Key Event Receipt Infrastructure

An Identity System Security Overlay for the Internet

A Trust Spanning Layer for the Internet

Samuel M. Smith Ph.D.
https://keri.foundation
https://keri.one/resources/
sam@keri.one

Resources

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

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

ToIP: KSWG, OWF License, within Linux Foundation, Decentralized Trust (meetings, specifications)
https://trustoverip.org/
https://lf-toip.atlassian.net/wiki/spaces/HOME/pages/56819755/KERI+Suite+Working+Group

Protocols and Specifications:
CESR: https://github.com/trustoverip/tswg-cesr-specification
KERI: https://github.com/trustoverip/tswg-keri-specification?tab=readme-ov-file
ACDC: https://github.com/trustoverip/tswg-acdc-specification
TSP: https://github.com/trustoverip/tswg-tsp-specification
VLEI: ISO
VVP: https://www.ietf.org/archive/id/draft-hardman-verifiable-voice-protocol-00.html
RAET: https://github.com/RaetProtocol/raet (2013)

Adoptions

GLEIF (vLEI): Organizational Identity (FSB Finance)
https://www.gleif.org/en/lei-solutions/gleifs-digital-strategy-for-the-lei/introducing-the-verifiable-lei-vlei

EBA P3DH: Banking
https://www.eba.europa.eu/risk-and-data-analysis/pillar-3-data-hub

DirectTrust: HISP Certification and vLEI
https://directtrust.org/

healthKERI: SASE for Health Care Data
https://healthkeri.com/

Provenant (GSMA): Telecom spam VVP and vLEI
https://provenant.net/

Cardano Foundation (Veridian): (Web3 Identity Layer)
https://cardanofoundation.org/veridian

Utah State: SEDI State Endorsed Digital Identity
S.B. 260 https://le.utah.gov/Session/2025/bills/introduced/SB0260.pdf

User Public/Private Key Management: *Then* (< 2015) and *Now*

Then: impossibility of user-managed private keys

Result: federated identity: IDPs with PKI-DNS/CA, OAuth

Now: mobile devices/HSMs with MFA, secure boot/ enclaves, password management apps

Result: practical user-managed private keys

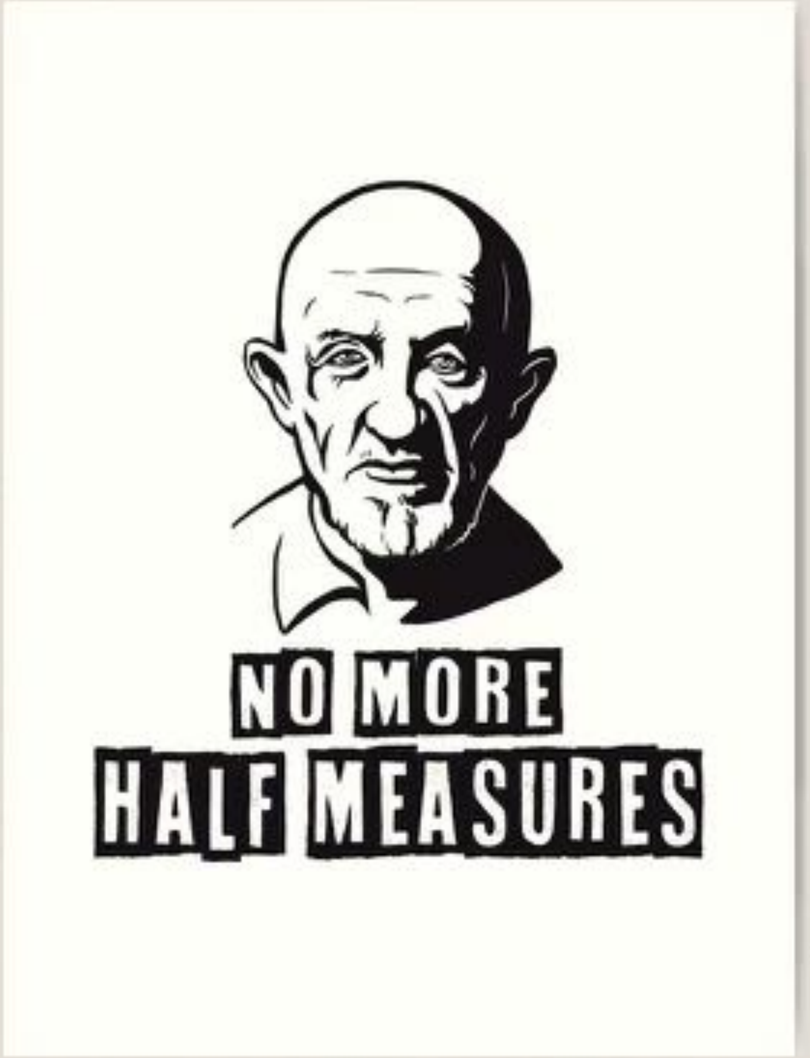
Then: *good enough* security from federated identity

Result: session security, perimeter security, weak auth-crypt for AuthN

Now: identity based on user-managed private key(s)

Result: KERI

Security First, Always



Minimally Sufficient Means

Adoptable *White Magic* Crypto

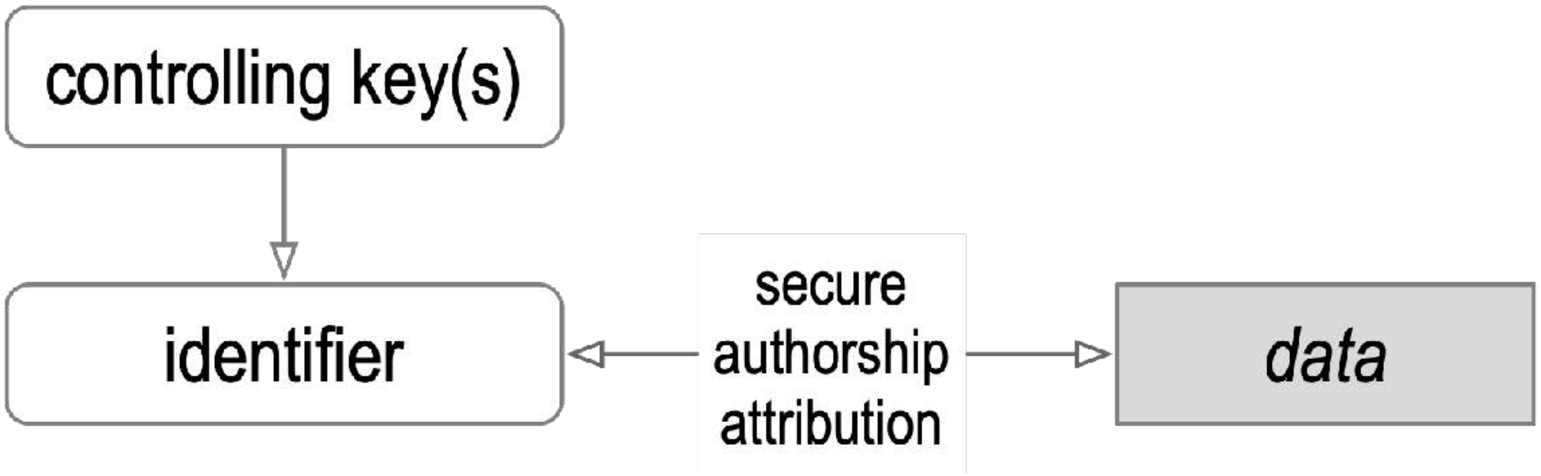
First Principles Approach to the Universal Secure Attribution Problem

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

Authentic data provenance by anyone to anyone from anyone, i.e. *universal*

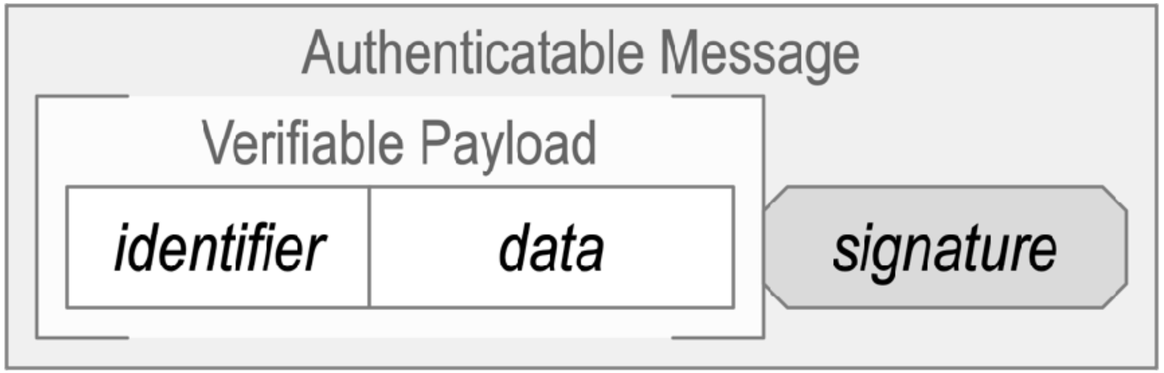
Cryptographically verifiable, non-repudiable secure attribution of any communication to its source

Solve authentic data provenance via universal secure attribution



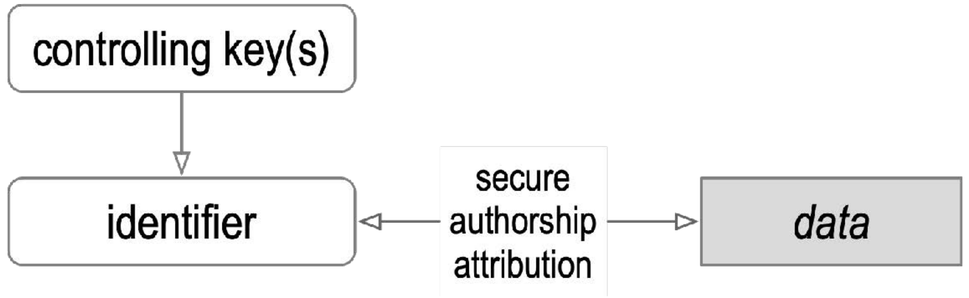
Identity (-ifier) System Security Overlay

Establish authenticity (securely attribute source) of message payload.



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

Private Key Management



The continued use of private keys *exposes* them to *side-channel* attacks.

This exposure *weakens* private keys over time (as the likelihood of a successful attack rises).

Thus, from time to time, private keys must be revoked and replaced, i.e., rotated

Conventional PKI must re-establish the root-of-trust with each rotation

Thereby making rotation highly vulnerable to attack

Compromised private keys result in loss of control over the identifier

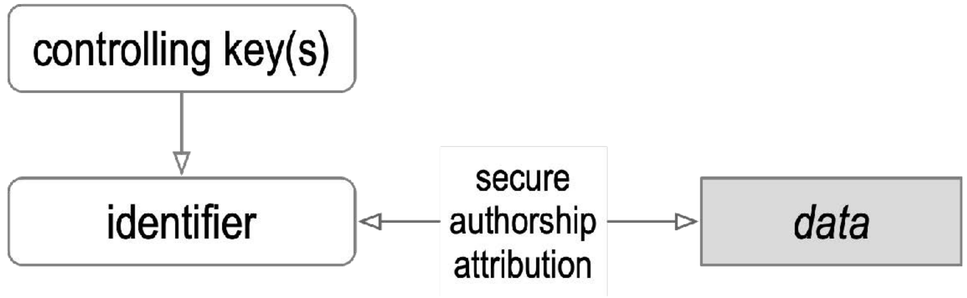
This breaks the chain-of-trust-of-control over the identifier

Key compromise recovery must detect and regain control over the identifier

Key rotation with key compromise recovery is the *hard problem* of private key management

KERI solves this hard problem

Identifier Theory

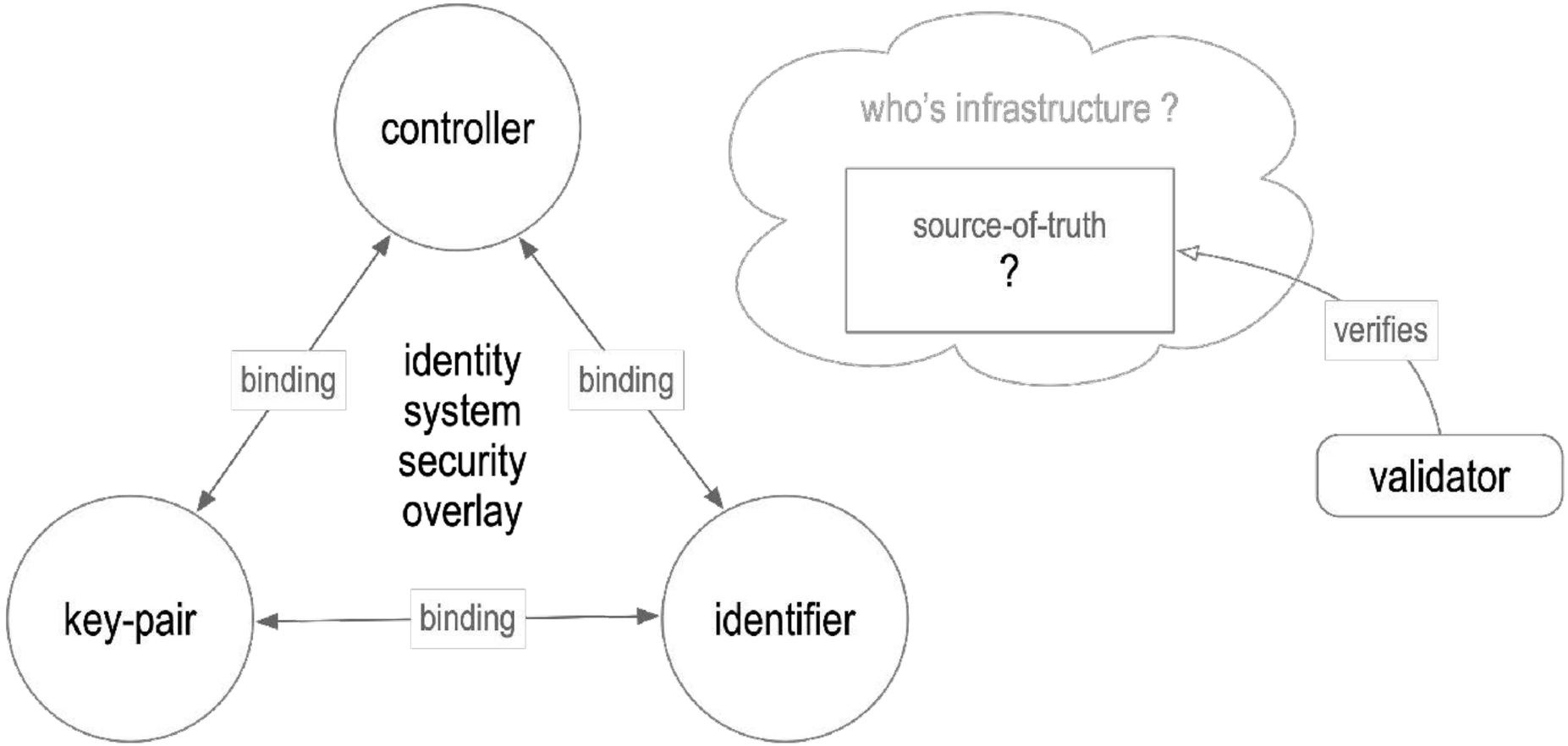


Short-term key-pair with public key as identifier = ephemeral identifier
Long-term key-pair with public key as identifier = less ephemeral identifier
Cryptographically derived identifier controlled by rotatable key-pair(s) = persistent (unbounded-term) identifier

Trust Basis of a Trust Domain

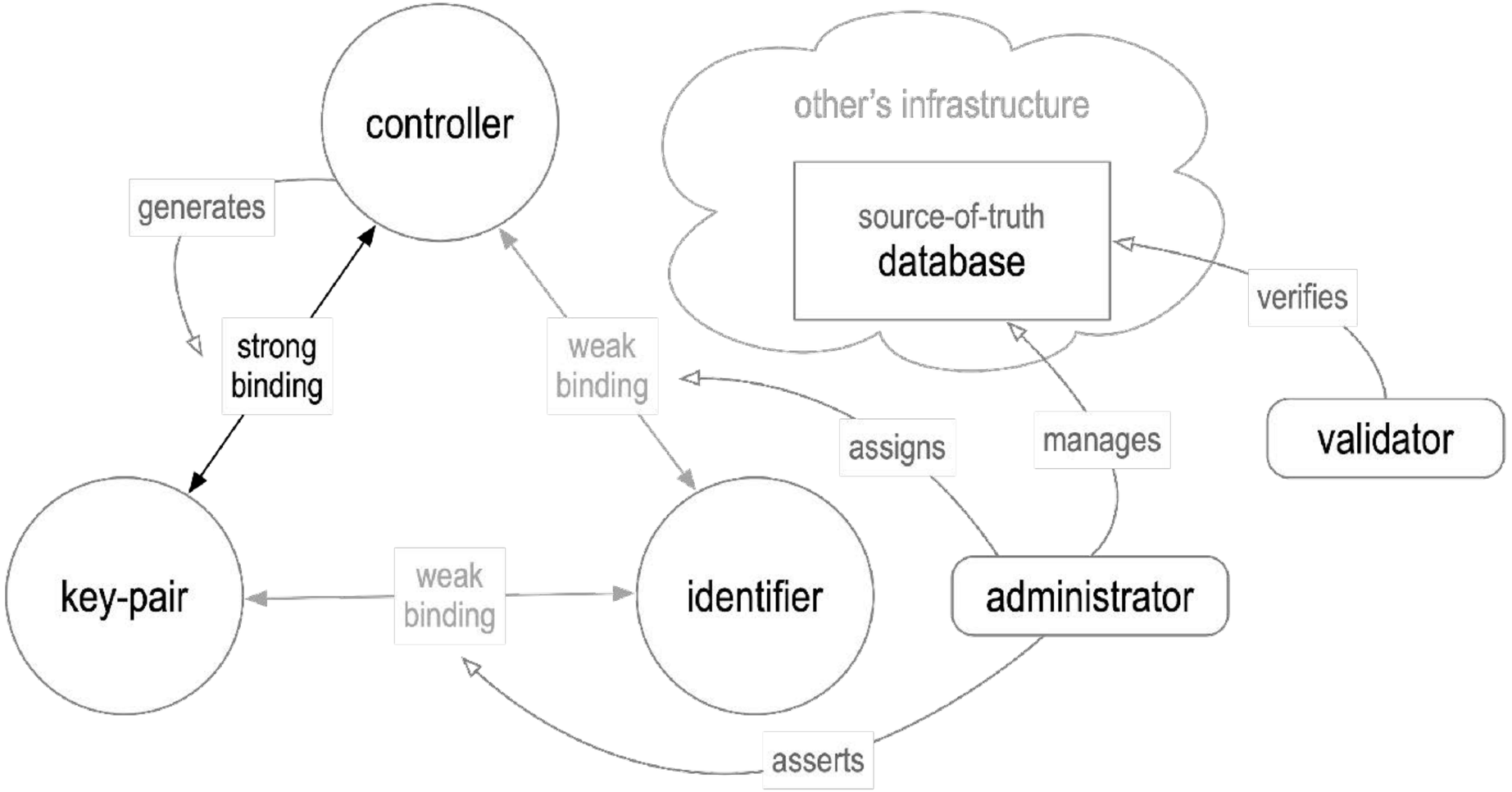
A *trust basis* binds controllers, identifiers, and key-pairs.

