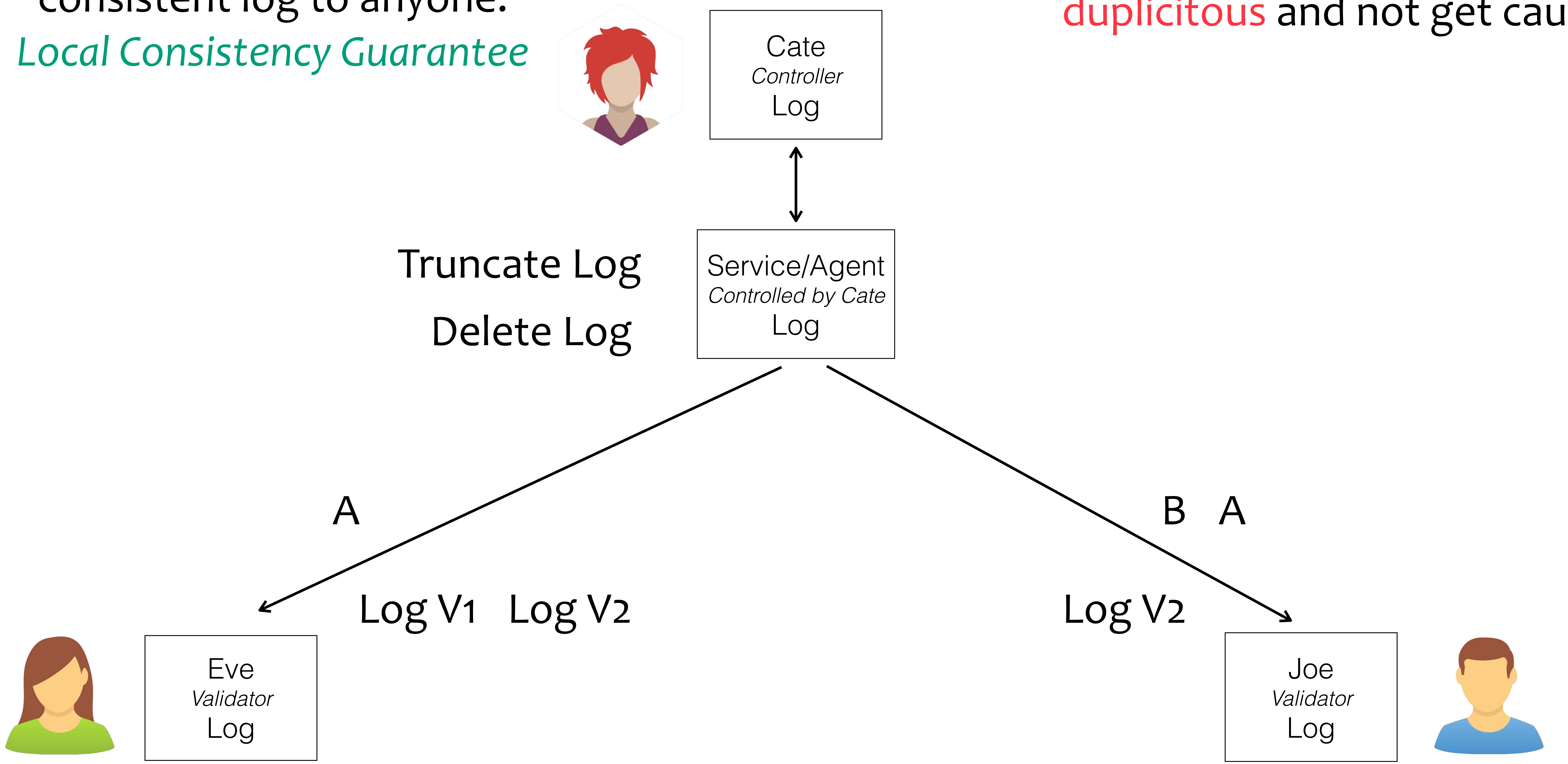
private (one-to-one) interactions

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?



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

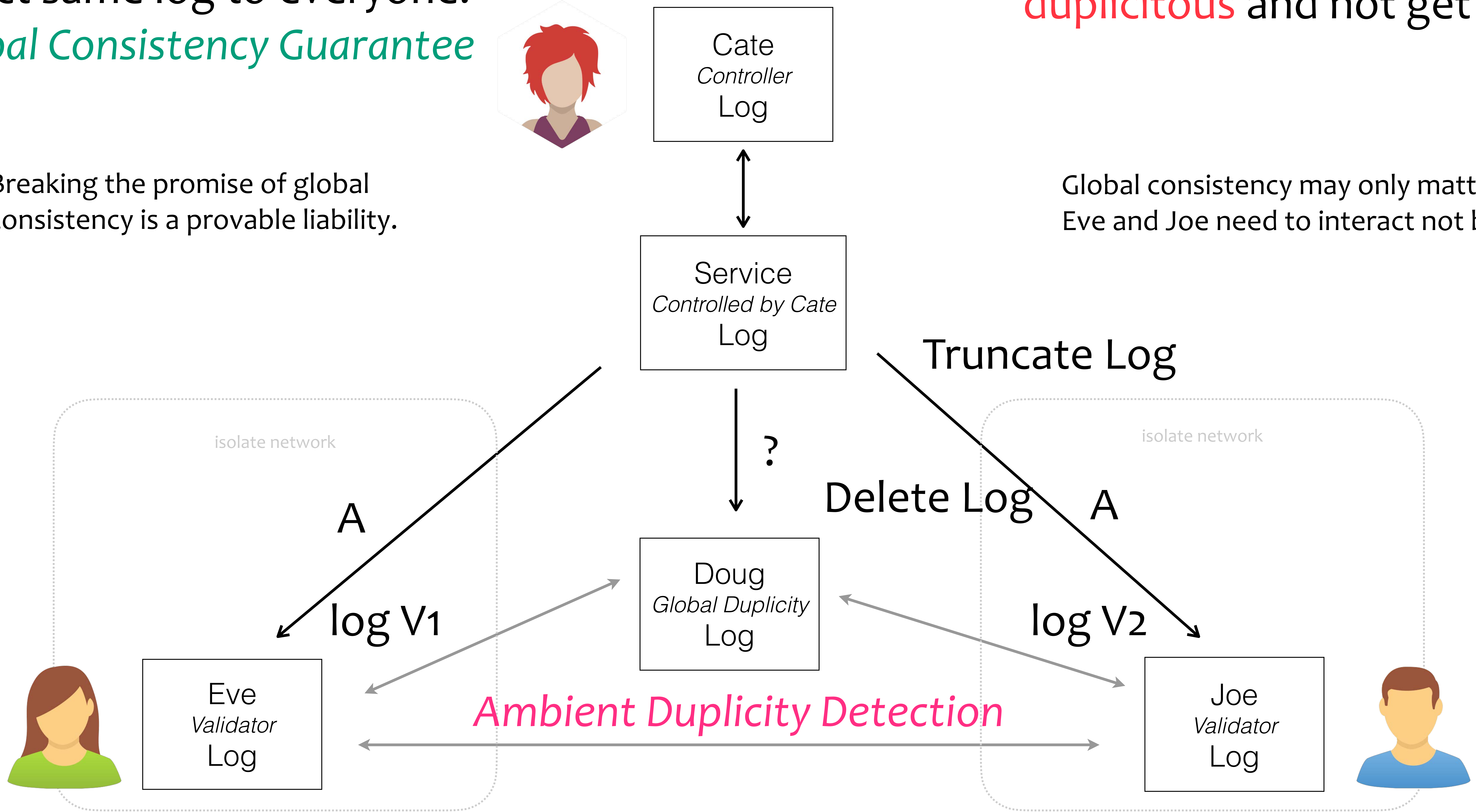
Duplicity Game

Service promises to provide exact same log to everyone.
Global Consistency Guarantee

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

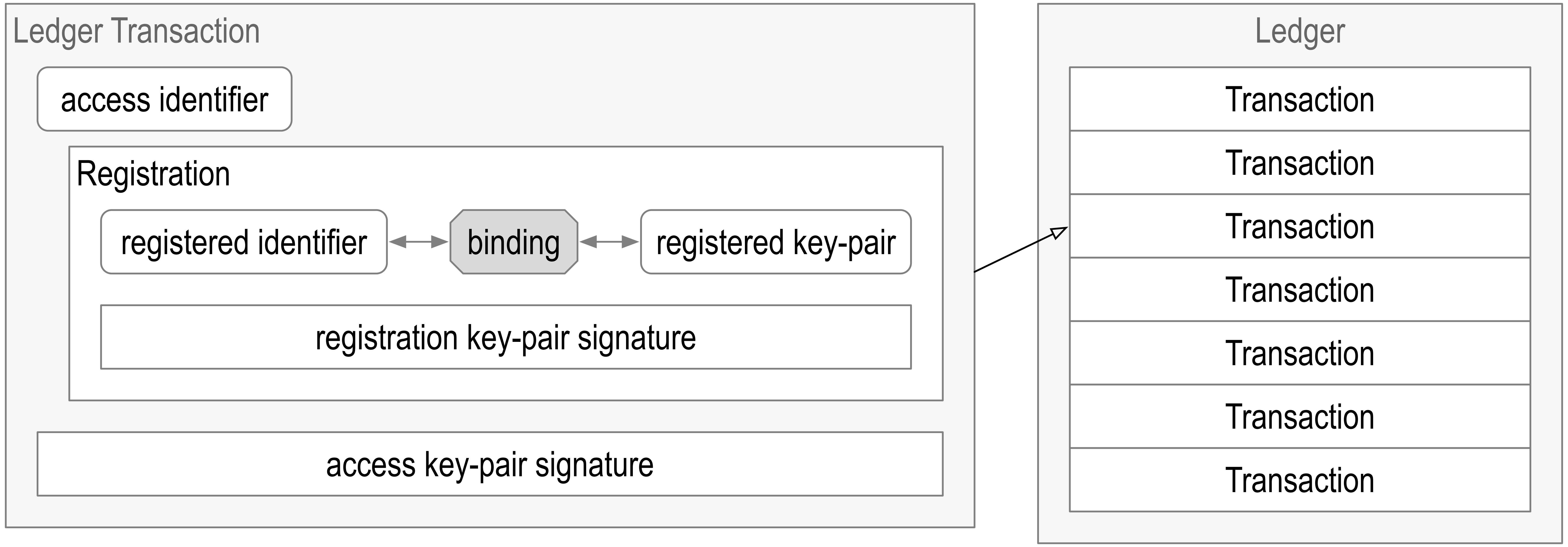
Breaking the promise of global consistency is a provable liability.

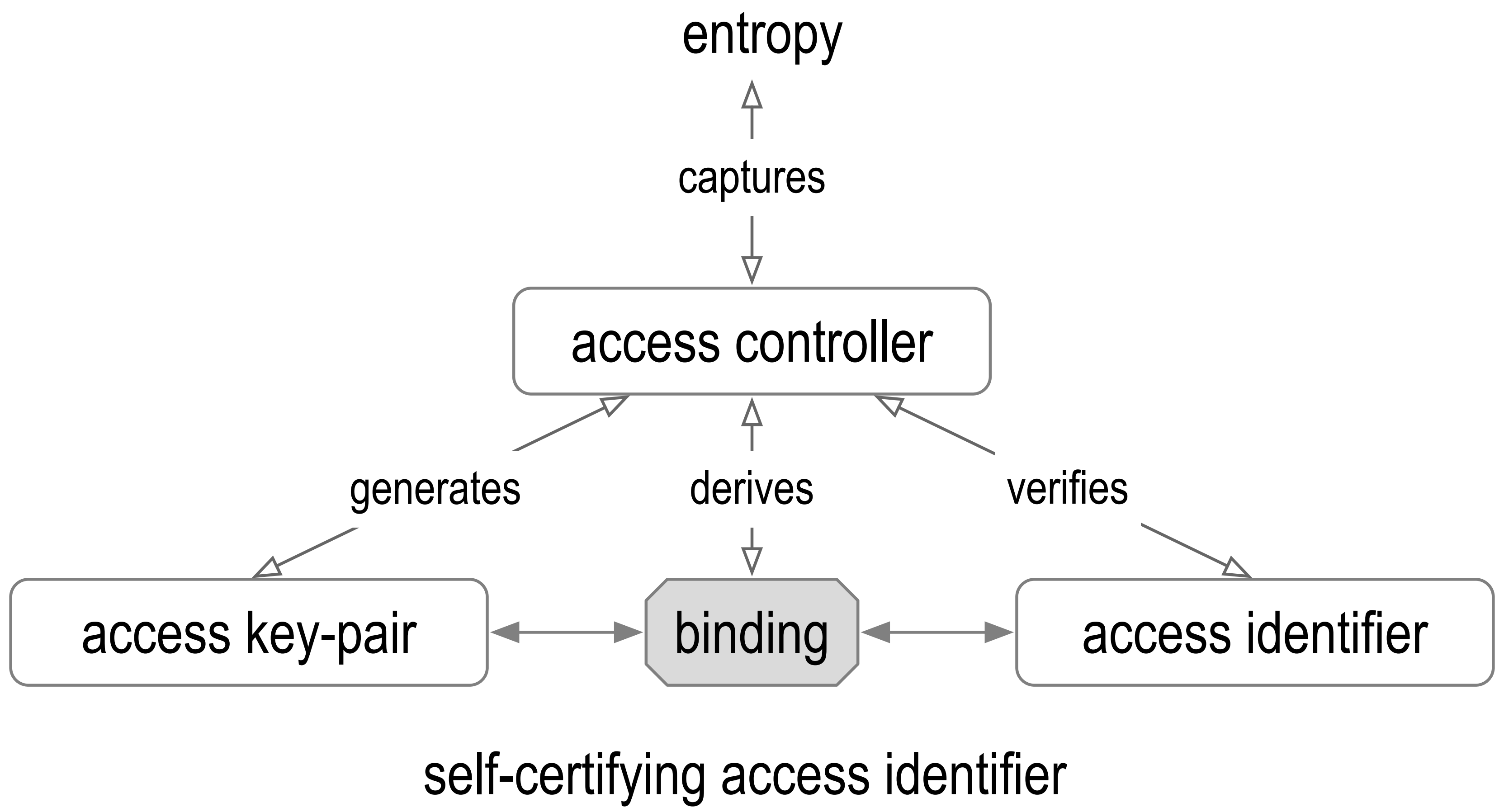
Global consistency may only matter after Eve and Joe need to interact not before.



