

# Key Event Receipt Infrastructure

## KERI-2

### A Secure Identifier Overlay for the Internet

Samuel M. Smith Ph.D.

[sam@prosapien.com](mailto:sam@prosapien.com)

IIW Spring 2020 April 28-30

<https://github.com/SmithSamuelM/Papers>

[https://github.com/SmithSamuelM/Papers/blob/master/presentations/KERI2\\_Overview\\_IIW\\_2020\\_A.pdf](https://github.com/SmithSamuelM/Papers/blob/master/presentations/KERI2_Overview_IIW_2020_A.pdf)

[https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI\\_WP\\_2.x.web.pdf](https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI_WP_2.x.web.pdf)

<https://github.com/decentralized-identity/keri>

<https://github.com/decentralized-identity/keri/blob/master/implementation.md>

<https://hackmd.io/orhyiJkLT721v4PCPkvQiA?both>

# Background References

## Self-Certifying Identifiers:

Girault, M., "Self-certified public keys," EUROCRYPT 1991: Advances in Cryptology, pp. 490-497, 1991

[https://link.springer.com/content/pdf/10.1007%2F3-540-46416-6\\_42.pdf](https://link.springer.com/content/pdf/10.1007%2F3-540-46416-6_42.pdf)

Mazieres, D. and Kaashoek, M. F., "Escaping the Evils of Centralized Control with self-certifying pathnames," MIT Laboratory for Computer Science,

<http://www.sigops.org/ew-history/1998/papers/mazieres.ps>

Kaminsky, M. and Banks, E., "SFS-HTTP: Securing the Web with Self-Certifying URLs," MIT, 1999

<https://pdos.csail.mit.edu/~kaminsky/sfs-http.ps>

Mazieres, D., "Self-certifying File System," MIT Ph.D. Dissertation, 2000/06/01

<https://pdos.csail.mit.edu/~ericp/doc/sfs-thesis.ps>

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>

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>

Smith, S. M., "Key Event Receipt Infrastructure (KERI) Design and Build", arXiv, 2019/07/03 revised 2020/04/23

<https://arxiv.org/abs/1907.02143>

Smith, S. M., "Key Event Receipt Infrastructure (KERI) Design", 2020/04/22

[https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI\\_WP\\_2.x.web.pdf](https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI_WP_2.x.web.pdf)

## 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>

# Human Basis-of-Trust “in person”

*I can know you – therefore I can trust you*



*“on the internet”*

*I can't really know you – therefore I can't really trust you*

# Replace human *basis-of-trust* with cryptographic *root-of-trust*.

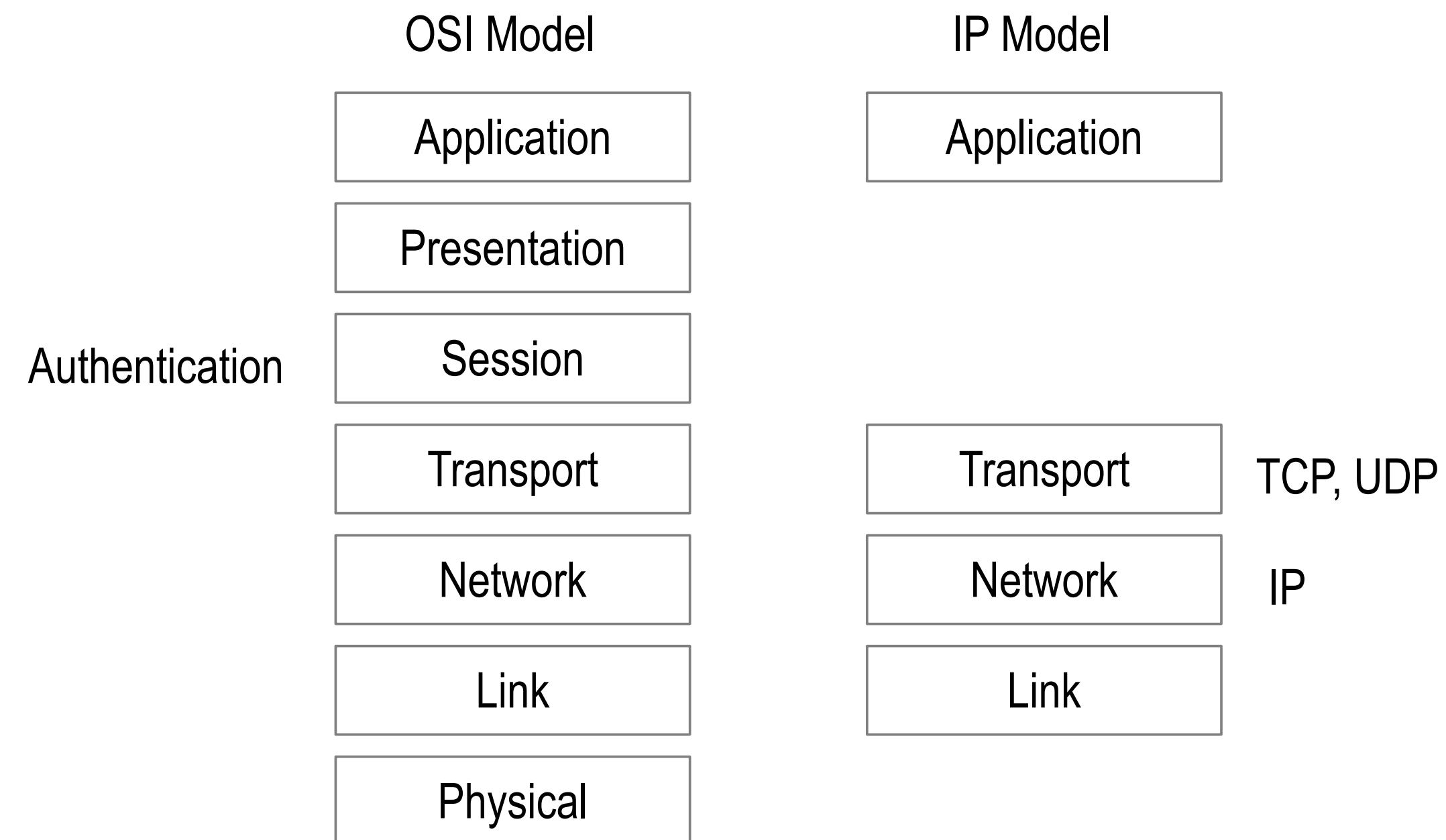
With verifiable digital signatures from asymmetric key crypto –  
we may not trust in “**what**” was said, but we may trust in “**who**” said it.

We may verify that the **controller** of a private key, (the **who**), made a statement  
but not the validity of the statement itself.

The root-of-trust is **consistent attribution** via verifiable integral non-repudiable statements

We may build trust over time in **what** was said via histories  
of verifiably attributable (to **whom**) consistent statements i.e. **reputation**.

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

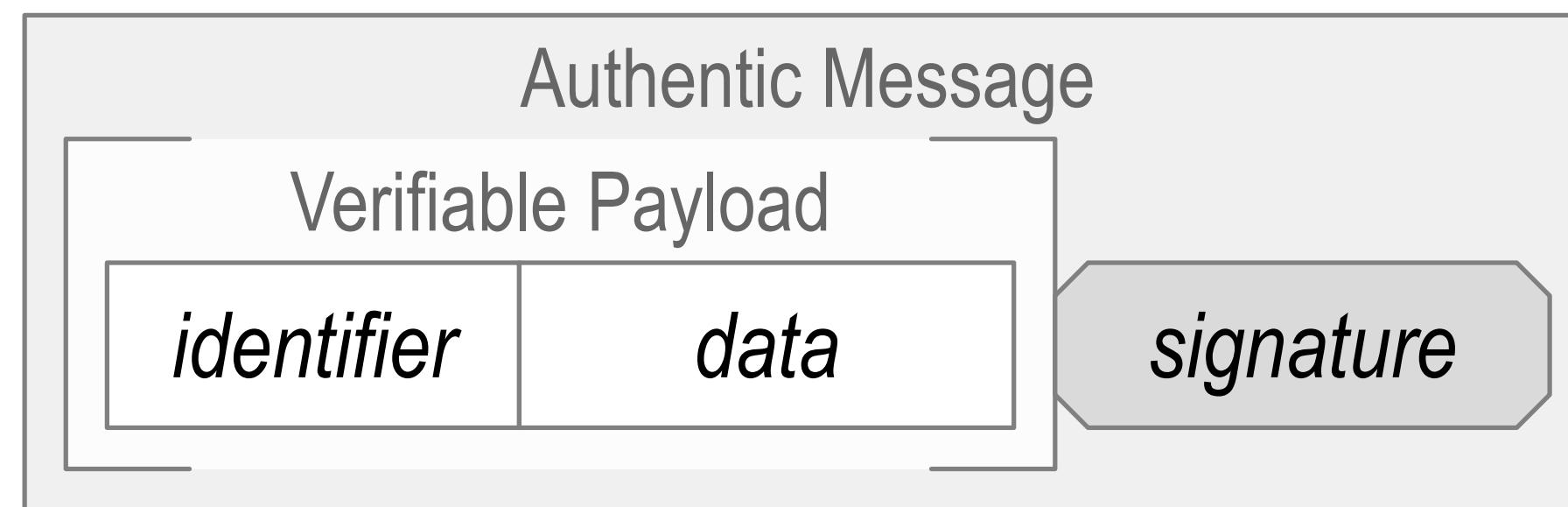
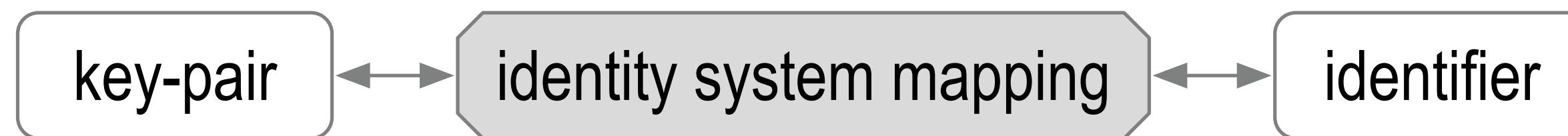


Instead ...

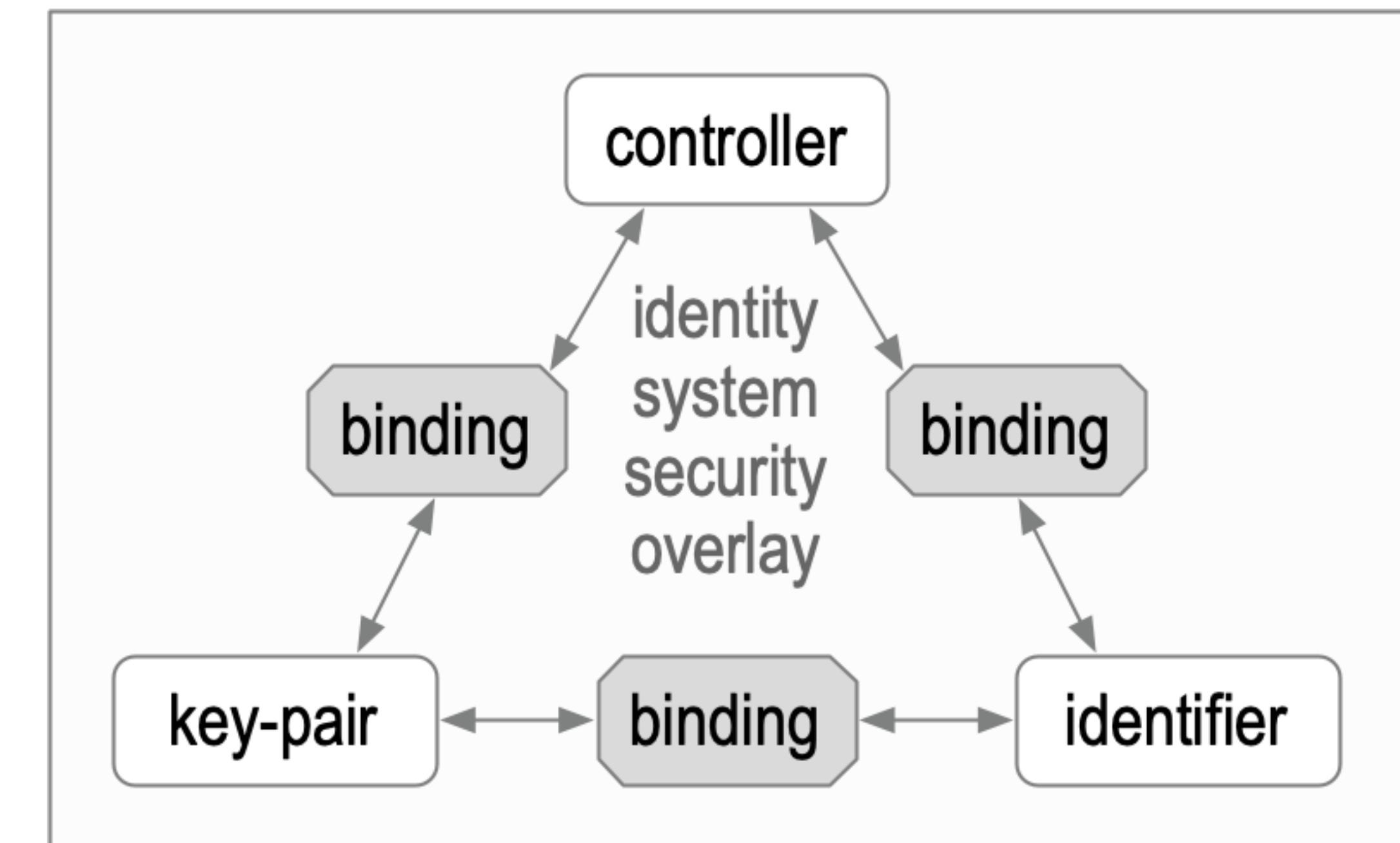
We use *bolt-on* identity system security overlays.  
(DNS-CA ...)

# Identity System Security Overlay

Establish authenticity of IP packet's message payload.

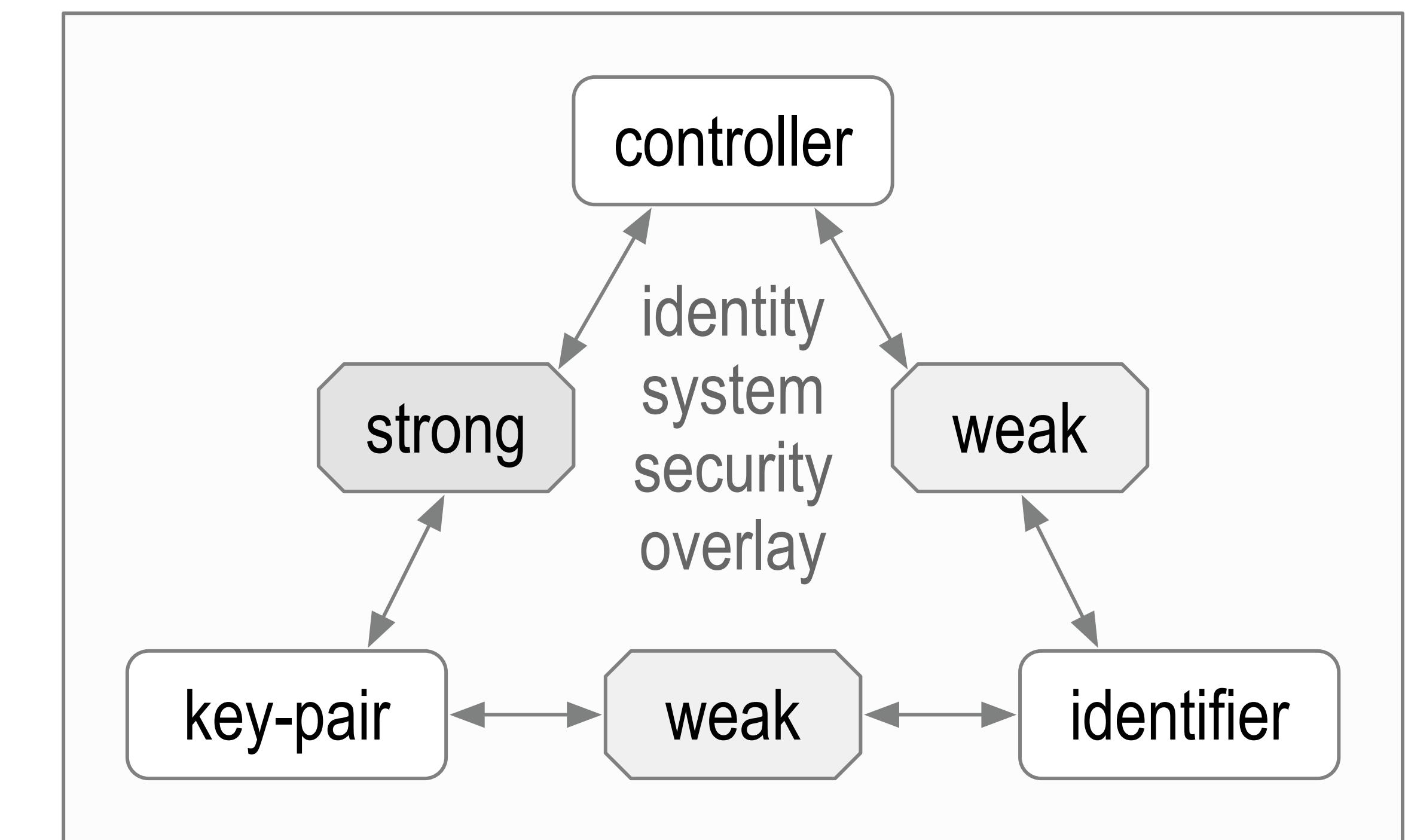
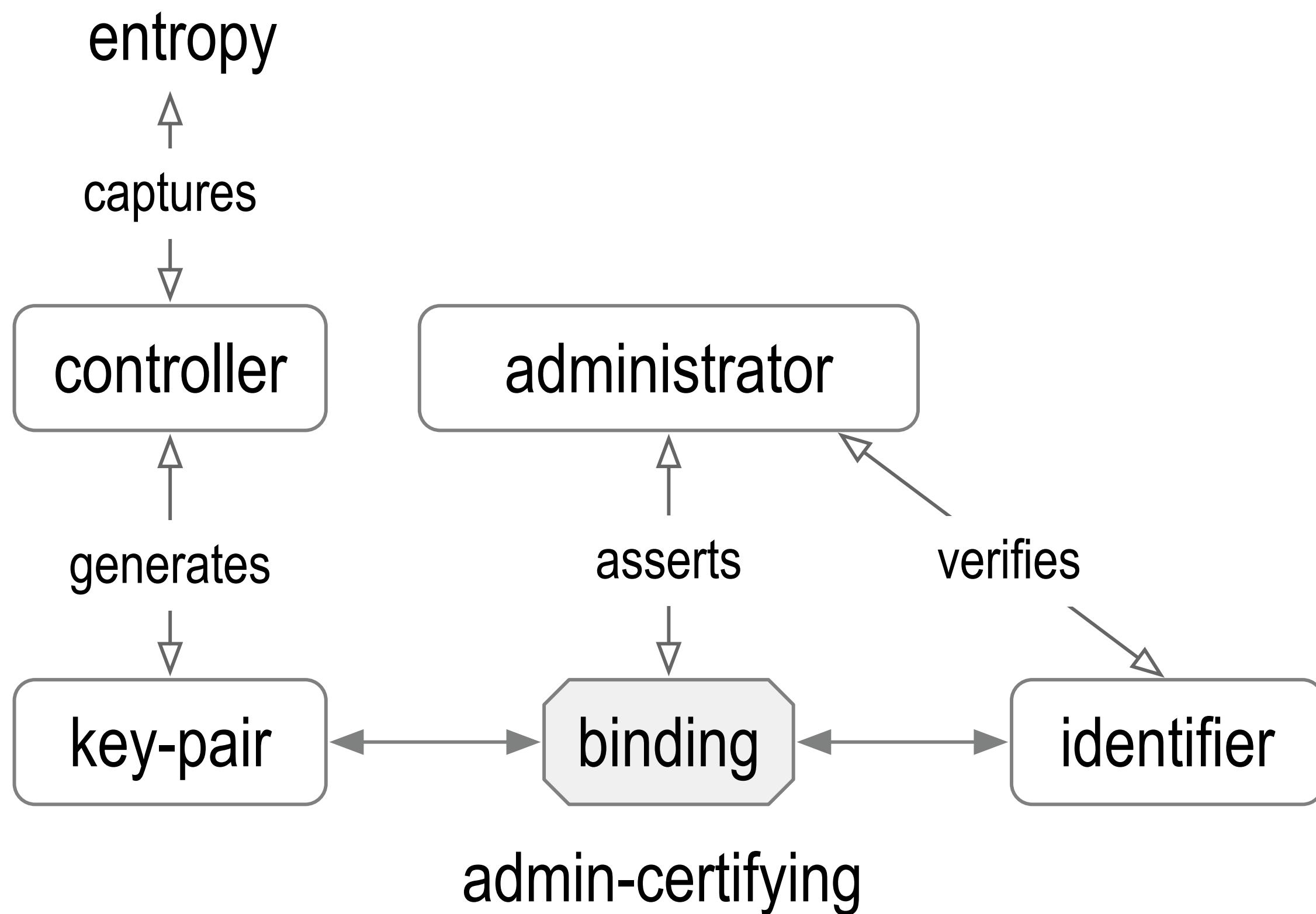