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



Administrative Trust Basis

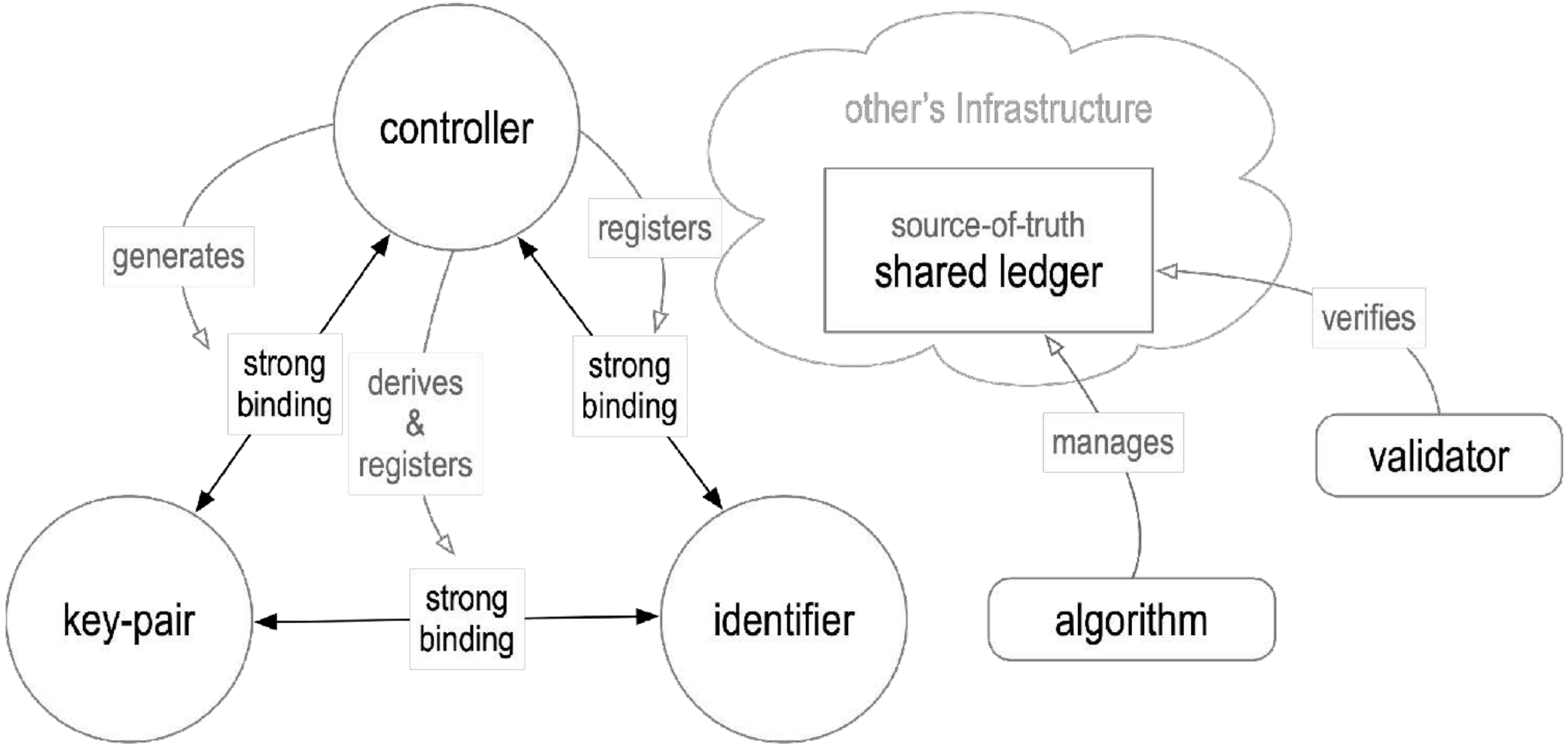
DNS/CA, OIDC IP



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

Algorithmic Trust Basis

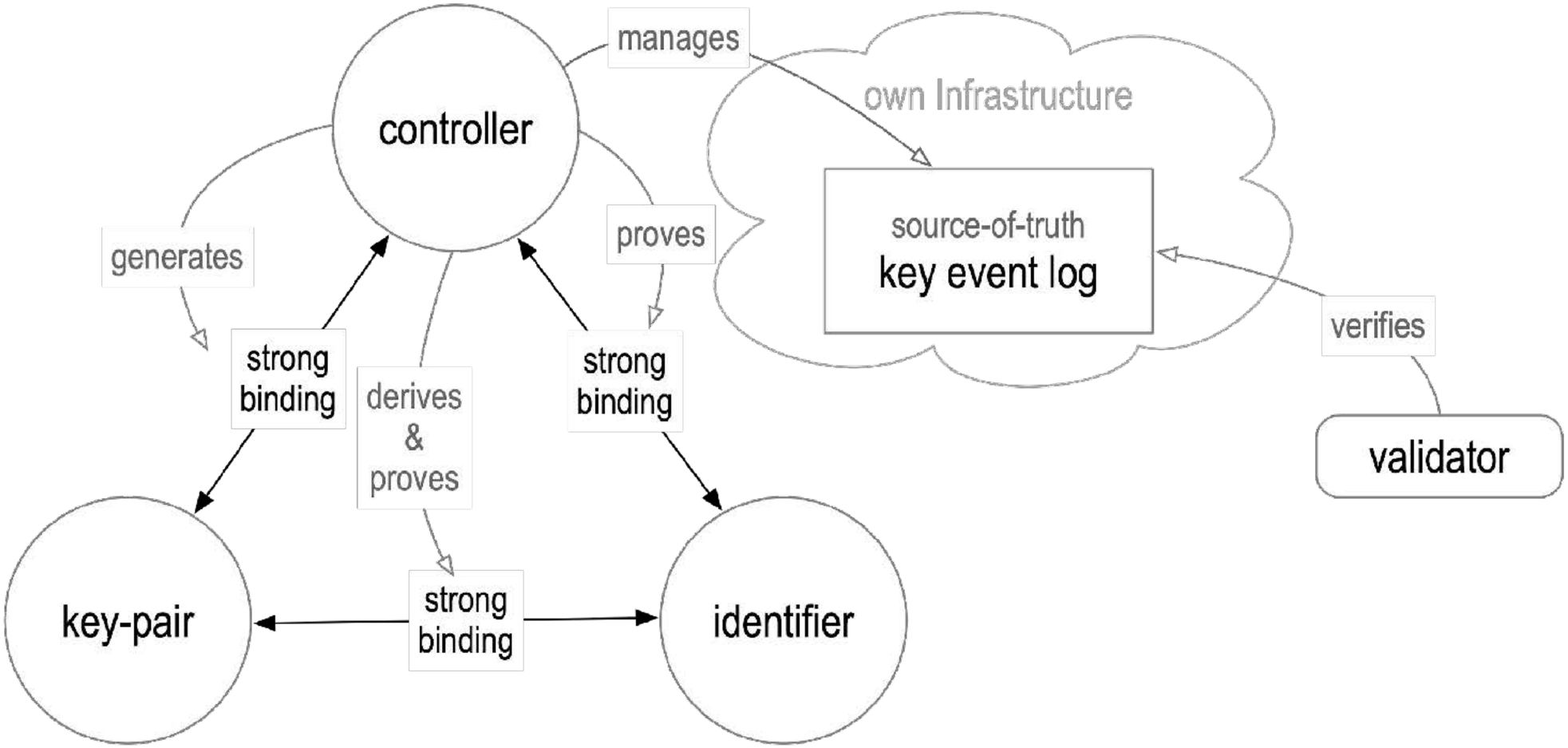
Shared distributed ledgers



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

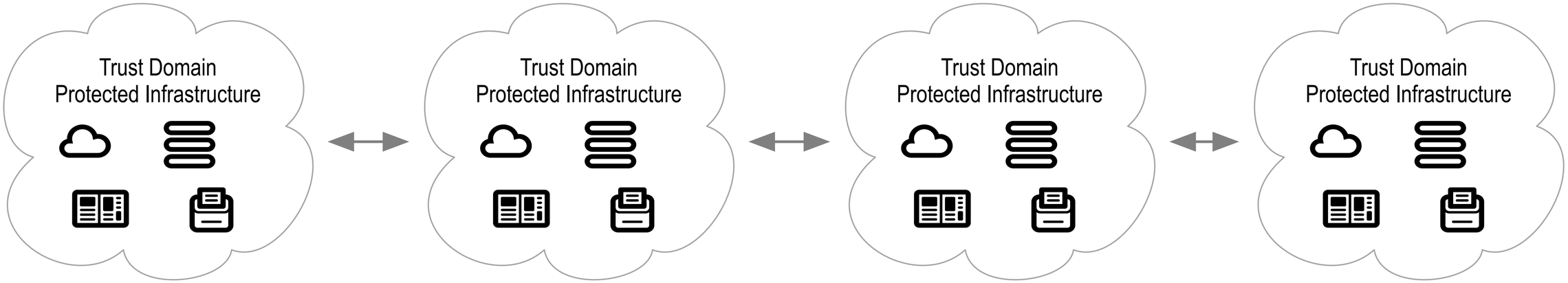
Autonomic Trust Basis

Cryptographic proofs via verifiable data structures



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

Autonomic Trust Basis Enables Solving the Hard Problem of Moving Data Across Trust-Domains



Globally portable, at-scale

No identity theft (fraud-free cross-domain)

No shared secrets as primary authenticators

- no passwords
- no DH encryption keys
- no bearer tokens
- no PII

Security Properties Teaser (*Spoiler Alert*)

All exploits must begin with the compromise of (asymmetric) private signing keys

Any exploit that compromises private key(s) must be nearly instantaneously detectable

Detected private key compromise must be automatically recoverable using quantum-safe one-time-use pre-rotated keys

Controller Key state may be made highly available via own witness pool

Private key compromise and hence exploitation may be made exponentially more difficult through a combination of three threshold structures: multi-signature, witness pools, and delegated identifiers

Validator's must be protected from dead (stale) key compromise via first seen policy enforced by own watcher network

Validators must be protected from live key compromise via duplicity evident property w.r.t. controller's key state

Reconciliation policies must enable exploited controllers to recover from a live key compromise with respect to validator 's

Validators may be protected from malicious eclipse attacks on key state via the duplicity-evident watcher network

Issuances (AuthZ, Entitlements) may be similarly protected by anchoring to the key state

Unbounded term issuances must be verifiable despite key rotation when anchored to the key state

High-stakes issuances may gain extra protection by interleaving anchoring with waiting periods that force in-stride detectability of live exploit attempts, thereby denying success.

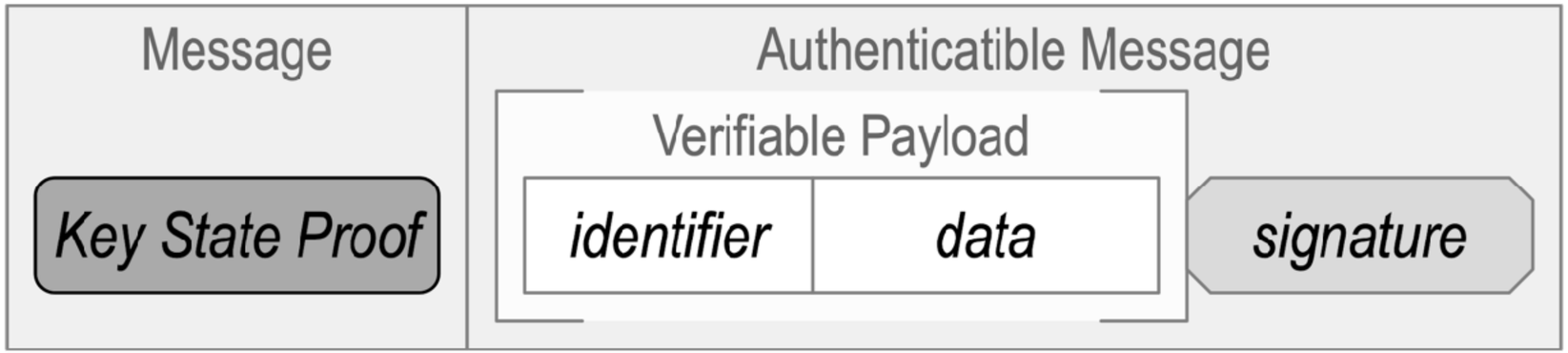
Everything must be provable via verifiable data structures, no trusted third parties, no shared secrets, no platform lock-in

Infrastructure must be highly portable, with no shared governance between the controller's witness network and the validator's watcher network

Identity (-ifier) System Security Overlay

persistent mapping via verifiable data structure of key state changes

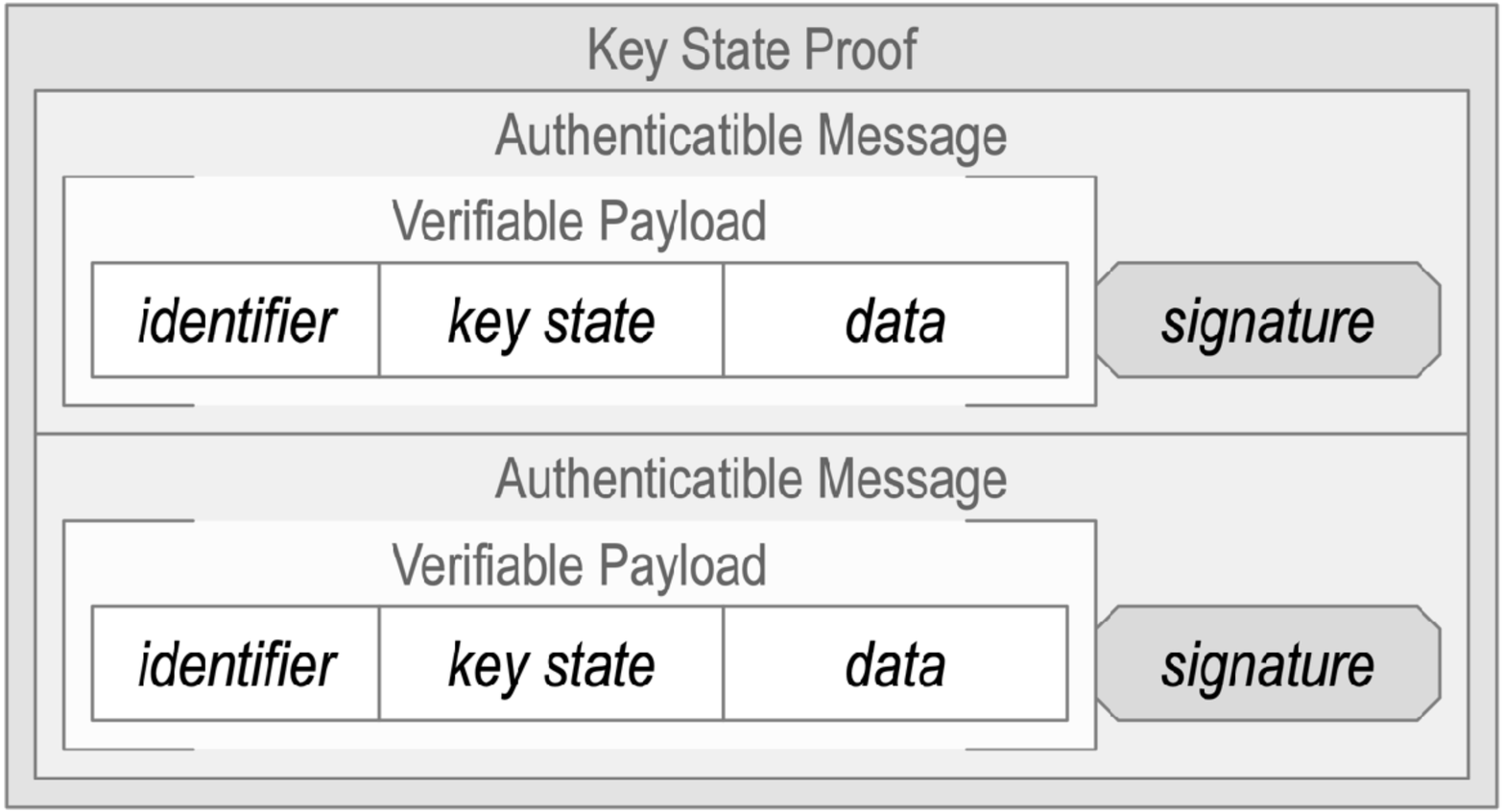
Establish authenticity (securely attribute source) of the message payload



