Live Appraisal of Key Compromise via Duplicity Evident Infrastructure

How to protect verifiers (and controllers) from imposters
without
blockchain and trusted third parties

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

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

Definitions

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 appraises 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.

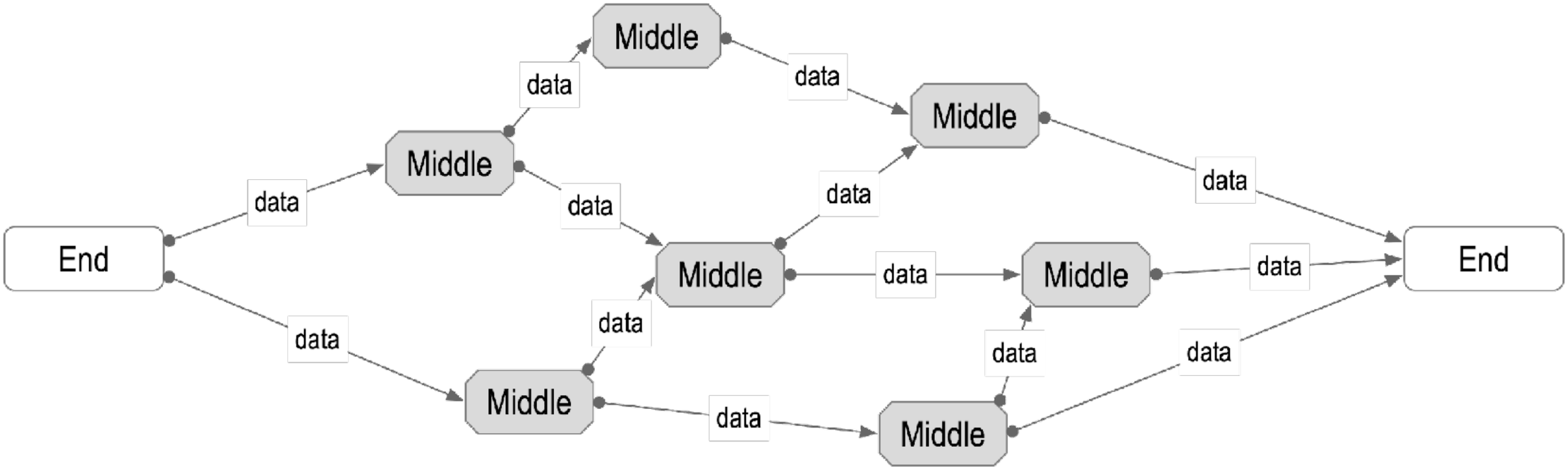
Duplicity Evident Infrastructure removes the hard reliance on trusted third parties to perform live appraisals.

This is the primary security innovation that KERI as a DKMI provides. It is the simplest known approach to solving the dynamic appraisability problem without relying on trusted third parties or blockchain.

Duplicity Evident(KERI) vs. Duplicity Hiding (blockchain) vs. Duplicity Fostering(DNS/CA)

End Verifiability

End-to-End Verifiability



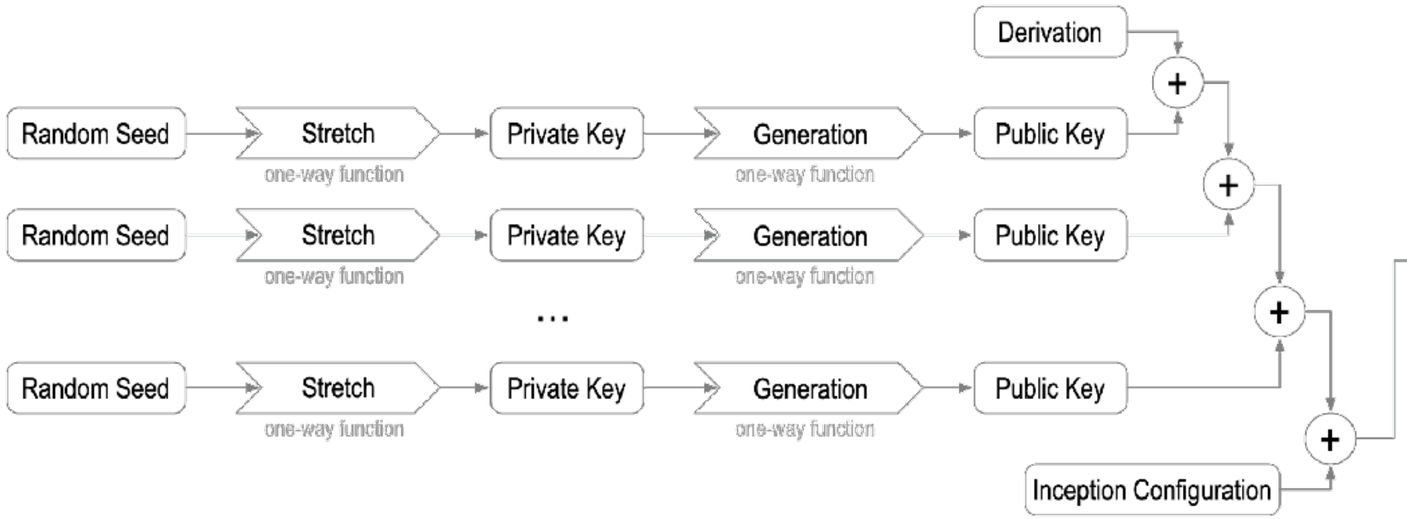
If the edges are secure, the security of the middle doesn't matter.

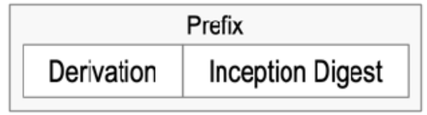
Ambient Verifiability: any-data, any-where, any-time by any-body

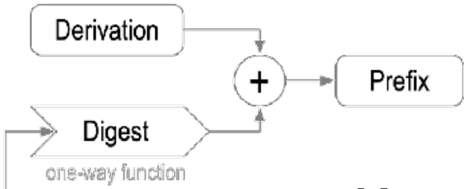
Zero-Trust-Computing

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

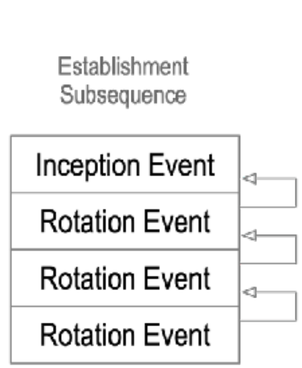
Cryptographic Root-of-Trust:
Self-Certifying Identifier (SCID) + Key Event Log + Infrastructure = Autonomic Identifier (AID)

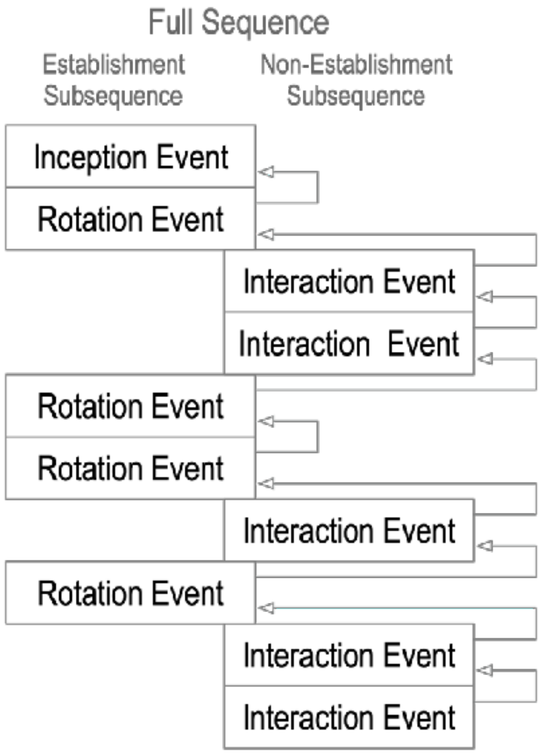


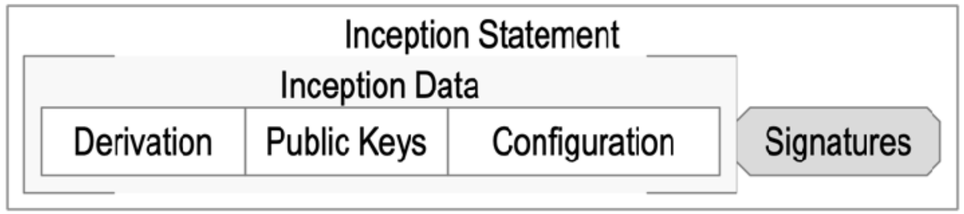




Key Event Log







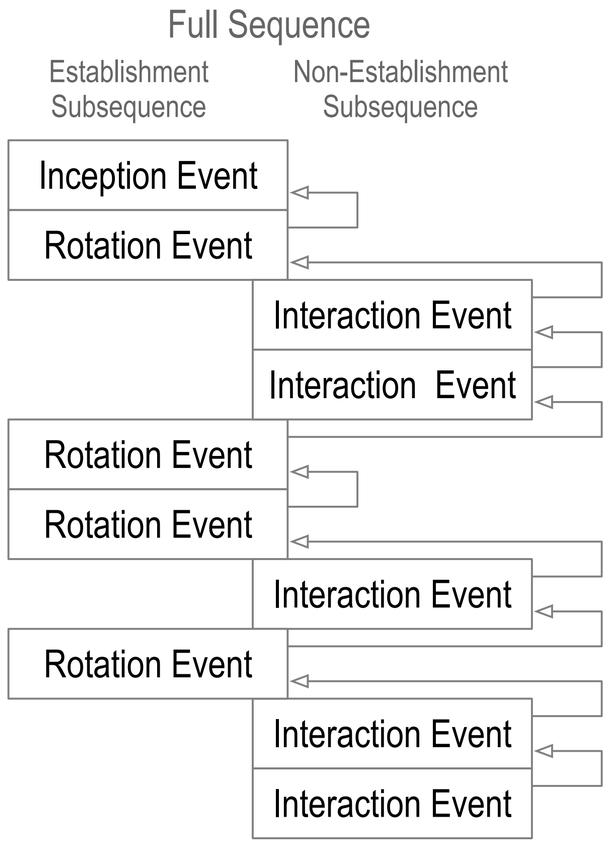
AID

`EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148`

Name-spaced AID

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

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.