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

Ledger Registration





The access identifier may have a self-certifying primary root-of-trust, but the registered identifier does not, even if its format appears to be self-certifying.

Autonomic Identifier (AID) and Namespace (AN)

auto nomos = self rule

autonomic = self-governing, self-controlling, etc.

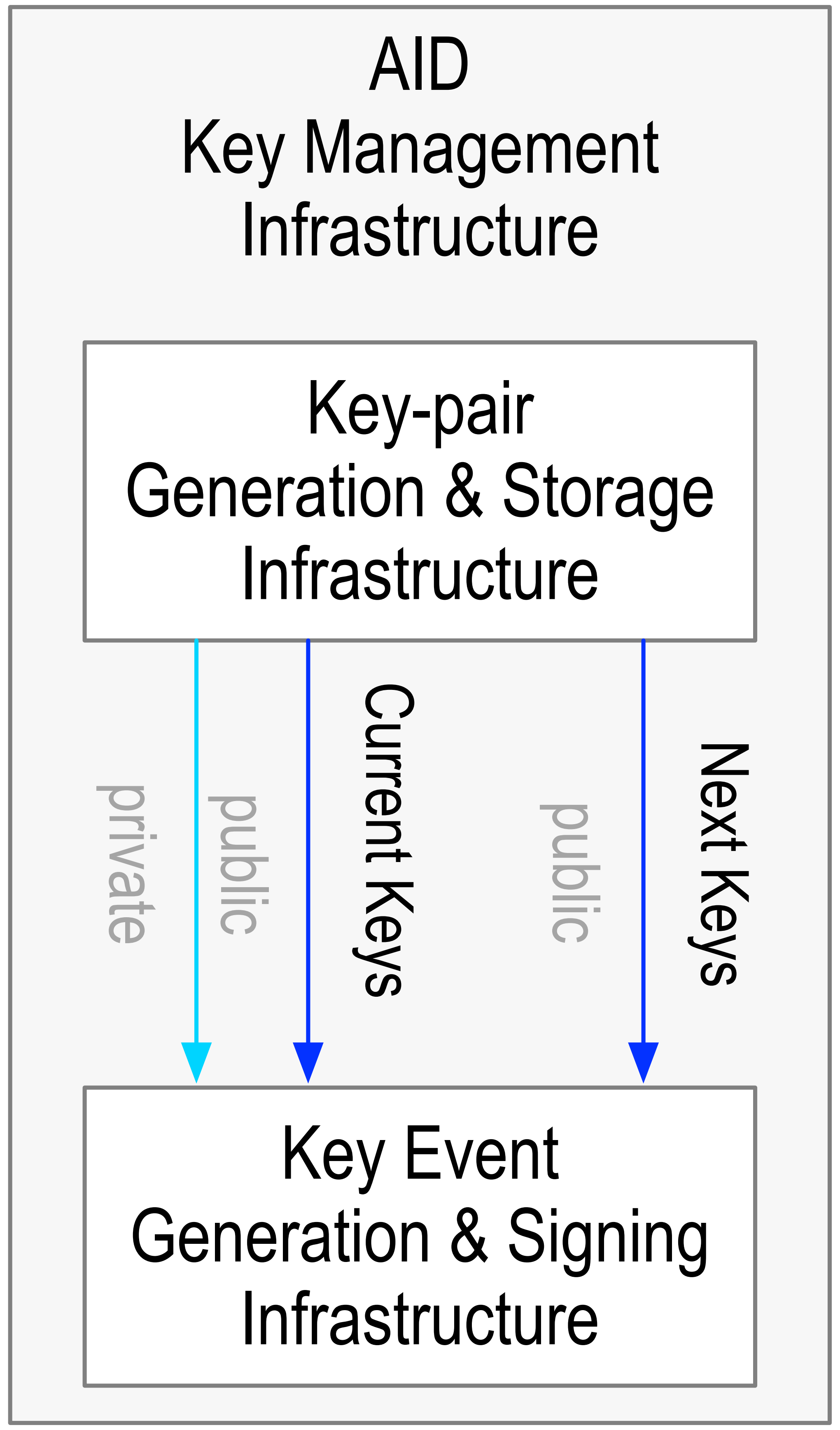
An *autonomic* namespace is

self-certifying and hence *self-administrating.*

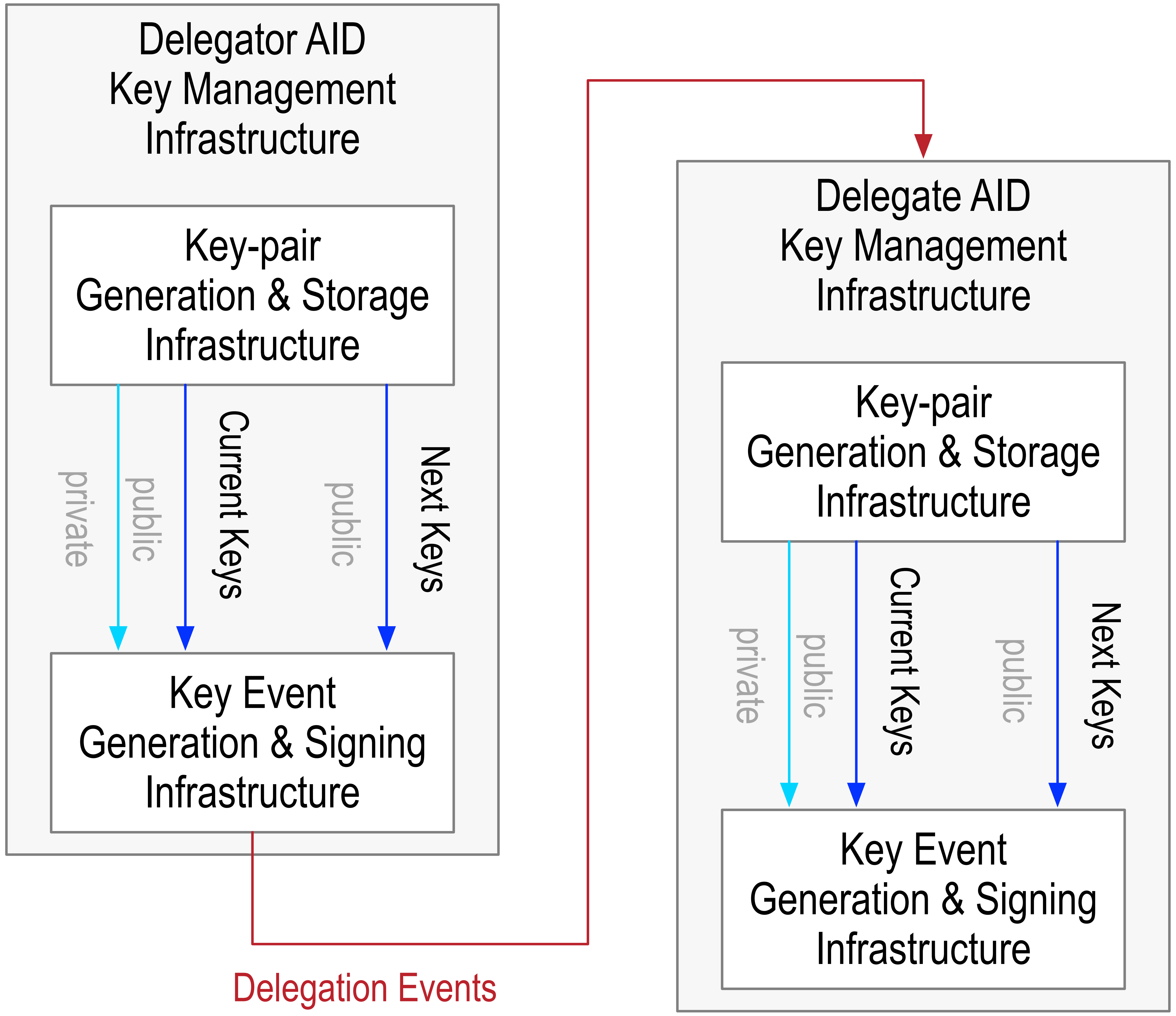
AIDs and ANs are *portable* = truly self-sovereign.

autonomic prefix = self-cert + UUID + URL = universal identifier

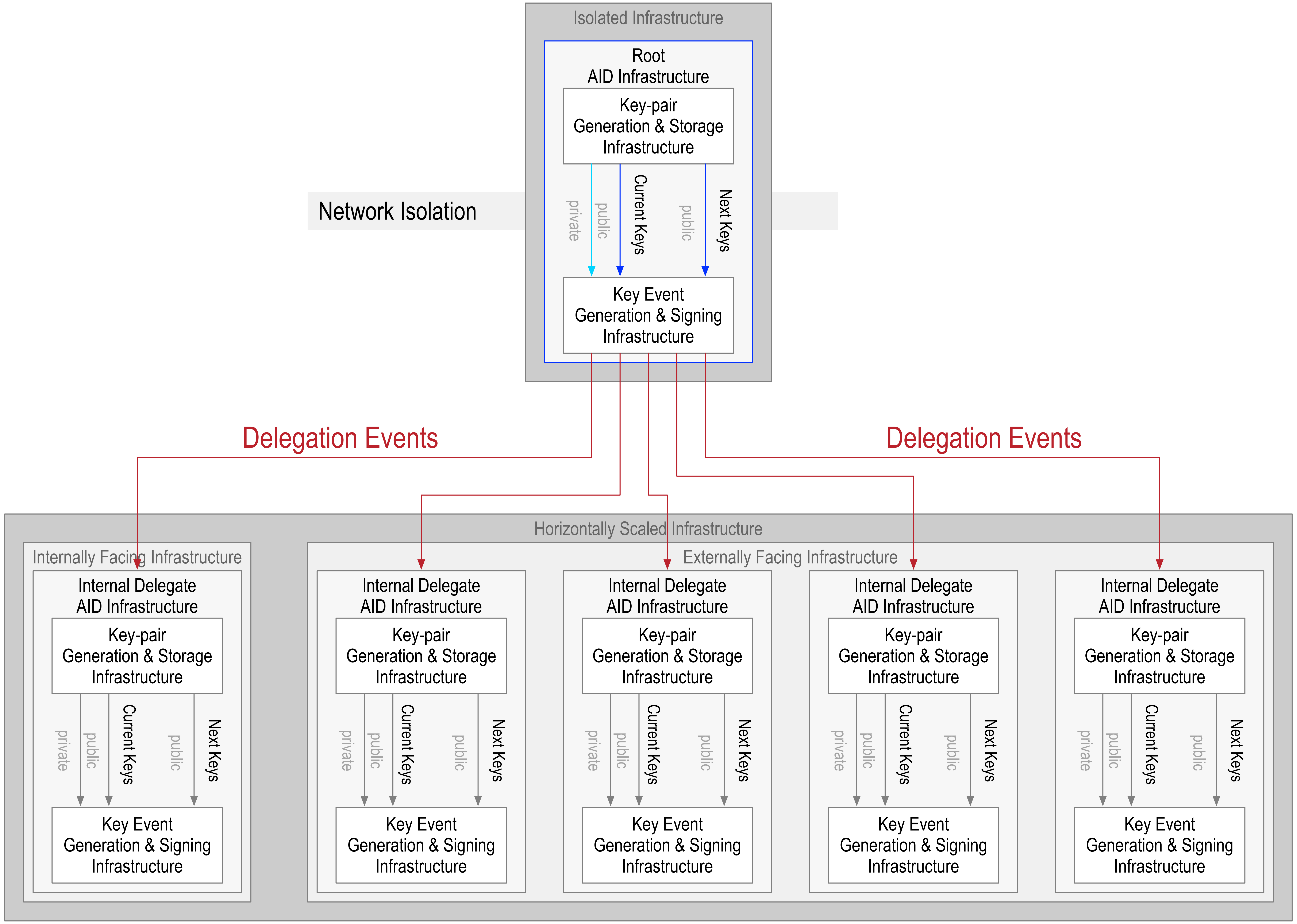
Decentralized Key Management Infrastructure (Univalent DKMI)