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

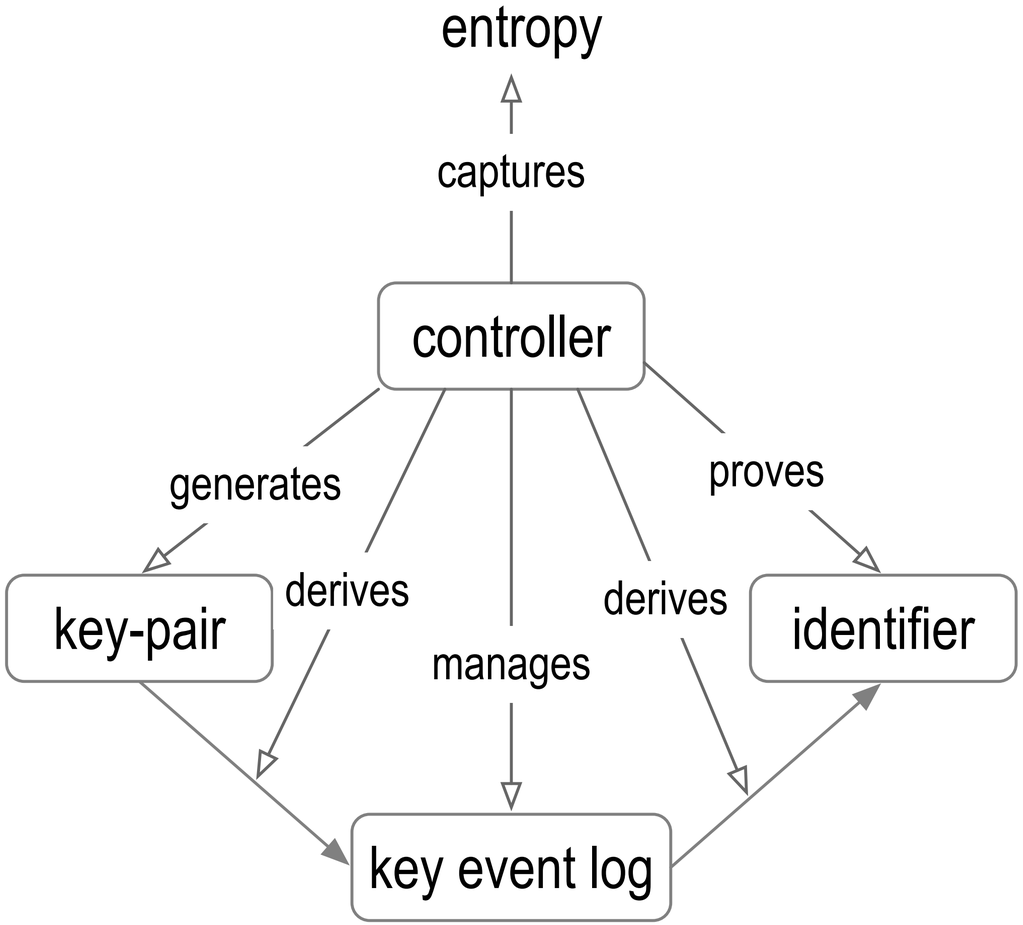
Key State Proof is Recursive Application of Overlay

Persistent mapping via a verifiable data structure of key state changes

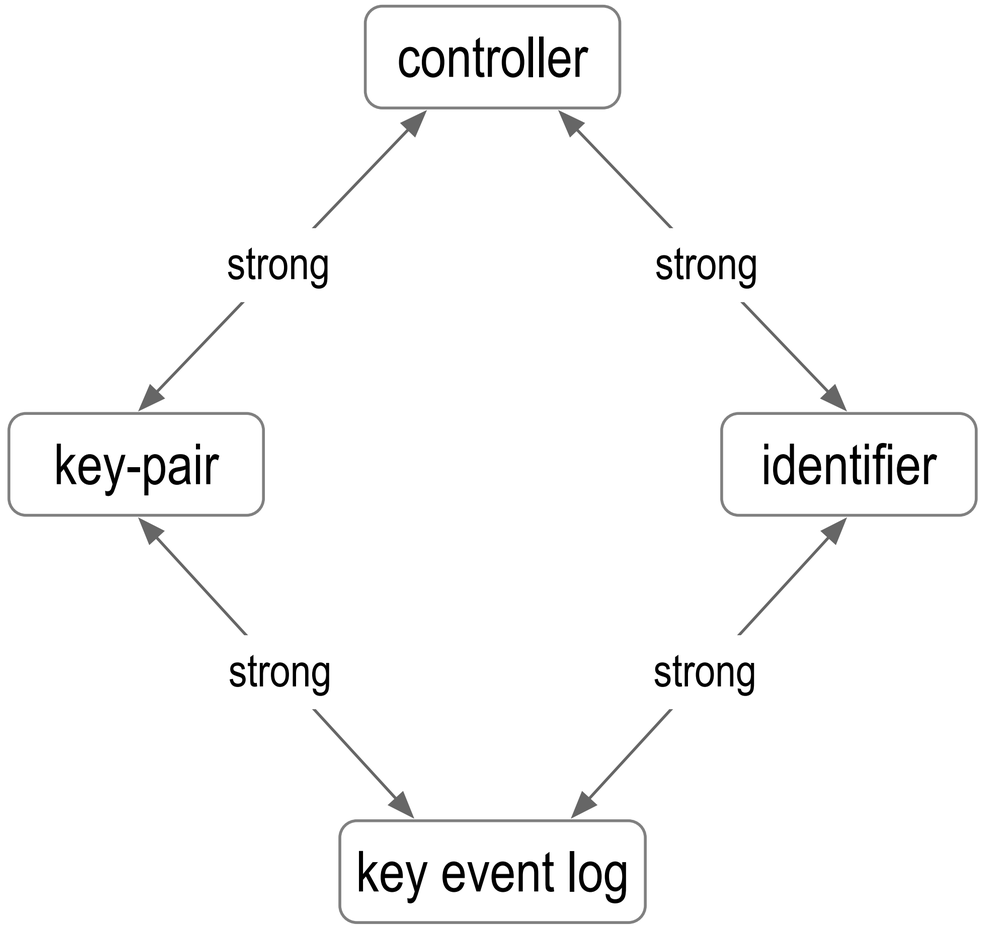


universally verifiable duplicity evident append-only backward and forward-chained key event log

Autonomic Identifiers (AIDs): Issuance and Binding
How to construct a cryptographic root-of-trust



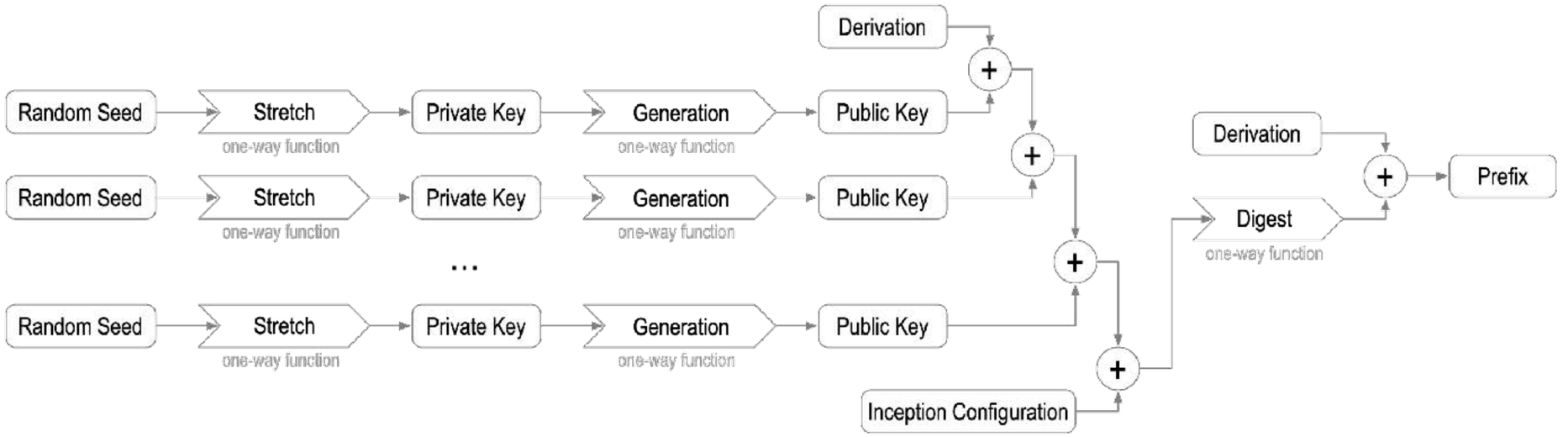
autonomic, self-managing

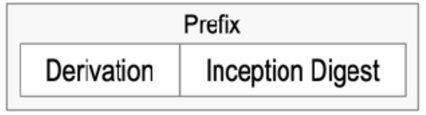


autonomic identifier binding tetrad

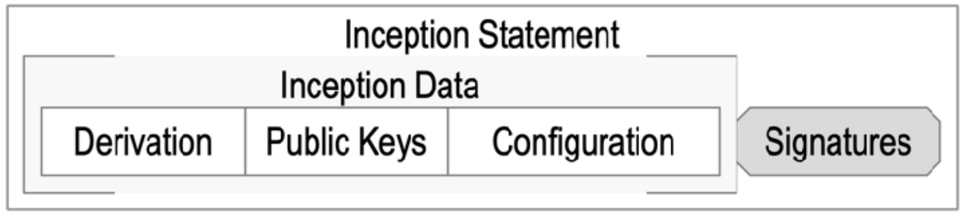
cryptographic root-of-trust with verifiable persistent control over the AID

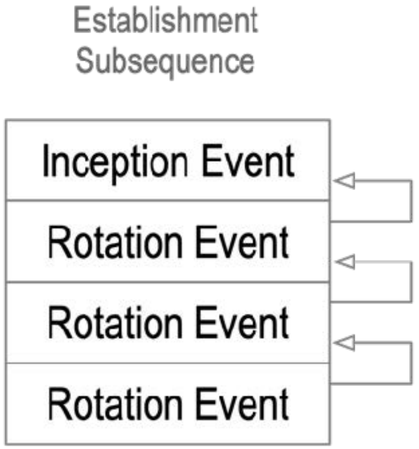
Cryptographic Root-of-Trust: Autonomic Identifier (AID)





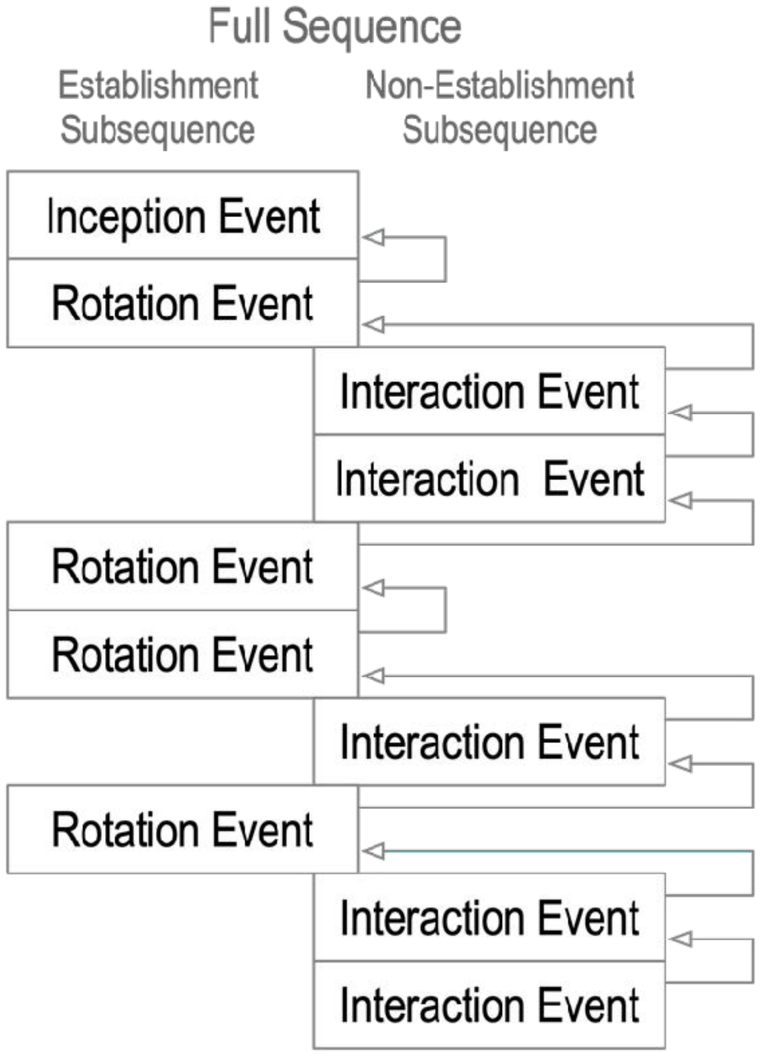
Key Event Log



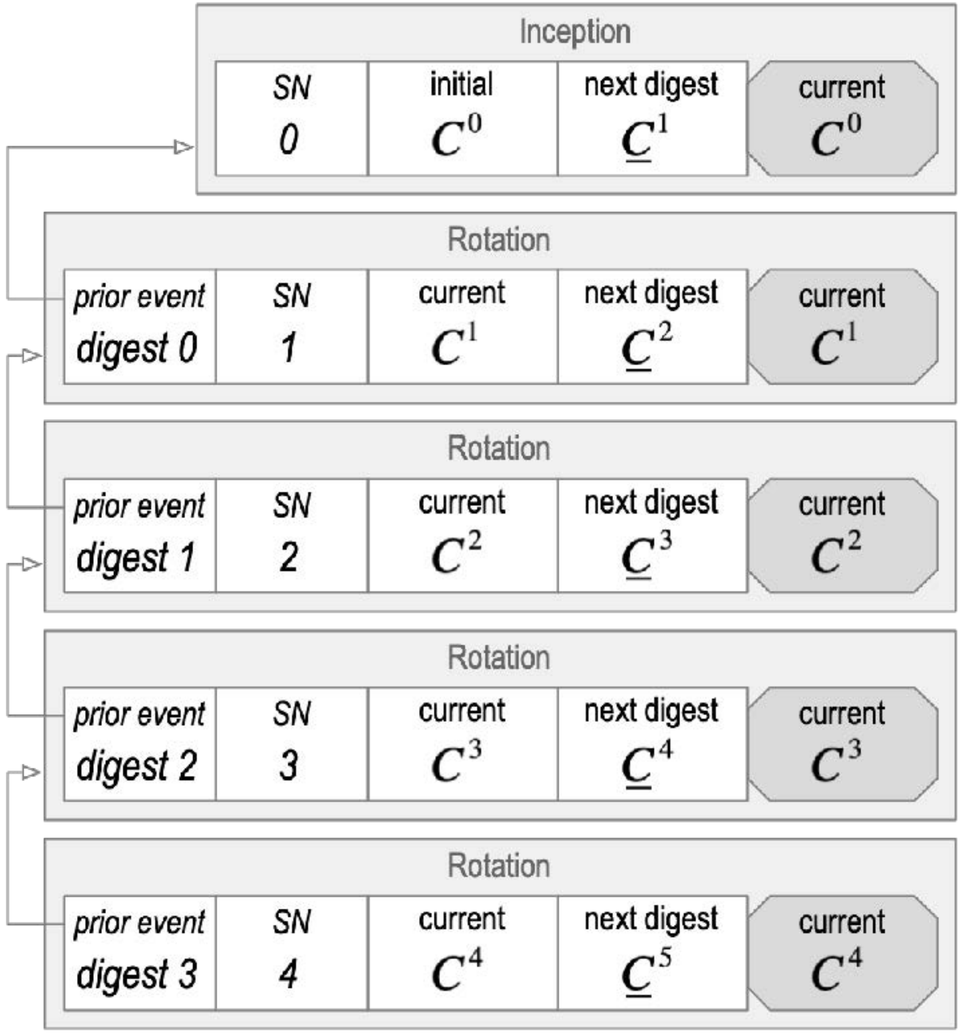


EL0JRaU2_RxFP0AL43wYn148Xq5YqaL6L48pf0fu7IUh

Key Pre-Rotation



KEL = duplicity evident verifiable data structure



Crypto-agility with pre-rotated digest(s) of *next* key(s) enable recovery from a surprise quantum attack

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

Inception Event

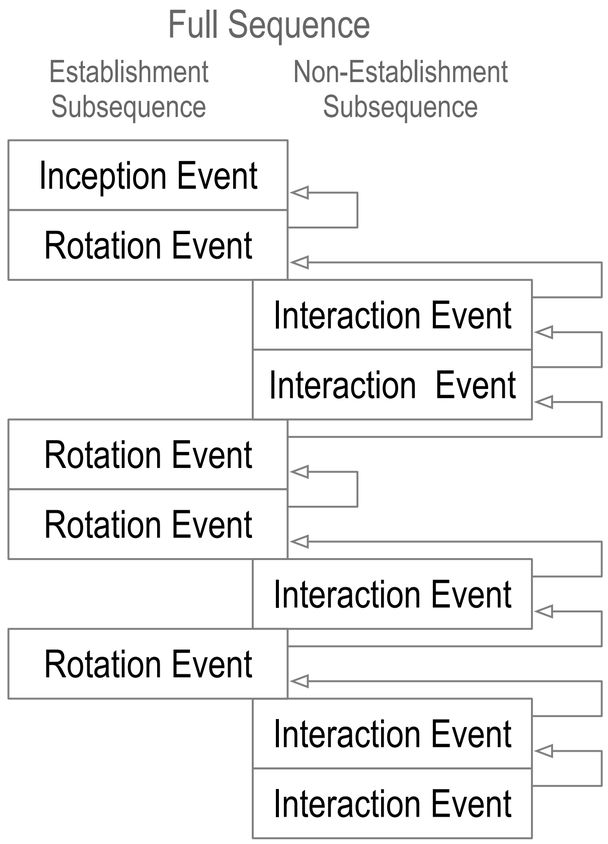
JSON

```
{
    "v": "KERICAACAAJSONAAKp.",
    "t": "icp",
    "d": "EPR7FWsN3tOM8PqfMap2FRfF4MFQ4v3ZXjBUcMVtvhmB",
    "i": "EPR7FWsN3tOM8PqfMap2FRfF4MFQ4v3ZXjBUcMVtvhmB",
    "s": "0",
    "kt": "2",
    "k":
    [
        "DBFiIgoCOpJ_zW_OO0GdffhHfEvJWb1HxpDx95bFvufu",
        "DG-YwInLUxzVDD5z8SqZmS2FppXSB-ZX_f2bJC_ZnsM5",
        "DGIAk2jkC3xuLIe-DI9rcA0naevtZiKuU9wz91L_qBAV"
    ],
    "nt": "2",
    "n":
    [
        "ELeFYMmuJb0hevKjhv97joA5bTfuA8E697cMzi8eoaZB",
        "ENY9GYShOjeh7qZUpIipKRHgrWcoR2WkJ7Wgj4wZx1YT",
        "EGyJ7y3TlewCW97dgBN-4pckhCqsni-zHNZ_G8zVerPG"
    ],
    "bt": "3",
    "b":
    [
        "BGKV6v93ue5L5wsgk75t6j8TcdgABMN9x-eIyPi96J3B",
        "BJfueFAYc7N_V-zmDEn2SPCoVFx3H20alWsNZKgsS1vt",
        "BAPv2MnoiCsgOnklmFyfU07QDK_93NeH9iKfOy8V22aH",
        "BA4PSatfQMw1lYhQoZkSSvOCrE0Sdw1hmmniDL-yDtrB"
    ],
    "c": ["DID"],
    "a": []
}
```