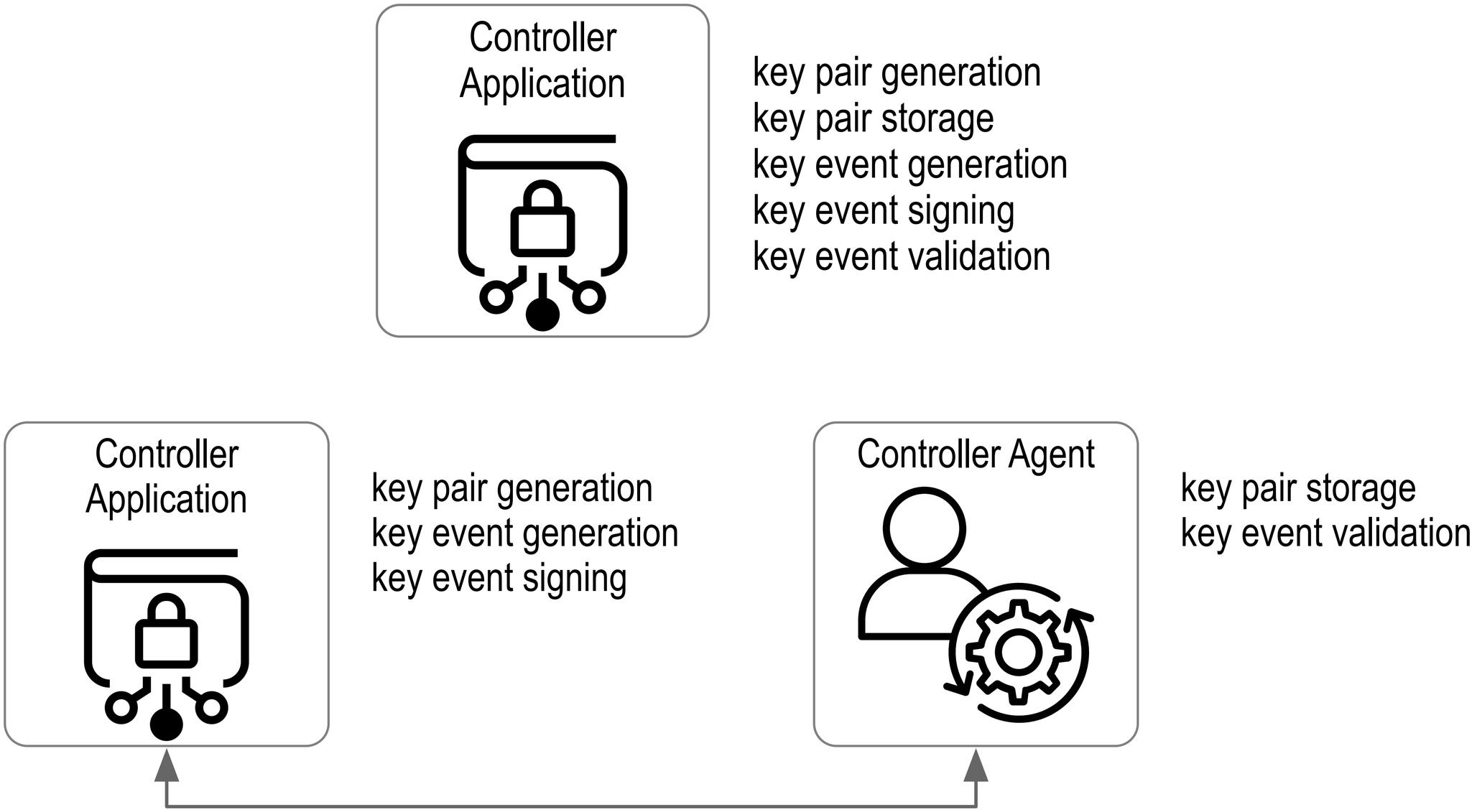
Log verification from self-certifying root-of-trust protects against internal inconsistency.

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

Duplicity detection protects against external inconsistency.

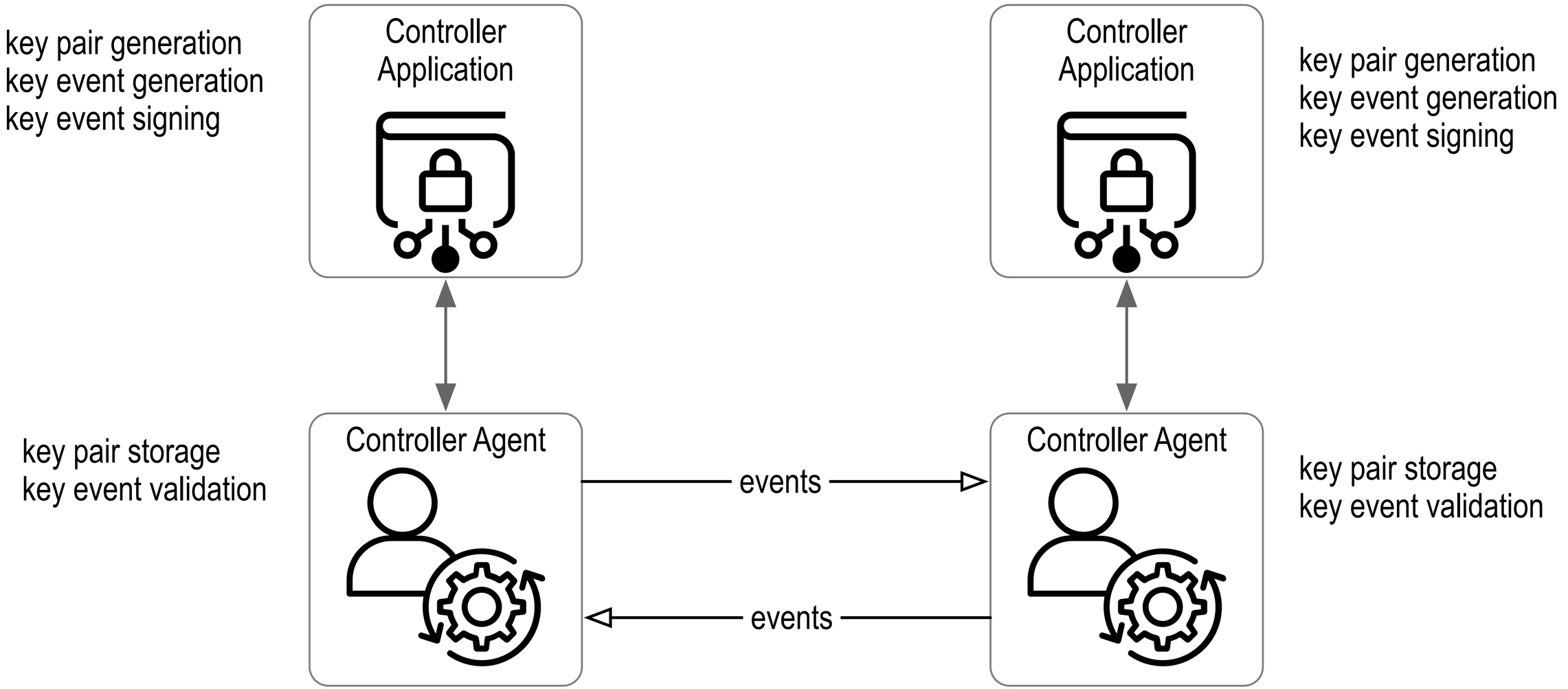
KERI provides duplicity evident DKMI

KERI Ecosystem Components: Controller Application and Agents

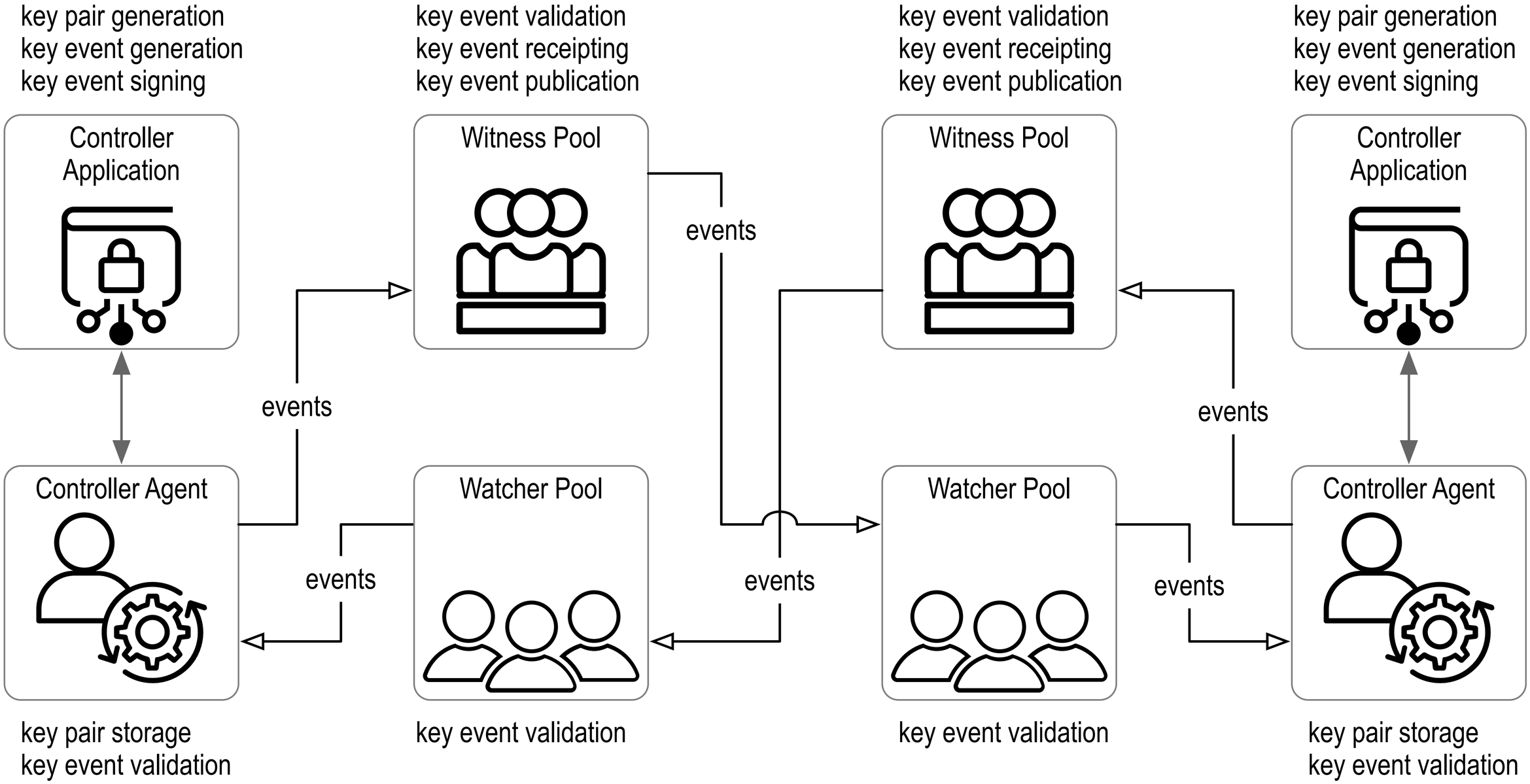


Modular, decentralized, web-based infrastructure without shared governance.