The overlay's security is contingent  
on the mapping's security.



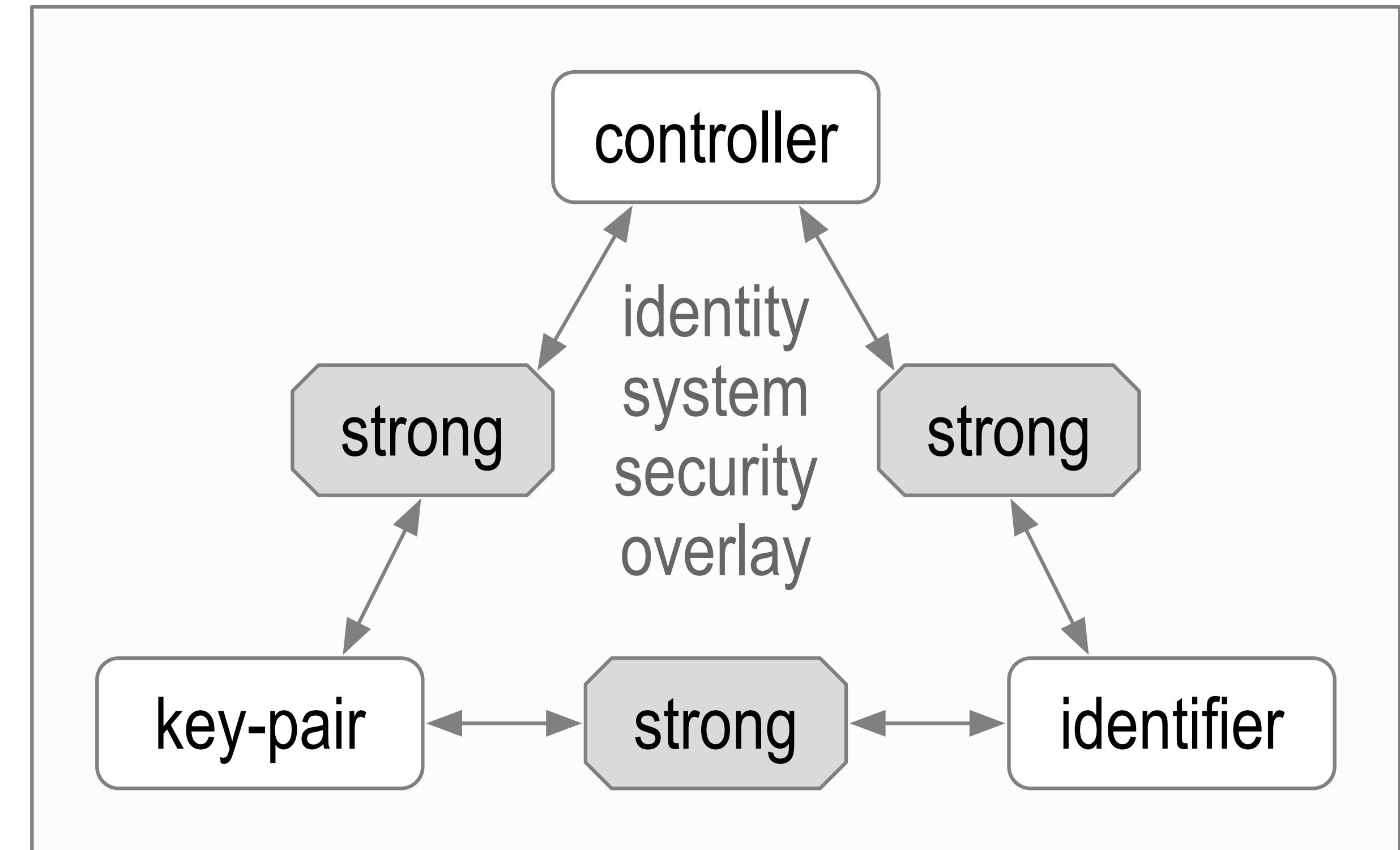
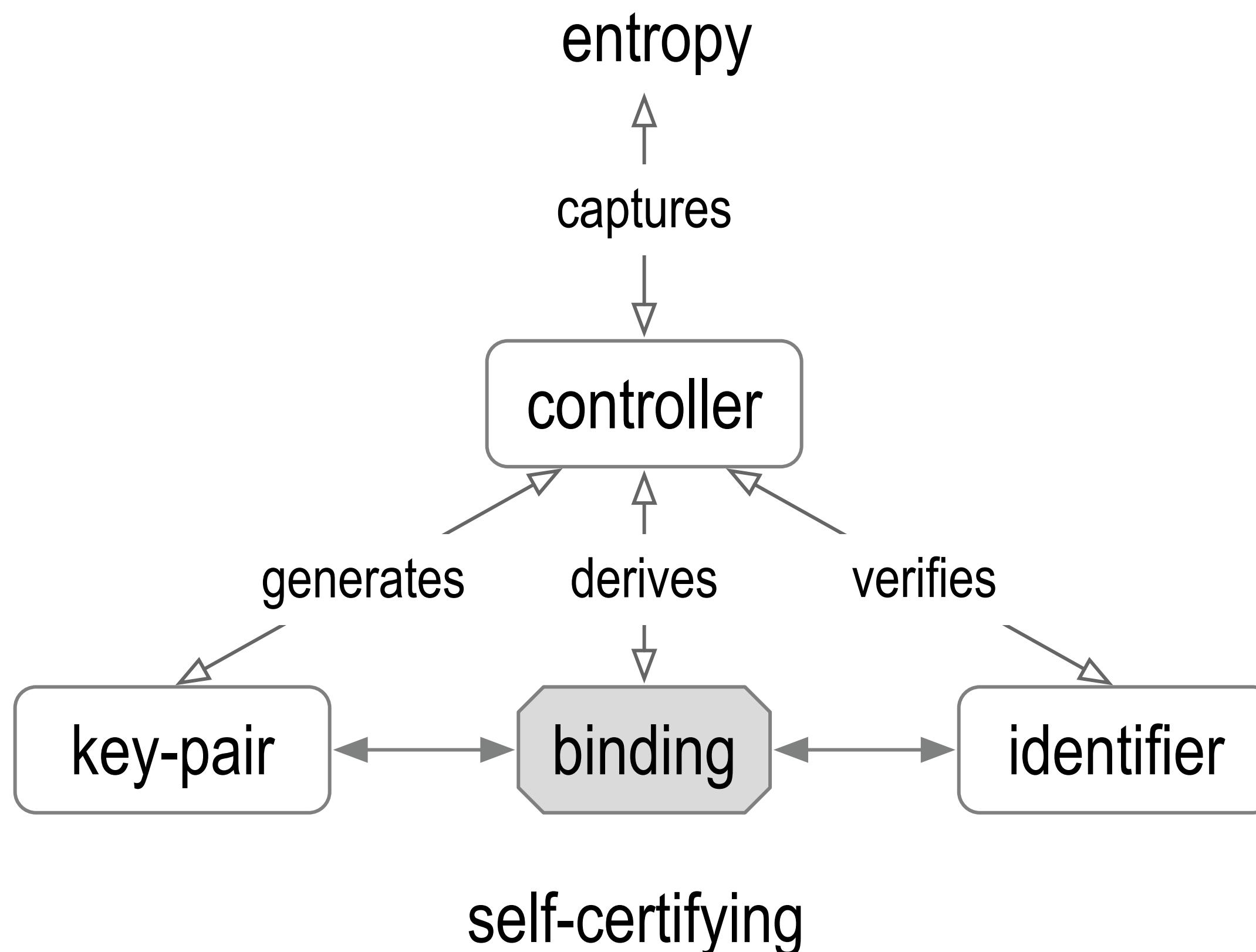
Identifier Issuance

# Administrative Identifier Issuance and Binding



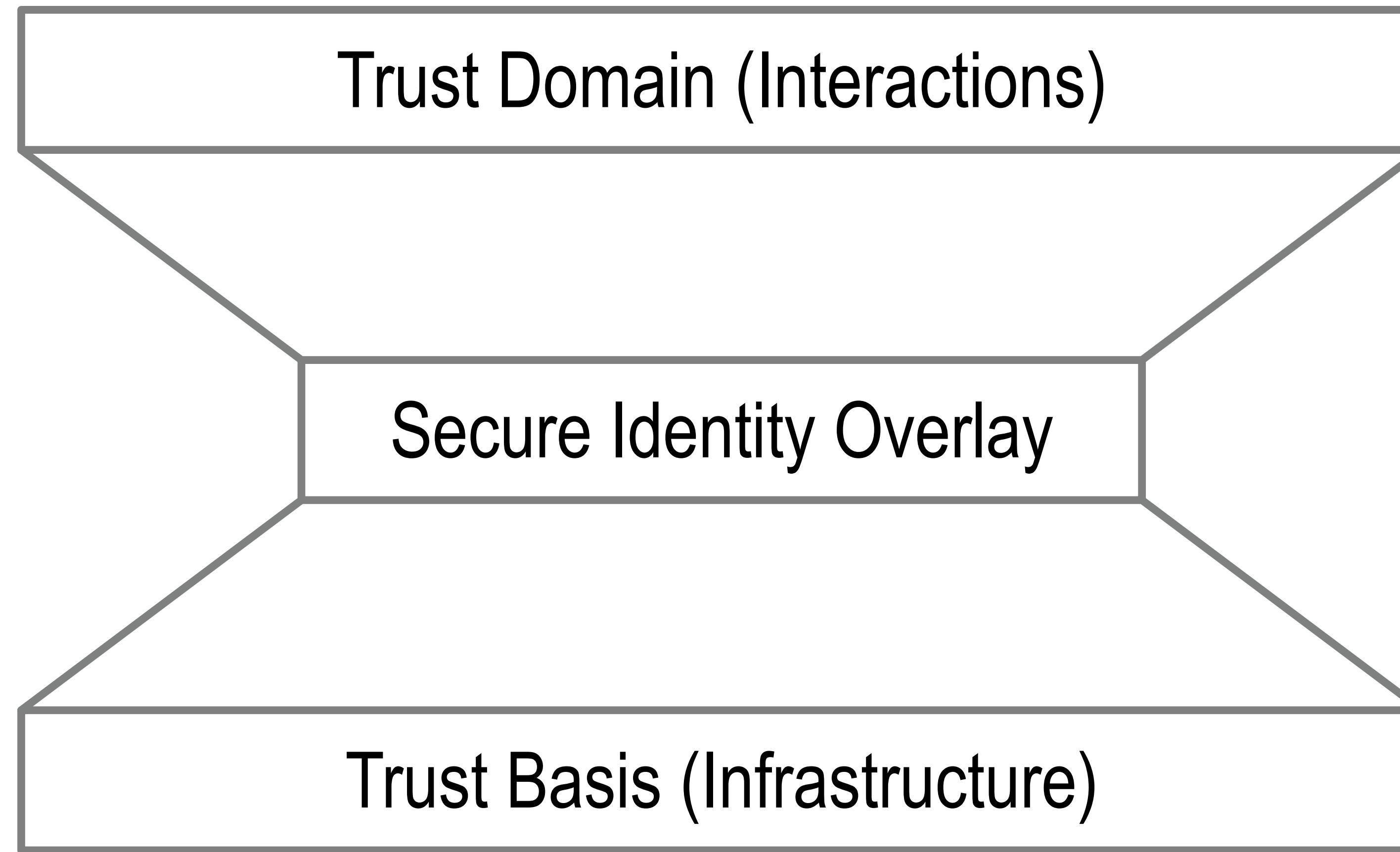
Admin-Certifying Identifier Issuance

# Self-Certifying Identifier Issuance and Binding

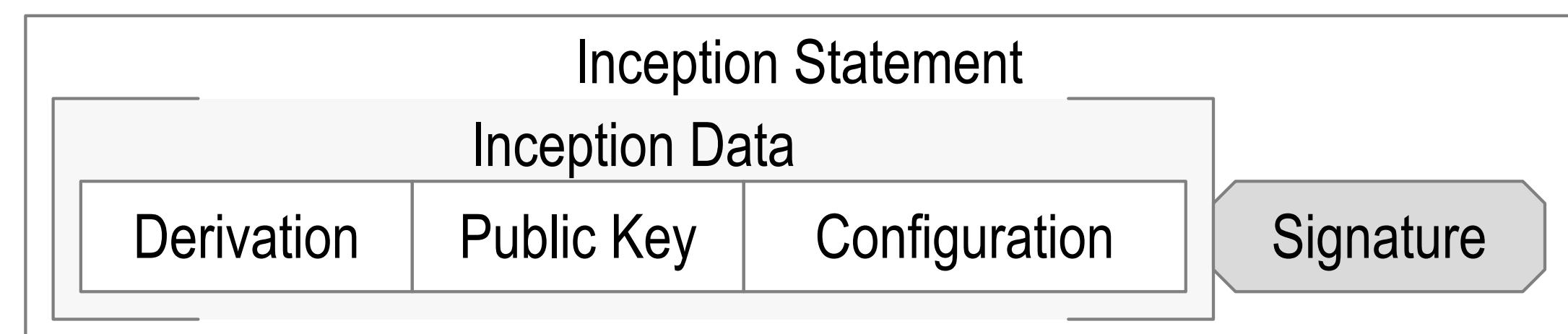
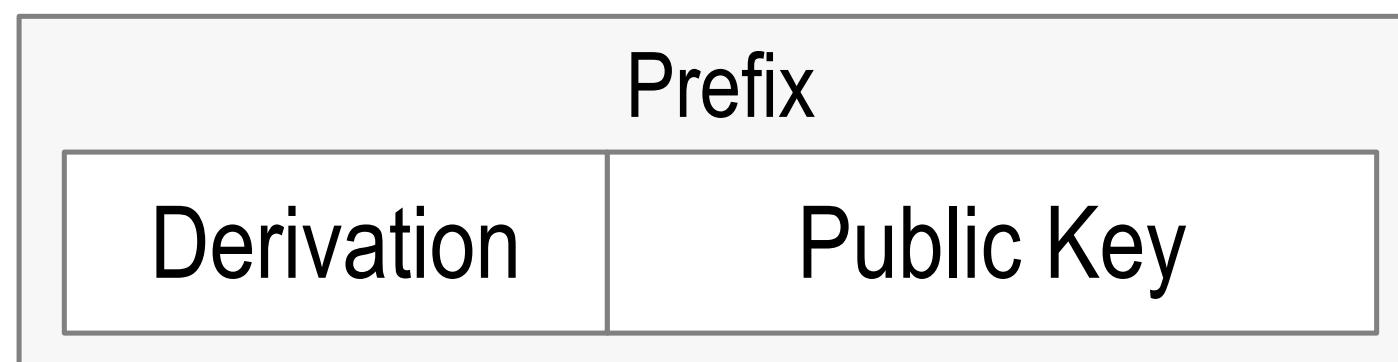
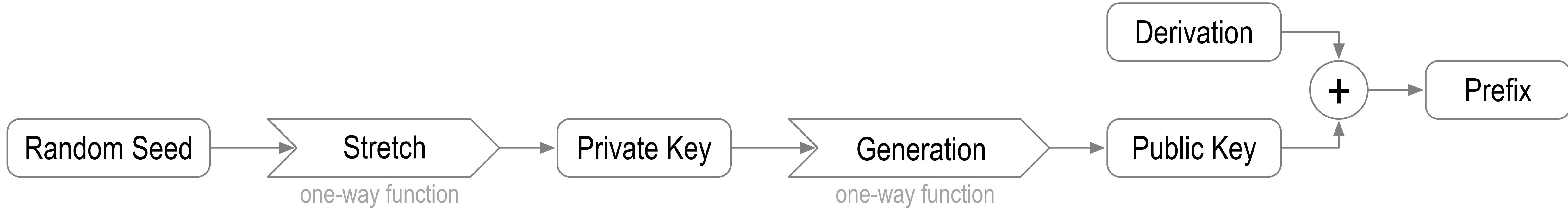