Hierarchical DKMI: Bivalent DKMI

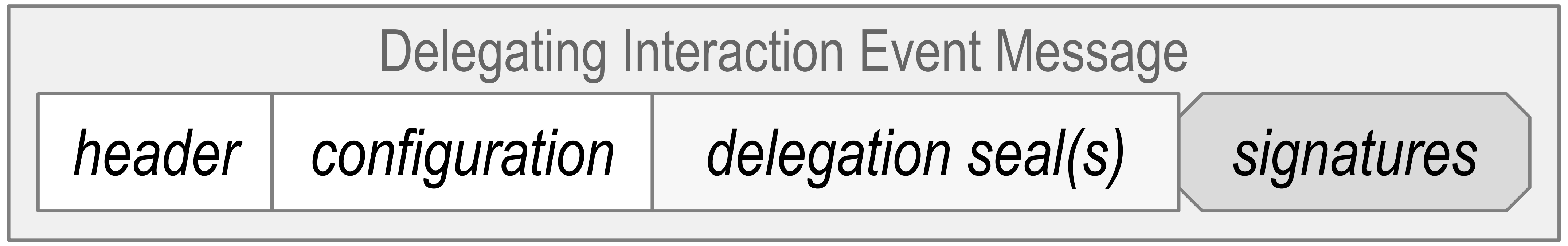


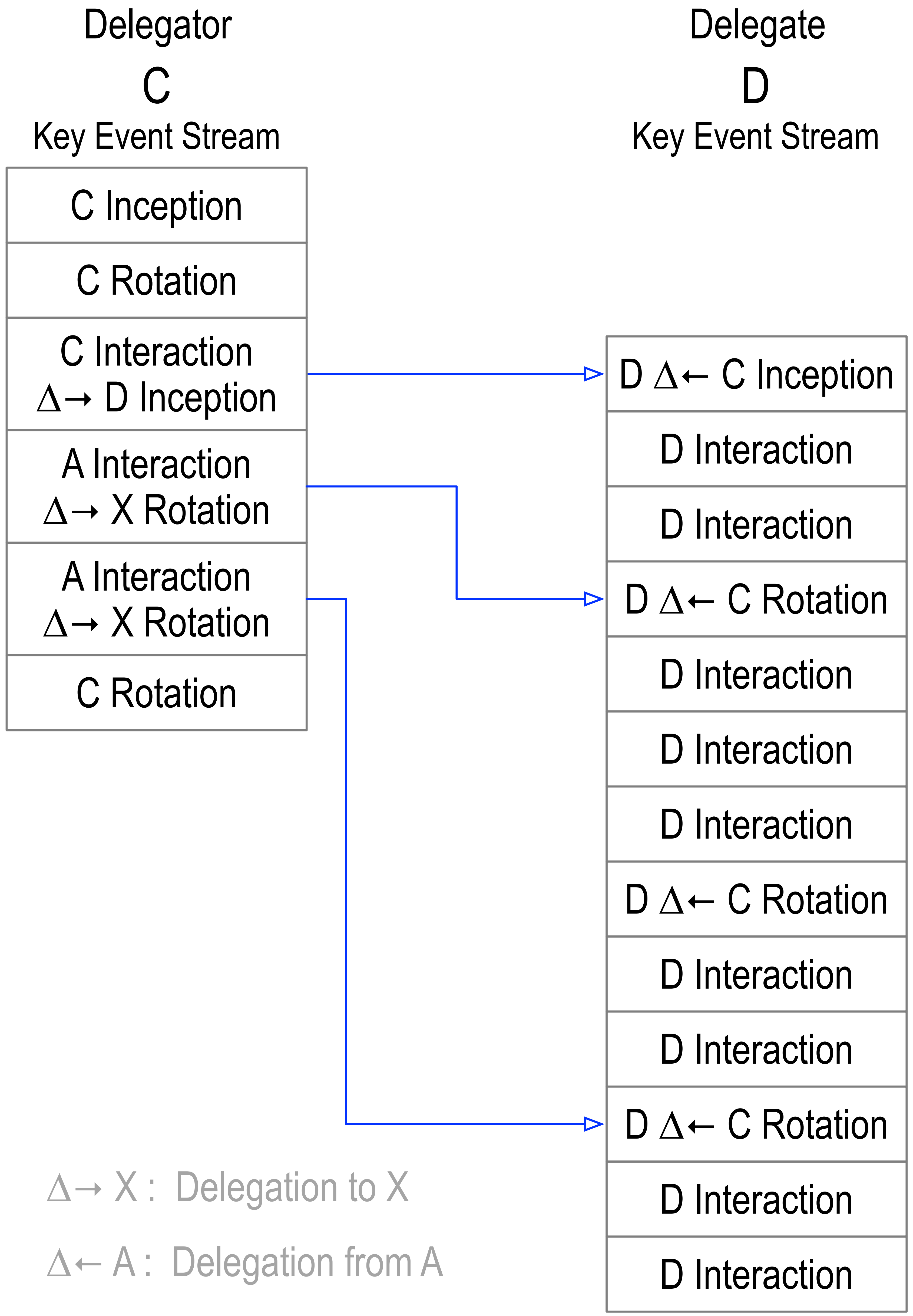
MultiValent Delegation



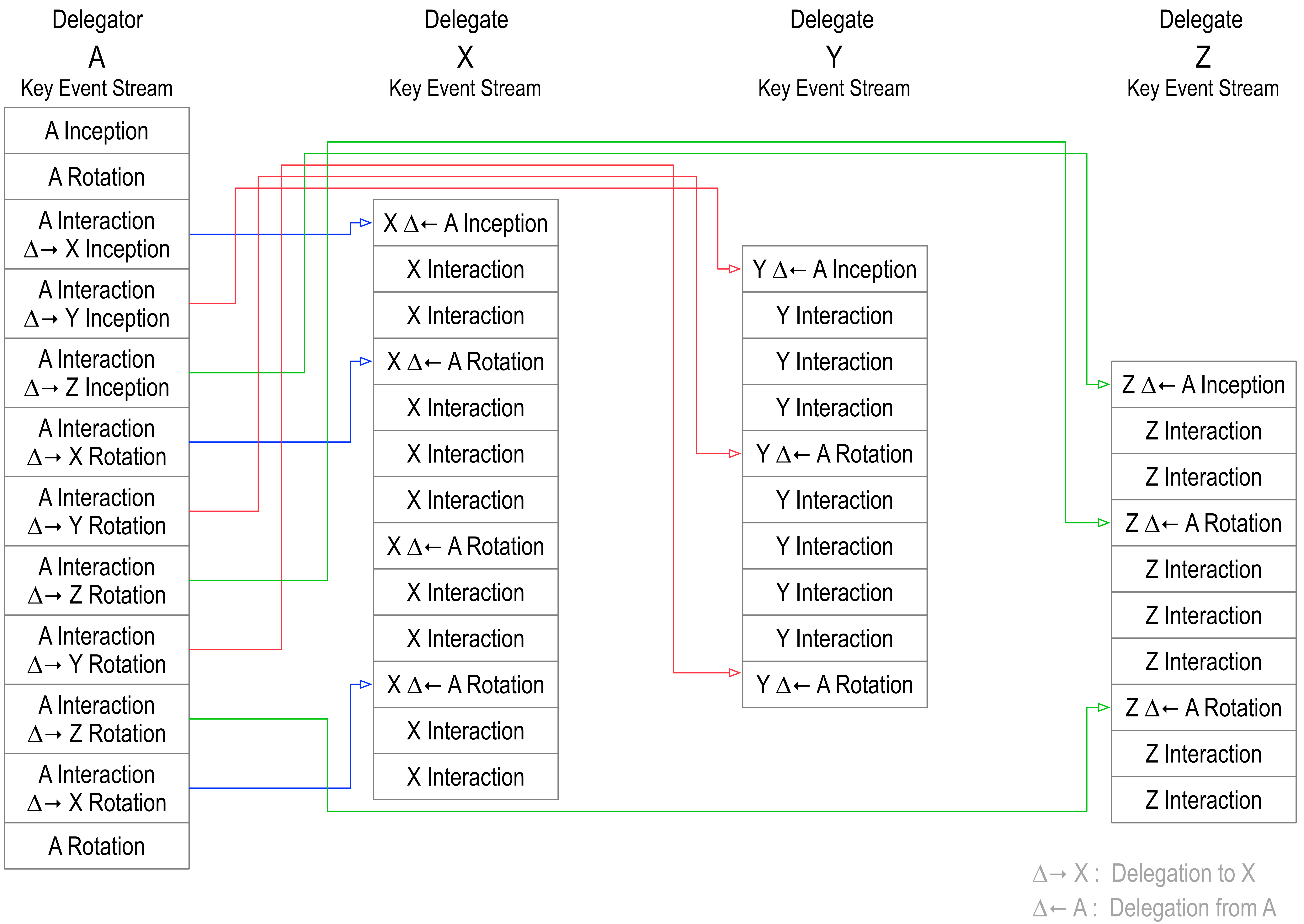
Delegation (Cross Anchor)