KERI Ecosystem Components: Peer-to-Peer Direct Mode

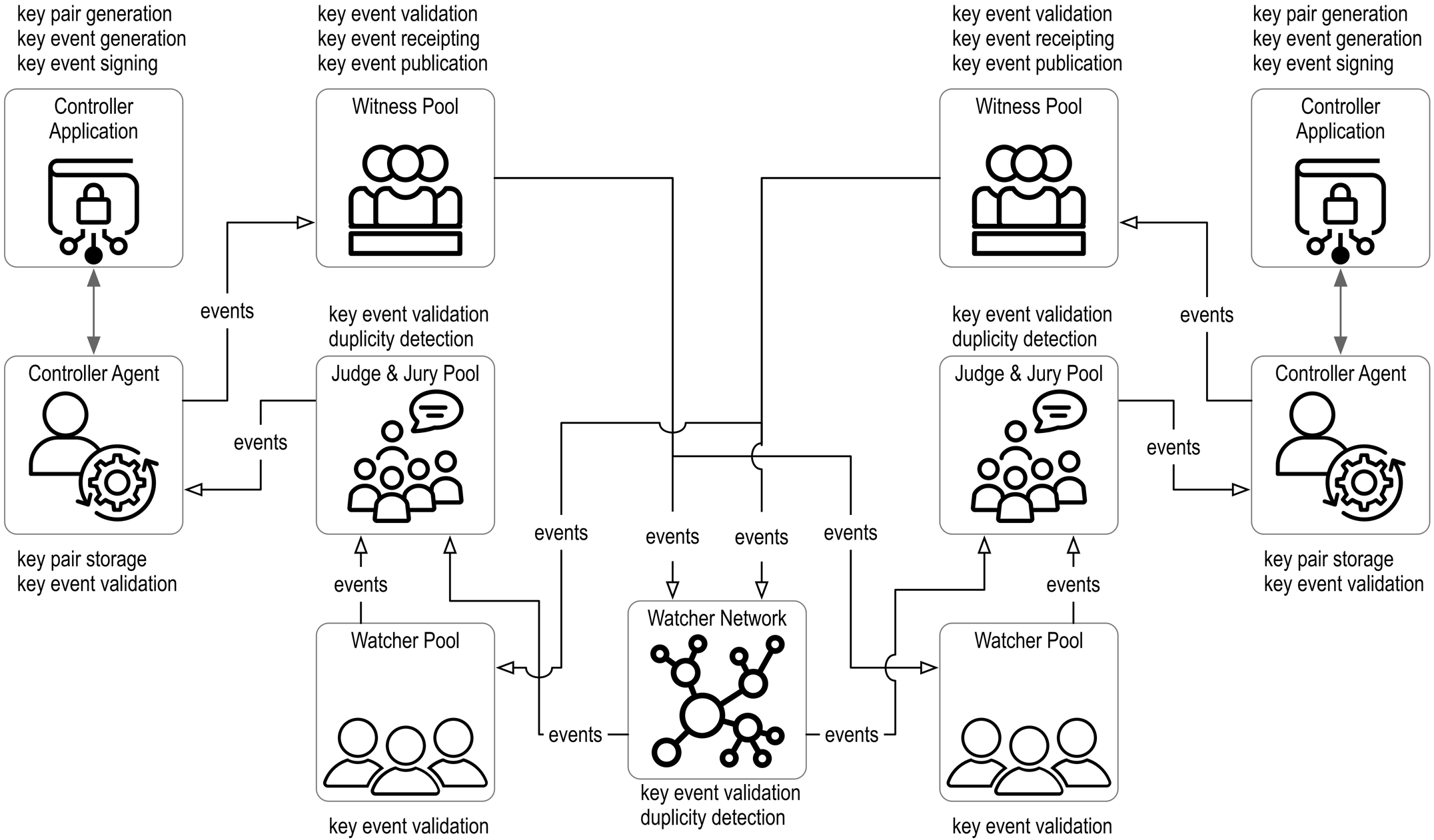


KERI Ecosystem Components: Witnesses and Watchers, Indirect Mode

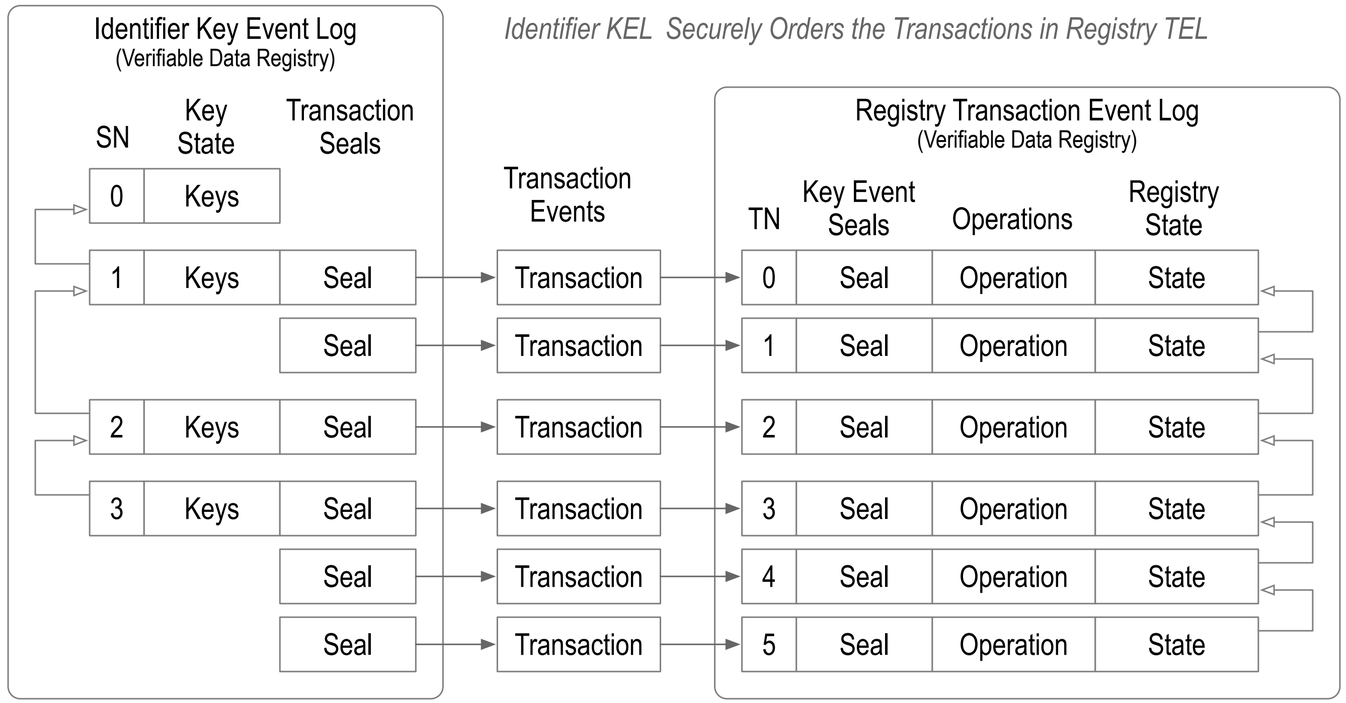


Modular decentralized web based infrastructure without shared governance

KERI Ecosystem Components: Witnesses and Watchers, Indirect Mode



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 (SAID) 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).

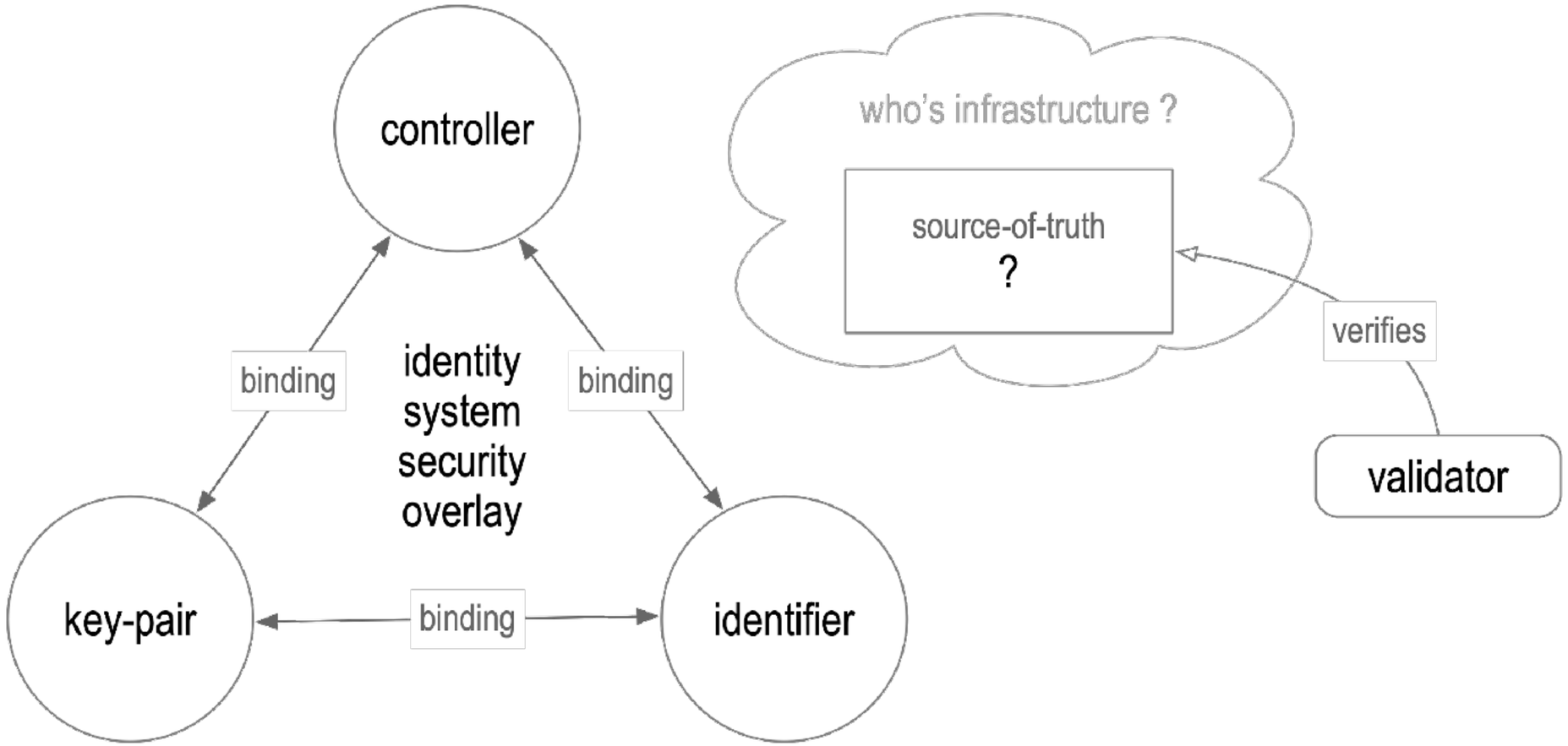
The transactions likewise contain a reference seal back to the key event authorizing the transaction.

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.