Self-Certifying Identifier Issuance

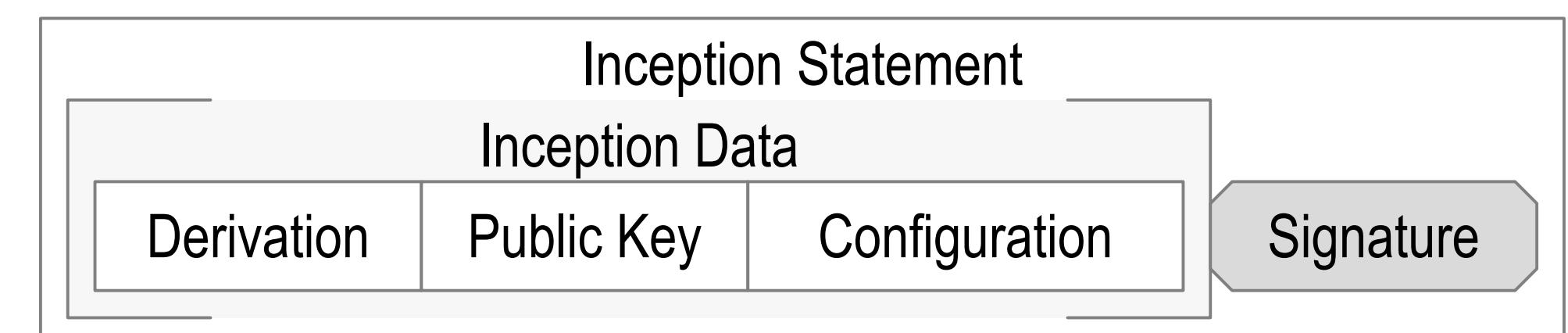
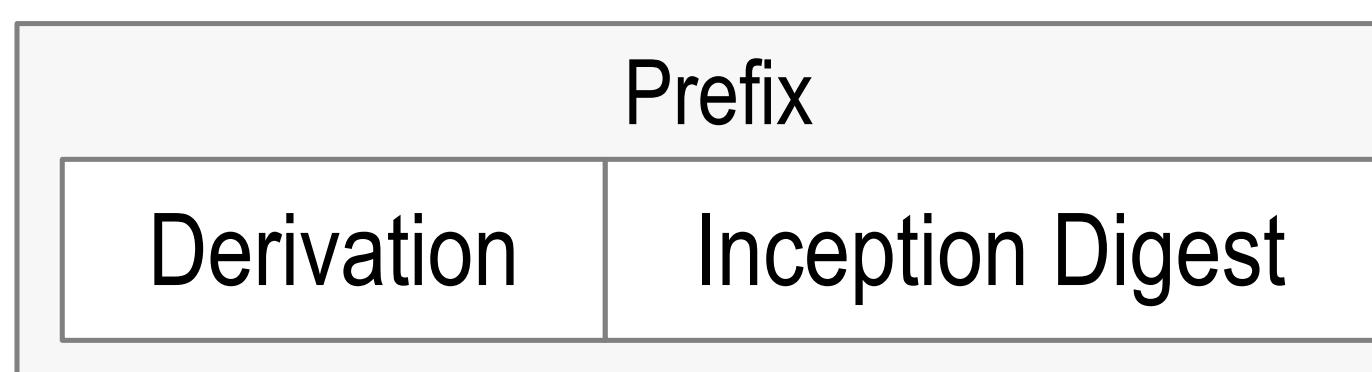
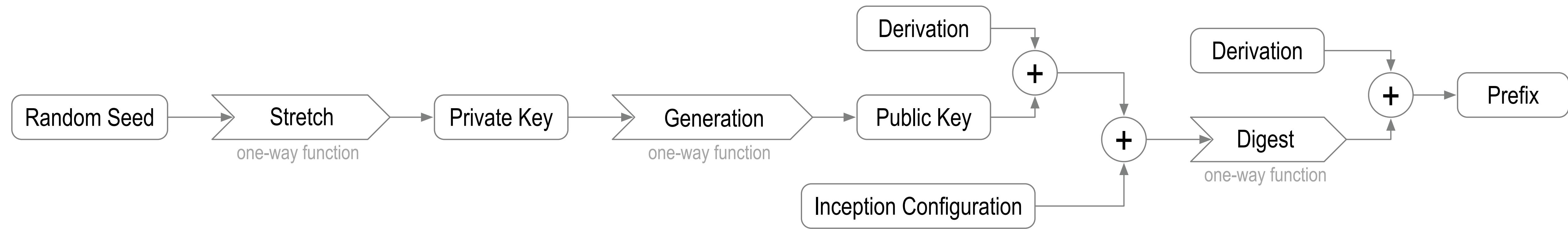
# Identity System Security Overlay



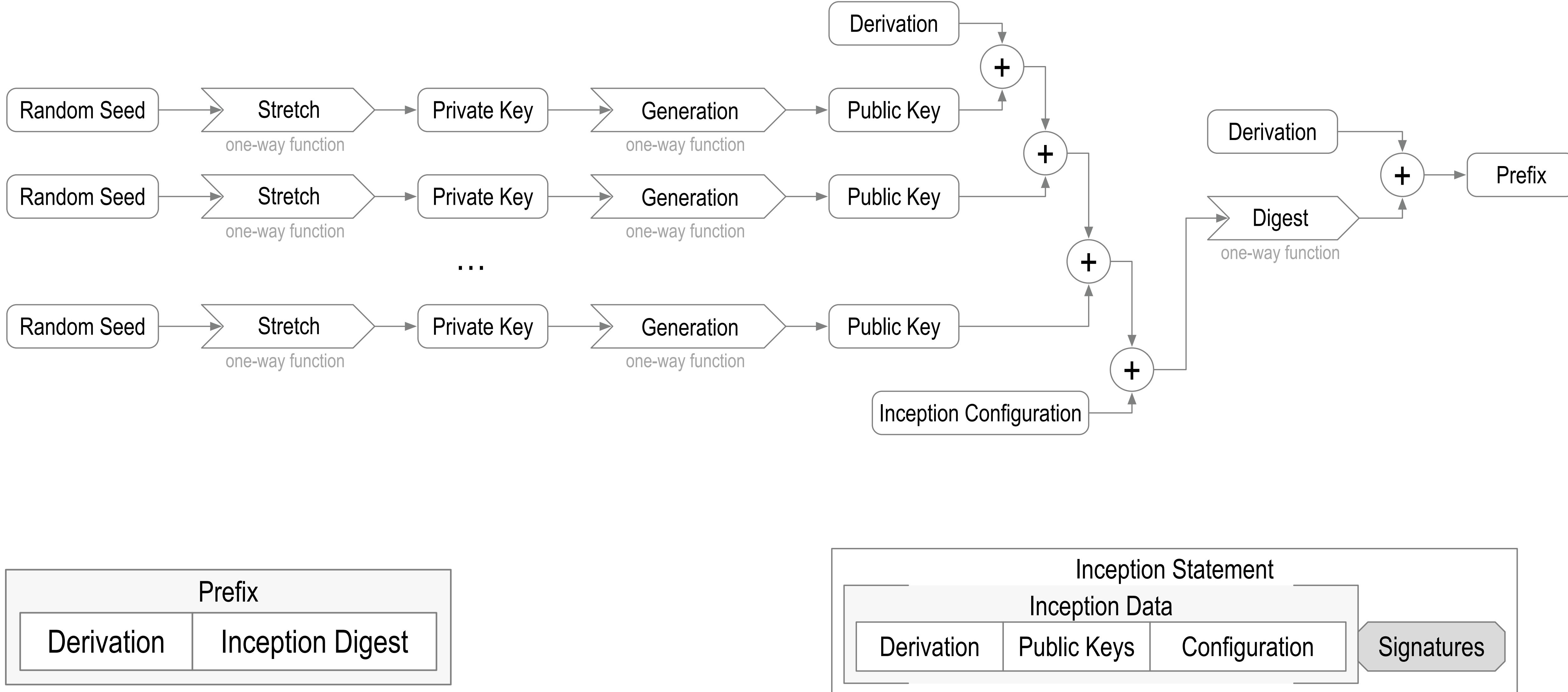
# Basic



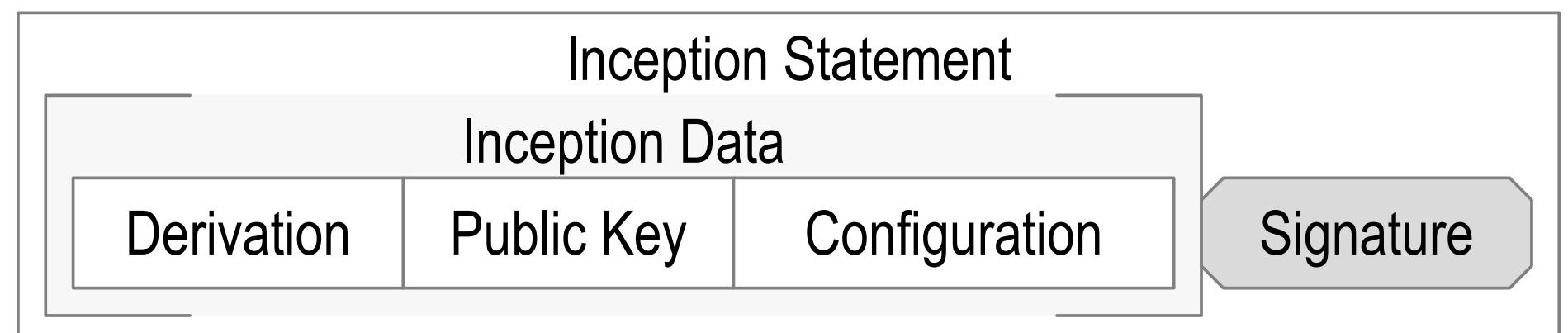
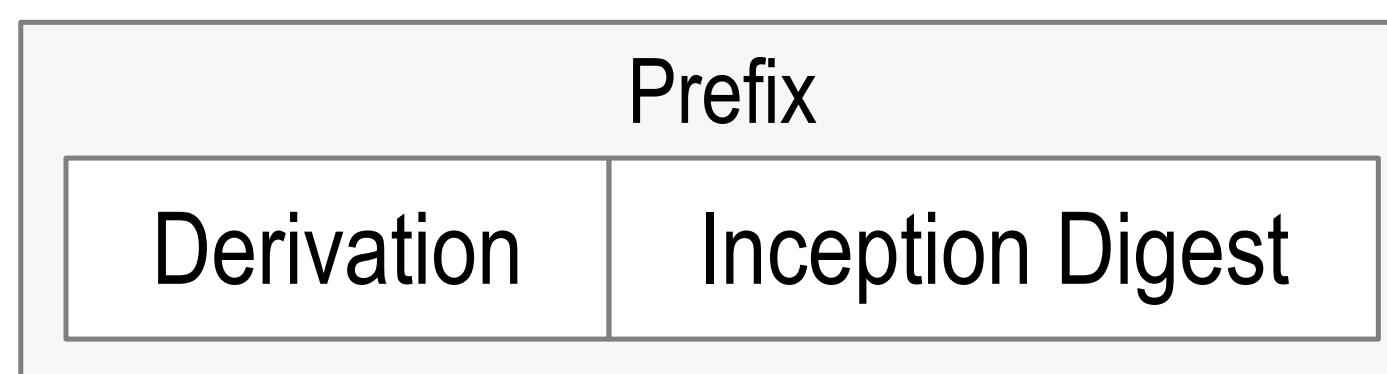
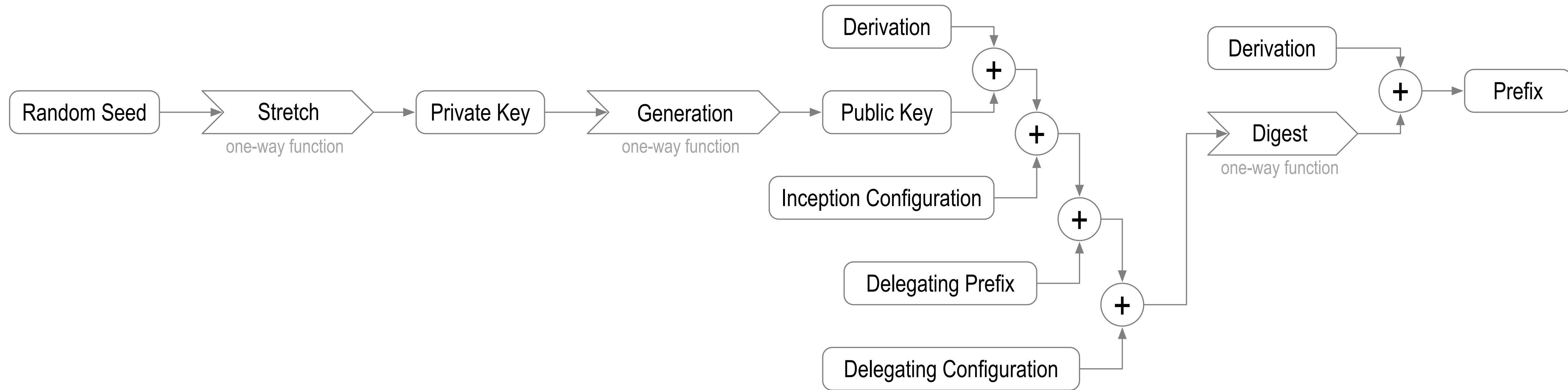
# Self-Addressing



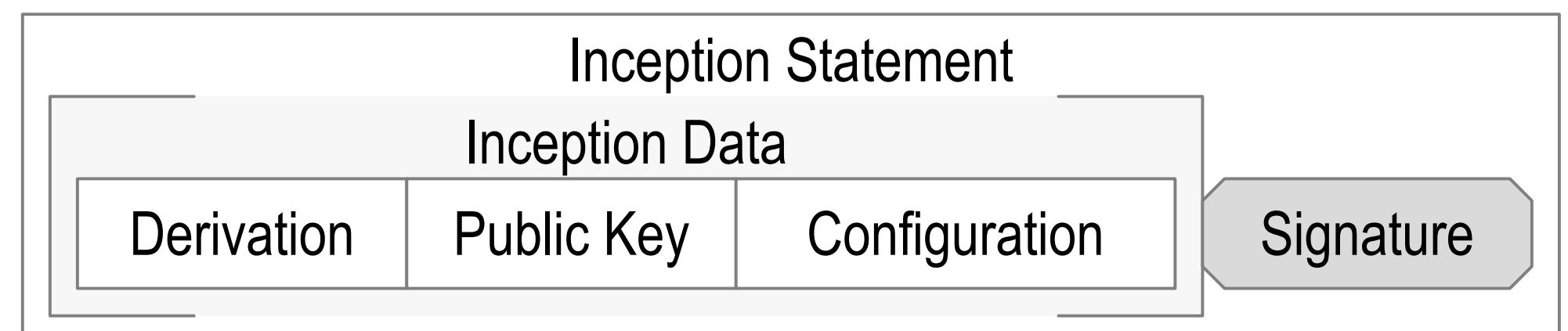
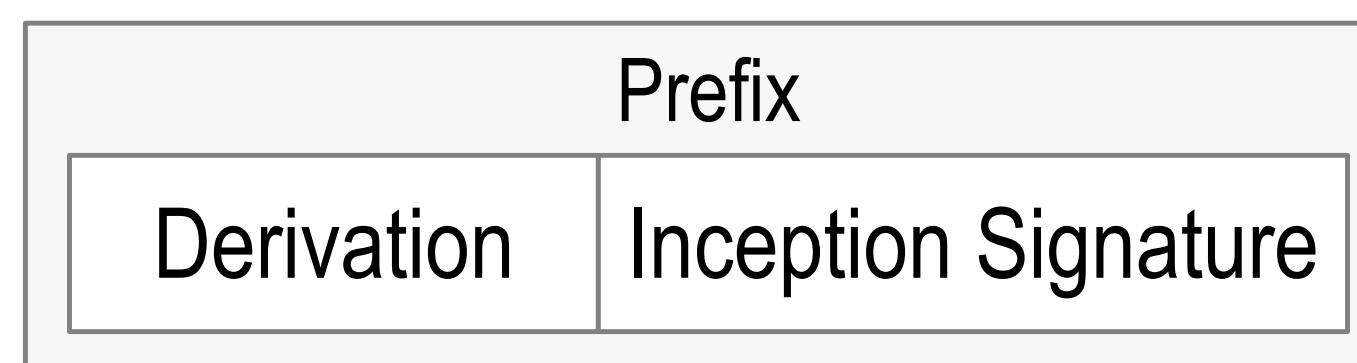
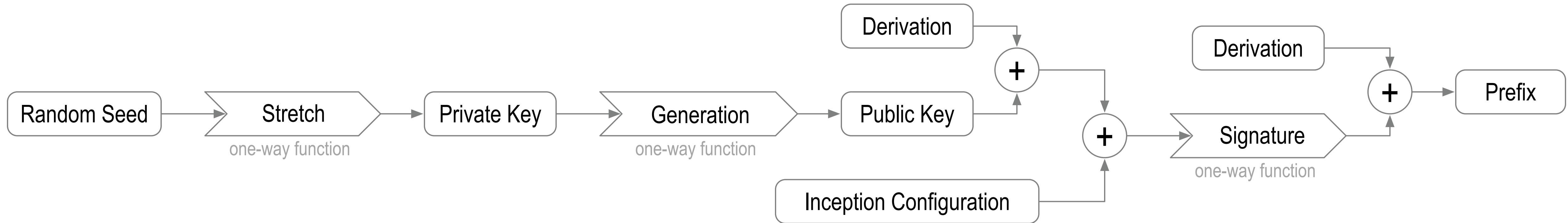
# Multi-Sig Self-Addressing