Interaction Delegation





Scaling Delegation via Interaction



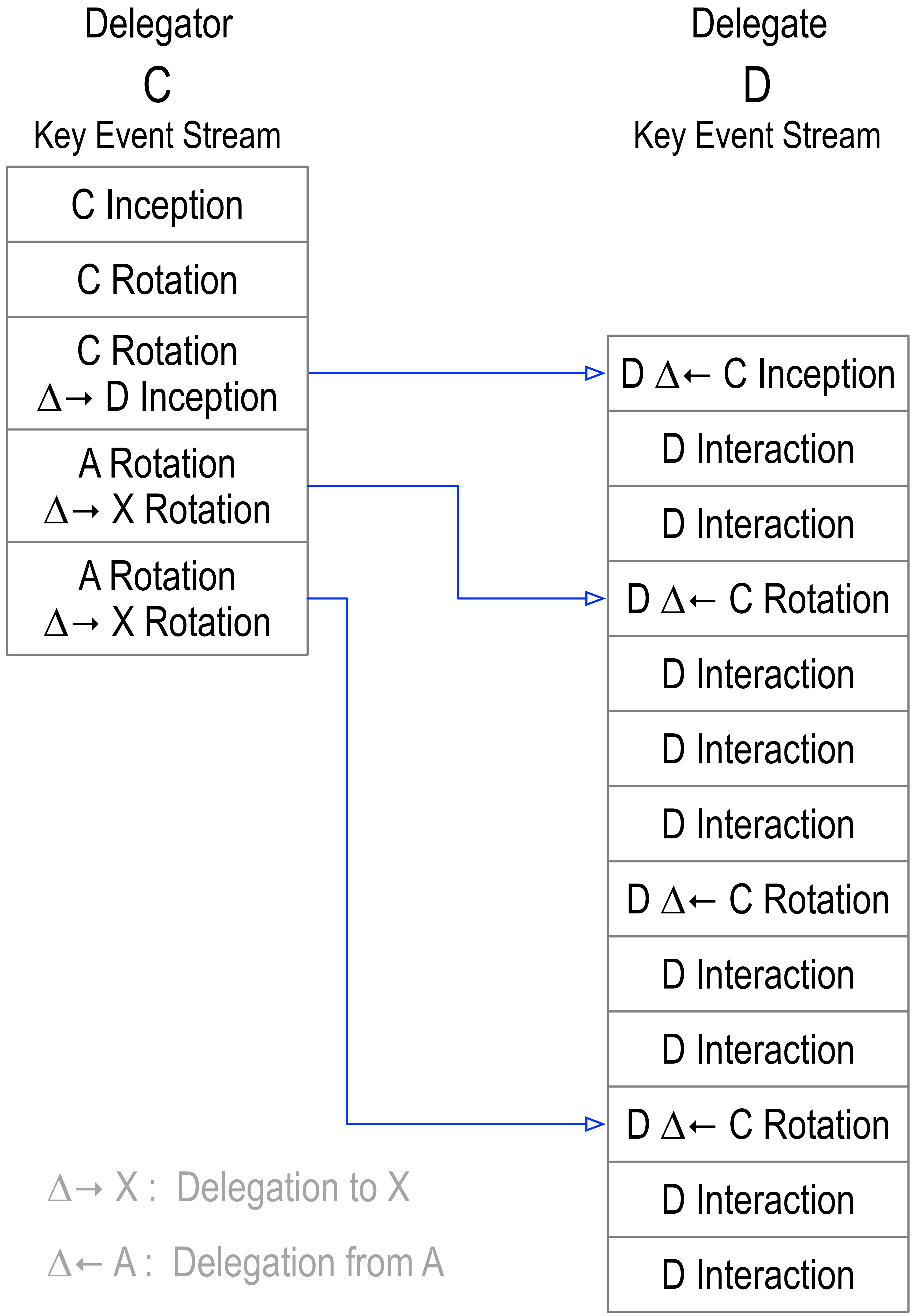
Rotation Delegation

Delegating Rotation Event Message

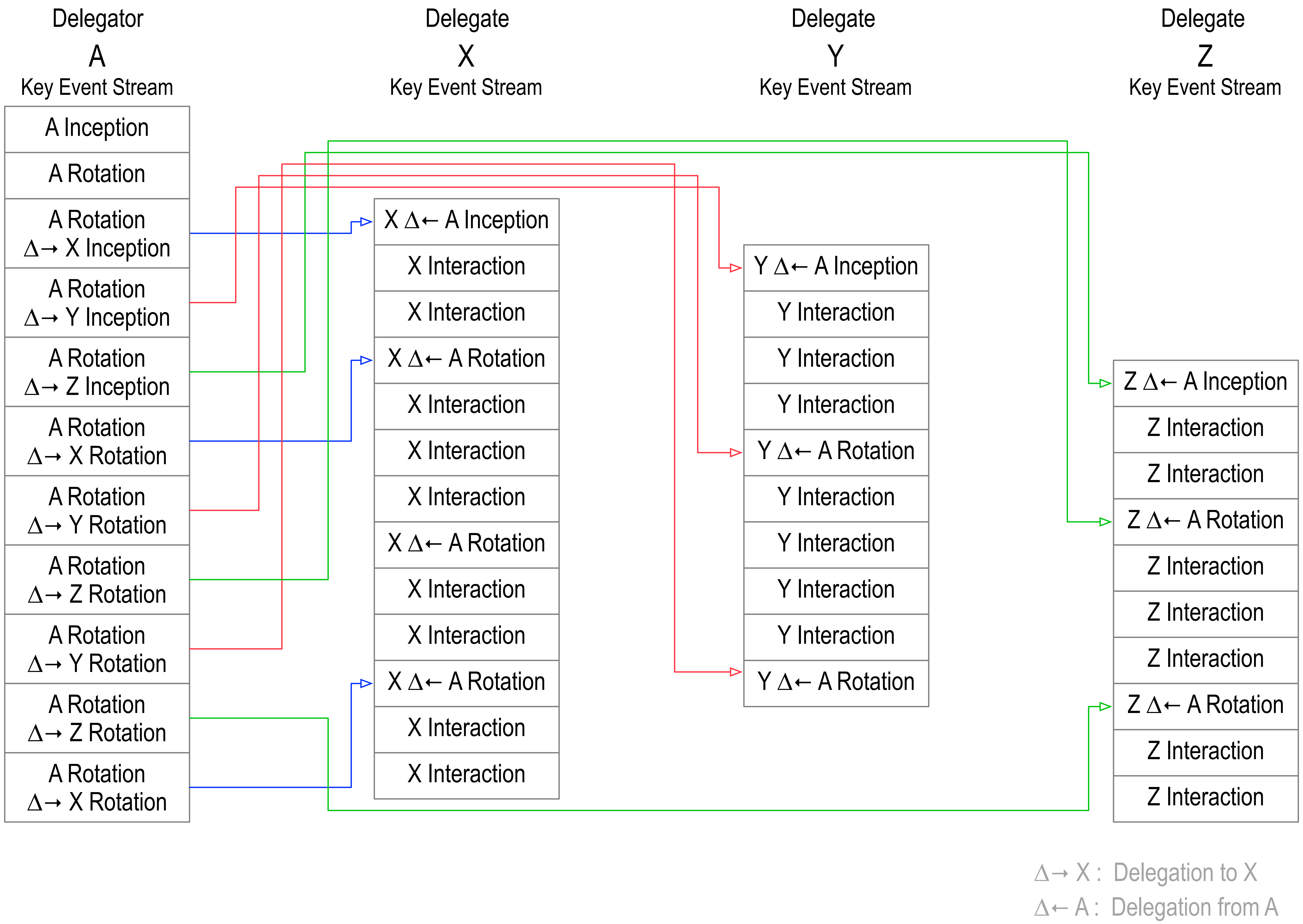
header	configuration	delegation seal(s)	signatures

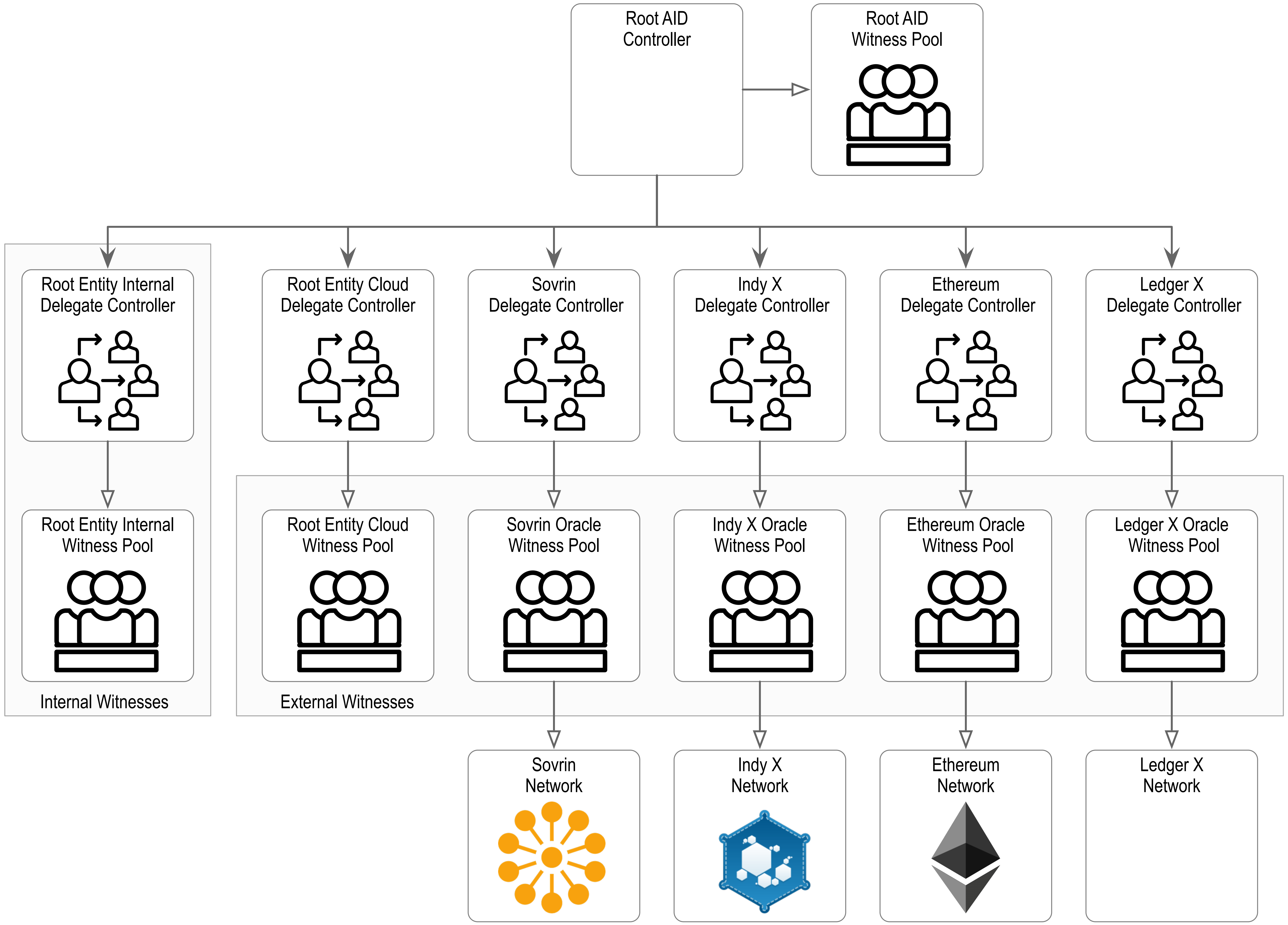
Delegated Event Message

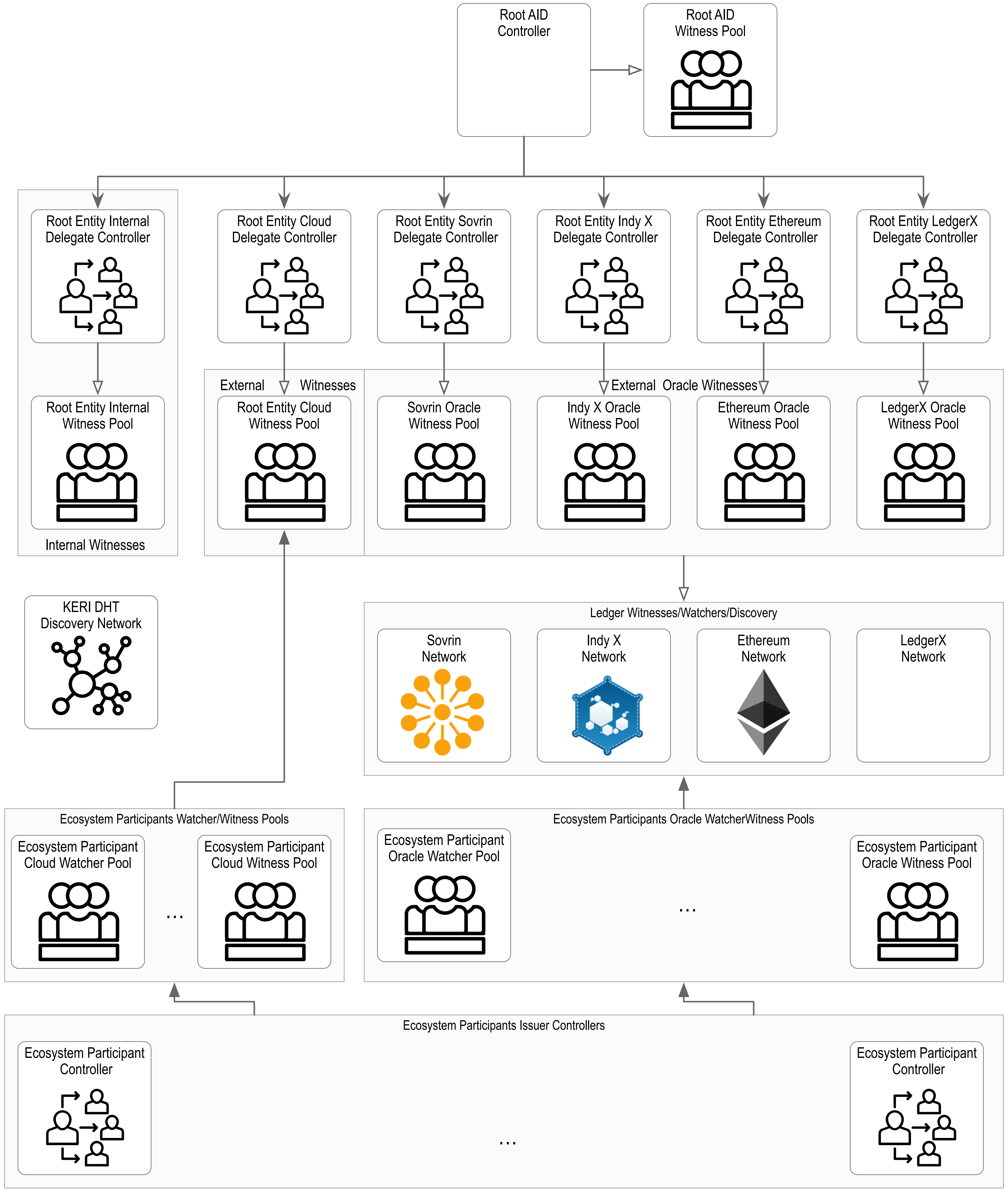
header	configuration	perms	delegation seal(s)	signatures