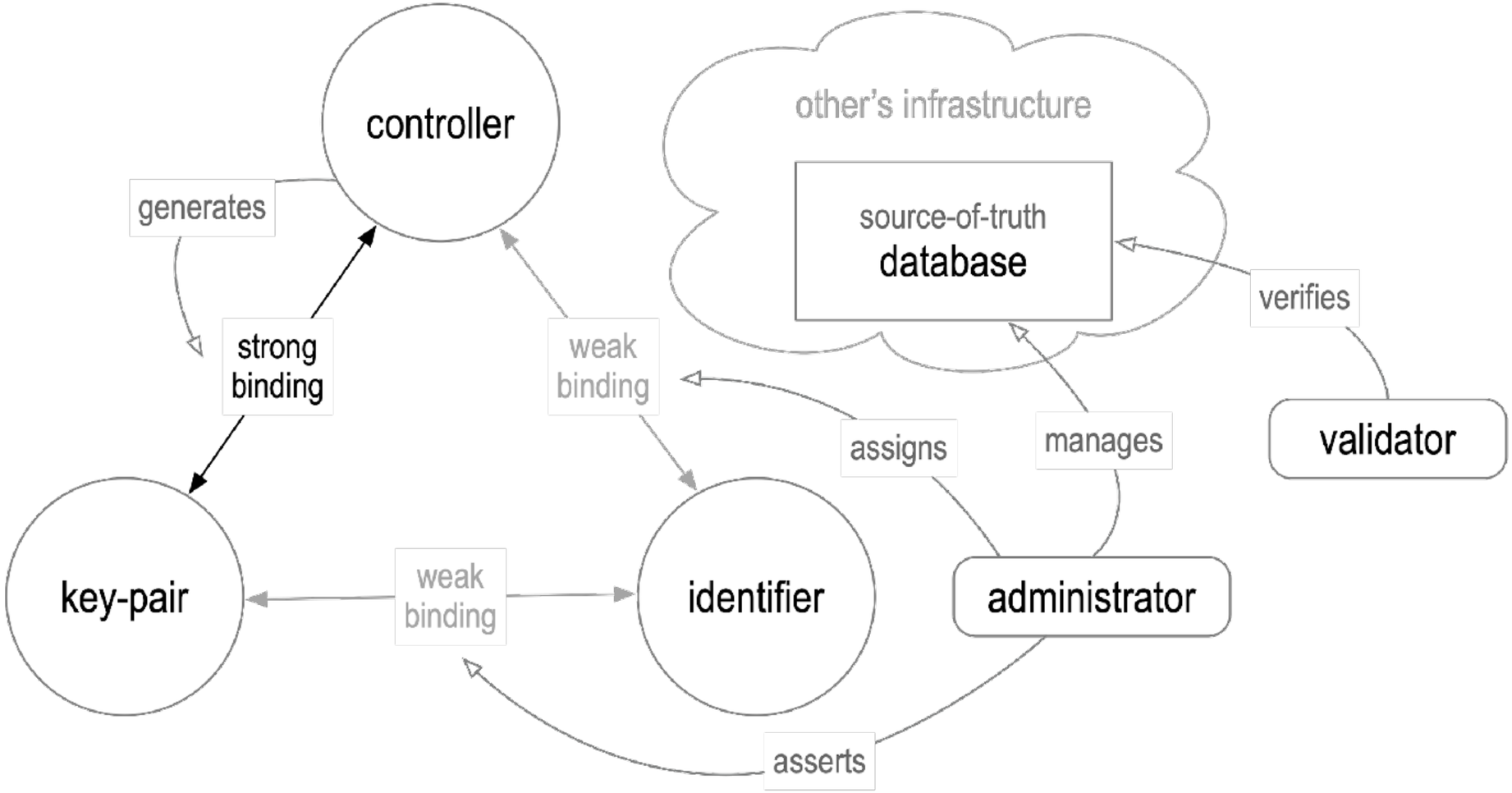
Any validator may therefore cryptographically verify the authoritative state of the registry.

Trust Basis of a Trust Domain



Administrative Trust Basis

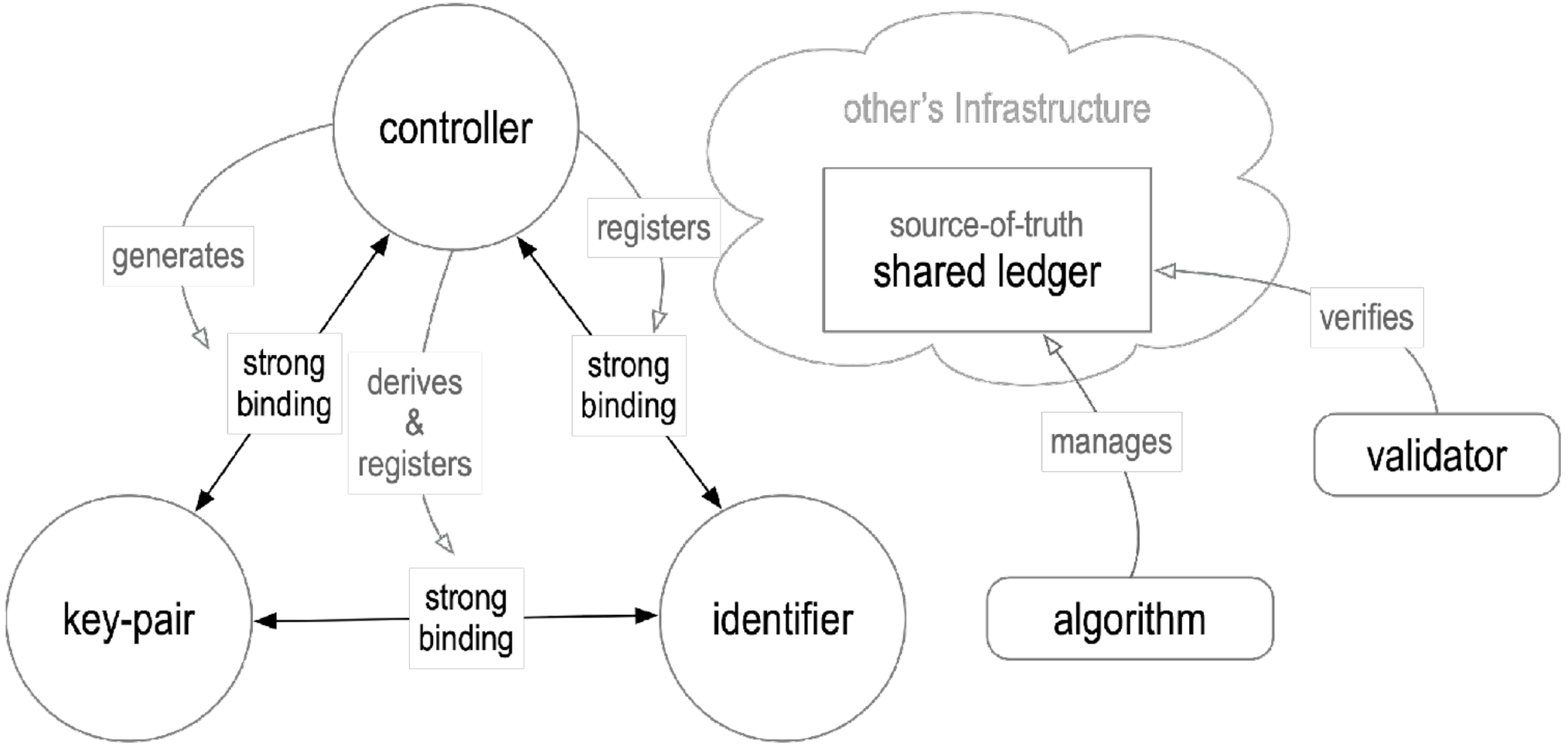
DNS/CA, OIDC IP



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

Algorithmic Trust Basis

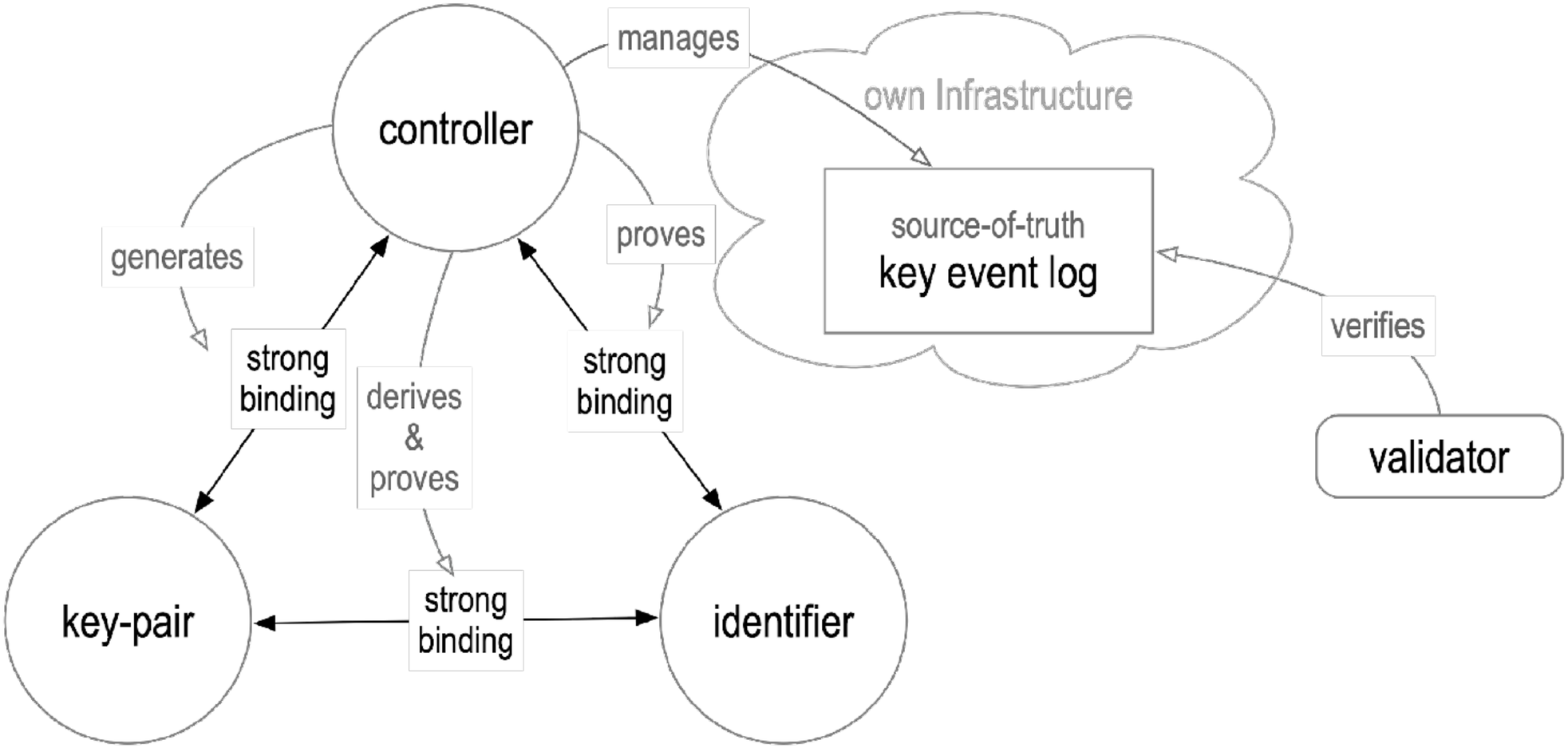
Shared distributed ledgers



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

Use Case Example: Automating the “Signing” of a Mortgage

Bank “Bob” (Lender/Mortgager) wishes to load funds to Individual “Ida” (Borrower/Mortgagee).

How does Bob utilize the duplicity evident infrastructure of Ida to make a live appraisal of compromise of Ida’s key state before transferring funds?

Identity assurance

Accountability/Liability incentives for Ida to be malicious vs. impersonated

Ida digitally commits to the mortgage agreement by binding (sealing/anchoring) SAID (hash) of the agreement in a TEL that is in turn bound (sealed/anchored) to Ida’s KEL, thereby binding the mortgage transaction event to Ida’s current key state.

Ida presents to Bob a bound mortgage agreement and proof of binding.

Bob performs live appraisal to determine evidence of duplicity.