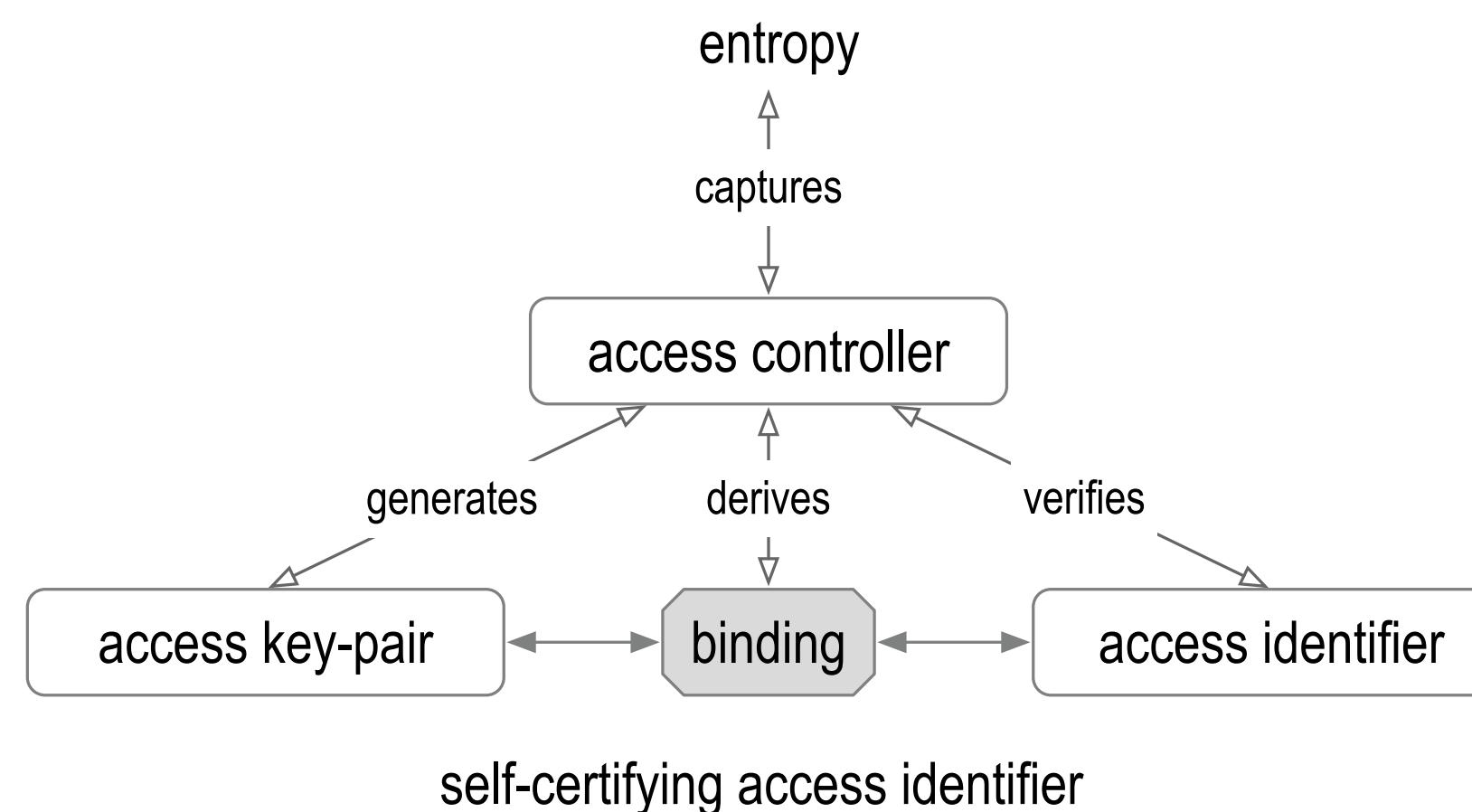
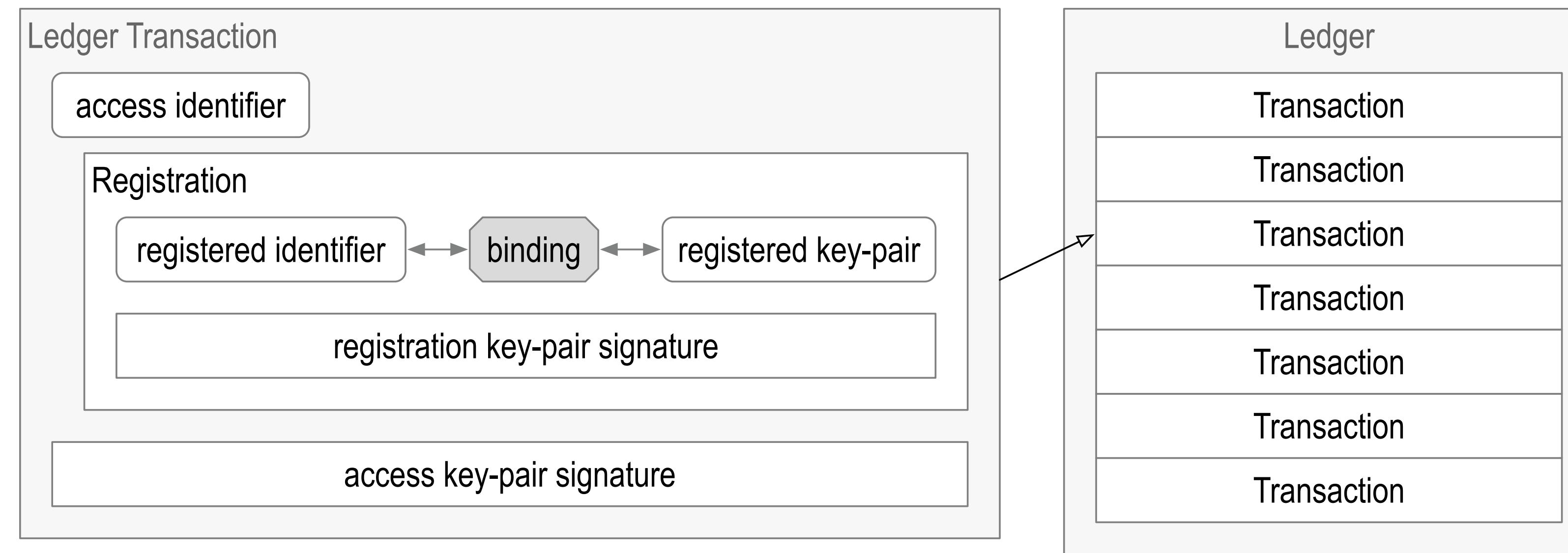
# Delegated Self-Addressing



# Self-Signing



# Ledger Registration



Access identifier may have self-certifying primary root-of-trust  
but registered identifier does not, even if its format appears 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*.

ANs are *portable* = truly self-sovereign.