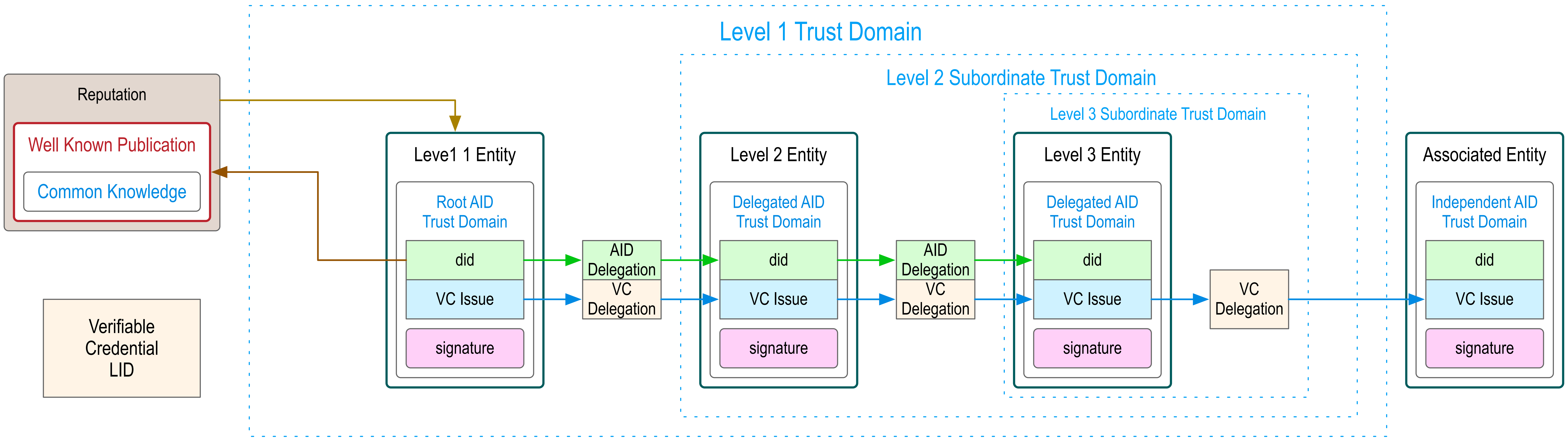
Scaling Delegation via Rotation




Root AID Controller
Root AID Witness Pool
Root Entity Internal Delegate Controller
Root Entity Cloud Delegate Controller
Sovrin Delegate Controller
Indy X Delegate Controller
Ethereum Delegate Controller
Ledger X Delegate Controller
Root Entity Internal Witness Pool
Root Entity Cloud Witness Pool
Sovrin Oracle Witness Pool
Indy X Oracle Witness Pool
Ethereum Oracle Witness Pool
Ledger X Oracle Witness Pool
Internal Witnesses
External Witnesses
Sovrin Network
Indy X Network
Ethereum Network
Ledger X Network



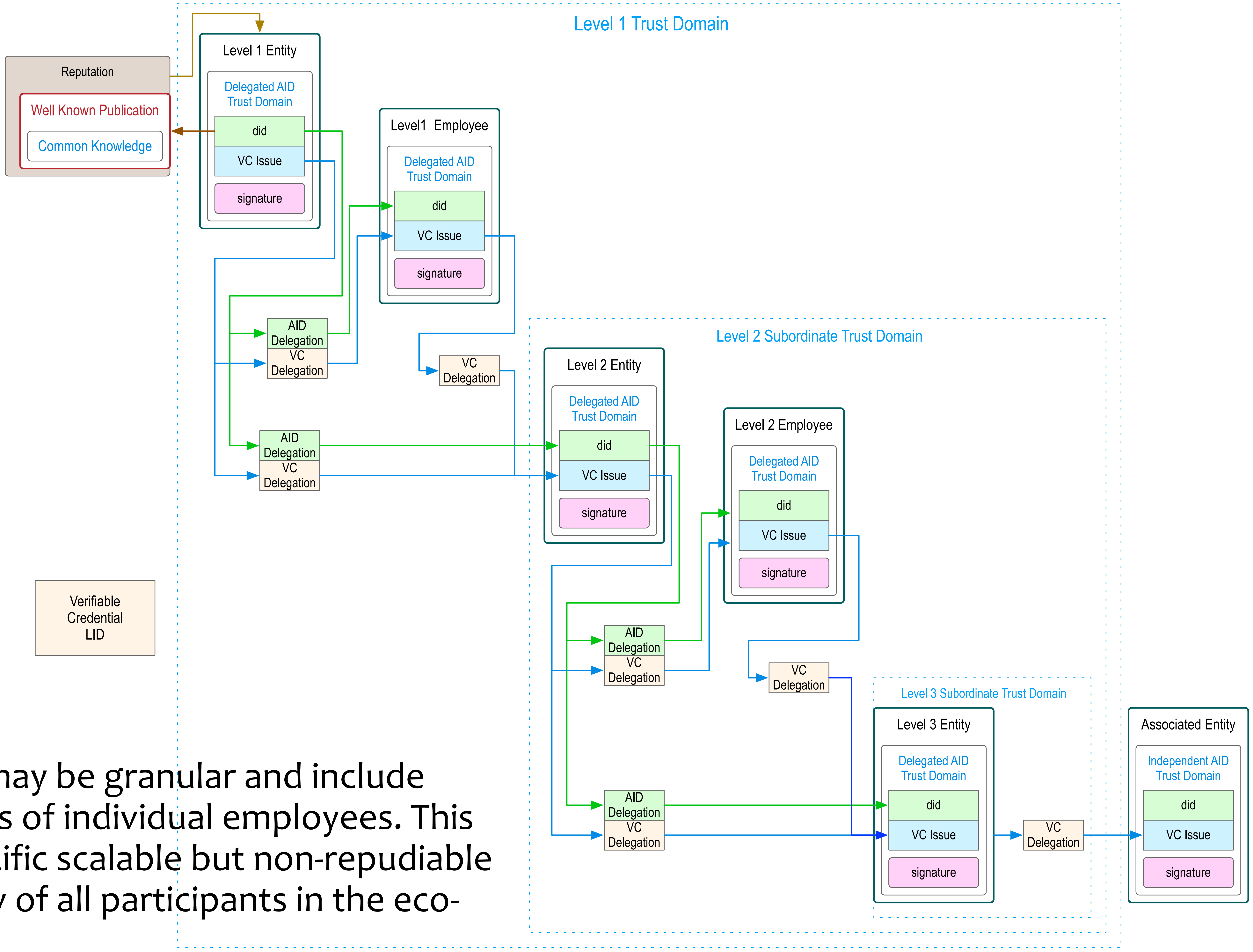
Root AID Controller
Root AID Witness Pool
Root Entity Internal Delegate Controller
Root Entity Cloud Delegate Controller
Root Entity Sovrin Delegate Controller
Root Entity Indy X Delegate Controller
Root Entity Ethereum Delegate Controller
Root Entity LedgerX Delegate Controller
External Witnesses
External Oracle Witnesses
Root Entity Internal Witness Pool
Internal Witnesses
Root Entity Cloud Witness Pool
Sovrin Oracle Witness Pool
Indy X Oracle Witness Pool
Ethereum Oracle Witness Pool
LedgerX Oracle Witness Pool
KERI DHT Discovery Network
Ledger Witnesses/Watchers/Discovery
Sovrin Network
Indy X Network
Ethereum Network
LedgerX Network
Ecosystem Participants Watcher/Witness Pools
Ecosystem Participant Cloud Watcher Pool
...
Ecosystem Participant Cloud Witness Pool
Ecosystem Participants Oracle WatcherWitness Pools
Ecosystem Participant Oracle Watcher Pool
...
Ecosystem Participant Oracle Witness Pool
Ecosystem Participants Issuer Controllers
Ecosystem Participant Controller
...
Ecosystem Participant Controller



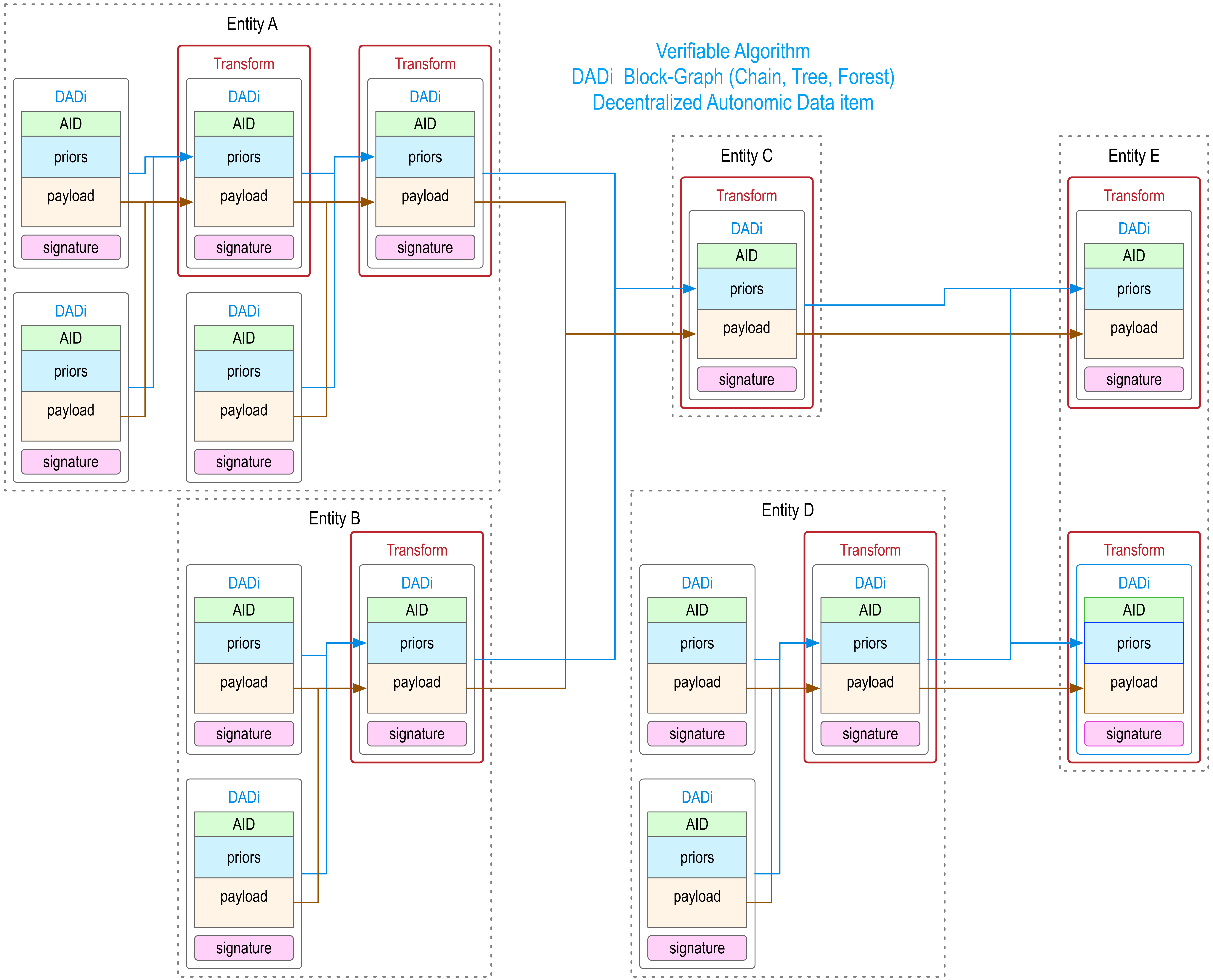
Each level of delegation forms a nested trust domain that is protected by the level above.
This increases ultimate security while enabling higher performance event issuance in lower layers.

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

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



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

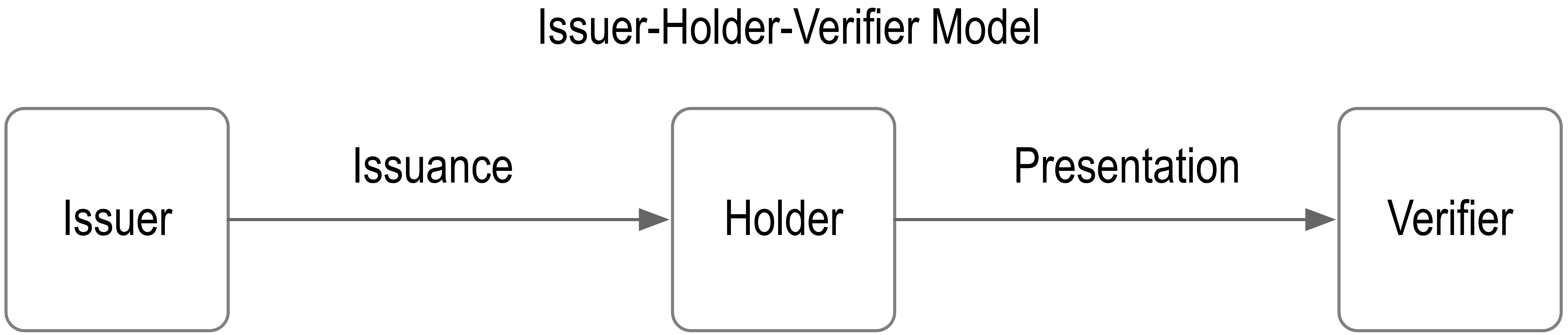


Verifiable Algorithm
DADi Block-Graph (Chain, Tree, Forest)
Decentralized Autonomic Data item
Entity A
Entity B
Entity C
Entity D
Entity E
Transform
DADi
AID
priors
payload
signature