* Verifies TEL, KEL, and Mortgage Agreement bindings

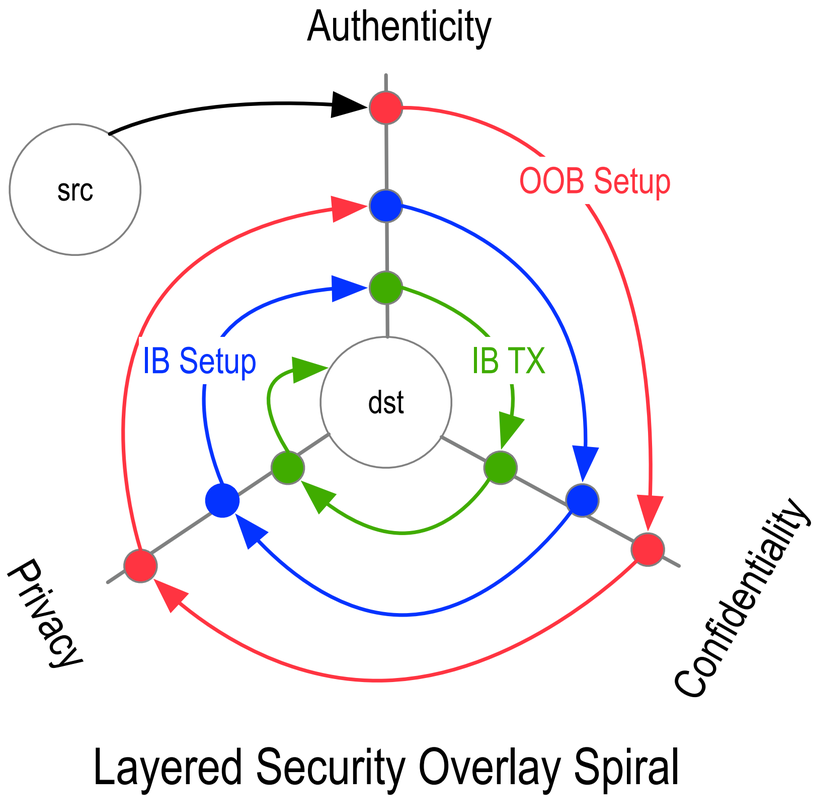
* Characterizes strength of bindings and the relative likelihood of compromise (multi-sig, witness, rotation vs interaction event, delegated vs non-delegated)

*Checks its watchers of Ida’s Witnesses; is there any evidence of duplicity?

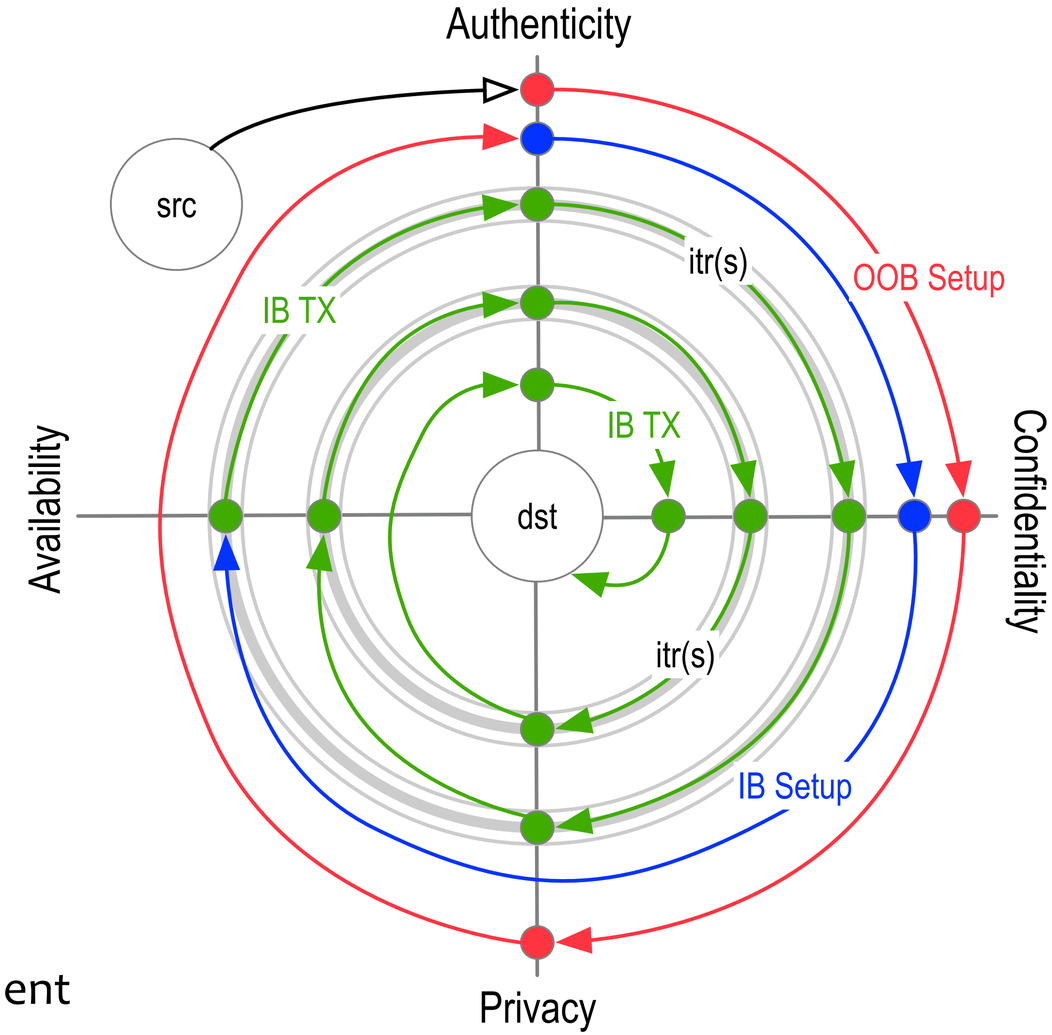
*Waits for some time (some meaningful multiple of network propagation time) for a recovery rotation by Ida that invalidates bindings. This enables Ida Watcher’s ongoing live appraisal to detect and recover from a compromise of her own infrastructure.