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

# Autonomic Identity System

*why, how* – *who* controls *what, when, and how?*

## Root-of-Trust

cryptographic autonomic identifier = *why, how*

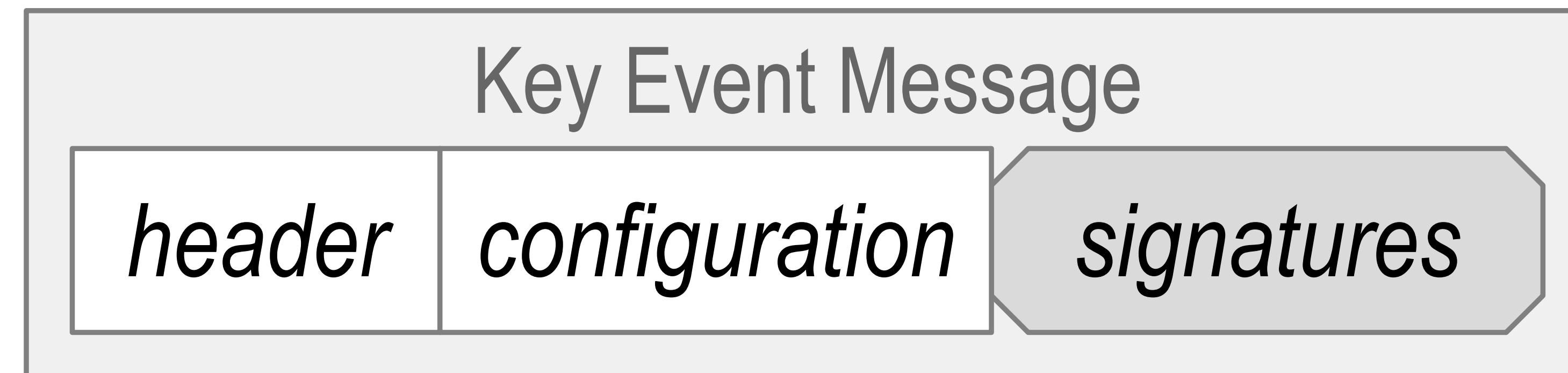
## Source-of-Truth

controller of the private key = *who*

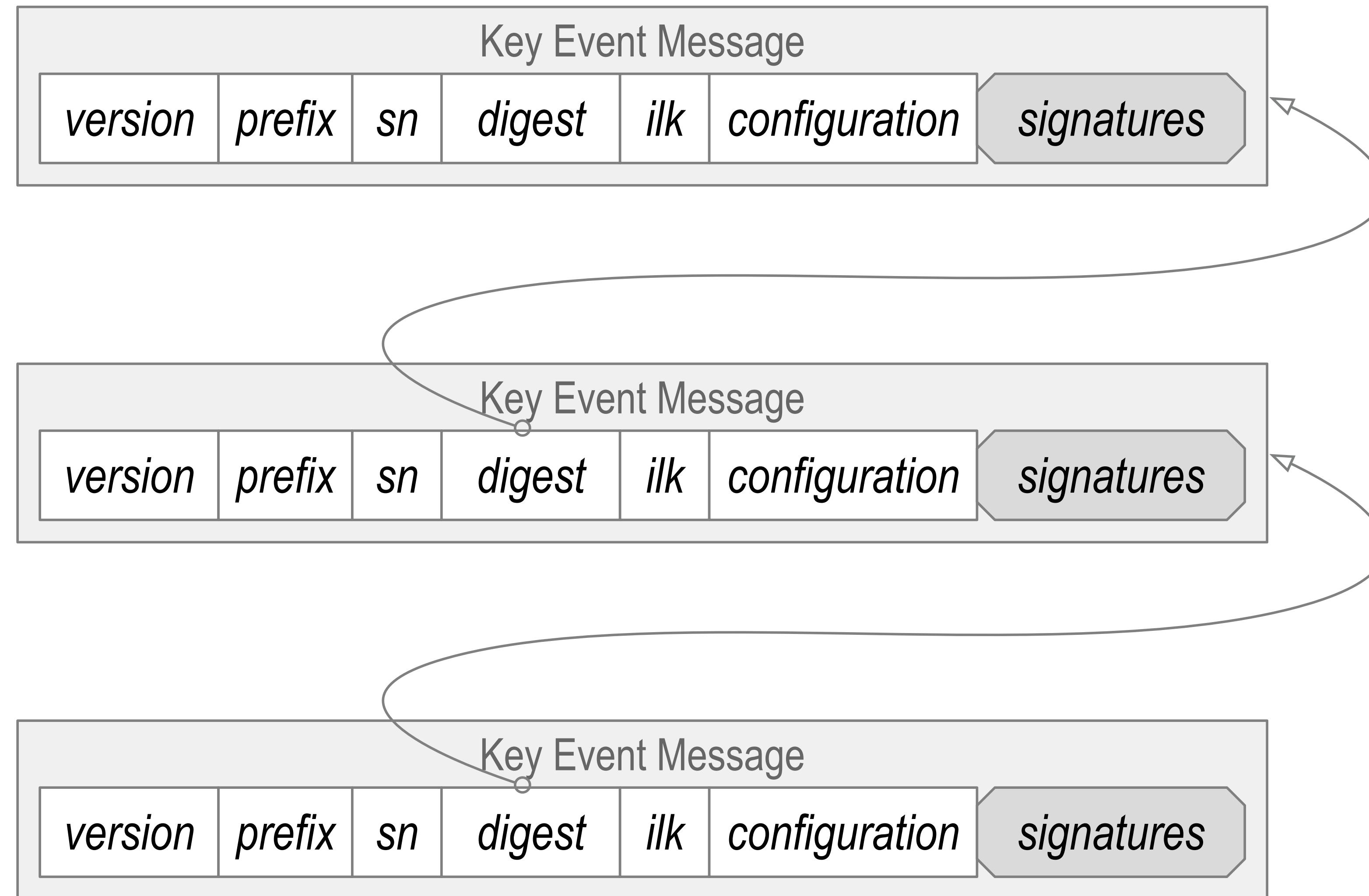
## Loci-of-Control

authoritative operation = *what, when, how*

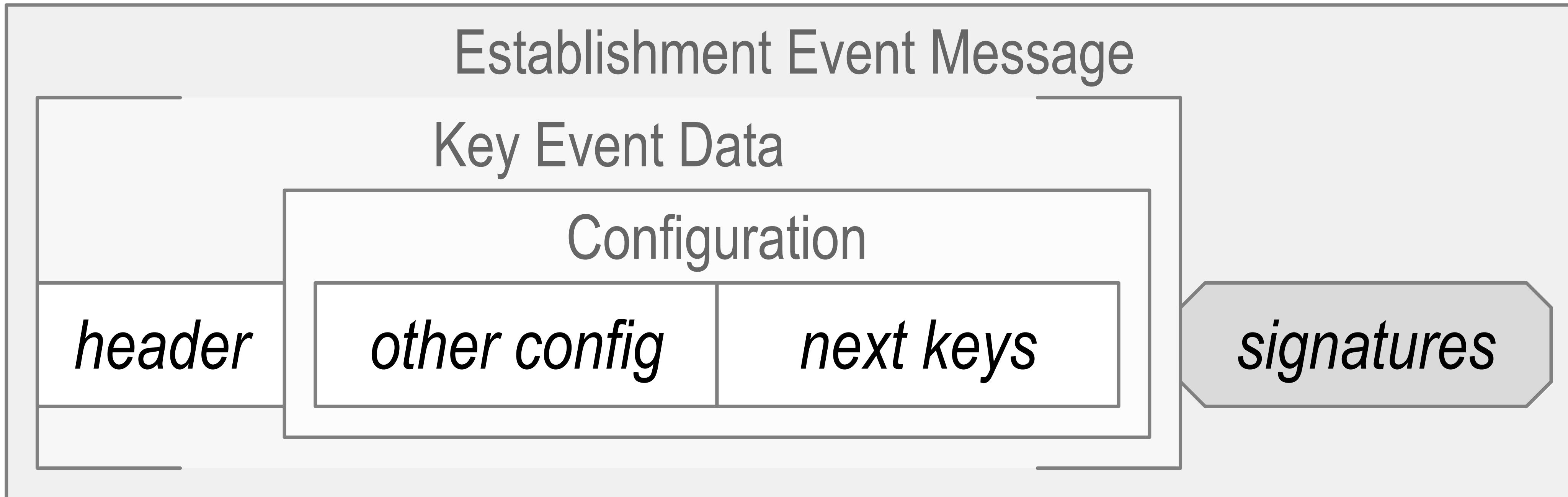
# Key Event Message



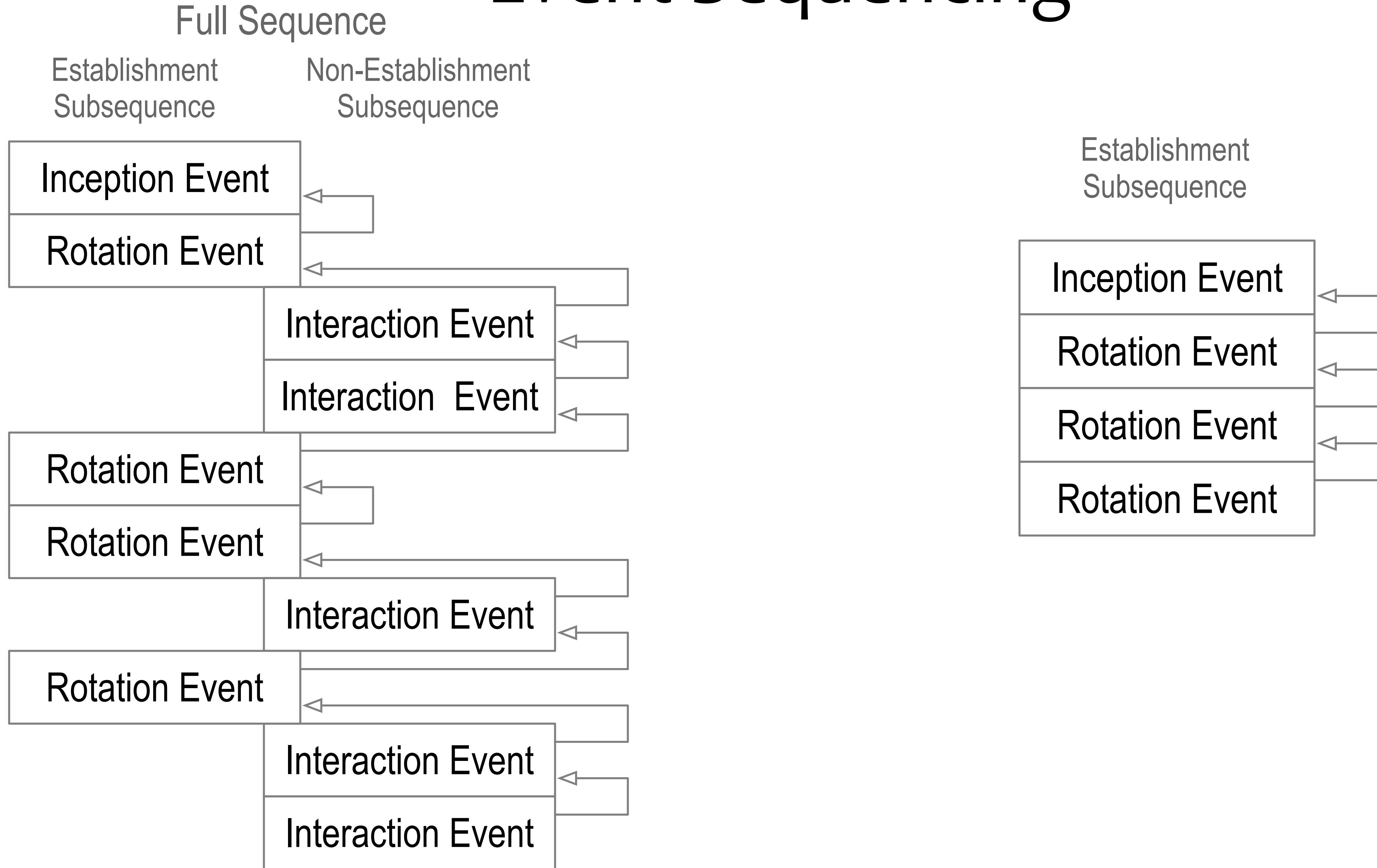
# Event Digest Chaining



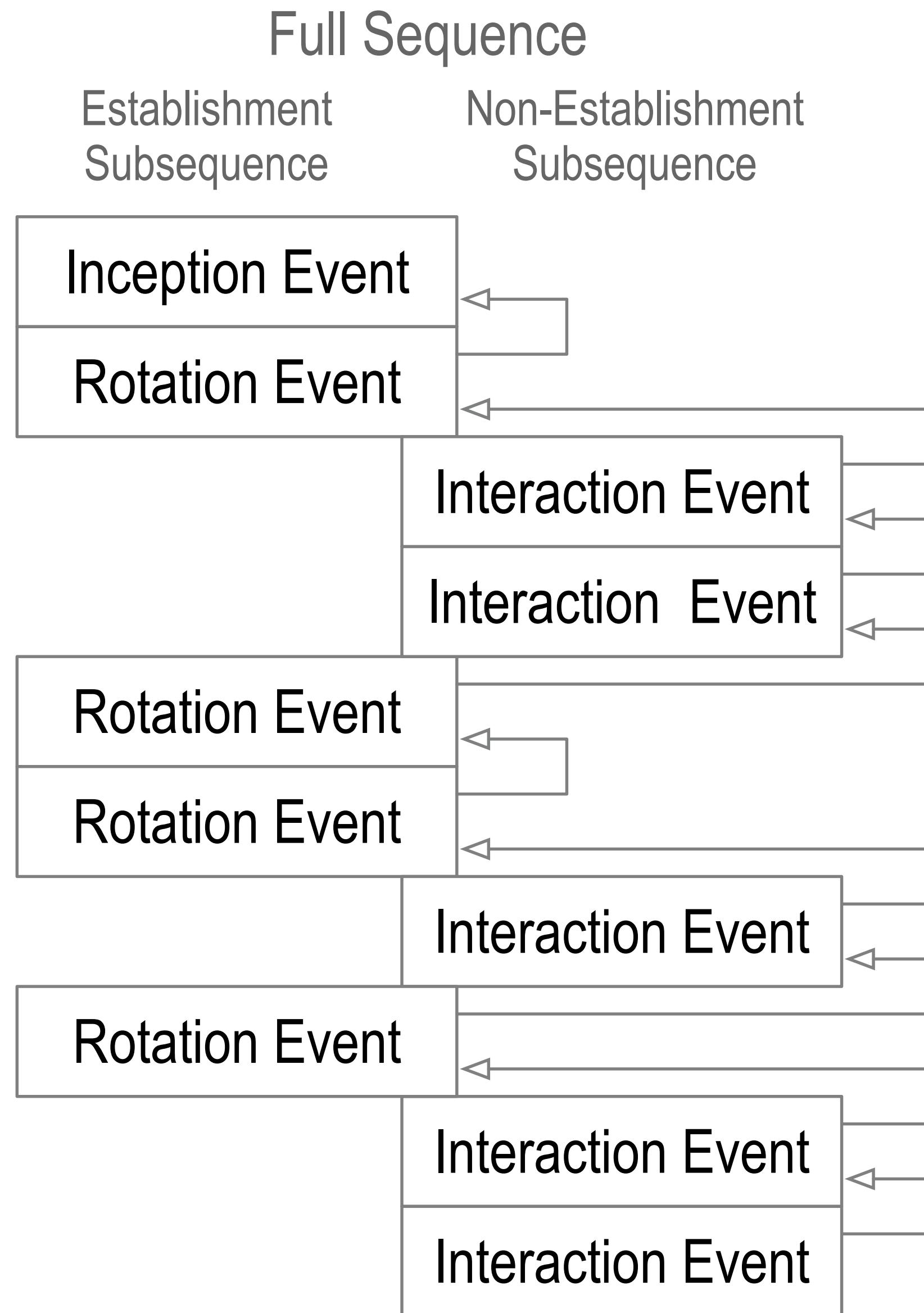
# Establishment Event



# Event Sequencing



# Inconsistency and Duplication



## Inconsistency vs. Duplication

*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.

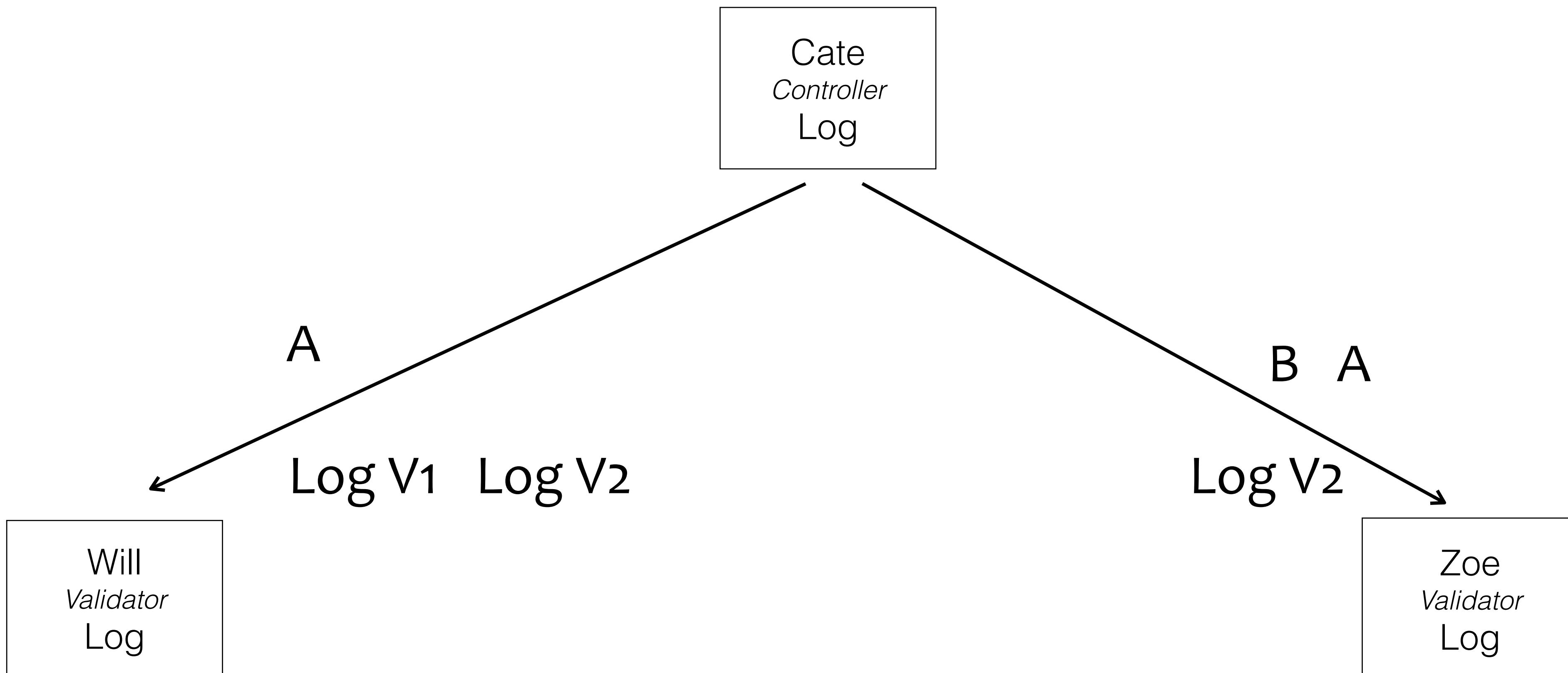
Duplicity detection protects against  
external inconsistency.

Cate promises to provide a consistent pair-wise log.

*Local Consistency Guarantee*

# Duplicity Game

How may Cate be *duplicitious* and not get caught?



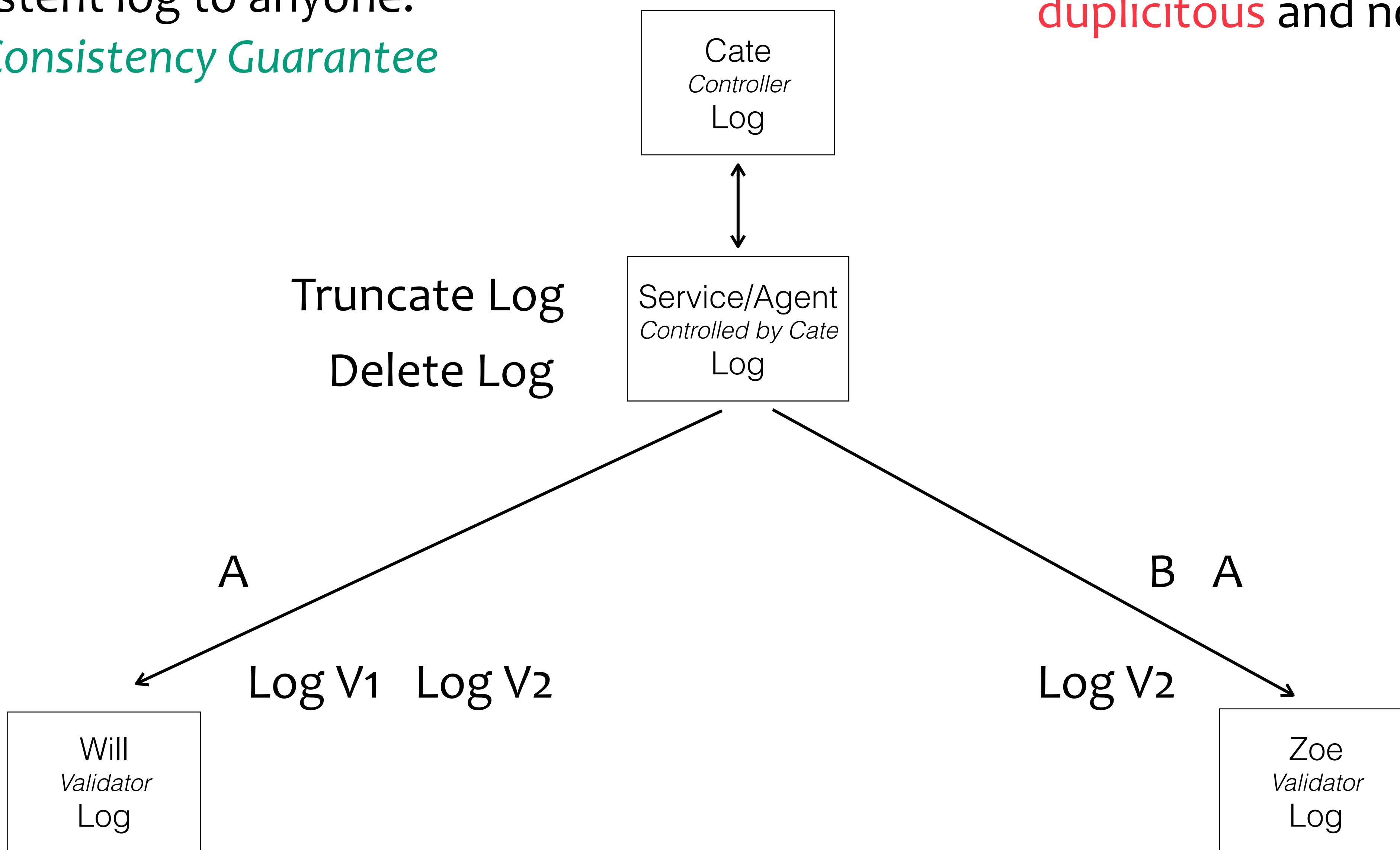
private (one-to-one) interactions

Service promises to provide a consistent log to anyone.

*Local Consistency Guarantee*

# Duplicity Game

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



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

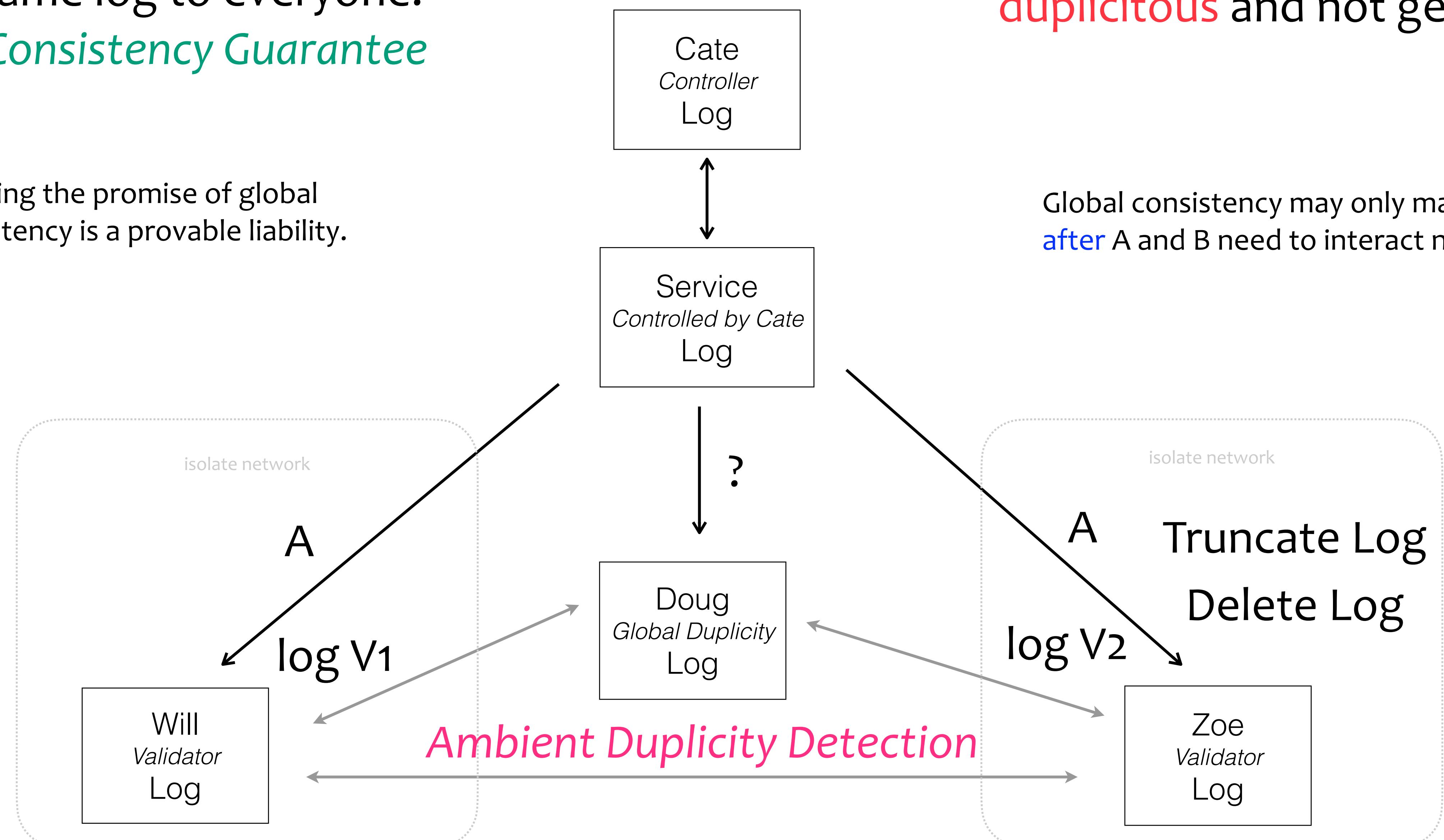
Service promises to provide exact same log to everyone.  
*Global Consistency Guarantee*

# Duplicity Game

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** A and B need to interact not before.



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

# KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority (provenance)* for each identifier.

A *proof* is in the form of an identifier's *key event receipt log (KERL)*.