Tripartite Authentic Data (VC) Model

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

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



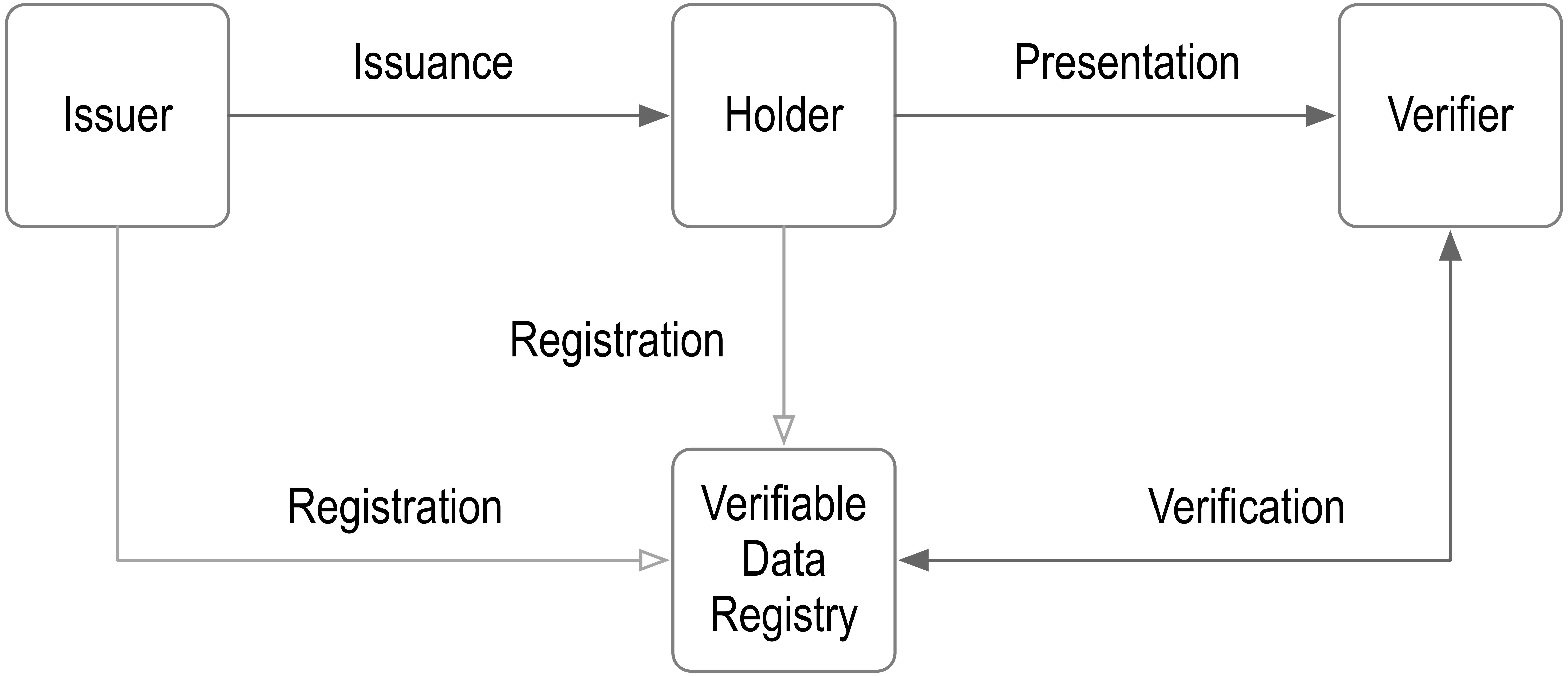
Tripartite Authentic Data (VC) Model with VDR

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

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

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

Issuer-Holder-Verifier Model with Verification at Verifiable Data Registry

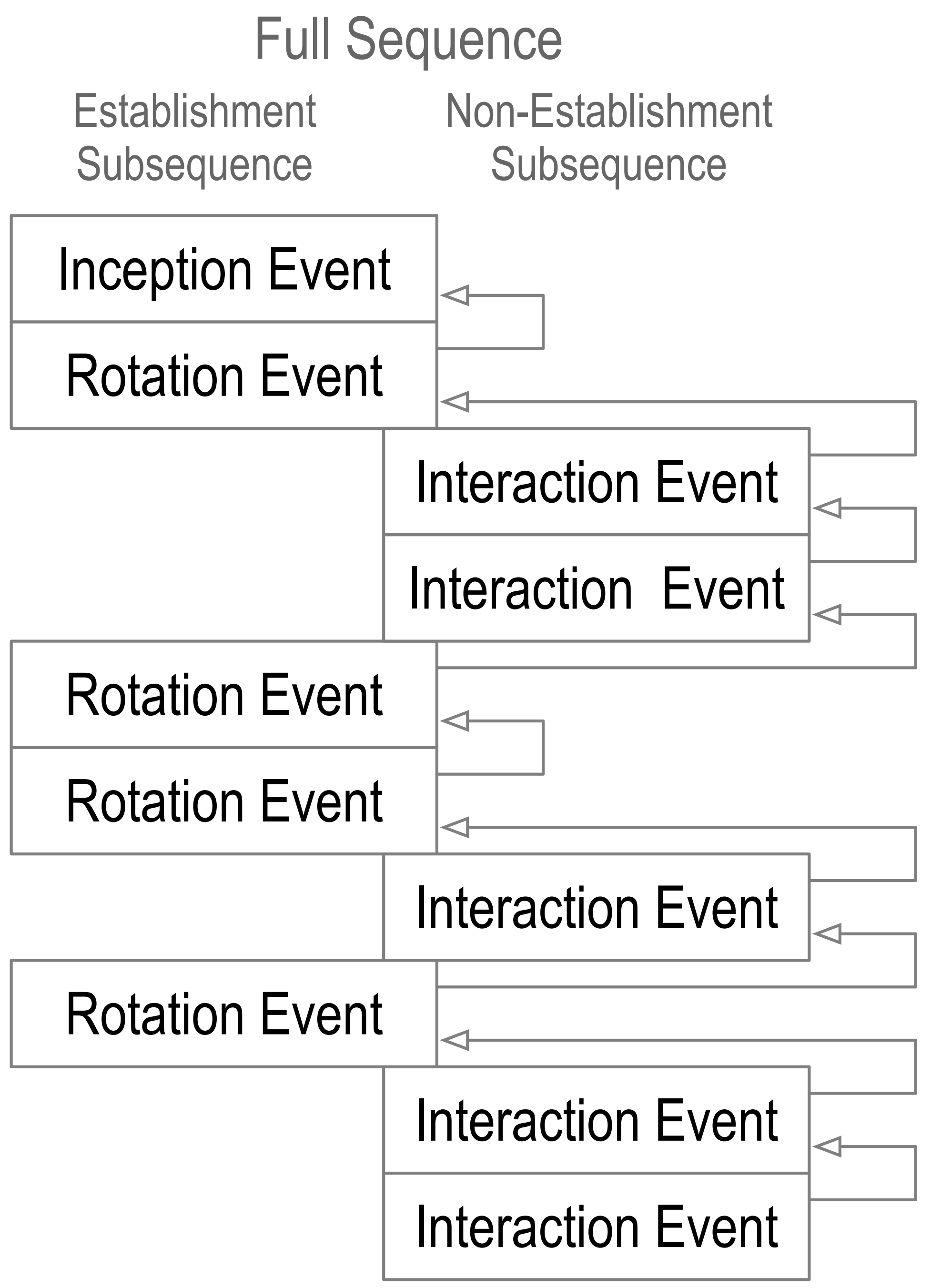


KERI VDRs vs. Shared Ledger VDRs

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

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

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



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

did:keri:ENqFtH6_cfDg8riLZ-GDvDaCKVn6clOJa7ZXXVXSWpRY

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

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

Seals include cryptographic digest of external transaction data.

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

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

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

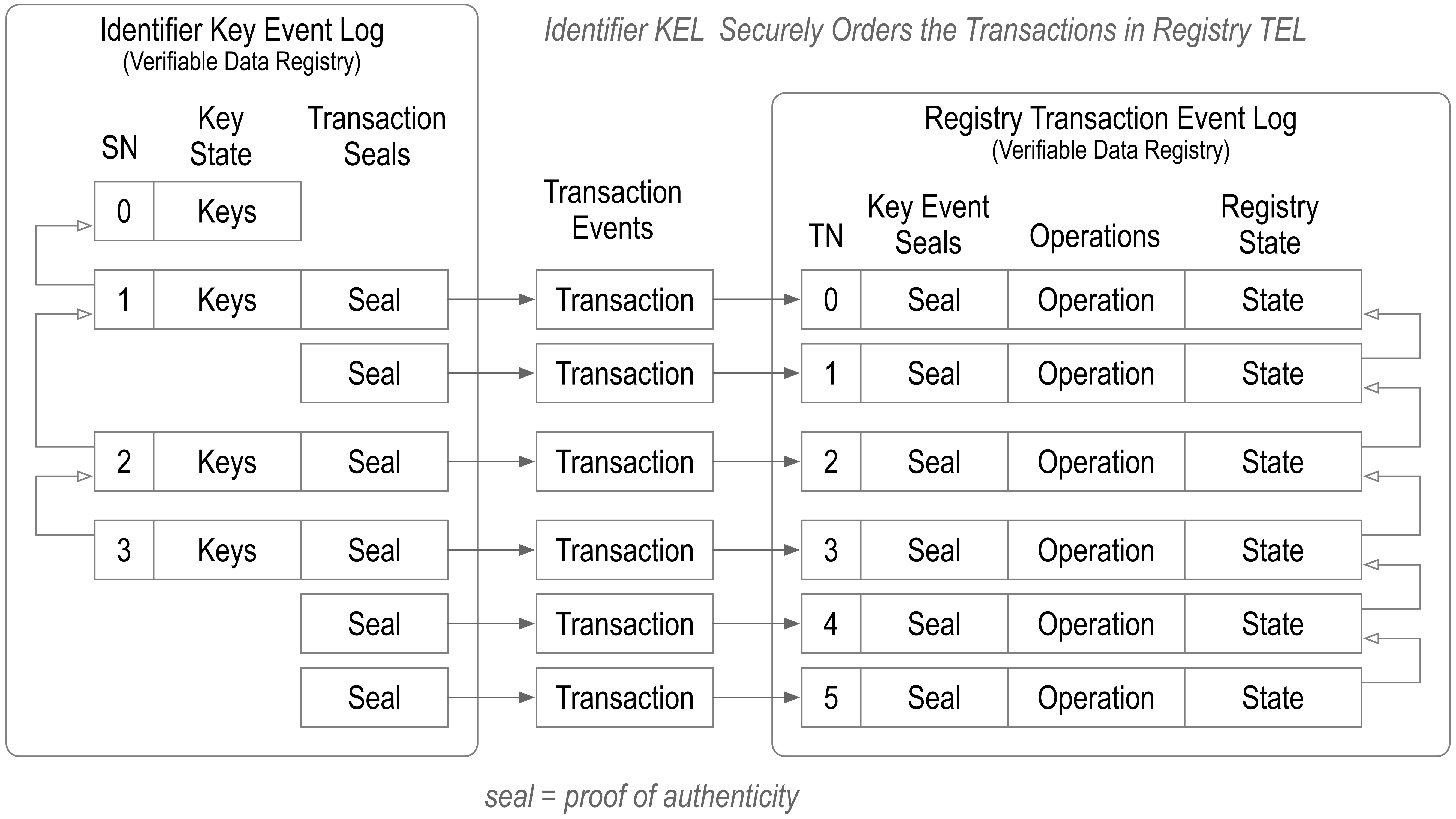
The transactions likewise contain a reference seal back to the key event authorizing the transaction.

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

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

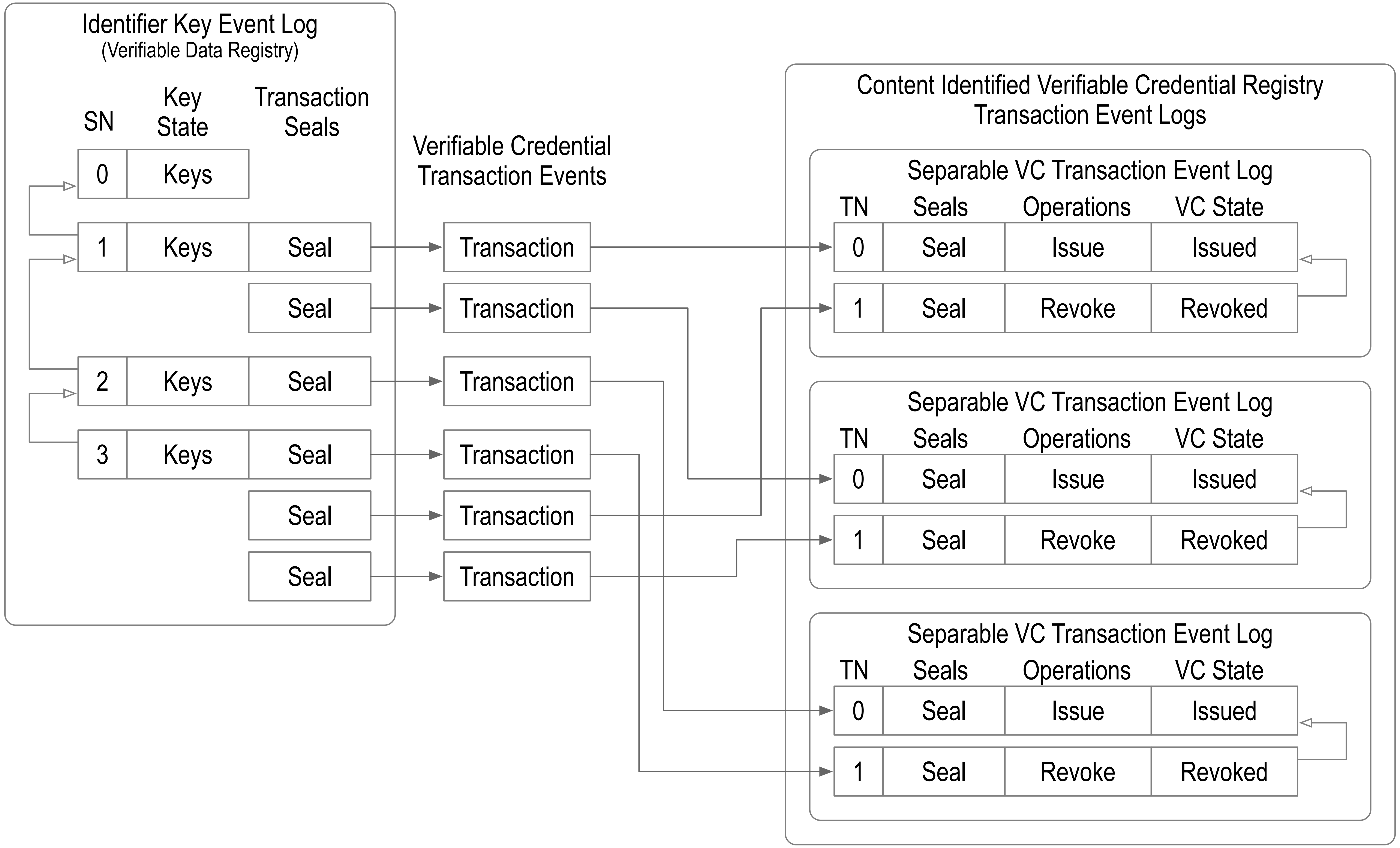
Any validator may therefore cryptographically verify the authoritative state of the registry.