CESR

```
-FCS # Key Event Counter FixBodyGroup count=146 quadlets
  0OKERICAACAA # 'v' version  Verser Tag10 proto=KERI vrsn=2.00
  Xicp # 't' message type Ilker Tag3 Ilk=icp
  EDZOA3y_b_0LG4_cfpKTbWU-_3eeYNM0w9iTkT7frTYs # 'd' SAID Diger Blake3_256
  EDZOA3y_b_0LG4_cfpKTbWU-_3eeYNM0w9iTkT7frTYs # 'i' AID Prefixer Blake3_256
  MAAA # 's' Number Short sn=0
  MAAC # 'kt' Tholder signing threshold=2
  -JAh # 'k' Signing Key List Counter GenericListGroup count=33 quadlets
    DBFiIgoCOpJ_zW_OO0GdffhHfEvJWb1HxpDx95bFvufu # key Verfer Ed25519
    DG-YwInLUxzVDD5z8SqZmS2FppXSB-ZX_f2bJC_ZnsM5 # key Verfer Ed25519
    DGIAk2jkC3xuLIe-DI9rcA0naevtZiKuU9wz91L_qBAV # key Verfer Ed25519
  MAAC # 'nt' Tholder rotation threshold=2
  -JAh # 'n' Rotation Key Digest List Counter GenericListGroup count=33 quadlets
    ELeFYMmuJb0hevKjhv97joA5bTfuA8E697cMzi8eoaZB # key digest Diger Blake3_256
    ENY9GYShOjeh7qZUpIipKRHgrWcoR2WkJ7Wgj4wZx1YT # key digest Diger Blake3_256
    EGyJ7y3TlewCW97dgBN-4pckhCqsni-zHNZ_G8zVerPG # key digest Diger Blake3_256
  MAAD # 'bt' Tholder Backer (witness) threshold=3
  -JAs # 'b' Backer (witness)List Counter GenericListGroup count=44 quadlets
    BGKV6v93ue5L5wsgk75t6j8TcdgABMN9x-eIyPi96J3B # AID Prefixer Ed25519N
    BJfueFAYc7N_V-zmDEn2SPCoVFx3H20alWsNZKgsS1vt # AID Prefixer Ed25519N
    BAPv2MnoiCsgOnklmFyfU07QDK_93NeH9iKfOy8V22aH # AID Prefixer Ed25519N
    BA4PSatfQMw1lYhQoZkSSvOCrE0Sdw1hmmniDL-yDtrB # AID Prefixer Ed25519N
  -JAB # 'c' Config Trait List Counter GenericListGroup count=1 quadlets
    XDID # trait Traitor Tag3 trait=DID
  -JAA # 'a' Seal List Counter GenericListGroup count=0 quadlets
```

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.

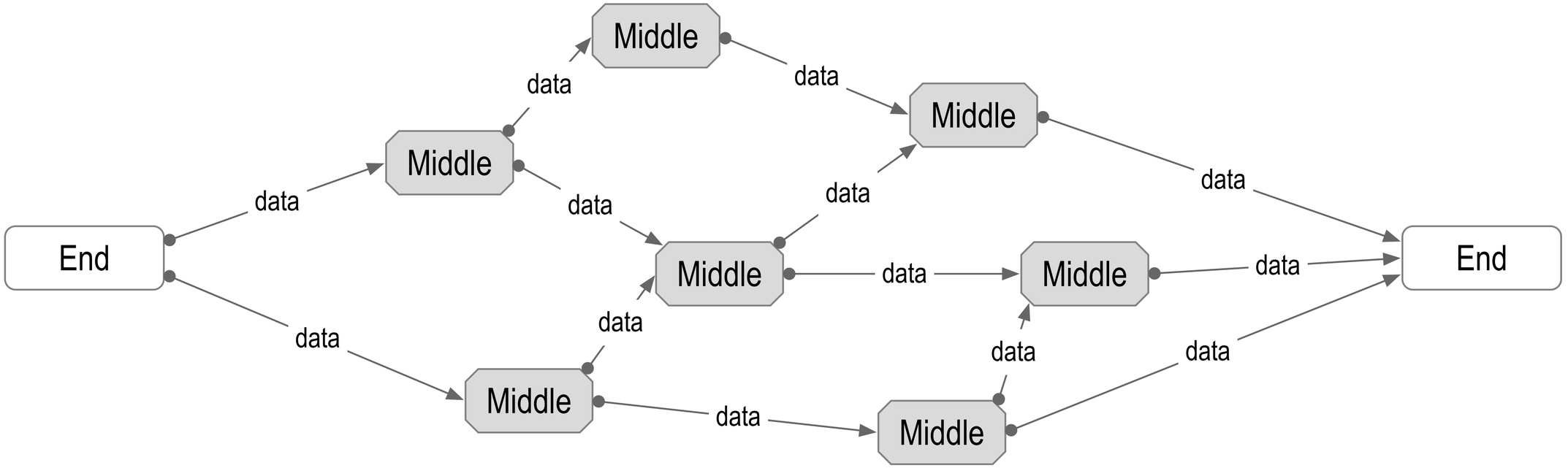
Externally inconsistent log with a purported copy of log but both verifiable = duplicitous.

Duplicity detection protects against external inconsistency.

KERI provides duplicity evident DKMI

End Verifiability

End-to-End Verifiability Authenticity

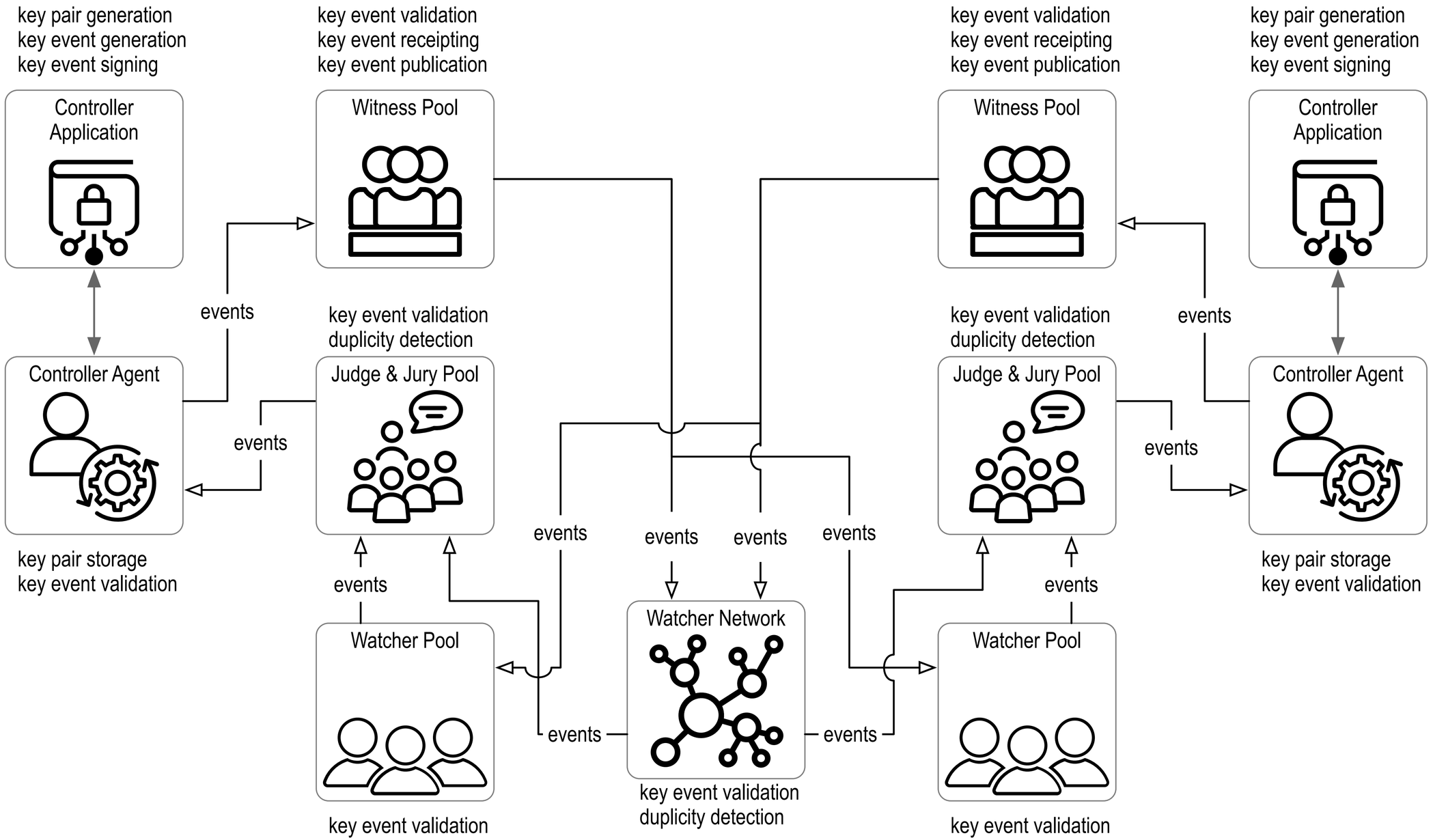


Its much easier to protect one's private keys than to protect all internet infrastructure

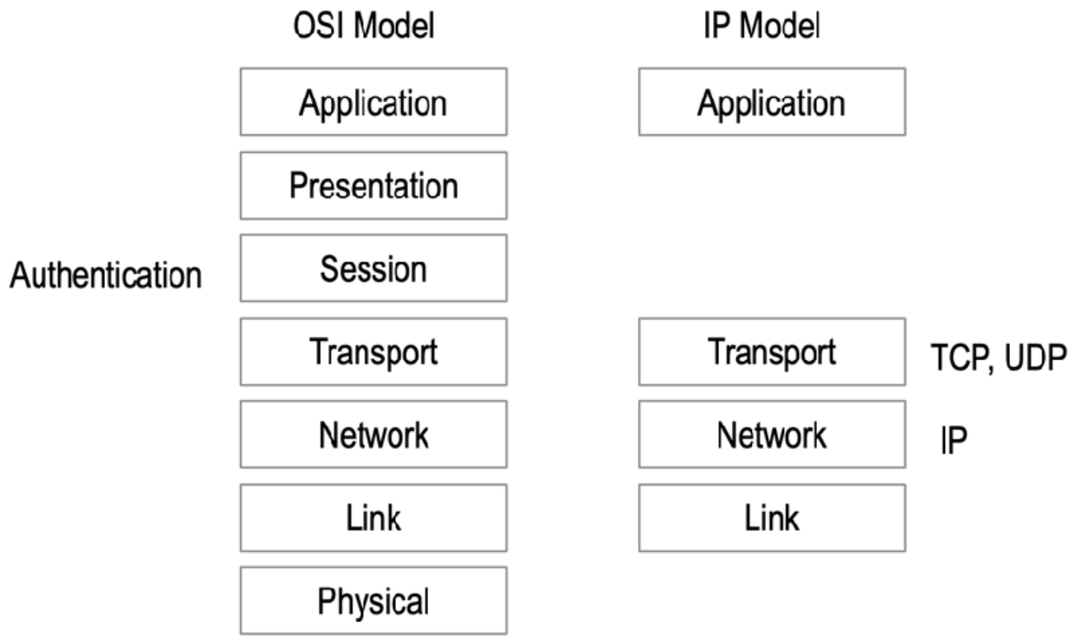
If the edges are secure, the security of the middle doesn't matter.

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

KERI Ecosystem



The Internet Protocol (IP) has no *security (trust)* layer.

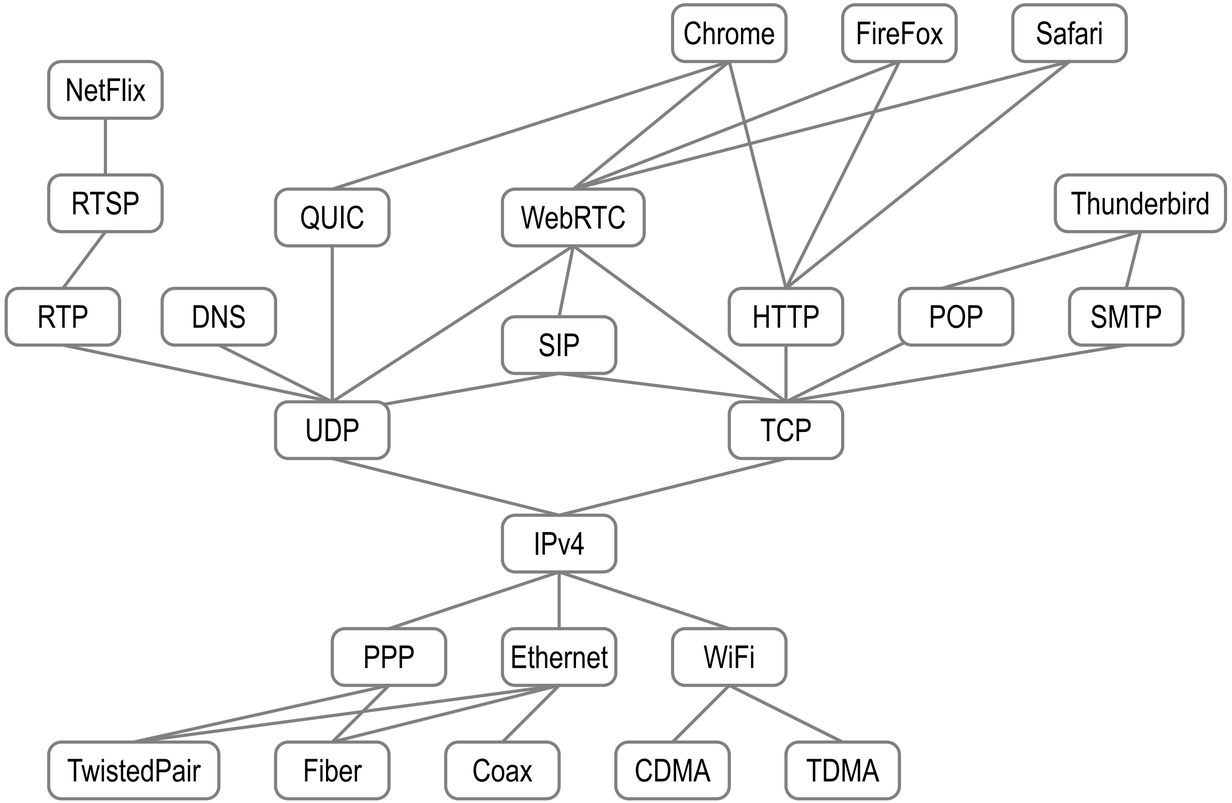


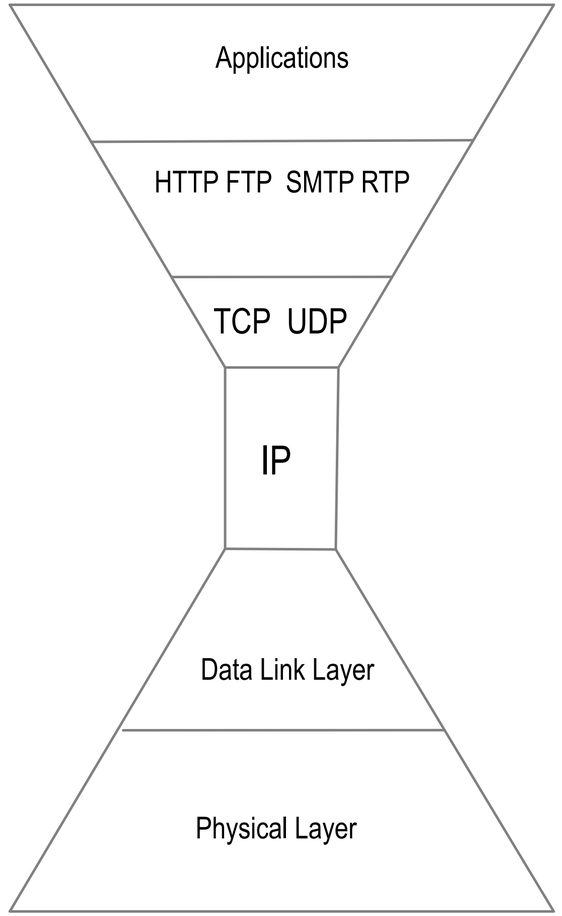
Instead …

We use *bolt-on* identity system security overlays.