KERLs are *End Verifiable*:

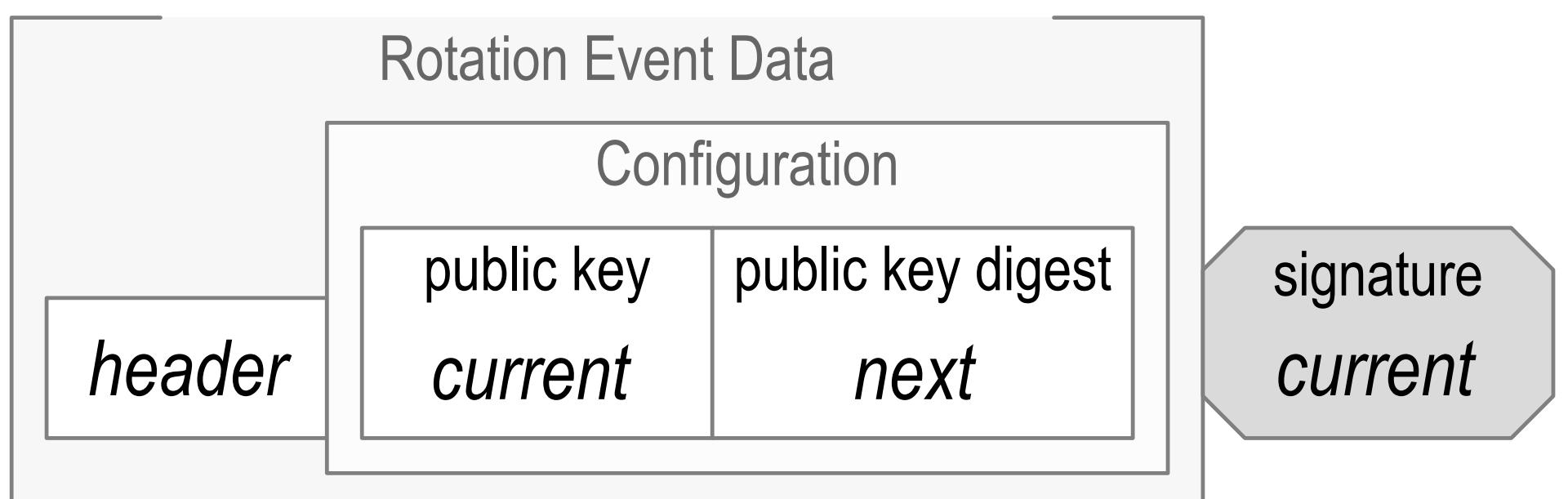
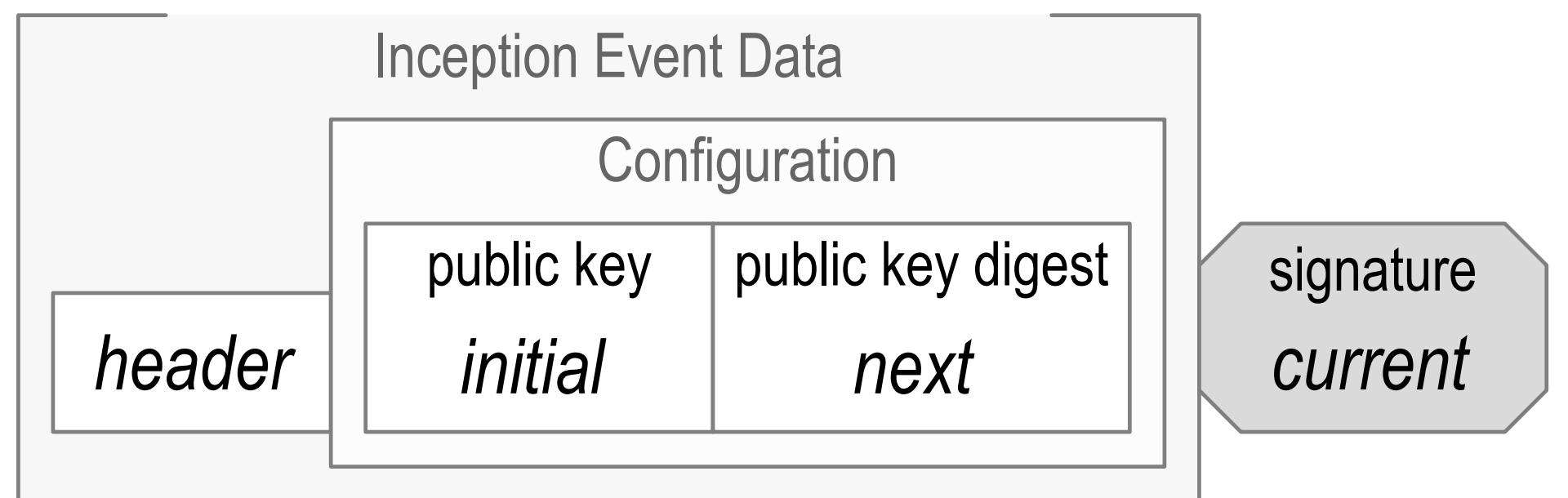
End user alone may verify. Zero trust in intervening infrastructure.

KERLs may be *Ambient Verifiable*:

Anyone may verify *anylog, anywhere, at anytime*.

KERI = self-cert root-of-trust + certificate transparency + KA<sup>2</sup>CE + recoverable + post-quantum.

# Pre-Rotation



Inception			
SN	initial	next digest	current
0	$C^0$	$\underline{C}^1$	$C^0$

Rotation			
SN	current	next digest	current
1	$C^1$	$\underline{C}^2$	$C^1$

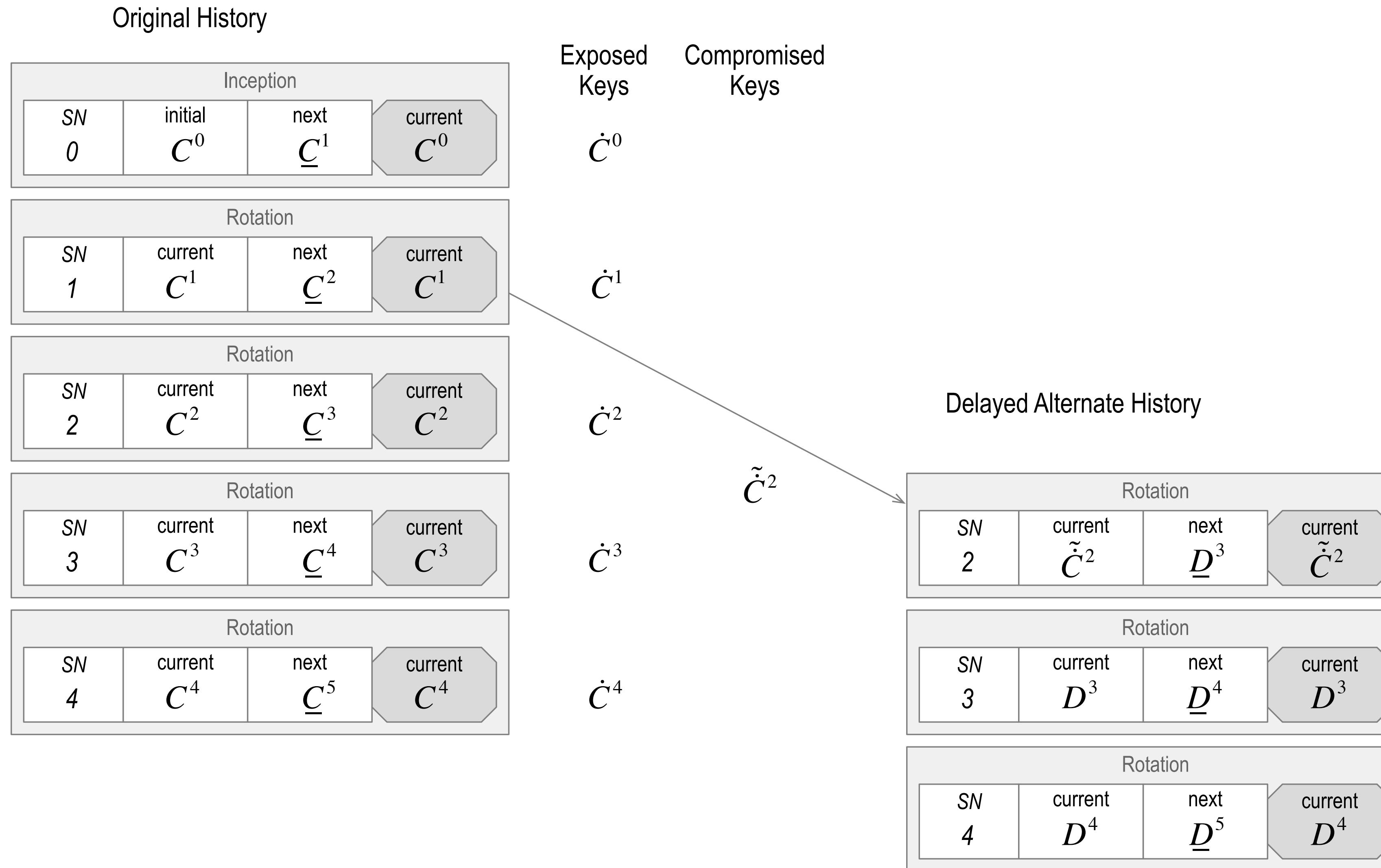
Rotation			
SN	current	next digest	current
2	$C^2$	$\underline{C}^3$	$C^2$