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

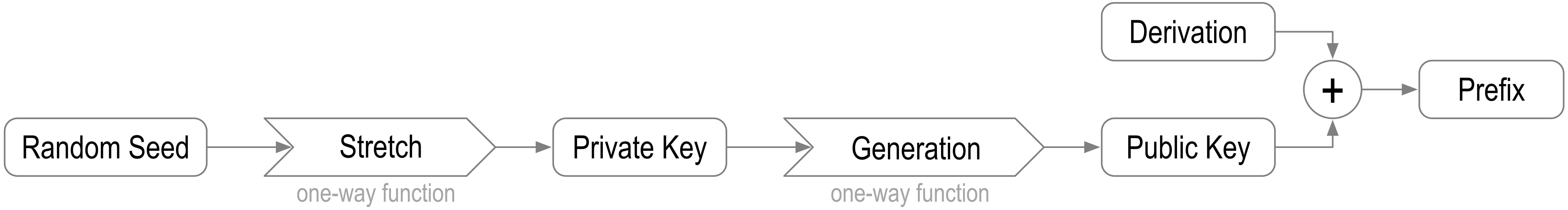


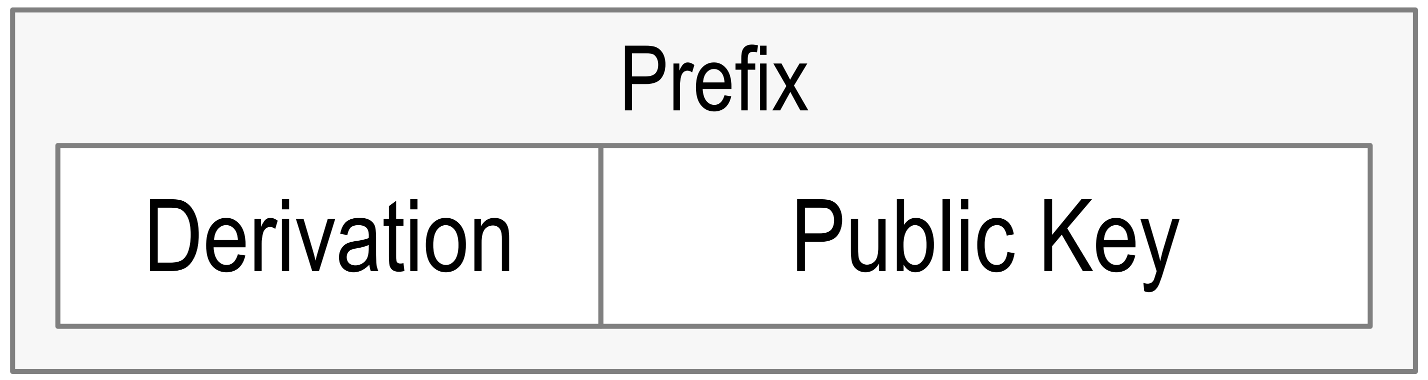
Registry with Separable VC Issuance-Revocation TELs

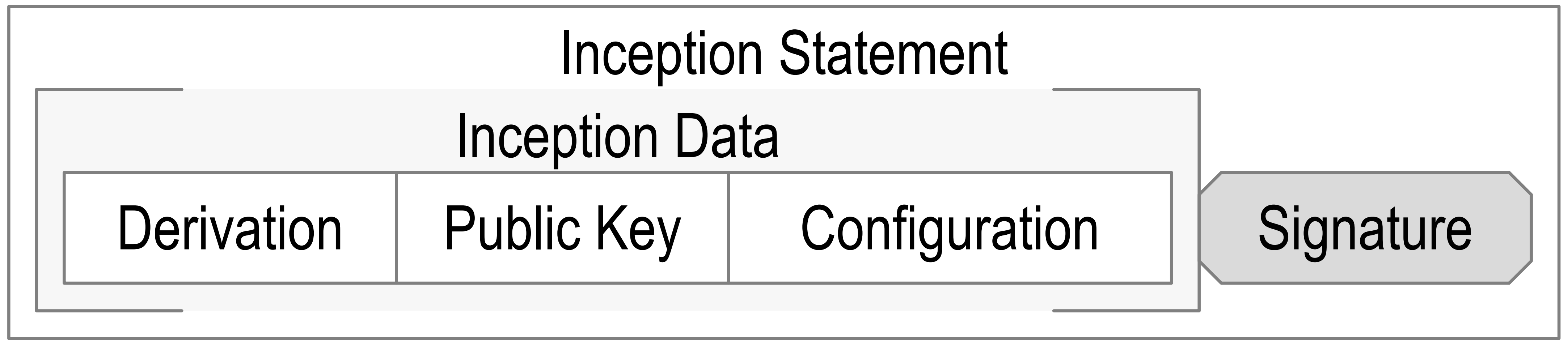
Each VC also has a uniquely identified issuer using a KERI AID.
Each VC may be uniquely identified with a content digest.
A full identifier for the VC may include its content digest but also be in the namespace of its issuer.
These may be used as database keys to lookup a VC and verify the content of a given VC.
This combination enables a separable registry of VC issuance-revocation state.
The state may employ a cryptographic accumulator for enhanced privacy



Basic SCID





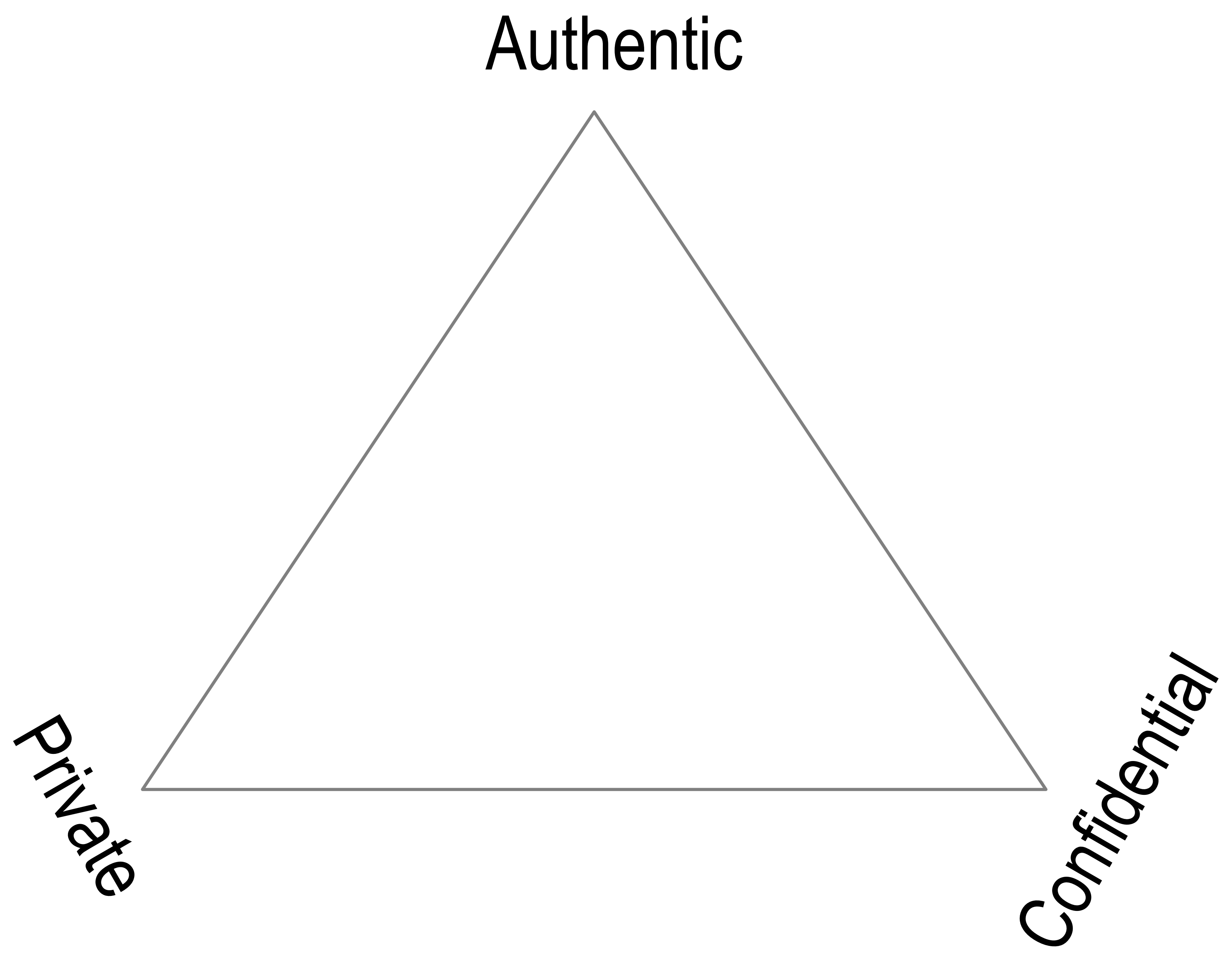


BDKrJxkcR9m5u1xs33F5pxRJP6T7hJEbhpHrUtlDdhh0

did:un:BDKrJxkcR9m5u1xs33F5pxRJP6T7hJEbhpHrUtlDdhh0/path/to/resource?name=secure#really

PAC Theorem

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



Trade-offs required!

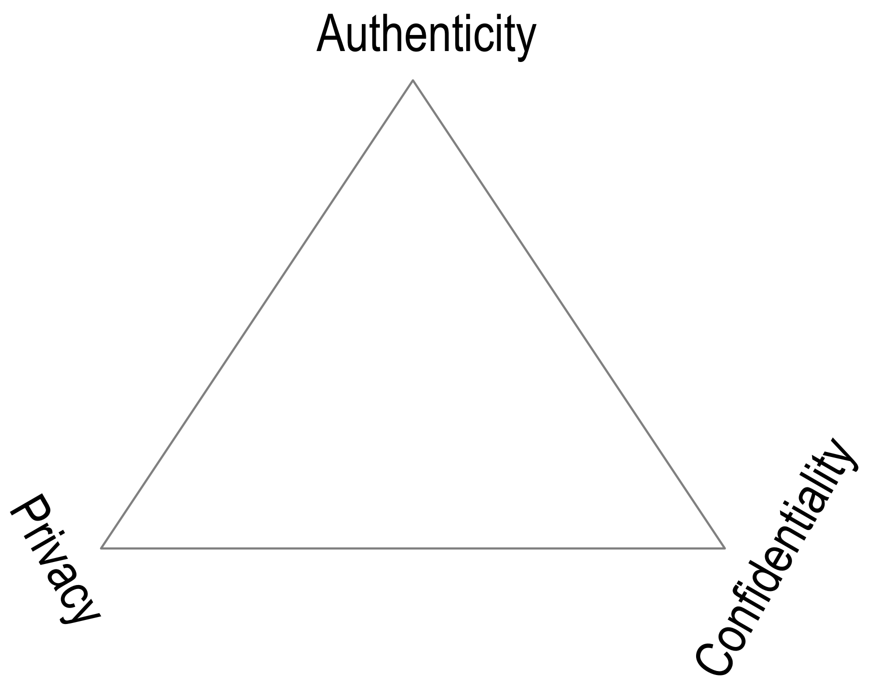
Definitions

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

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

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

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



Proving Authenticity

Non-repudiable Proof:

a statement's author cannot successfully dispute its authorship

Asymmetric key-pair digital signature

Repudiable Proof:

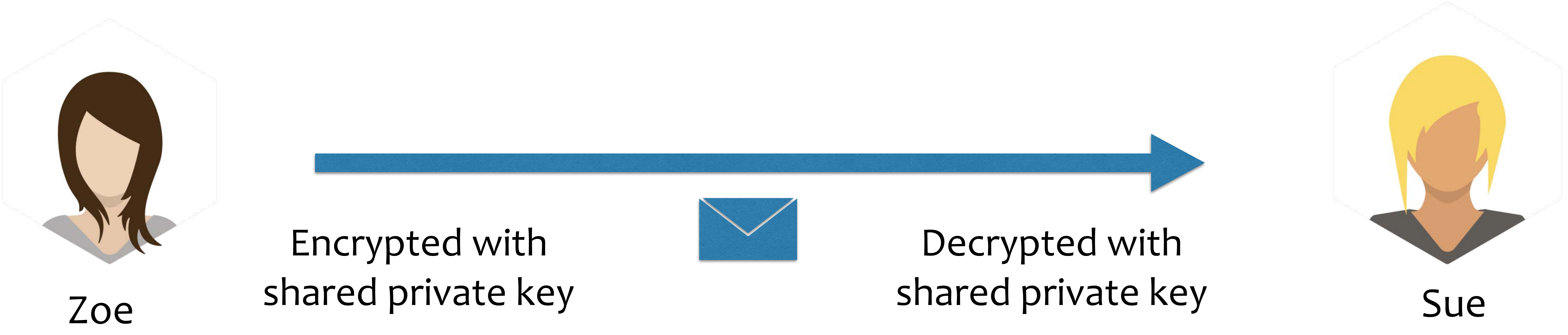
a statement's author can successfully dispute its authorship

DH shared symmetric key-pair encryption (auth crypt)

Non-Repudiable Authenticity

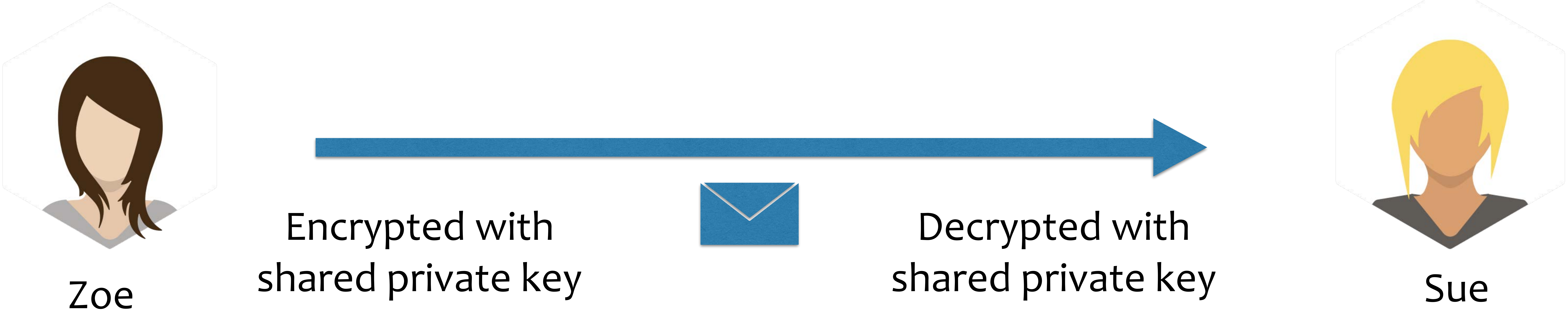
Non-repudiable authenticity is *zero-trust*

Repudiable Authenticity



Repudiable authenticity requires trust (is not zero-trust)

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

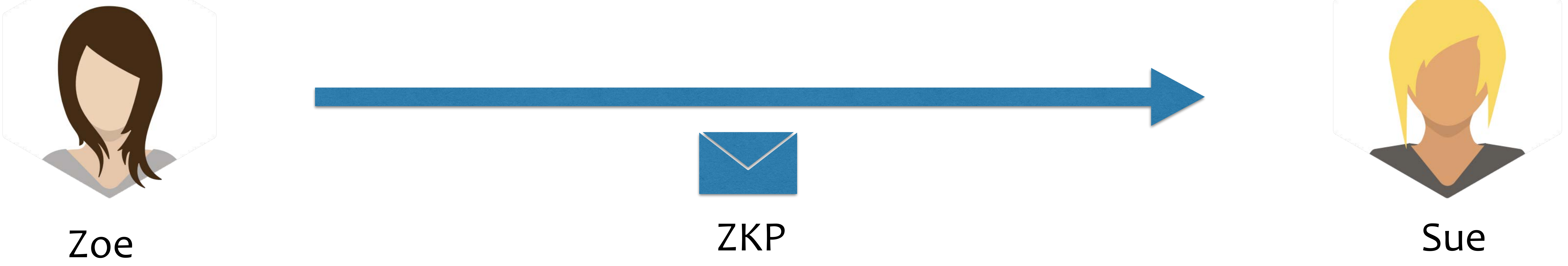


Non-Repudiable authenticity has recourse.

Best fits current business and regulatory eco-systems.

Zero Knowledge Proof?

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.

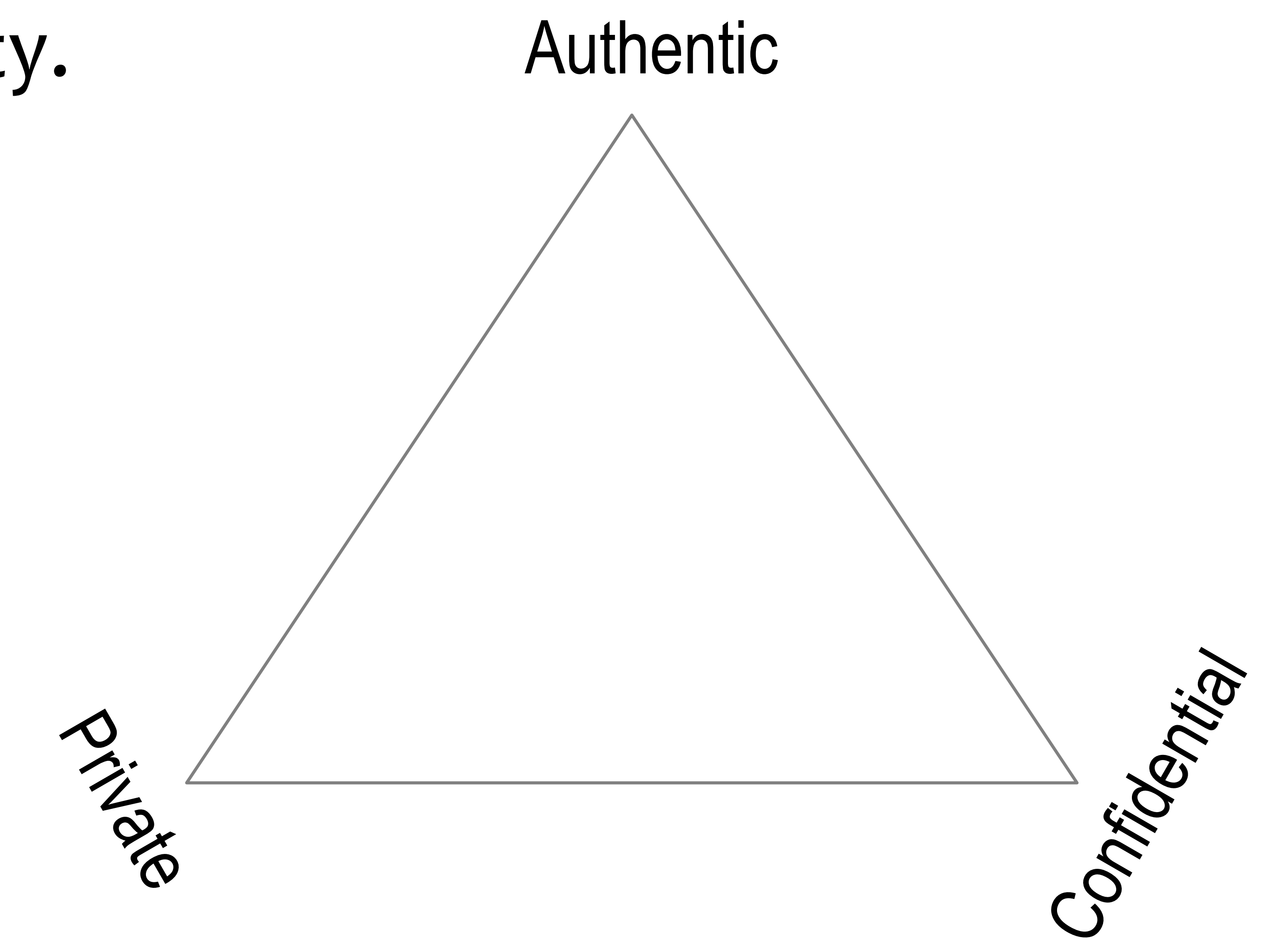


Authentic ZKP: Is the information proven in a repudiable or non-repudiable manner?

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.

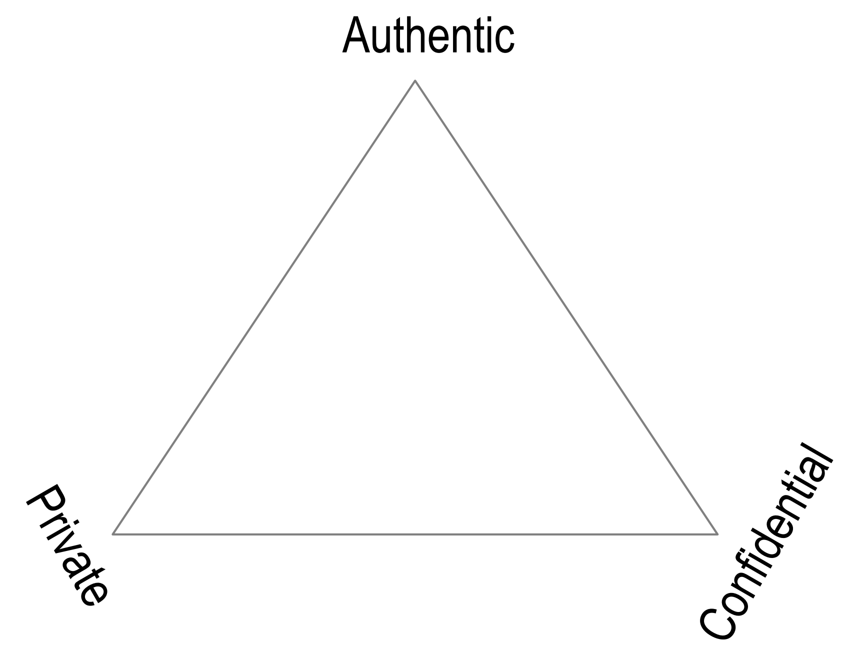
Non-repudiation means any party to conversation can proof to any other party exactly what was said by whom.
This means that technologically there is no way to prevent disclosure by any party to some third party.
We can incentivize confidentiality by imposing a liability on the parties to the disclosure 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.

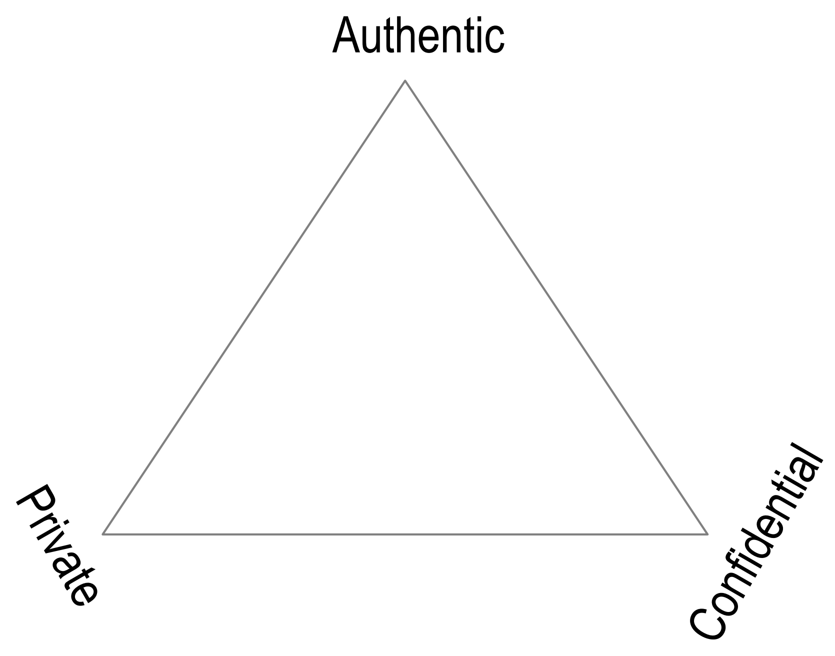
Authenticity
Confidentiality
Privacy



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.