(PKI-DNS-CA, OAuth, …)

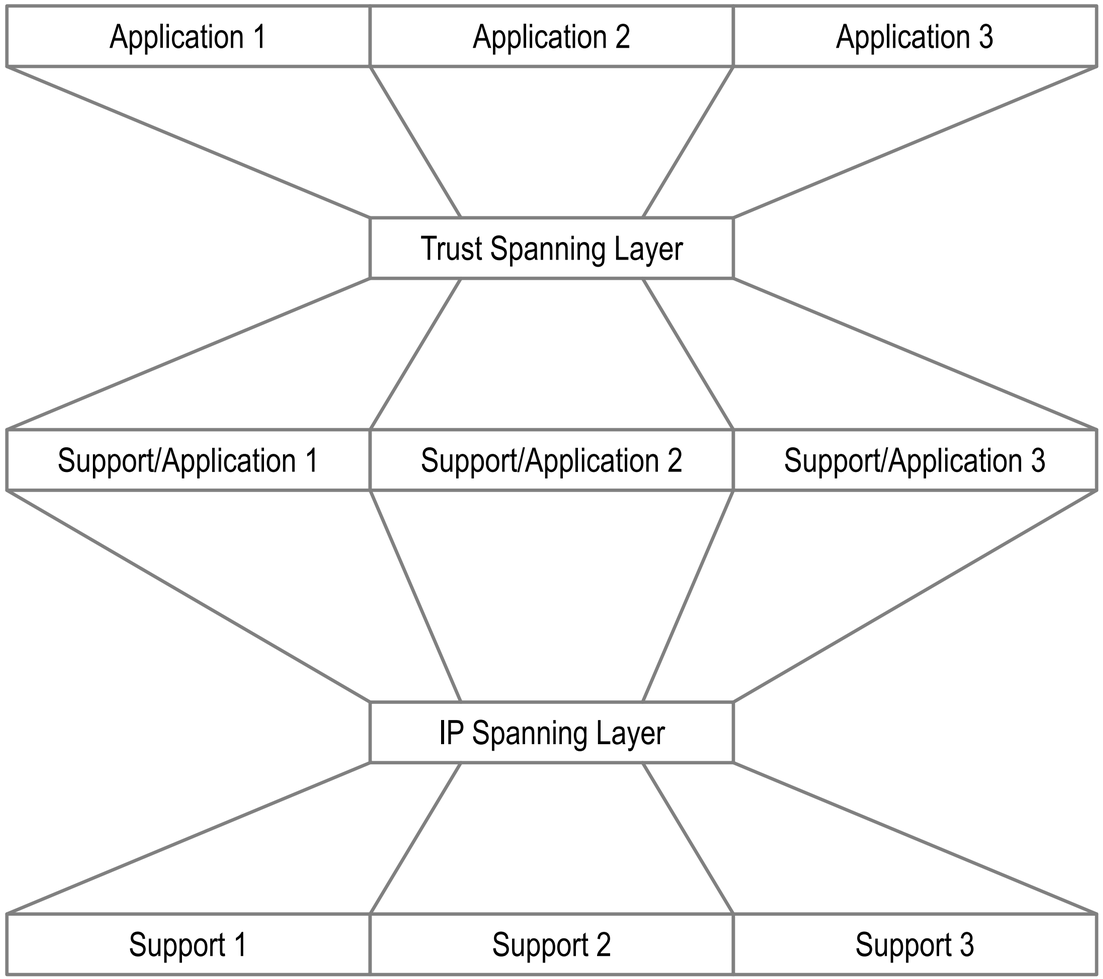
Waist = IP Spanning Layer

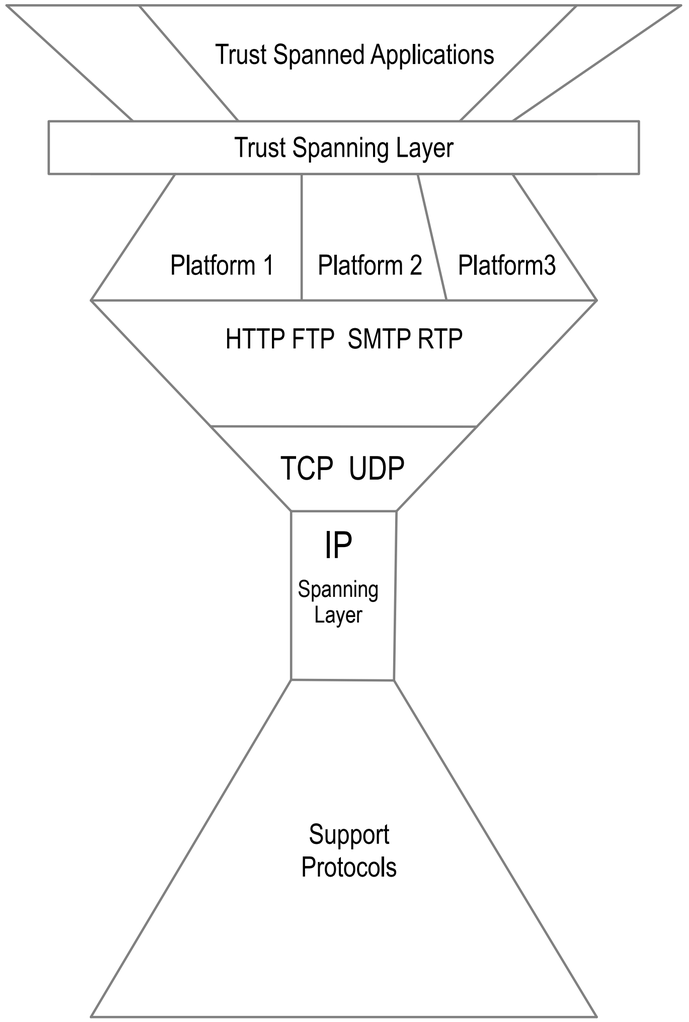




https://web.archive.org/web/20050415042854/http://www.csd.uch.gr/~hy490-05/lectures/Clark_interoperation.htm

Waist and Neck — Trust Spanning Layer





Why Should You Care!

Organizational Identity.

Path to a much simpler, more secure, fully decentralized, trust-spanning layer for the internet.

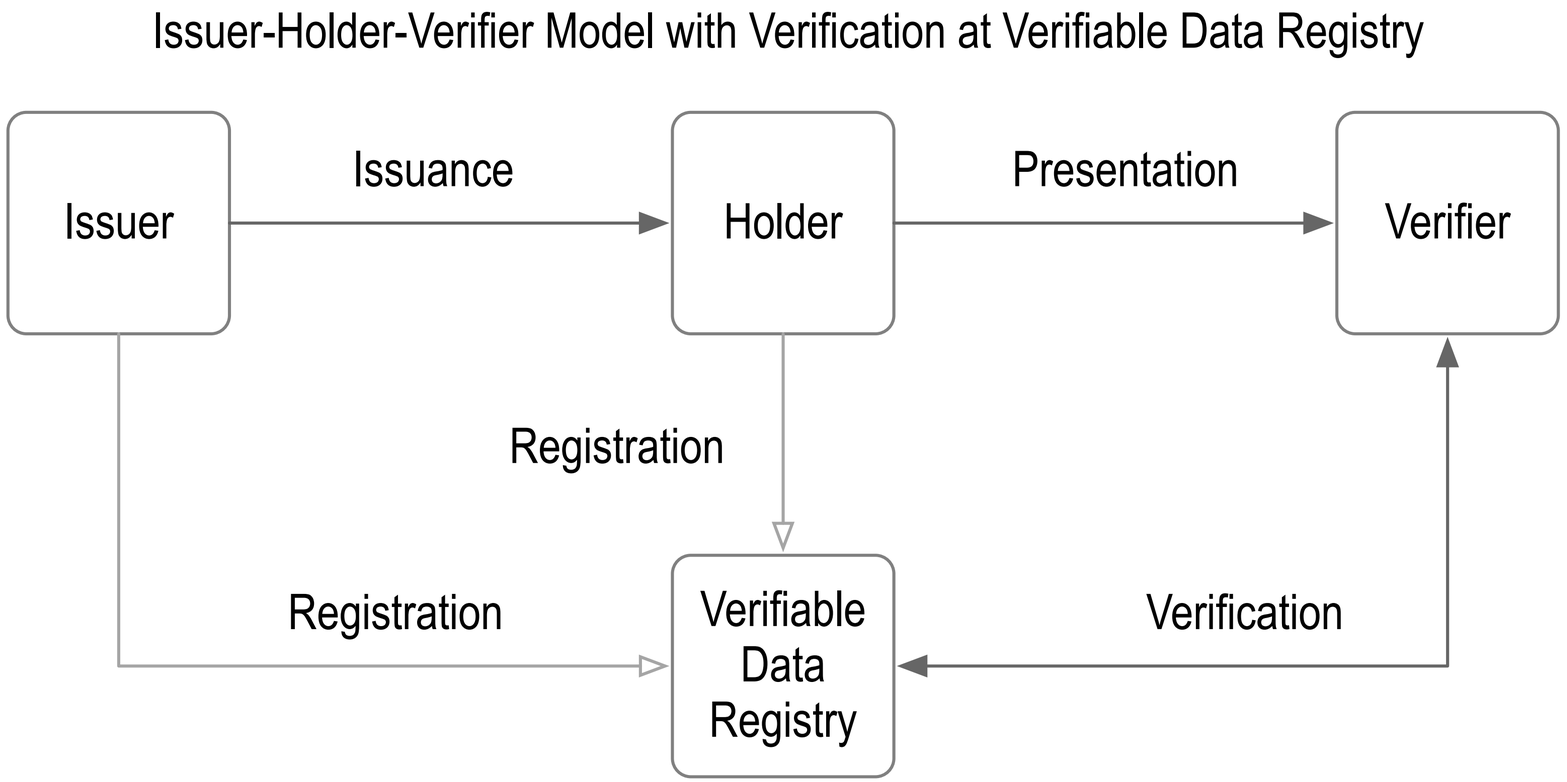
All data supply chains, including AI data supply chains, are targets of 5GW, but the PKI that protects them is highly vulnerable to 5GW. KERI is 5GW “safe” by design.

Broad adoption of the KERI-Suite has been hindered by the relative immaturity of its tooling, primarily due to limited resources rather than its technical architecture.

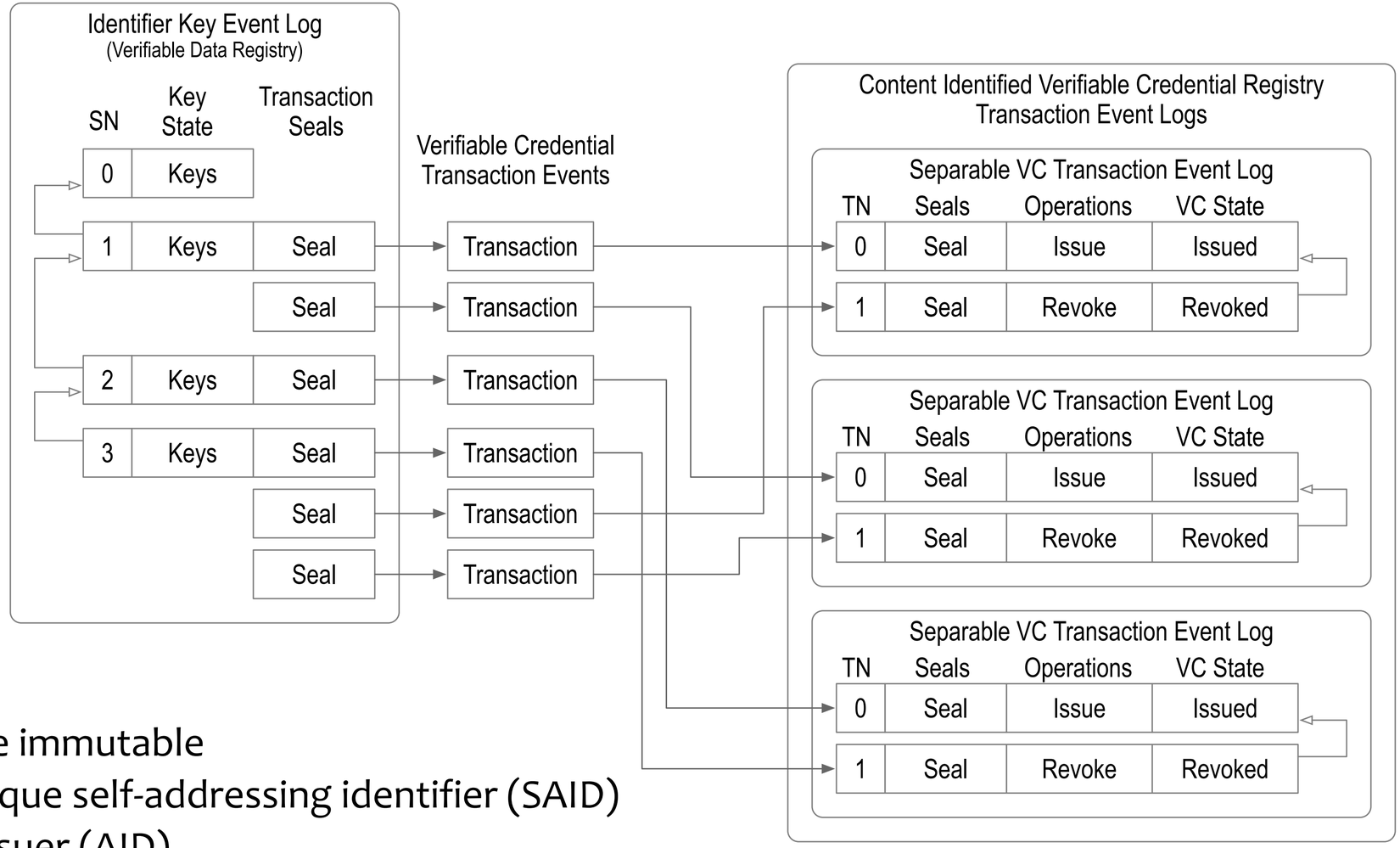
The KERI Foundation’s mission is to provide open source (Apache-2) mature tooling for *one-click-KERI — wallet, witness, watcher, web, wizard* — to better foster adoption.

Open-Loop Tripartite Model for AuthZ

Verifiable Data Registry (VDR) enables decentralized but interoperable discovery and verification of Issuance state.



KEL Anchored Issuance-Revocation Registry with Separable ACDC TELs



All ACDC Schema are immutable

Each ACDC has a unique self-addressing identifier (SAID)

Each ACDC has an Issuer (AID)

Each ACDC may have an Issuee (AID).

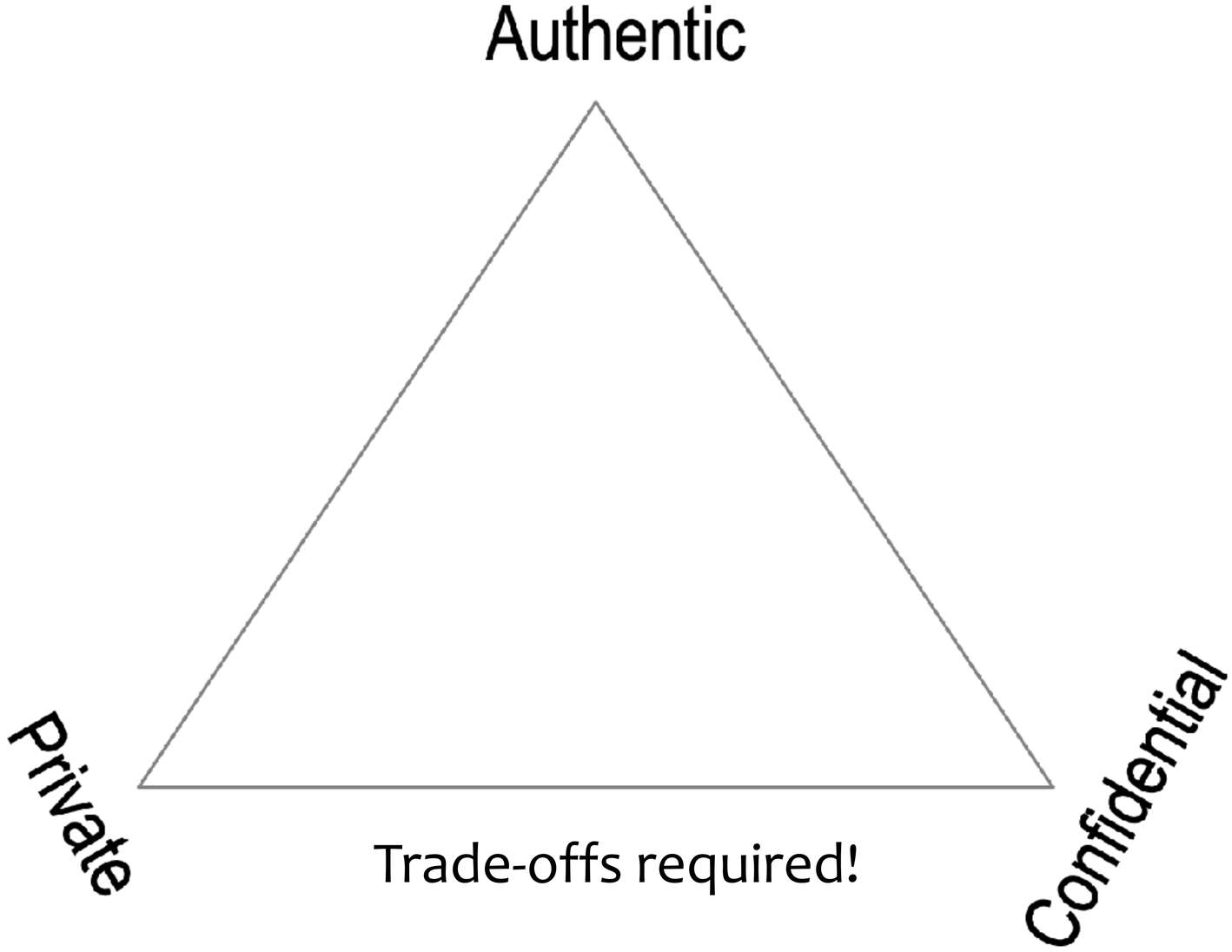
Each ACDC may have edge(s) the link to other ACDC's forming a delegated chain (tree) of authority

Each ACDC has a TEL whose state is bound to the Issuer's key-state and may be bound to the Issuee's key state

This provides TADAAA! (tail authenticated, delegated, attenuable, aggregable authorization)

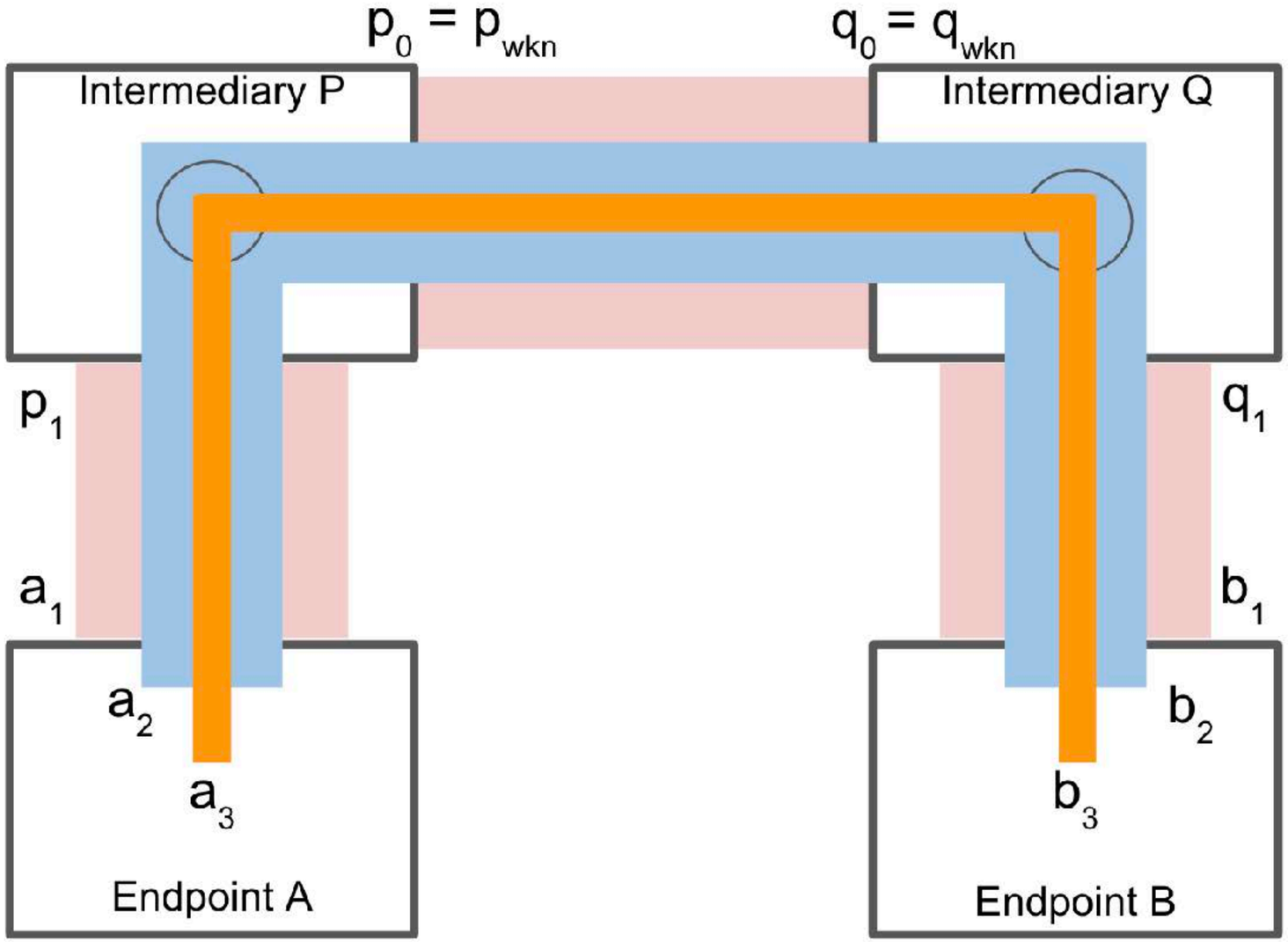
PAC Theorem (Tri-lemma)

One can have any two of the three – privacy, authenticity, confidentiality at the highest level but not all three.