Rotation			
SN	current	next digest	current
3	$C^3$	$\underline{C}^4$	$C^3$

Rotation			
SN	current	next digest	current
4	$C^4$	$\underline{C}^5$	$C^4$

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

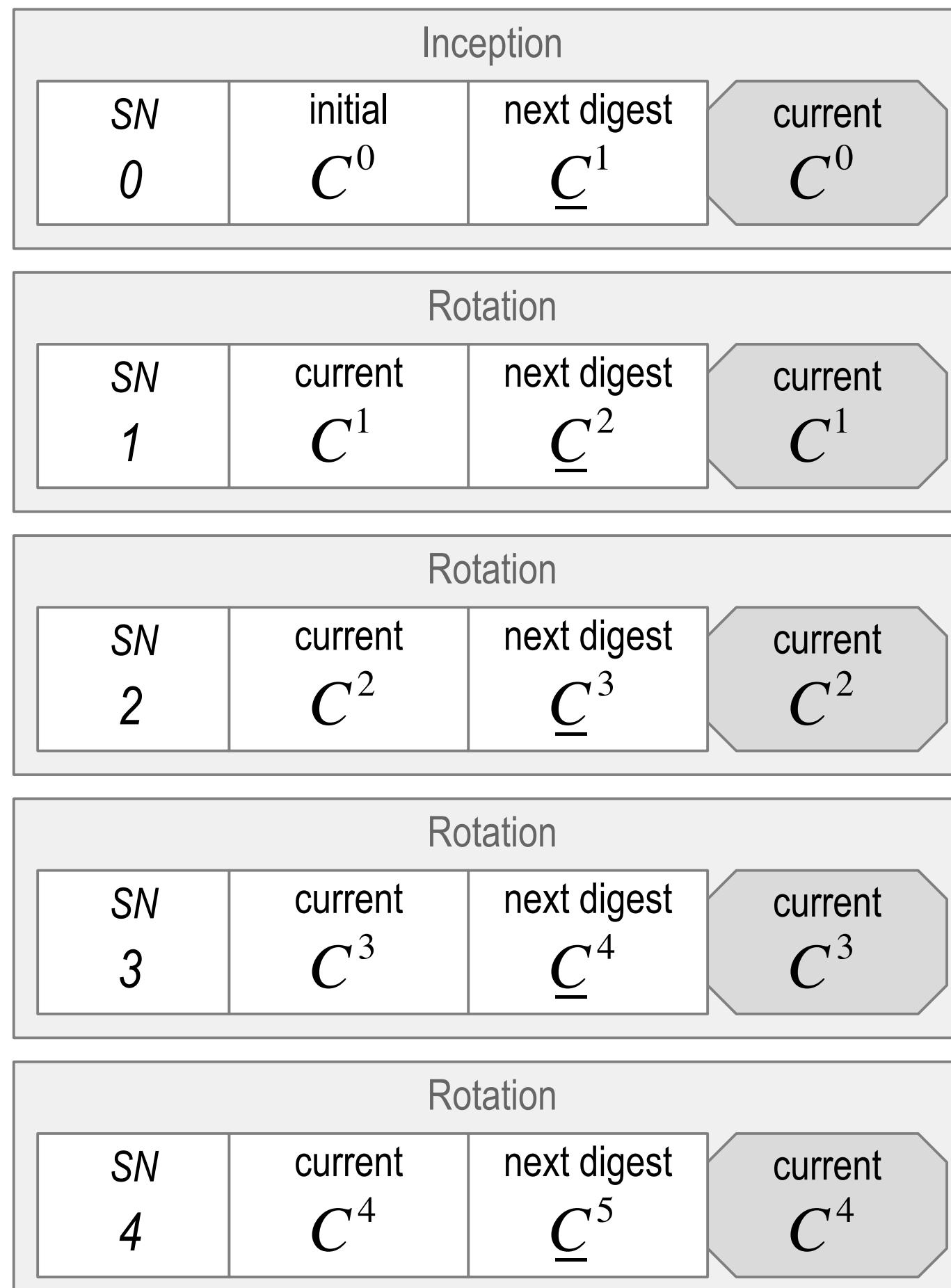
# Dead Exploit



Any copy of original history protects against successful *dead exploit*

# Live Exploit

Original History



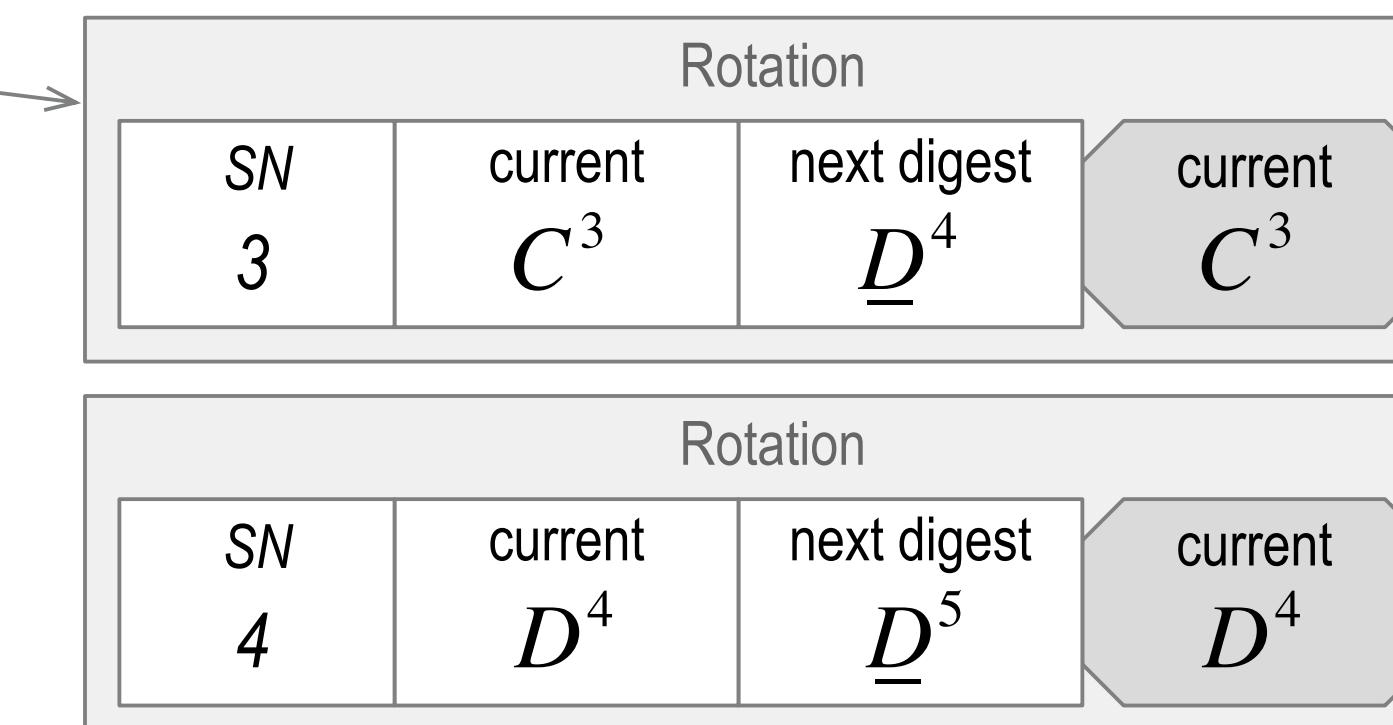
Exposed Keys  
Compromised Keys

$\dot{C}^0$

$\dot{C}^1$

$\dot{C}^2$

Preemptive Alternate History



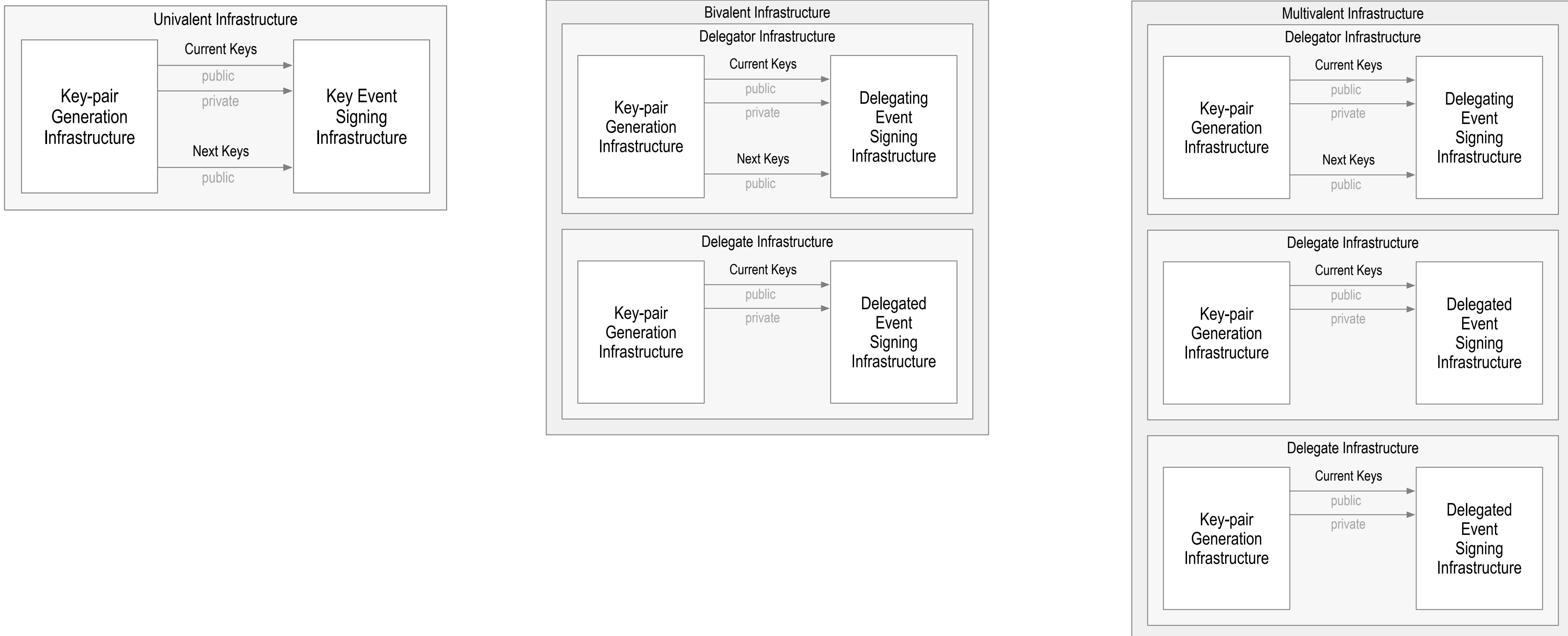
$\tilde{C}^3$

$\dot{C}^3$

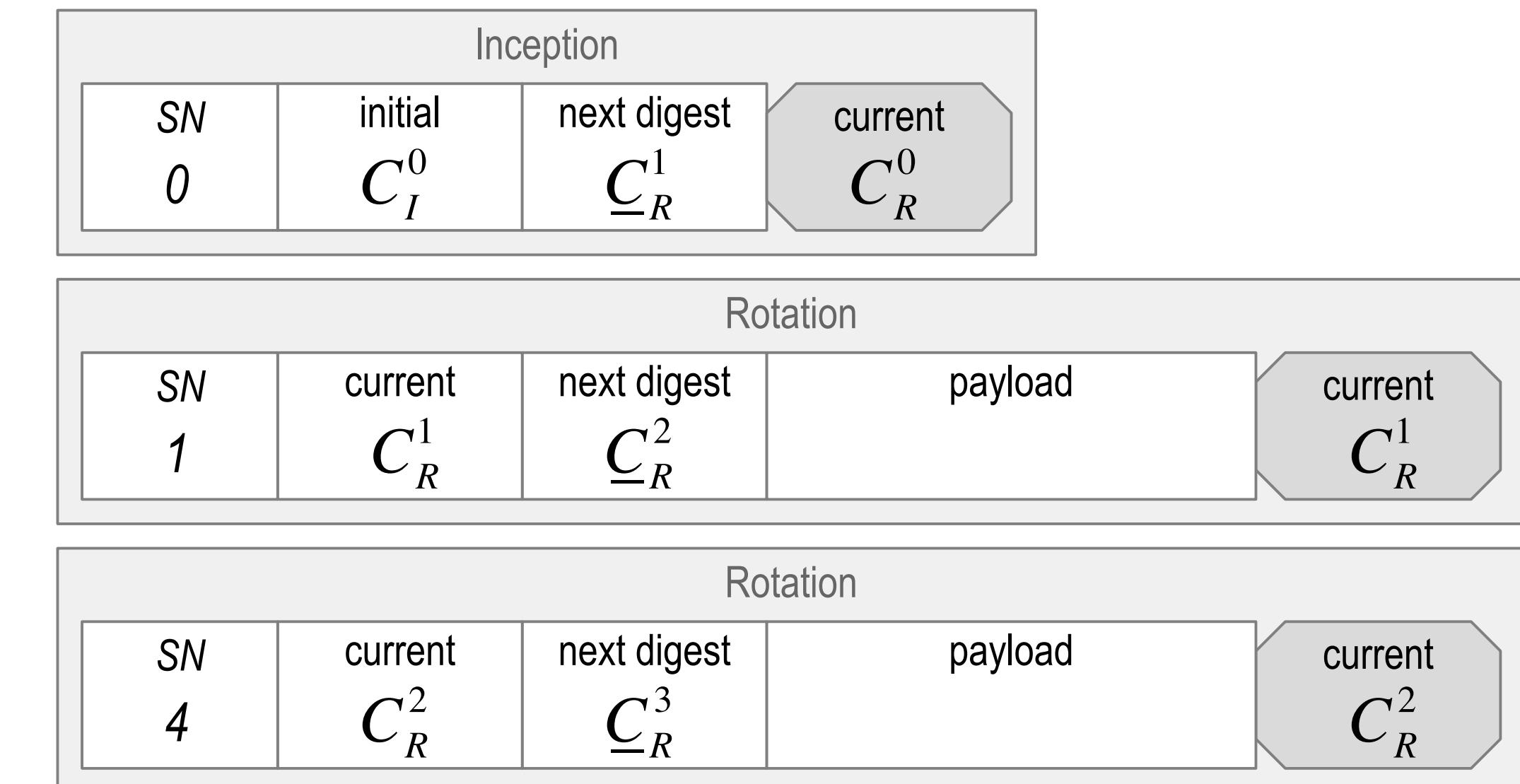
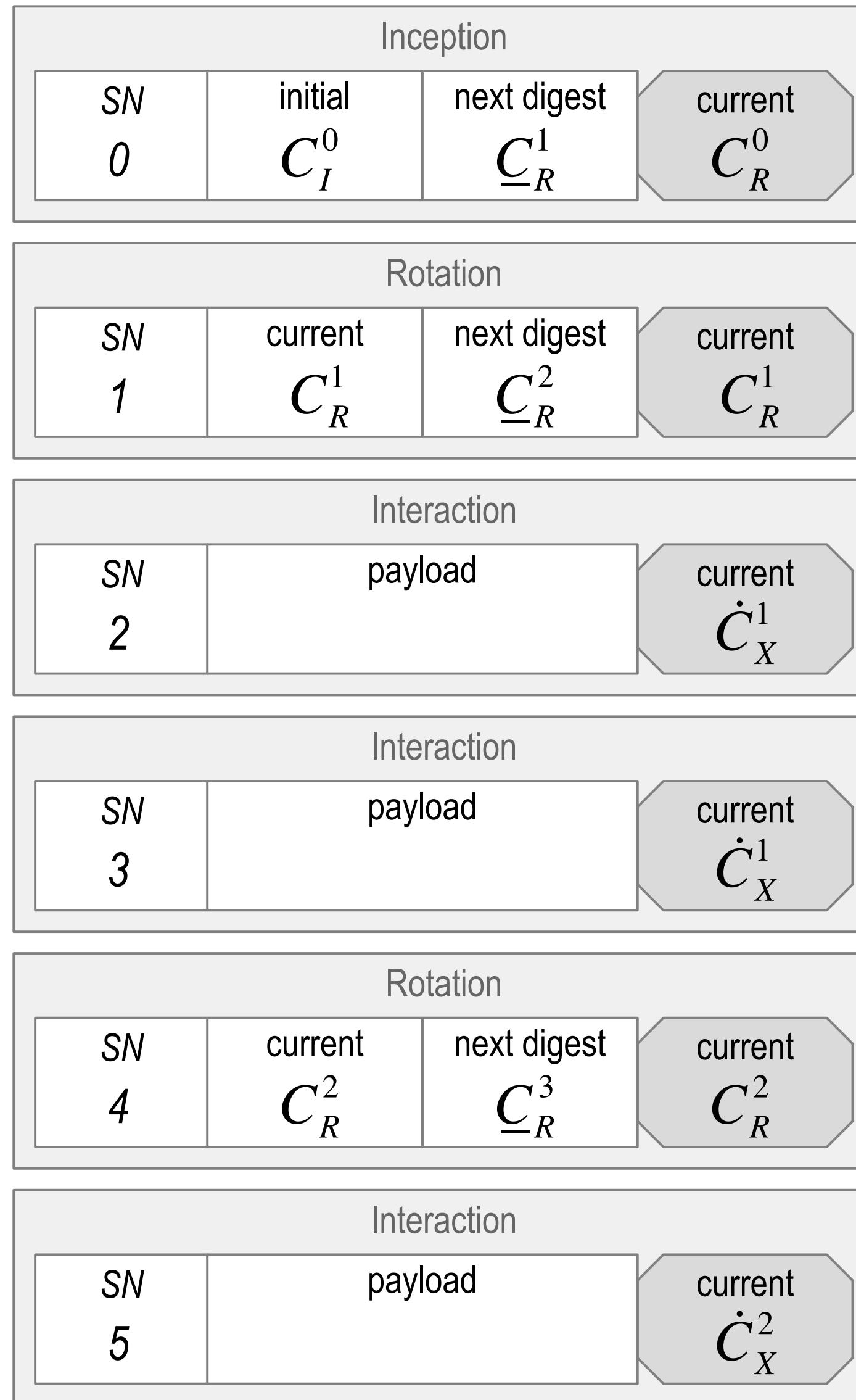
$\dot{C}^4$

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

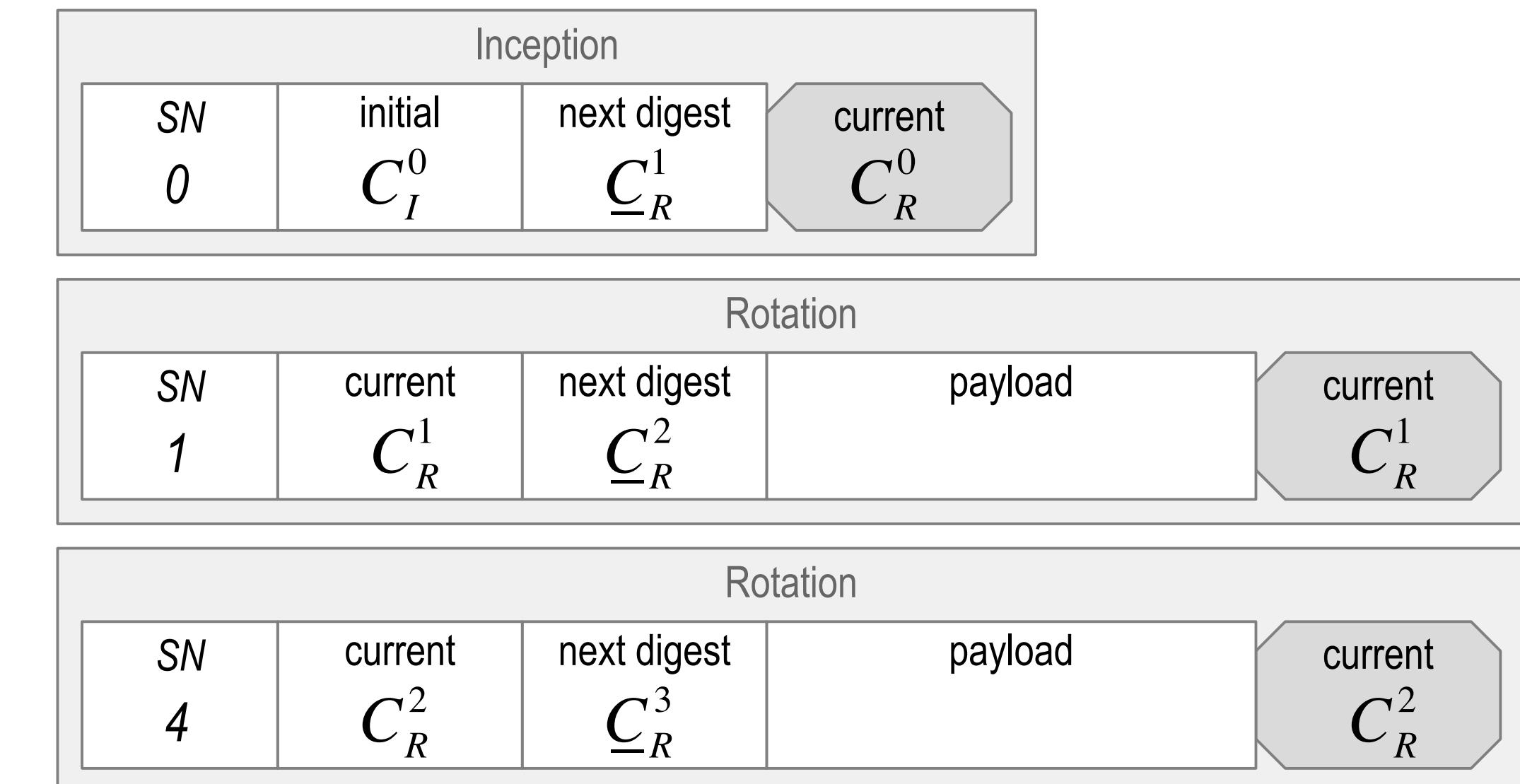
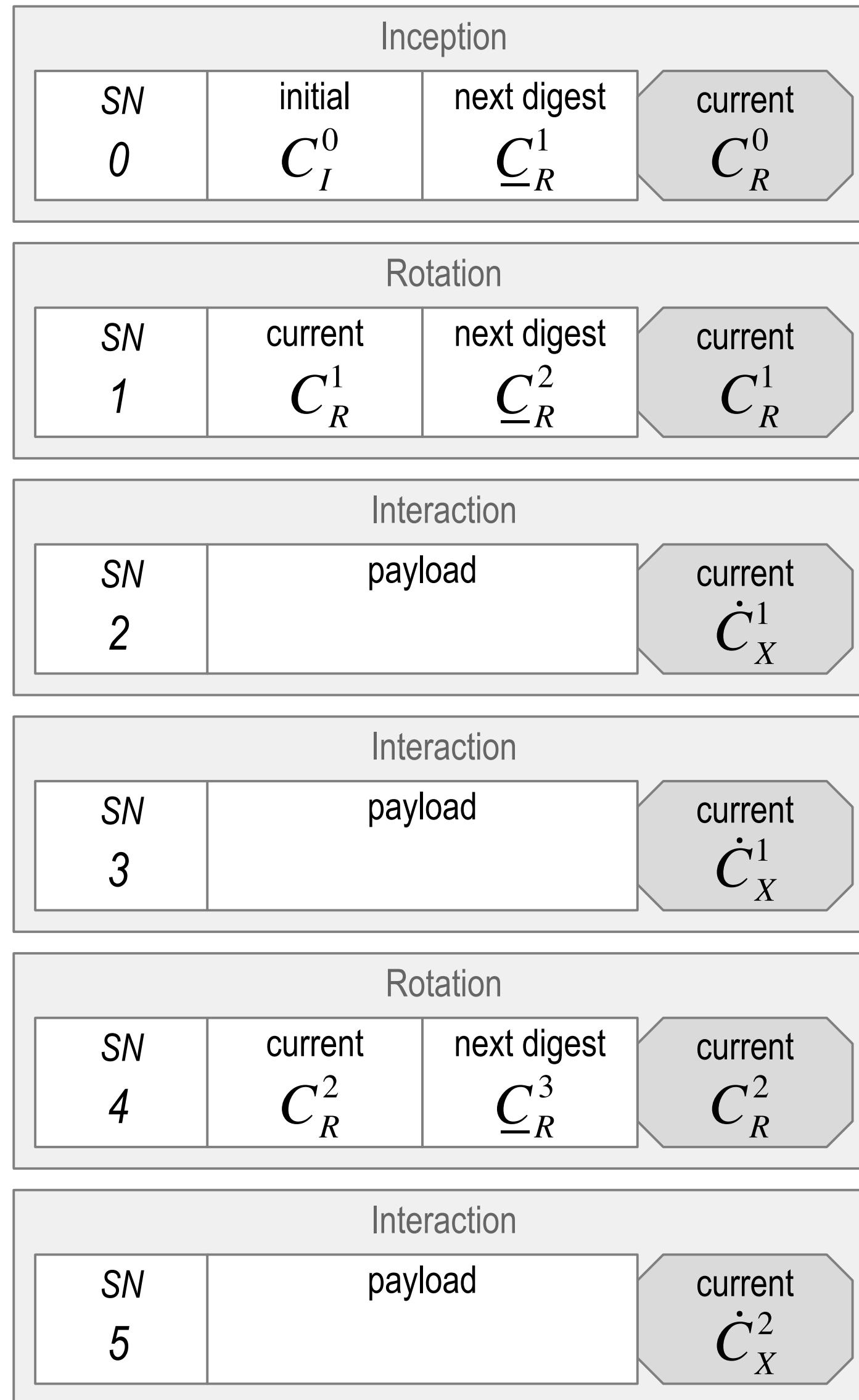
# Key Infrastructure Valence