Backup Slides

Setups Matter in Layered Security Overlays



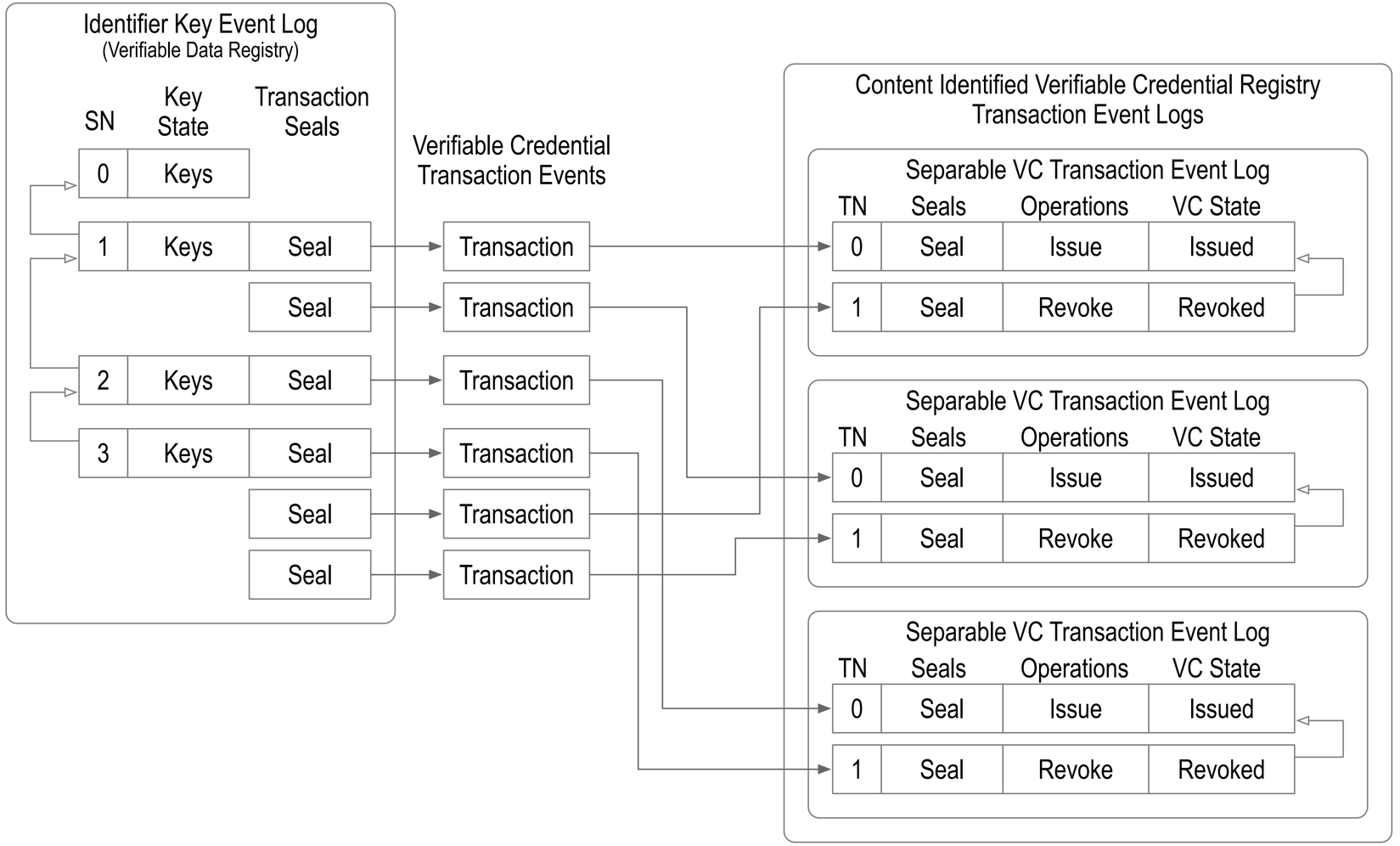
Layered Security Overlay Spiral



Layered Security Overlay Spiral

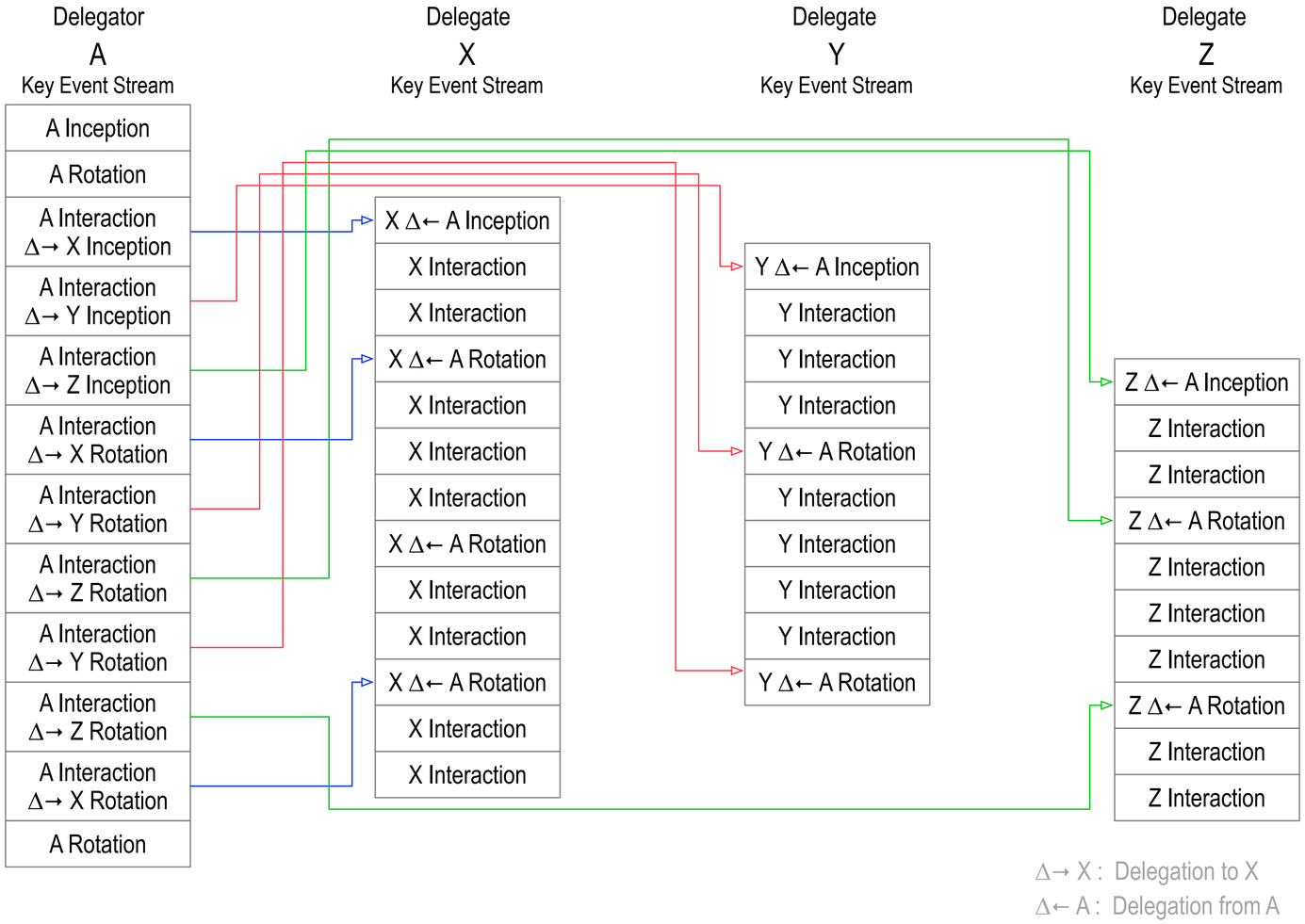
Setups may be application dependent & hence overlay ordering dependent
Each setup requires one OOBA factor to protect against MITM attack
Setups may be costly, especially those with repeated OOBAs
Interactive setups are not as scalable as non-interactive setups
Trade-offs required

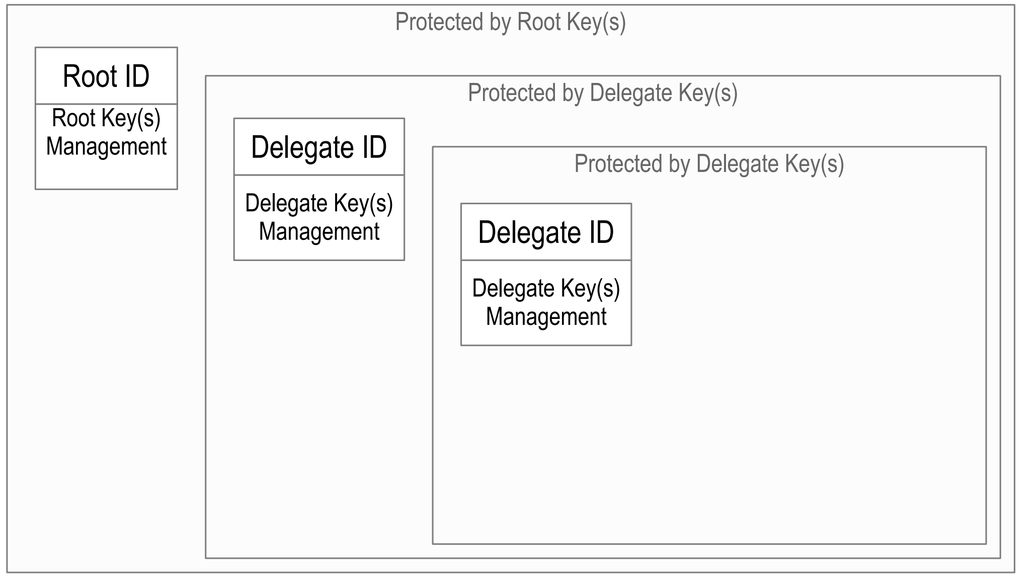
KEL Anchored Issuance-Revocation Registry with Separable VC TELs



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

Identifier Delegation: Scaling & Protection





Hard Problems & Solutions

Moving Data Across Trust Domains.

No Shared Secrets

- No passwords
- No shared encryption keys
- No bearer tokens
- No shared private keys

Key Management (rotation)

True Zero-Trust = Sign Everything

Global Portability At-Scale

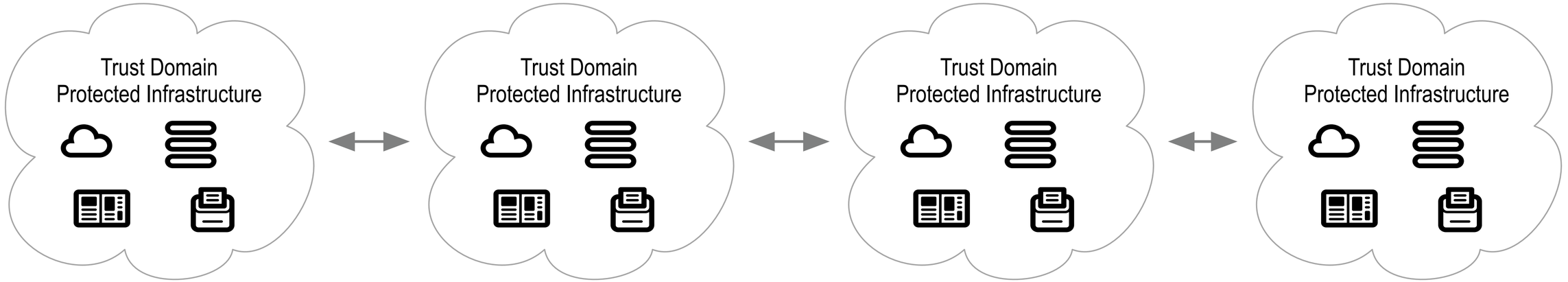
Trust Spanning Protocol (TSP)(SPAC)

Authentic Chained Data Container (ACDC)

Key Event Receipt Infrastructure (KERI)

Composable Event Streaming Representation (CESR)

GLEIF vLEI



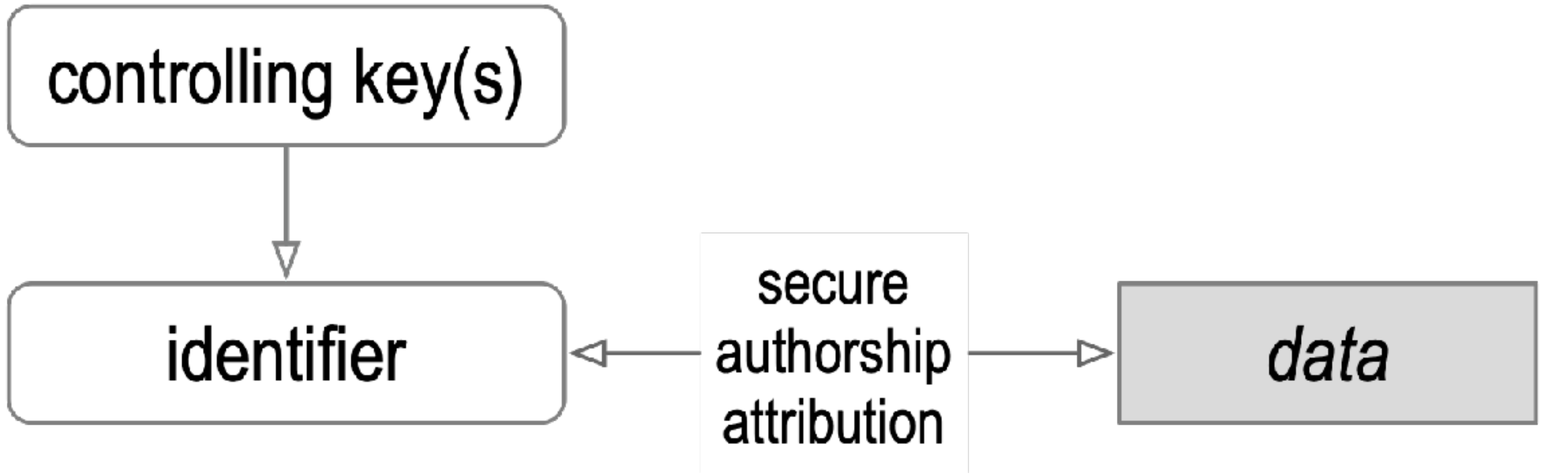
Universal Secure Attribution Problem

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

= Verifiable secure attribution of any communication to its source

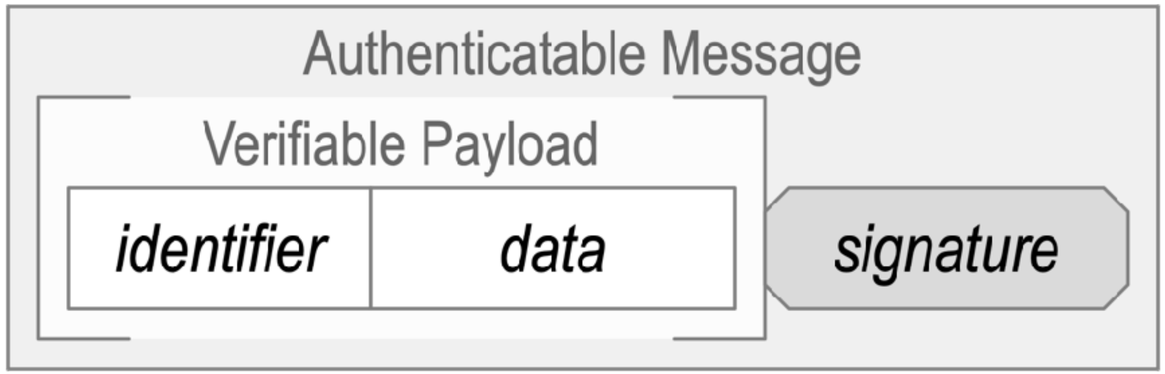
= Authentic data provenance by anyone to anyone from anyone

Solve data provenance to solve security



Identity (-ifier) System Security Overlay

Establish authenticity of IP packet's message payload.



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

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?

Cate
Controller
Log

A

Log V1 Log V2

B A

Log V2

Eve
Validator
Log

Joe
Validator
Log

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

Log V1 Log V2

B A

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?

Cate
Controller
Log

Breaking the promise of global consistency is a provable liability.