TSP (SPAC) Protocol

Triple-nested tunneling protocol that minimizes surveillable metadata

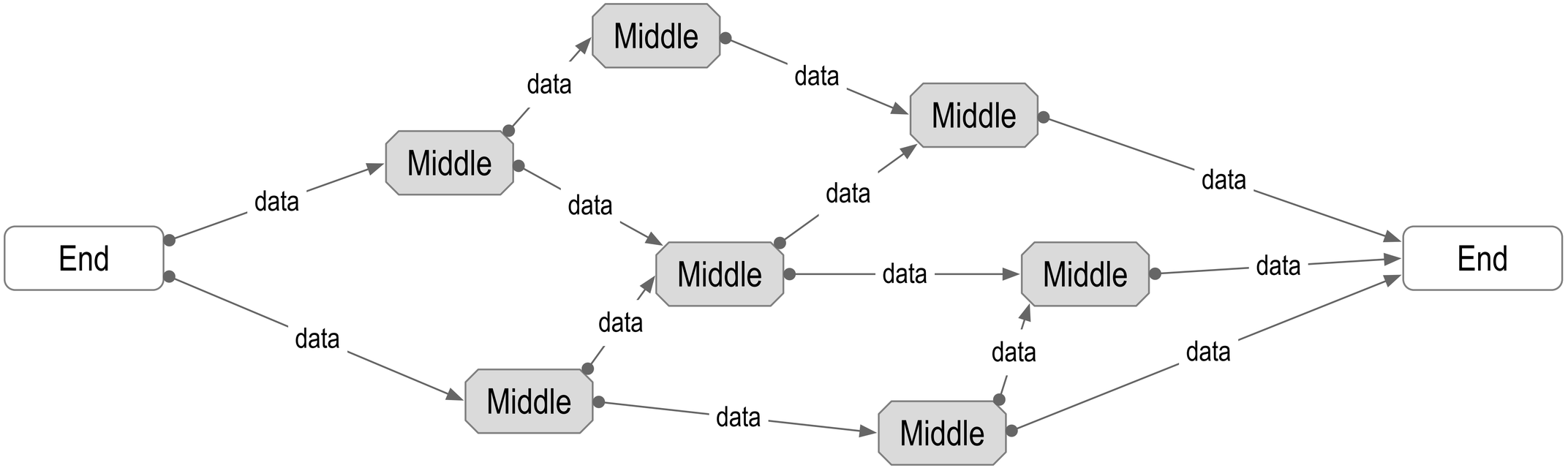


covert vs. clandestine surviellability

Dual: End Verifiability and End-Only Viewability

End-to-End Verifiability Authenticity

Only-at-End Viewability Confidentiality



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

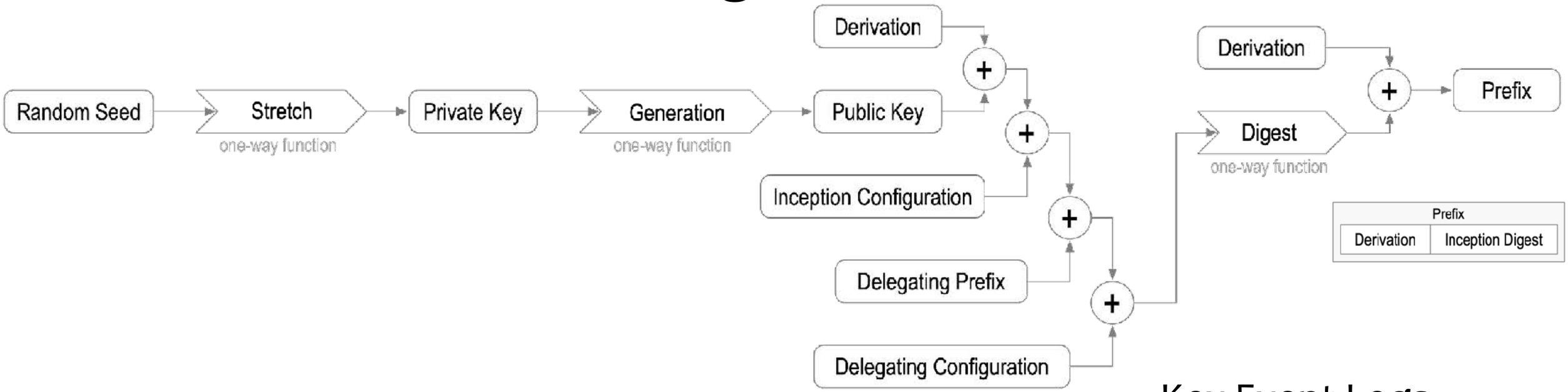
If the edges are secure, the security of the middle doesn't matter.

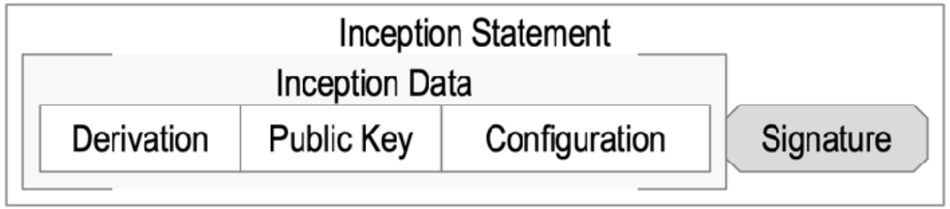
Ambient Verifiability: any-data, any-where, any-time by any-body

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

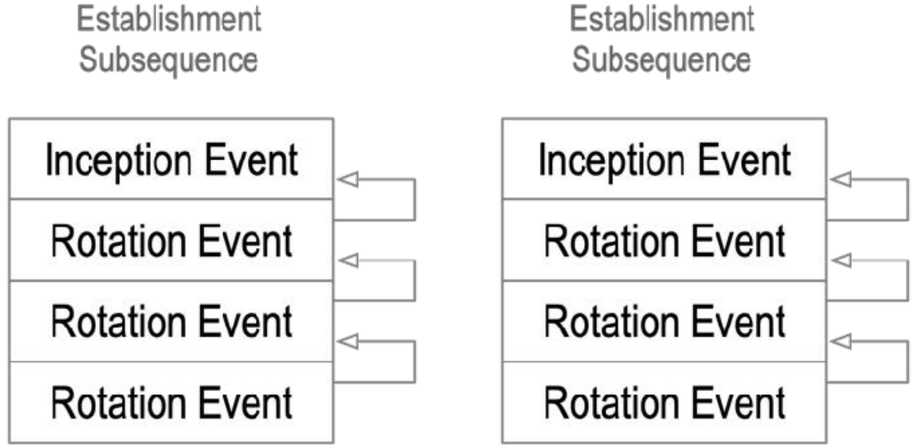
Backup Slides

Delegated AID



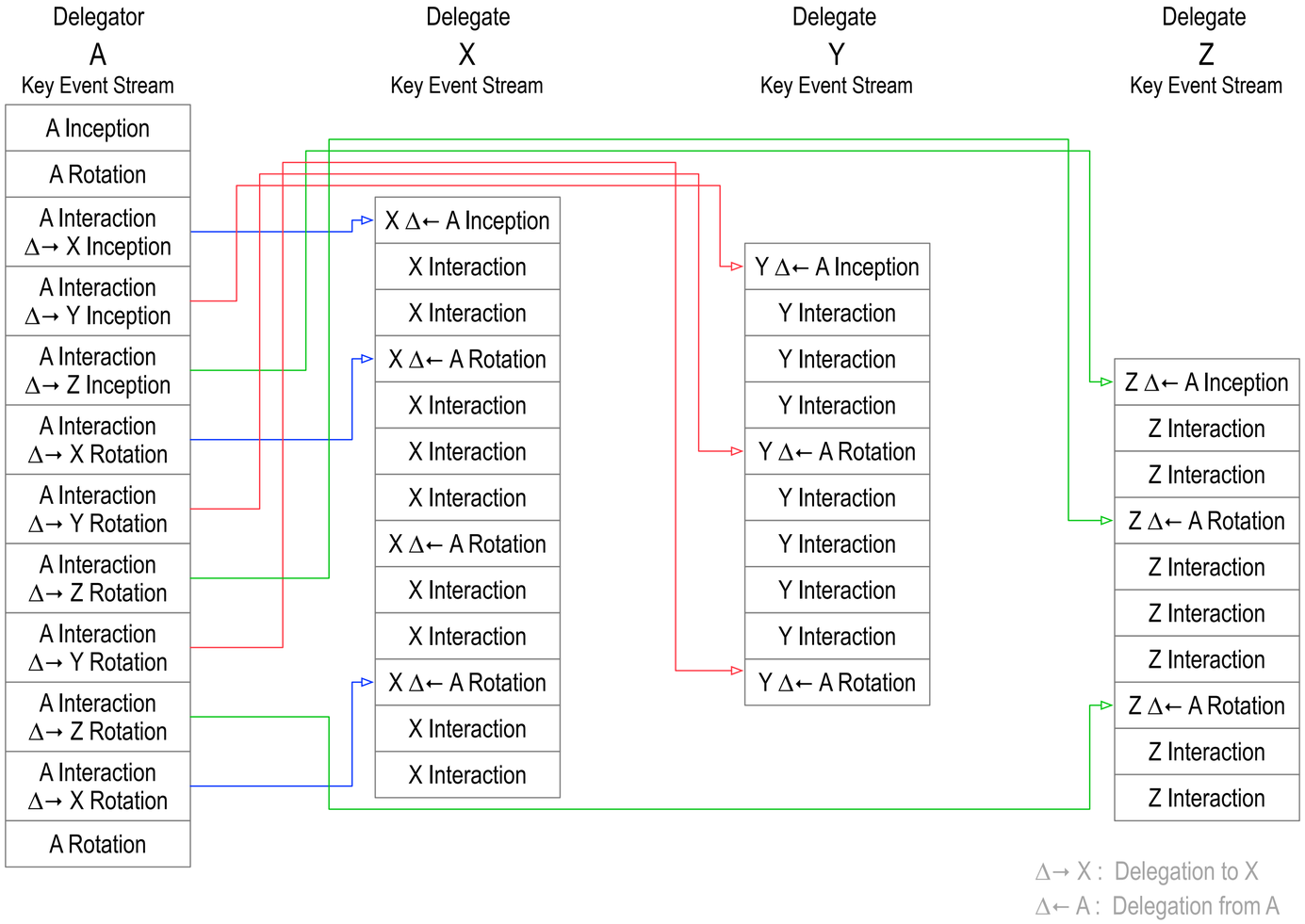


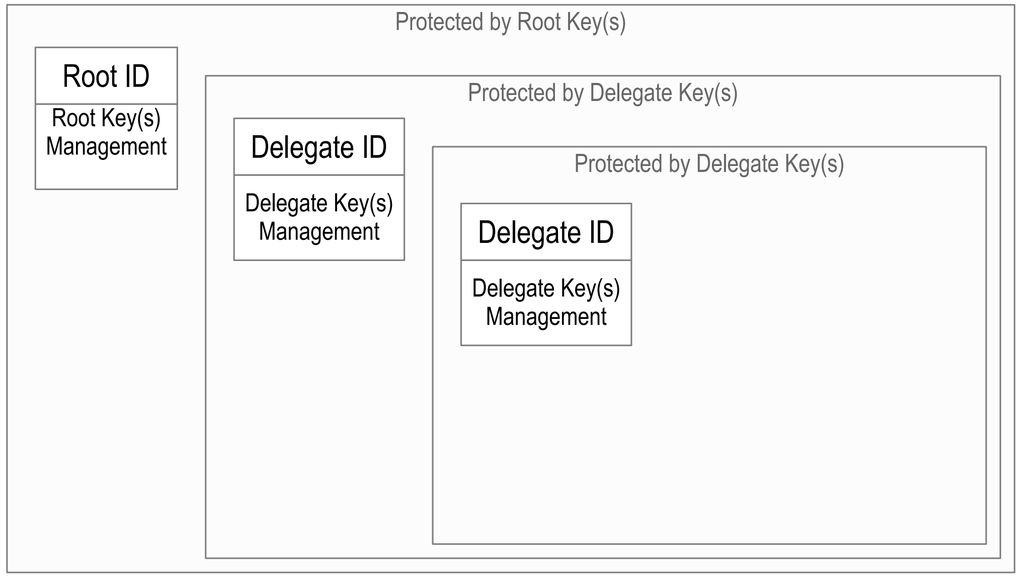
Key Event Logs



EFP0AL43wYn148Xq5YqaL6L48pf0fu7IUhL0JRaU2_Rx

Identifier Delegation: Scaling & Protection





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

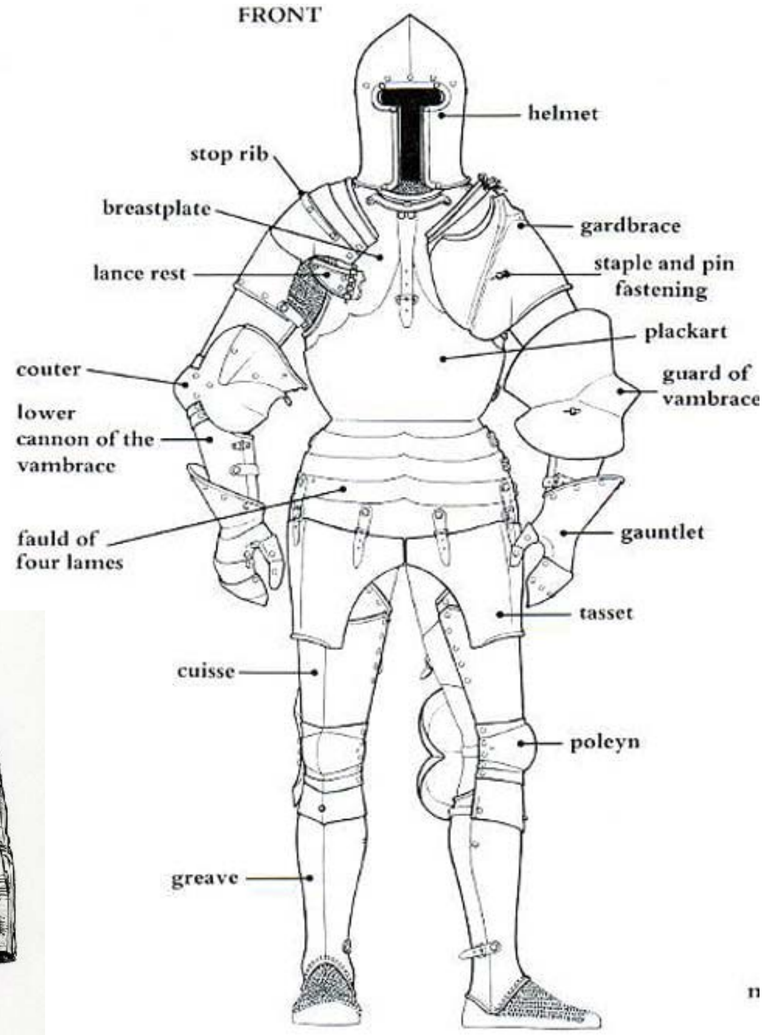
Ambient Duplicity Detection

Joe
Validator
Log

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

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.

Zero-Trust Architectures & Computing

Never trust, always verify.

Perimeter-less security model.

Data is signed and/or encrypted both in motion and at rest.

No such thing as true zero-trust.

All architectures lie on the zero-trust spectrum given by:

the ratio of trusted surface to verifiable surface.