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

Service
Controlled by Cate
Log

Truncate Log

?

isolate network

isolate network

A

Delete Log

A

Doug
Global Duplicity
Log

log V1

log V2

Eve
Validator
Log

Ambient Duplicity Detection

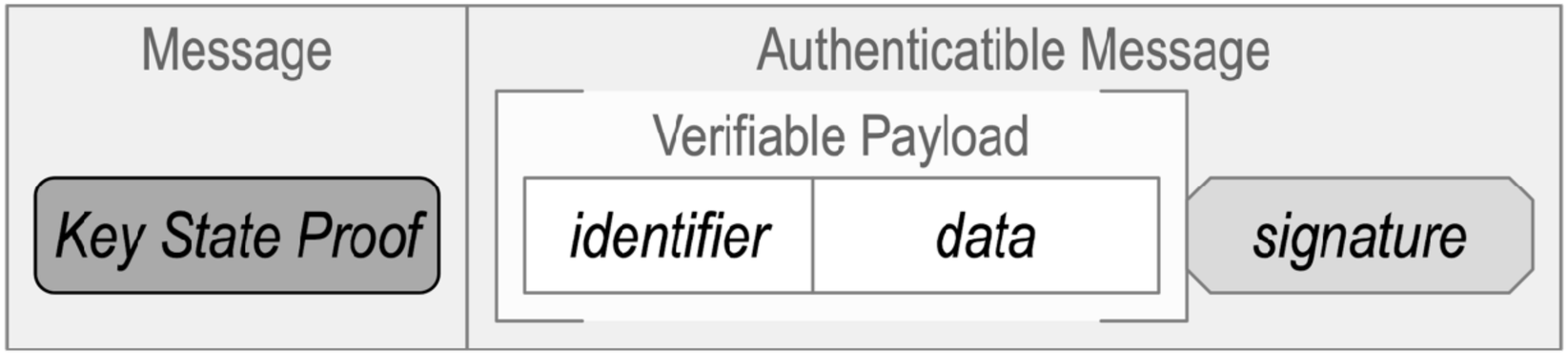
Joe
Validator
Log

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

Identity (-ifier) System Security Overlay

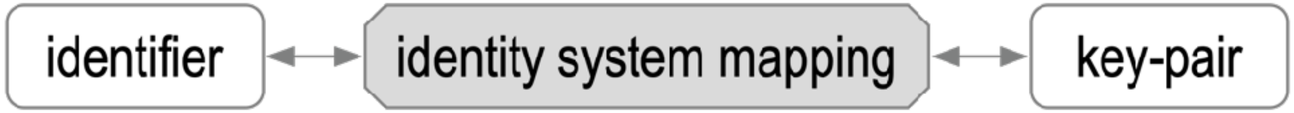
persistent mapping via verifiable data structure of key state changes

Establish authenticity of IP packet's message payload.

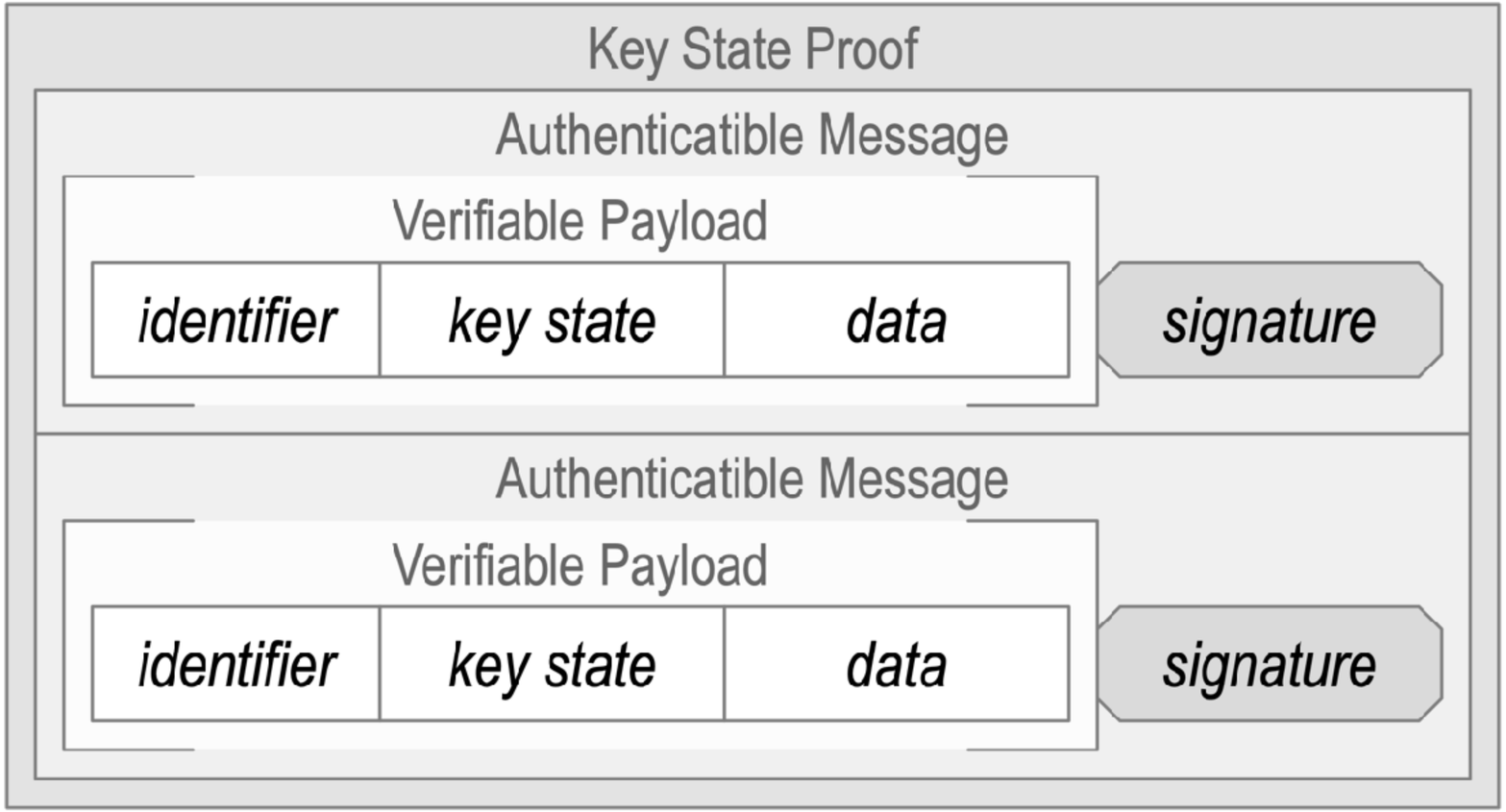


The overlay's security is contingent on the mapping's security.

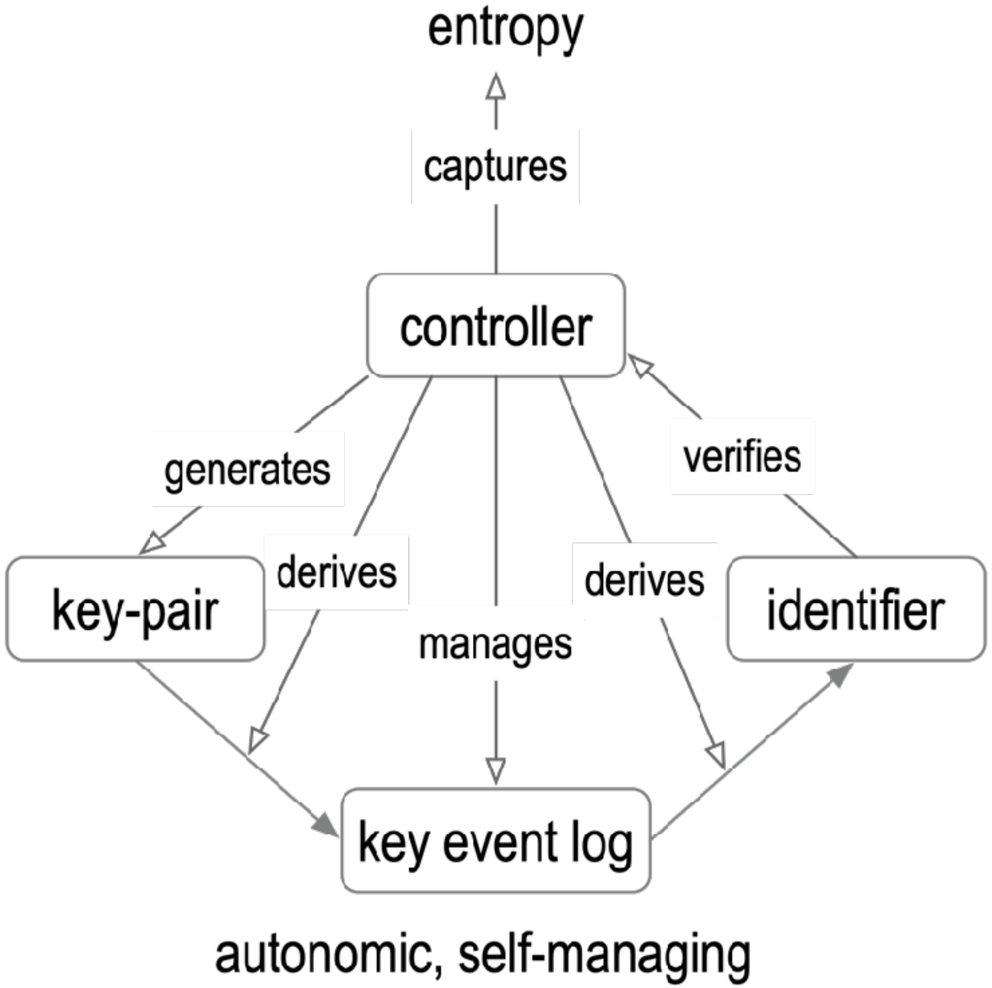
Key State Proof is Recursive Application of Overlay



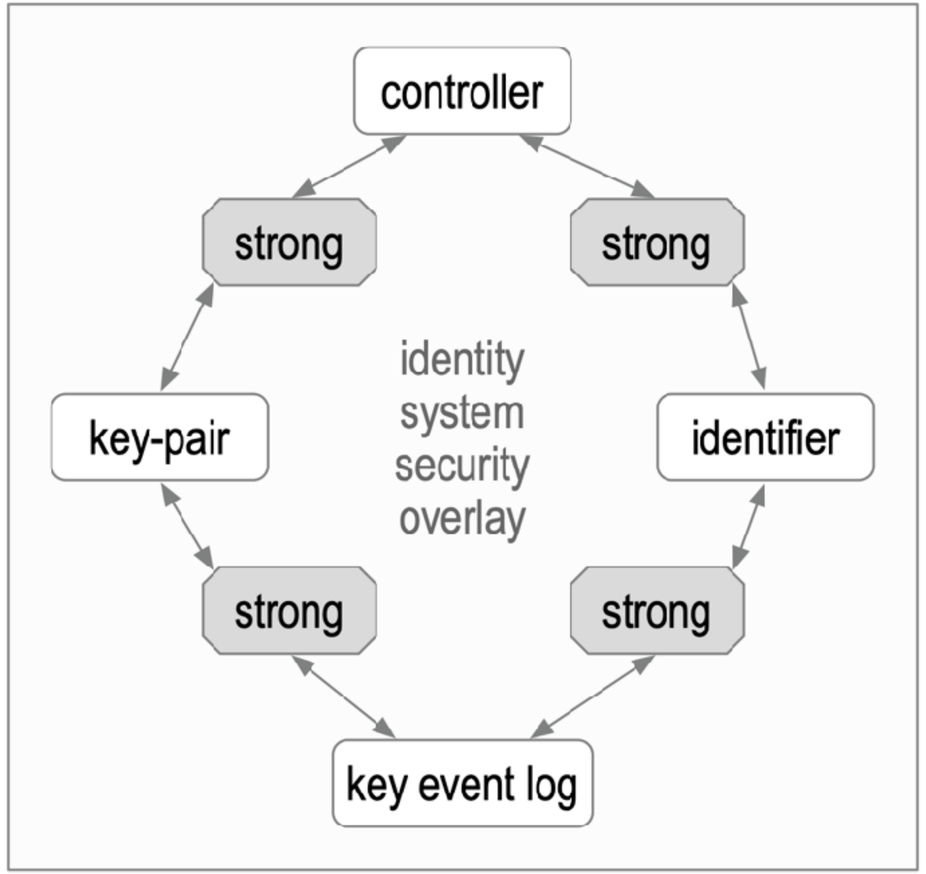
Persistent mapping via verifiable data structure of key state changes