End goal = all data has end-to-end verifiable authenticity (provenance)

Resources:

NIST: Developing a Framework to Improve Critical Infrastructure Cybersecurity 04/08/2013 Zero Trust Model for Information Security, Forrester Research. http://csrc.nist.gov/cyberframework/rfi_comments/040813_forrester_research.pdf

https://www.nist.gov/cyberframework Zero Trust Networks 2017 Gilman & Barth https://www.amazon.com/Zero-Trust-Networks-Building-Untrusted/dp/1491962194/ref=sr_1_1?

s=books&ie=UTF8&qid=1499871379&sr=1-1&keywords=zero+trust+networks

Organizational Identity

Zero-trust architecture

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

Protocol not Platform

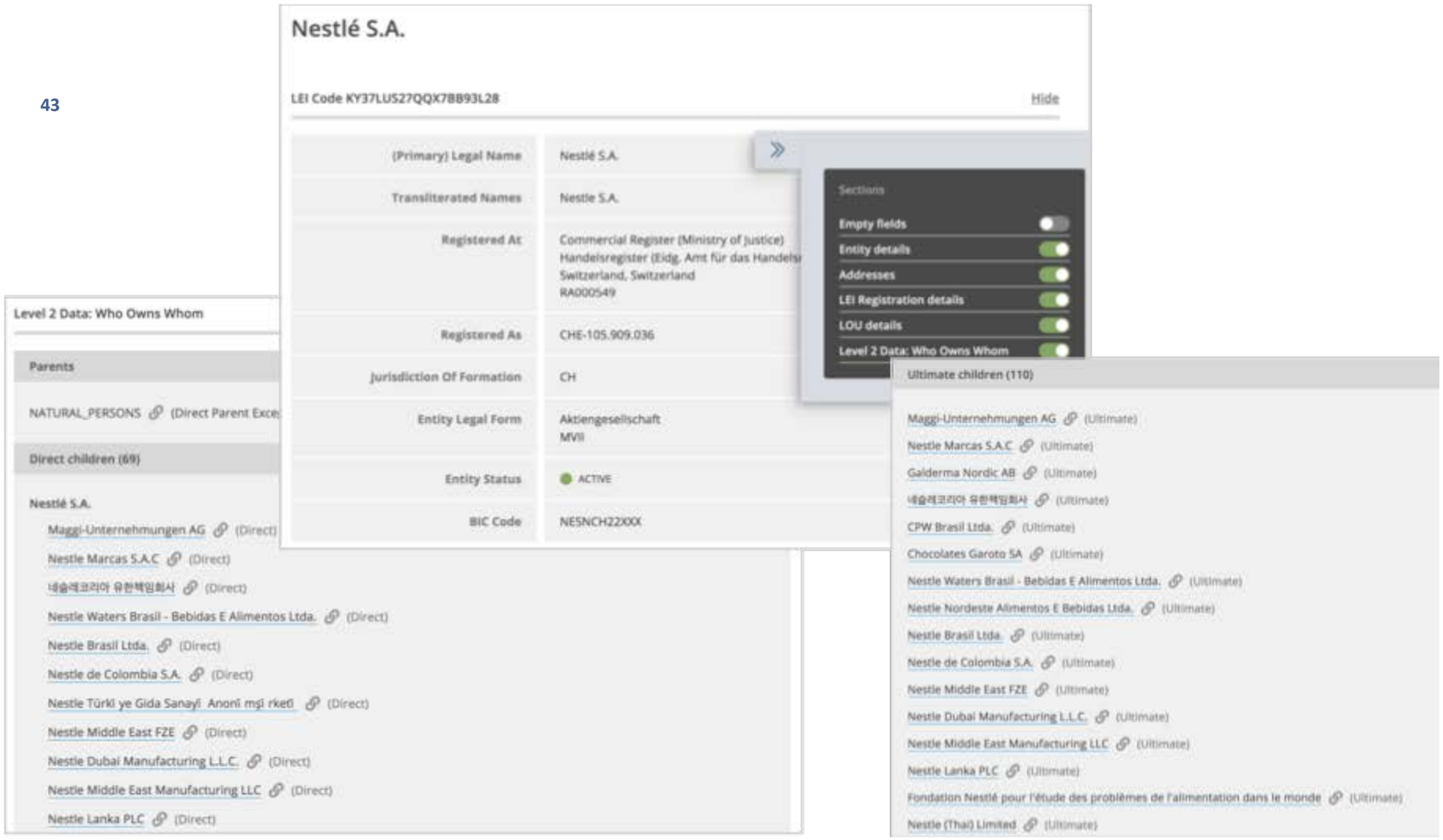
Delegable Authority

Multi-sig DPKI

Authentic Chained Data Containers

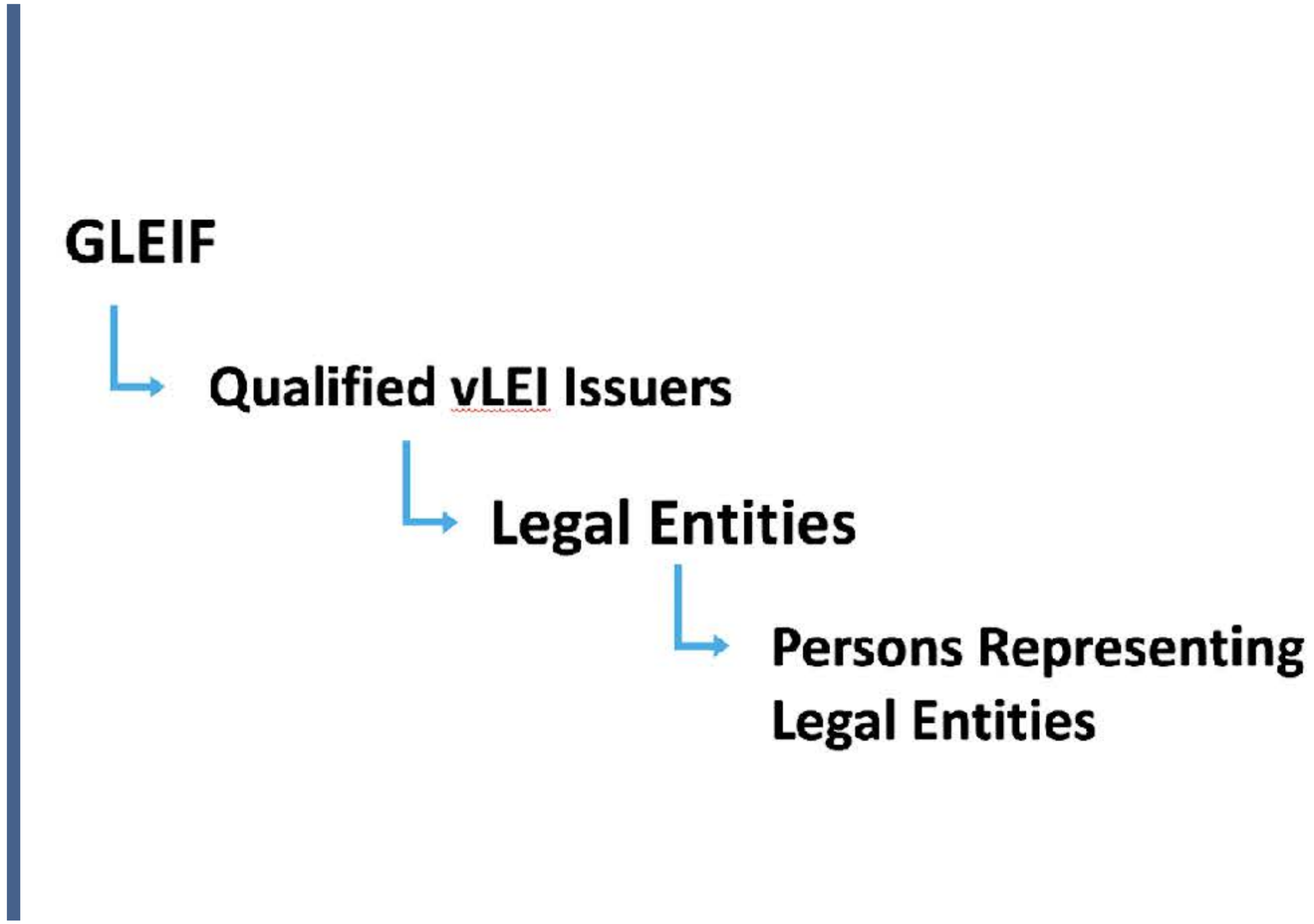
The Legal Entity Identifier – the LEI

- 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

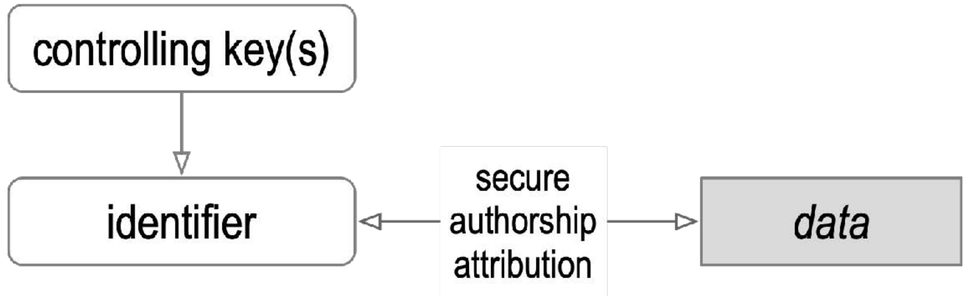


The LEI as a Verifiable Credential – the vLEI Trust Chain

- 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



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.

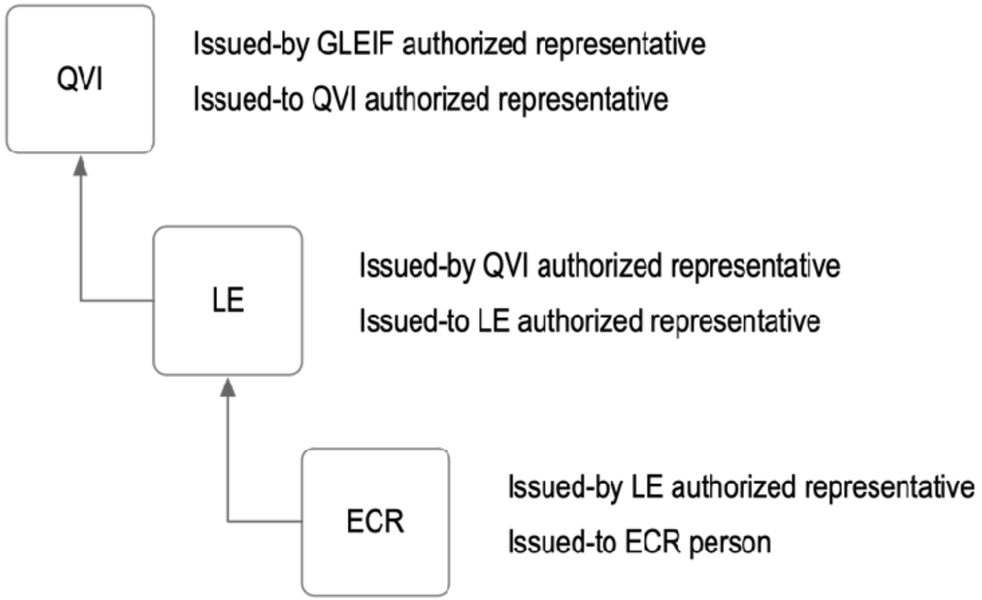
KERI solves the *secure attribution* problem with zero trust.

GLEIF vLEI Credential Example

Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Engagement Context Role (ECR) Credential

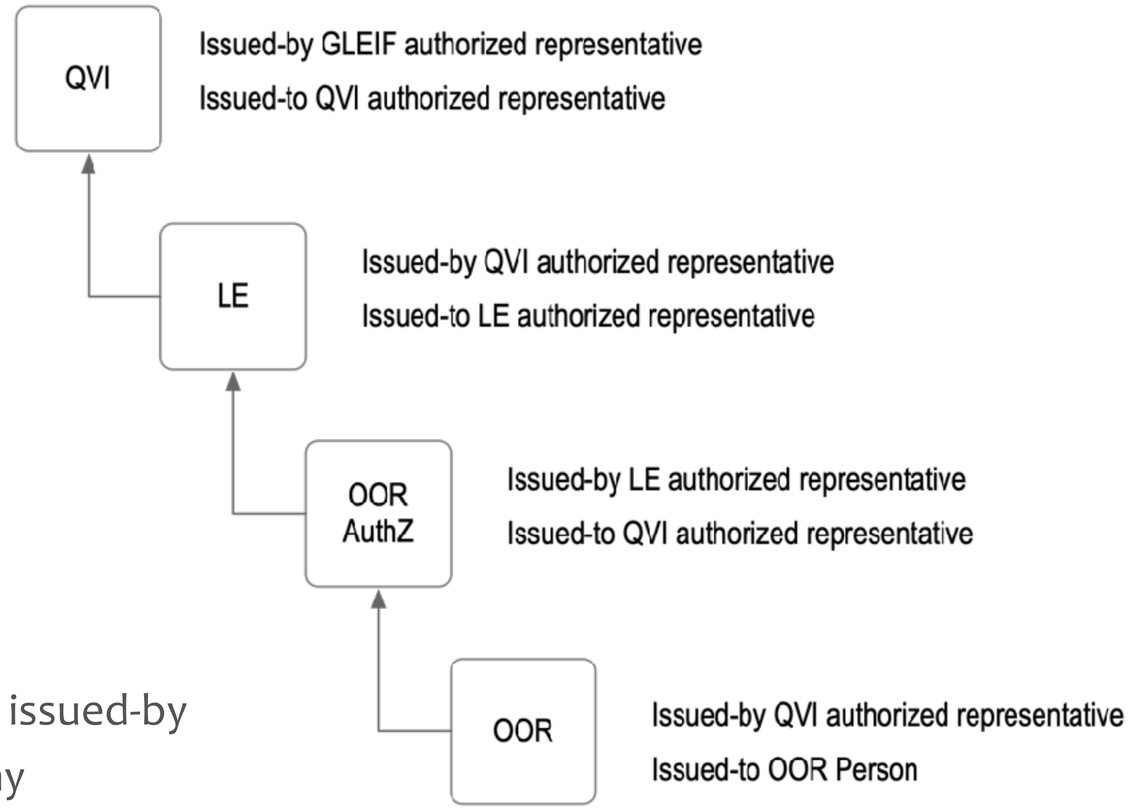


Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Official Organizational Role Authorization (OOR-AuthZ) Credential

Official Organizational Role (OOR) Credential



Anyone in the chain-of-authority can revoke the credential issued-by them. This breaks the chain and thereby may invalidate any authorizations or attestations that are chained from their credential.

GLEIF vLEI Authorized Attestation Example

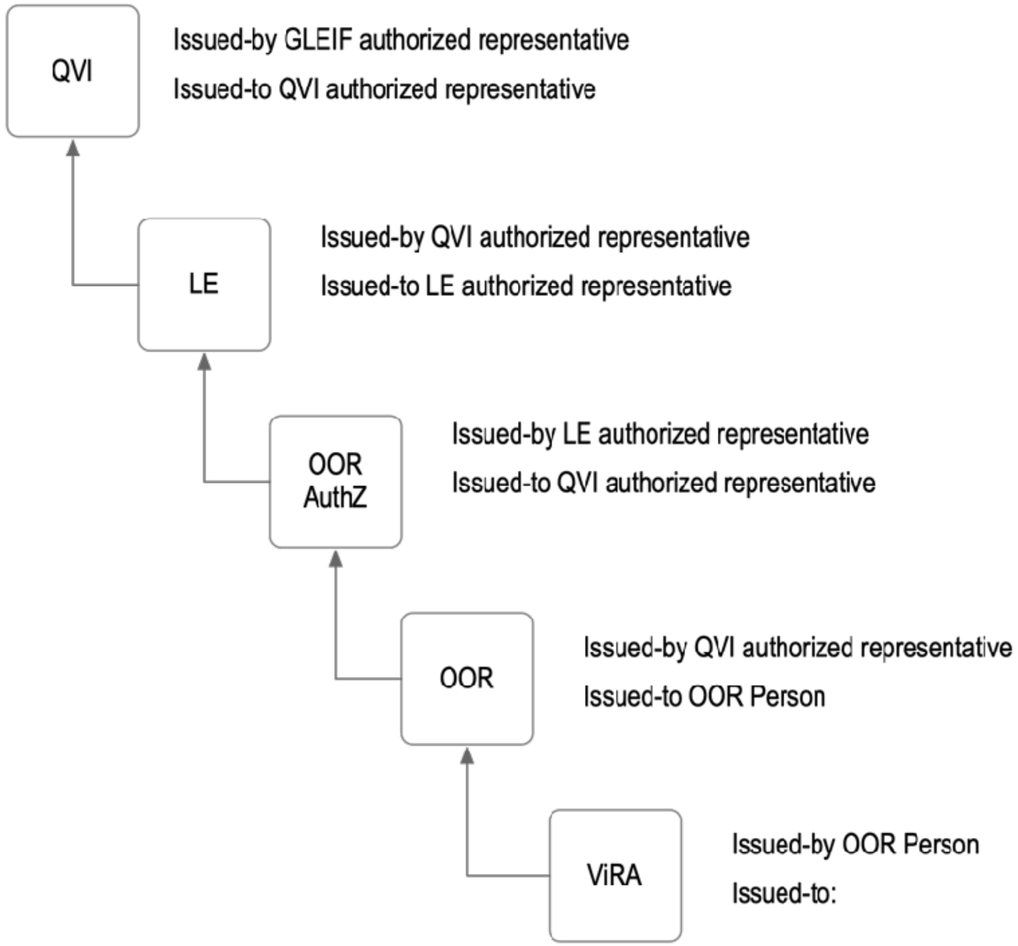
Qualified vLEI Issuer (QVI) Credential

Legal Entity (LE) Credential

Official Organizational Role Authorization (OOR-AuthZ) Credential

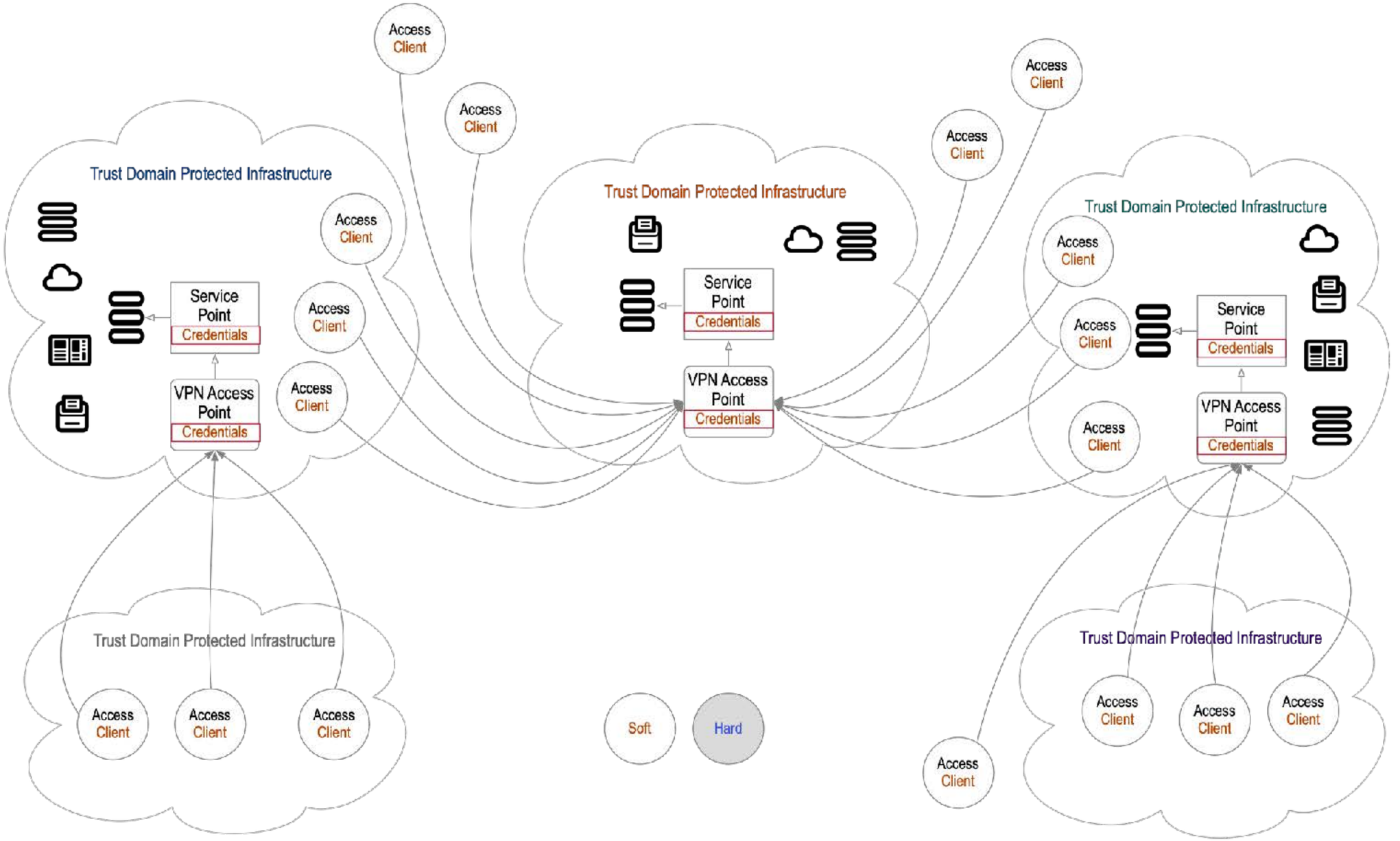
Official Organizational Role (OOR) Credential

Verifiable IXBRL Report Attestation (ViRA)



Anyone in the chain-of-authority can revoke the credential issued-by them. This breaks the chain and thereby may invalidate any authorizations or attestations that are chained from their credential.