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



autonomic, self-managing

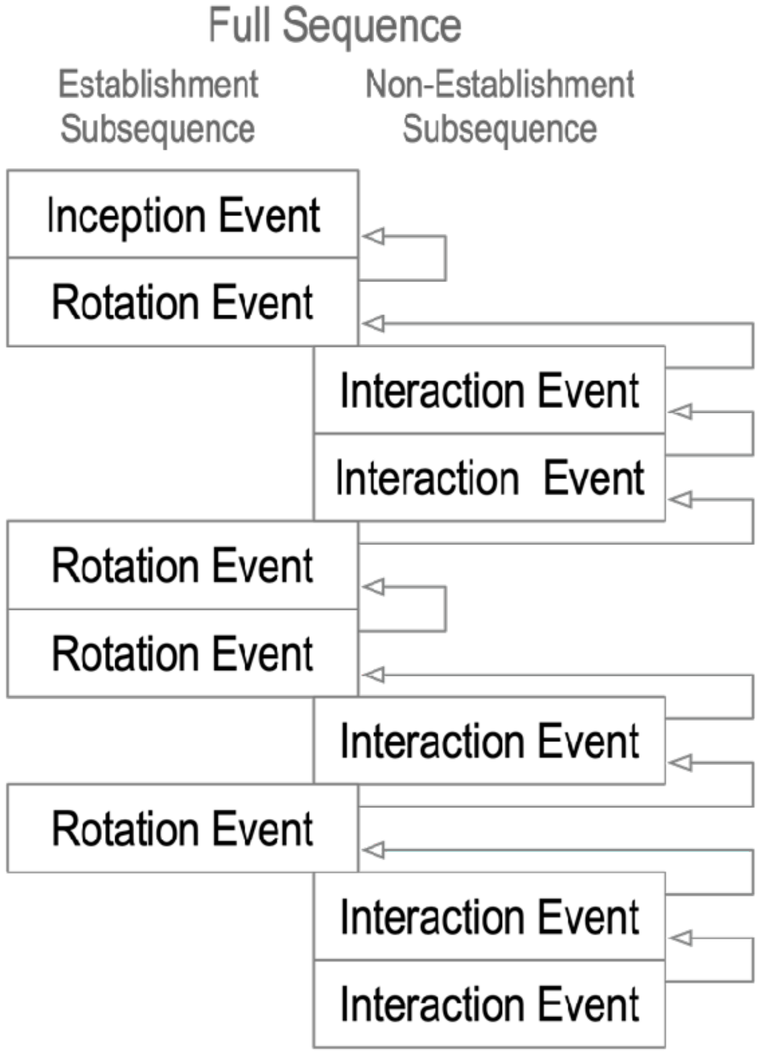


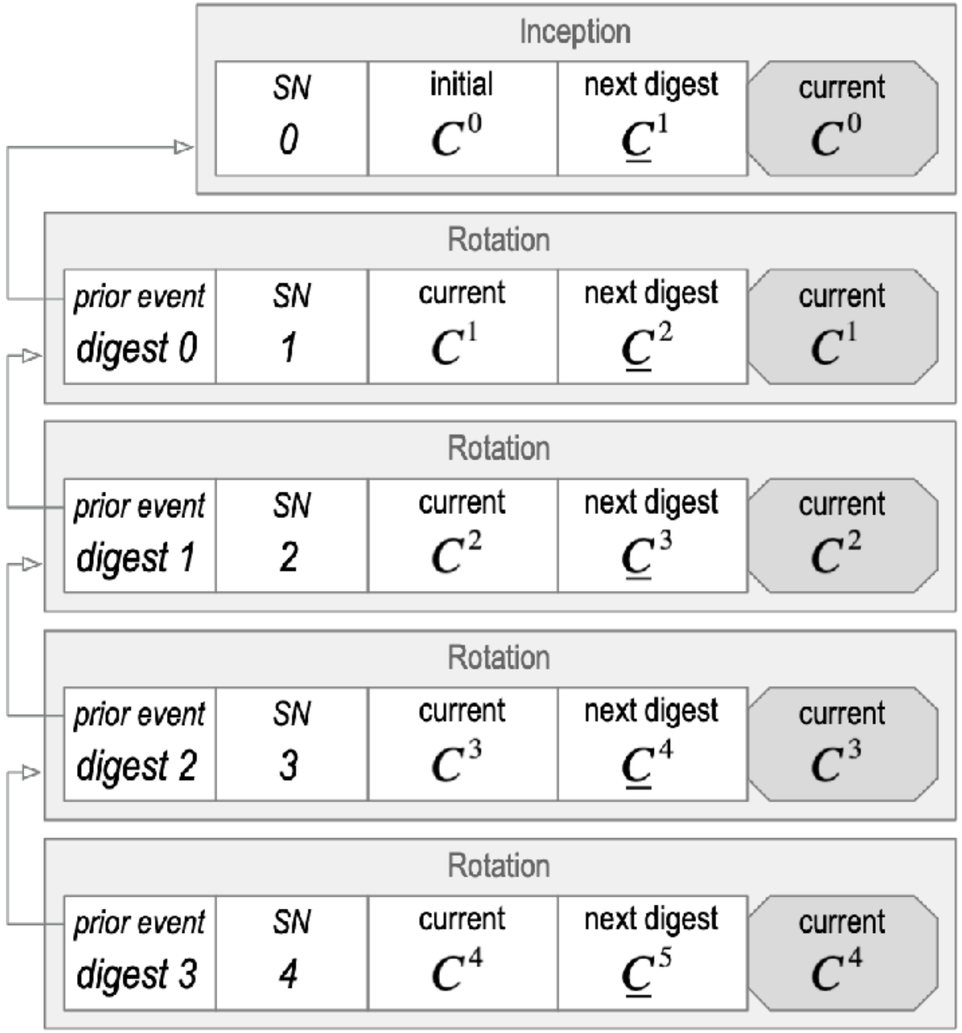
Autonomic Identifier Issuance Tetrad

cryptographic root-of-trust with verifiable persistent control

Solution: Key Pre-Rotation

duplicity evident
verifiable data
structure

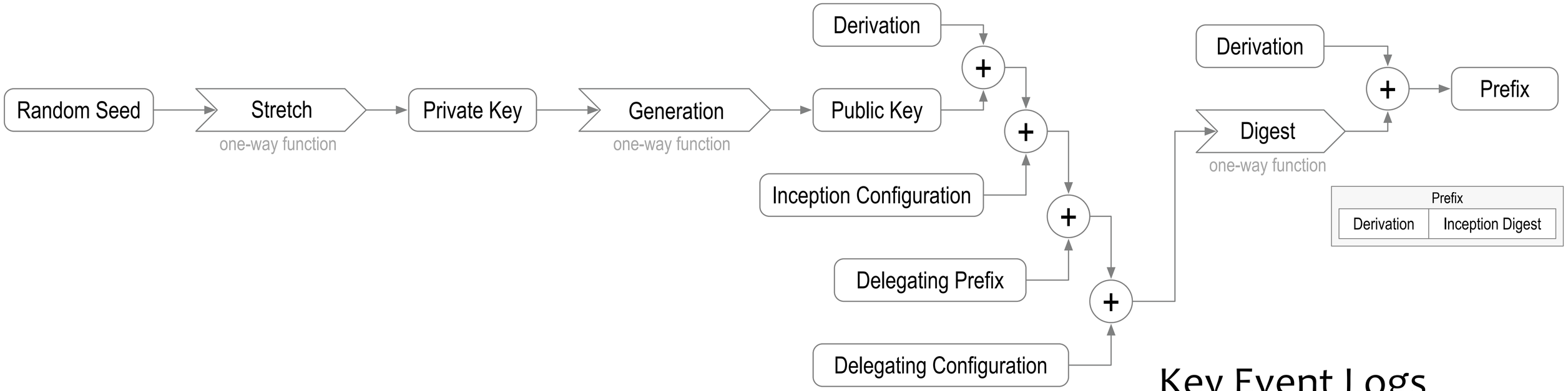


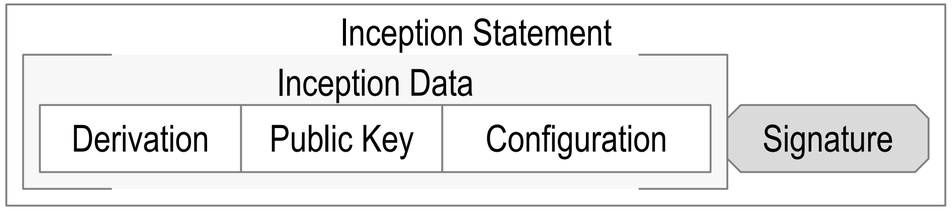


Digest of *next* key(s) makes pre-rotation post-quantum secure

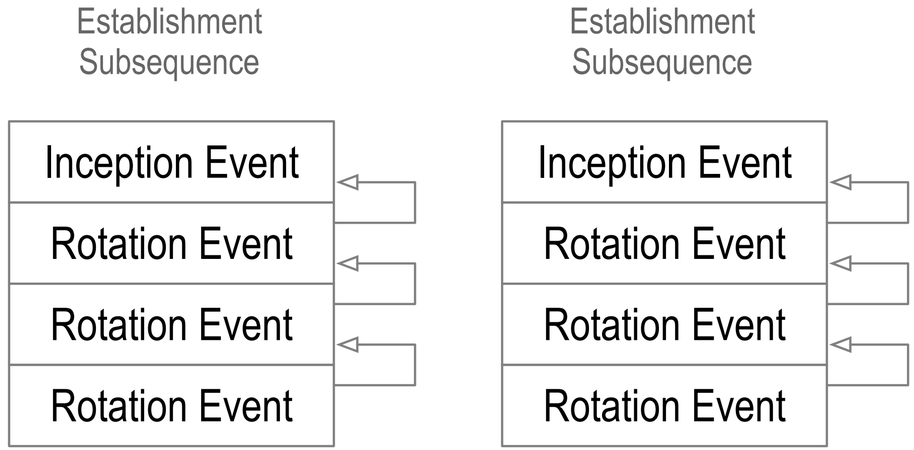
D.J. Bernstein: https://cr.yp.to/hash/collisioncost-20090517.pdf

Delegated Identifiers





Key Event Logs



EXq5YqaL6L48pf0fu7IUhL0JRaU2_RxFP0AL43wYn148

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

Resources

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

KERI/ACDC Community: (meetings, open source code Apache2, specification drafts)
https://github.com/WebOfTrust
https://github.com/WebOfTrust/keri

ToIP: (meetings, specifications OWF License)
https://trustoverip.org/
https://wiki.trustoverip.org/display/HOME/ACDC+(Authentic+Chained+Data+Container)+Task+Force
https://wiki.trustoverip.org/display/HOME/Trust+Spanning+Protocol+Task+Force

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

healthKERI:
https://healthkeri.com/

KERI GLEIF TRUST Over IP FOUNDATION ISO DirectTrust® healthKERI