Conventional health data attack surface



healthKERI

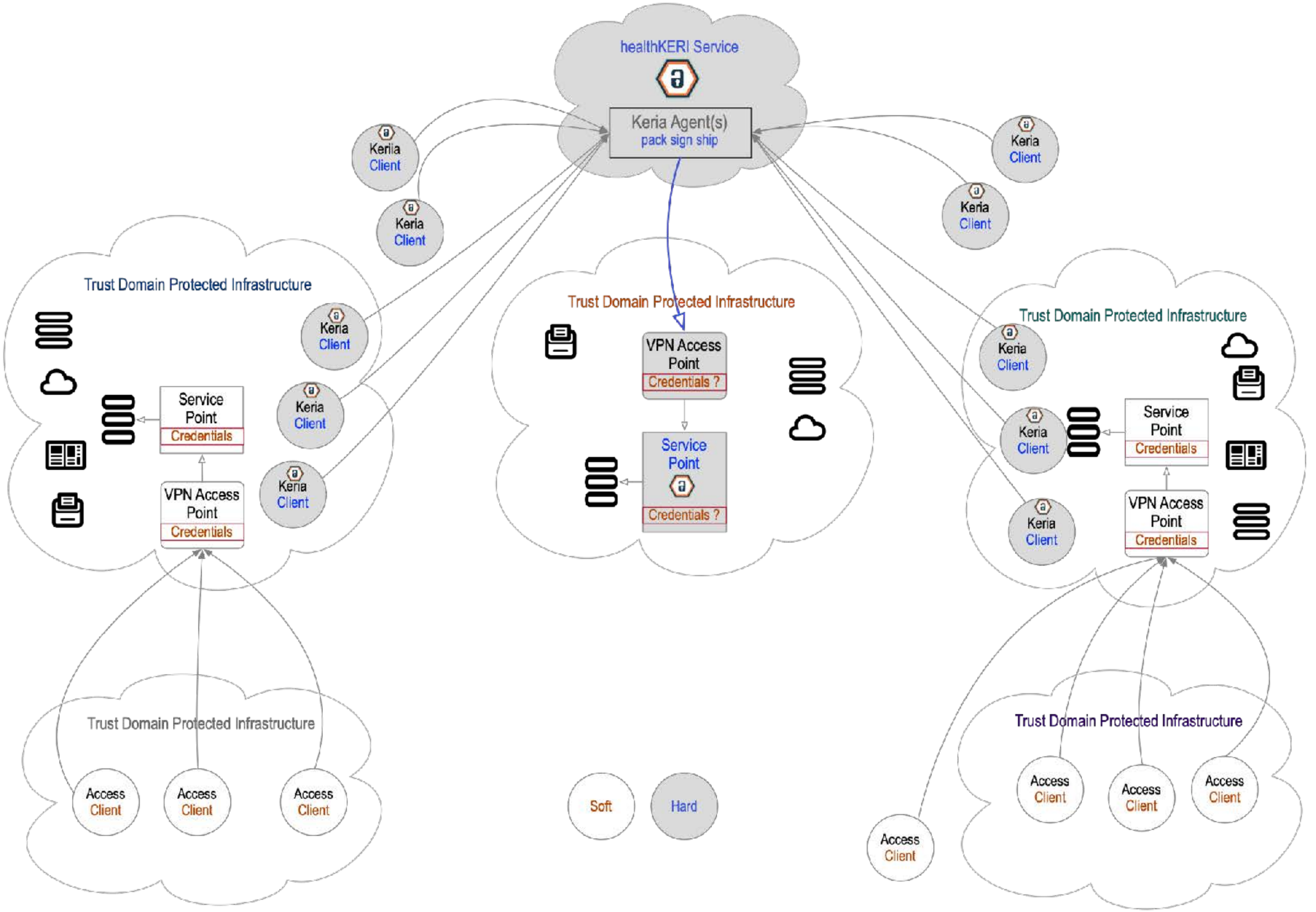
attack surface

Mission Survivability Gains:

Lower Vulnerable (soft) Edge Count

Simplify Configuration

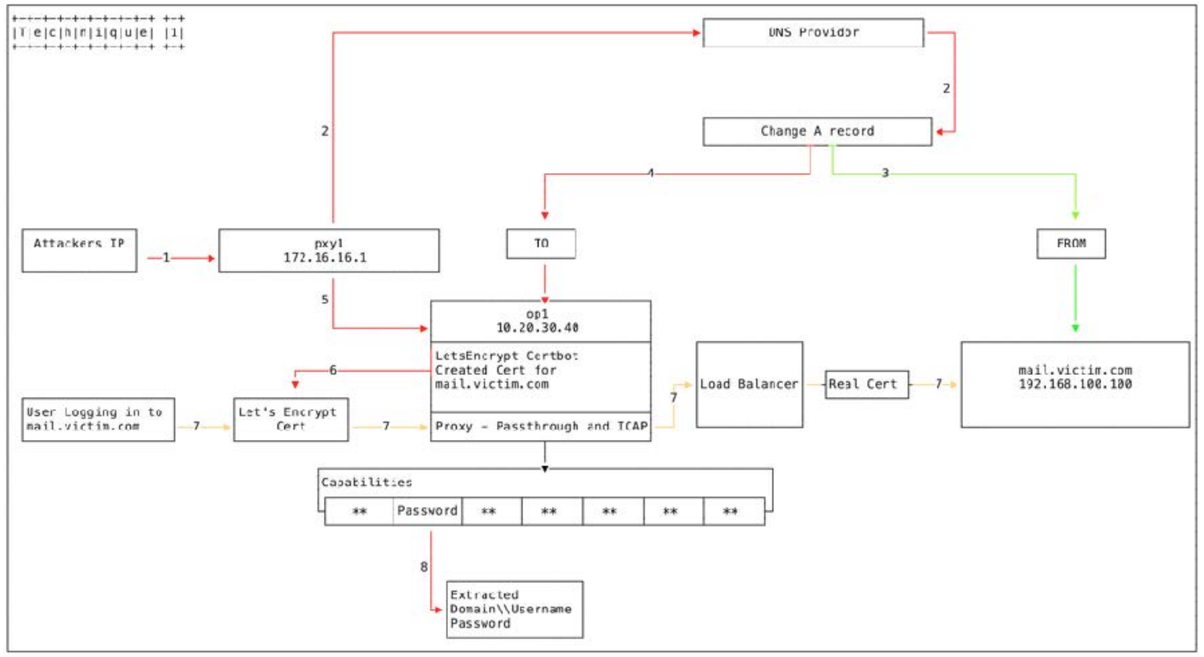
Lessen Reliance on Shared Secrets

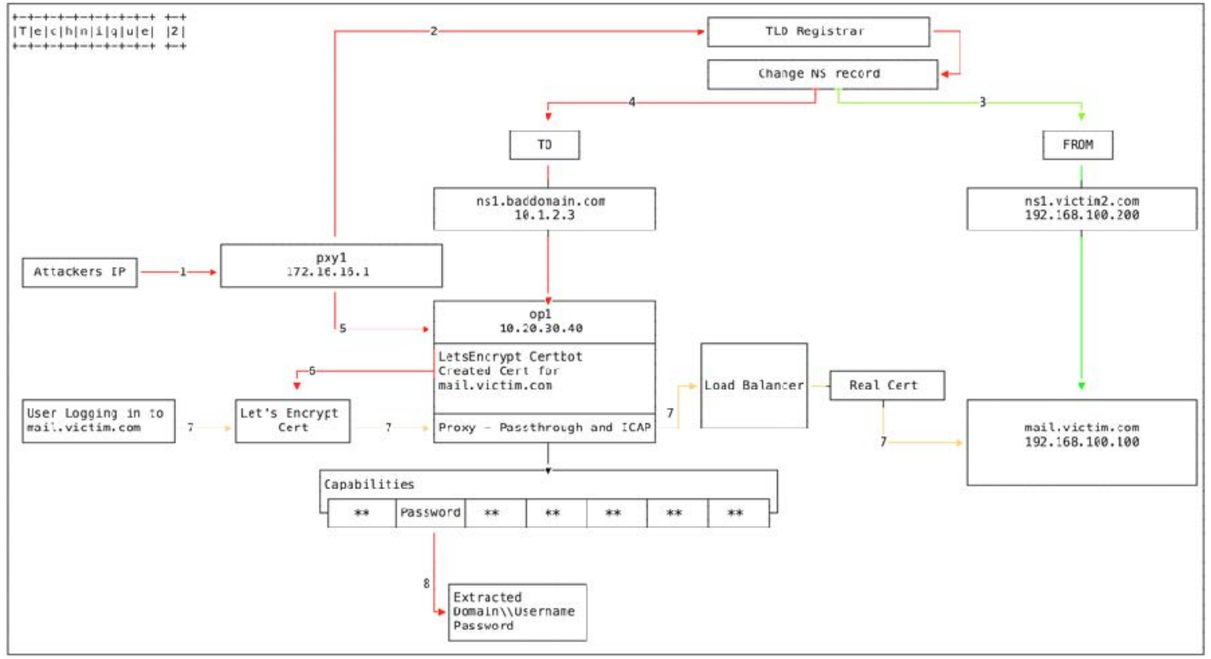


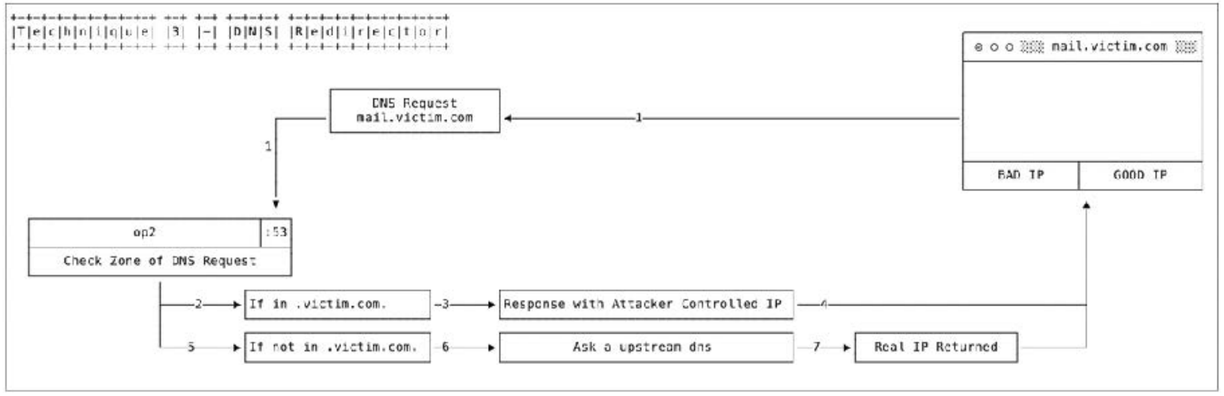
DNS Hijacking

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

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







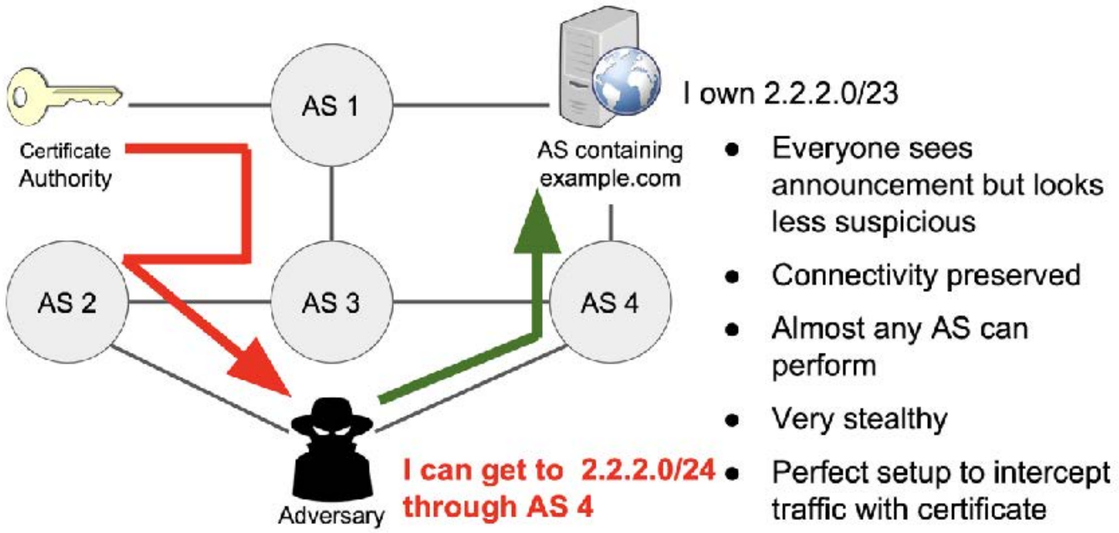
BGP Hijacking: AS Path Poisoning

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

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

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

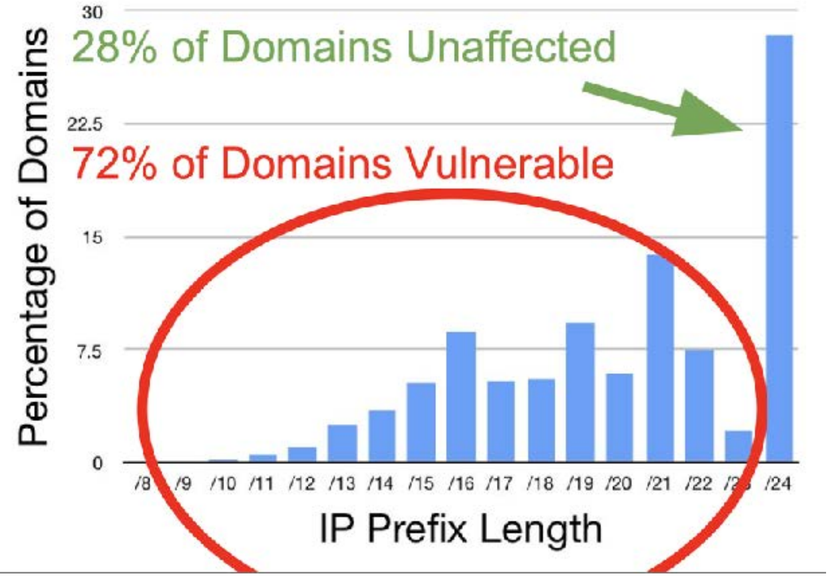
AS path poisoning



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

Vulnerability of domains: sub-prefix attacks

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



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)

Trust Domain

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

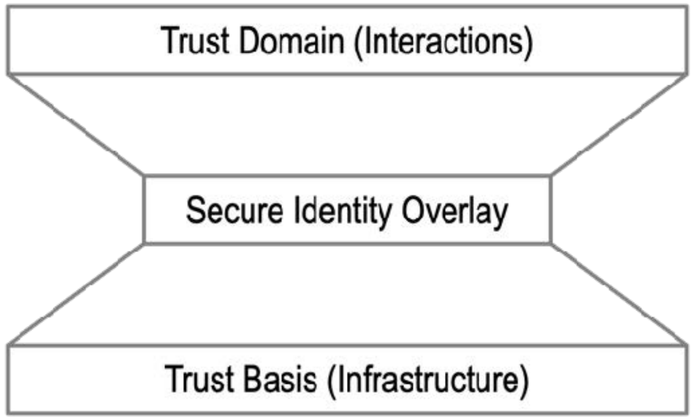
A *secondary* root-of-trust is *replaceable*.

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.*



Control over Trust Bases and Domains

Want decentralized control over trust domains or at least the trust bases.

Shared control over a trust domain is less decentralized than non-shared identifier specific control over a trust domain.

A shared primary (non-replaceable) root-of-trust aka shared ledger is the trust basis for one trust domain.

It has shared governance which is more decentralized but it’s not solving the hard problem of moving value between trust domains under different control.