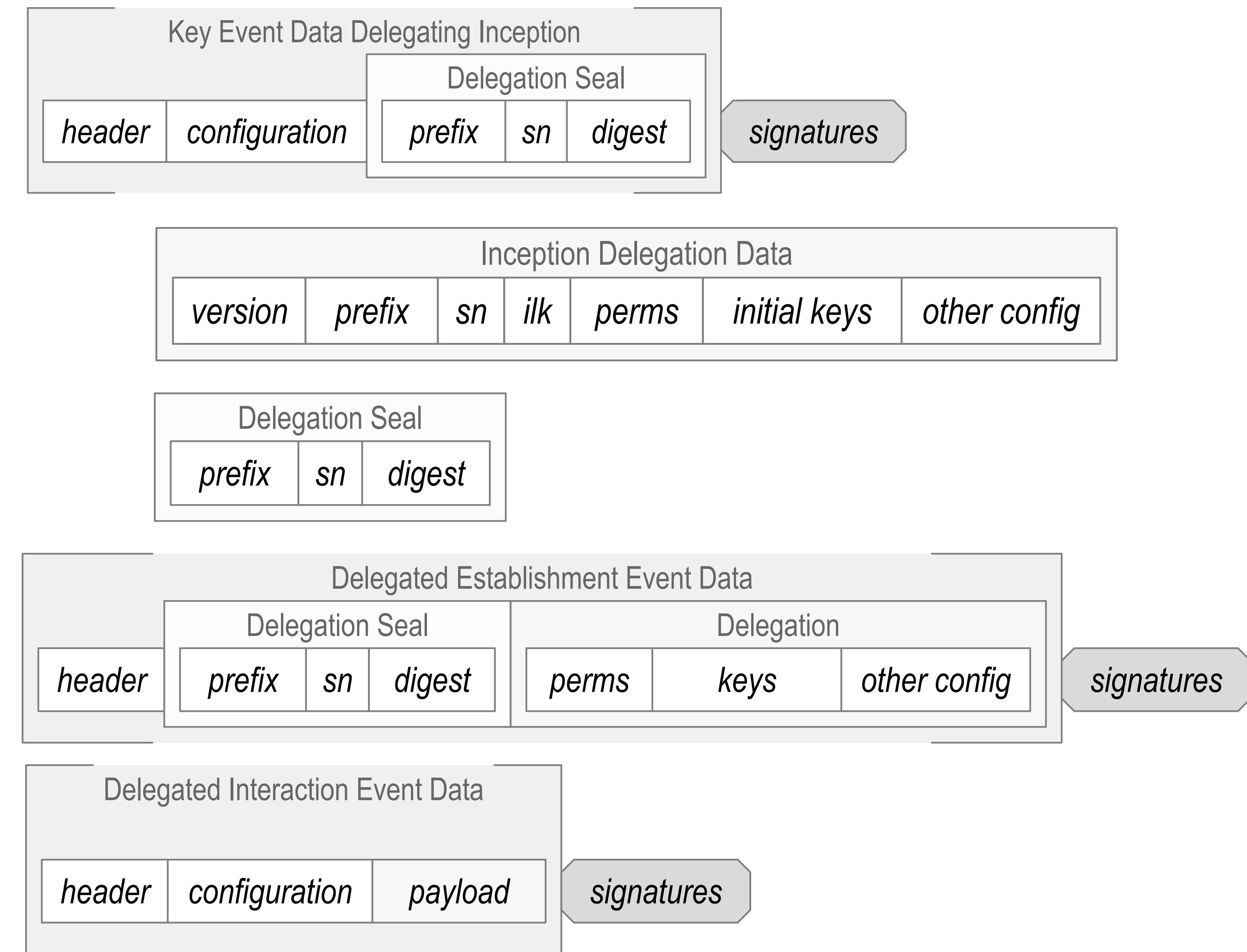
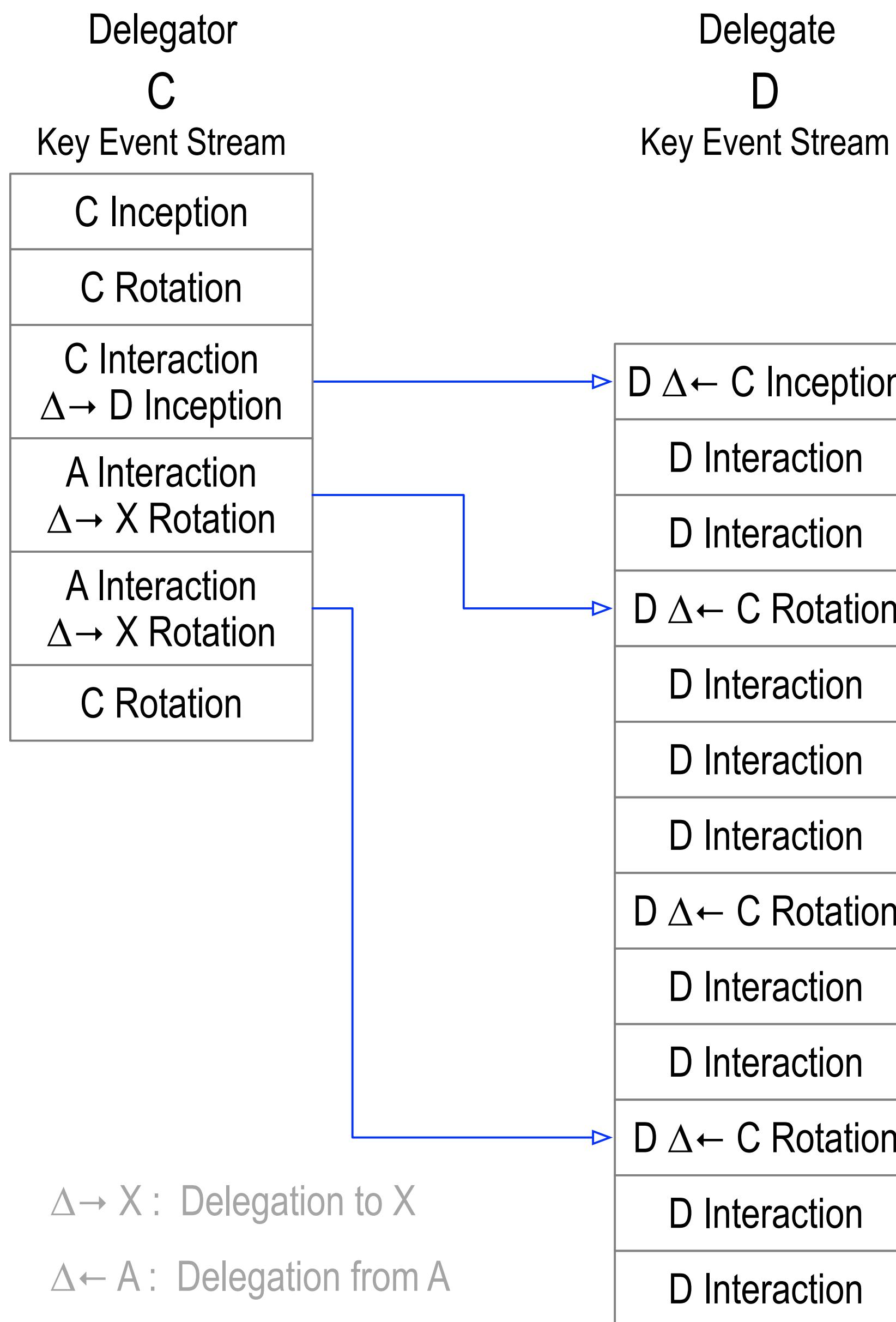
# Repurposed Keys



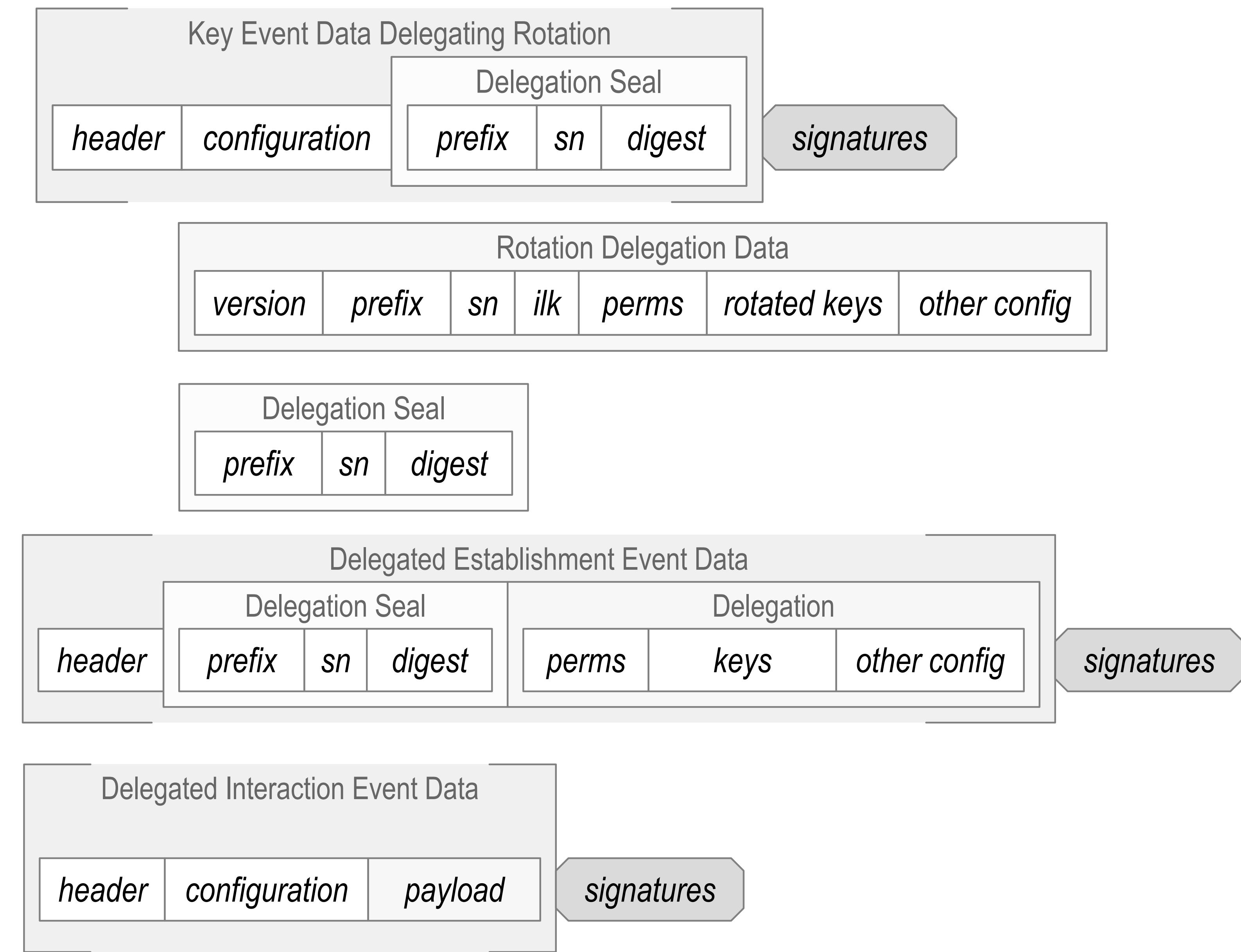
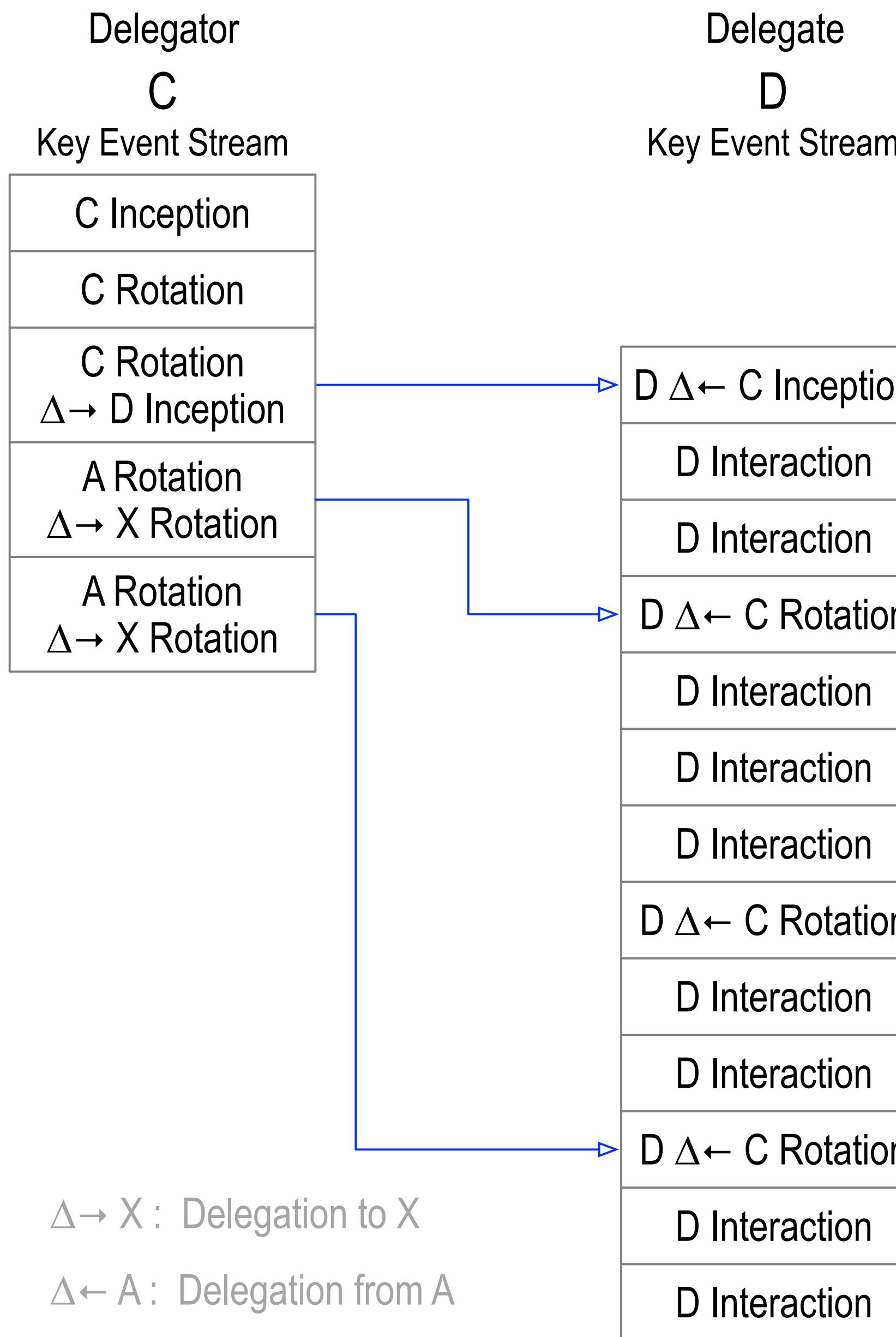
# Repurposed Keys



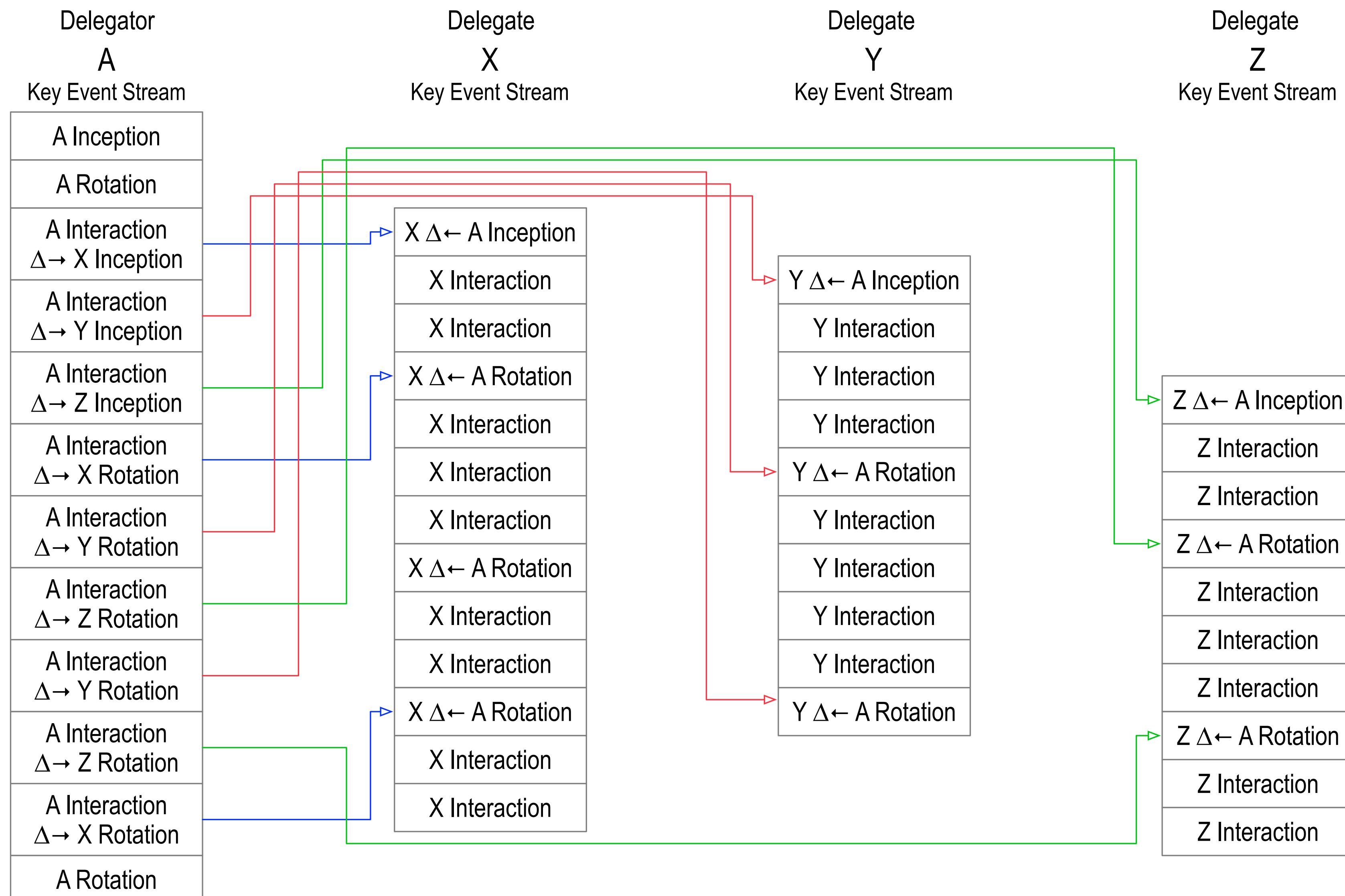
# Interaction Delegation



# Rotation Delegation



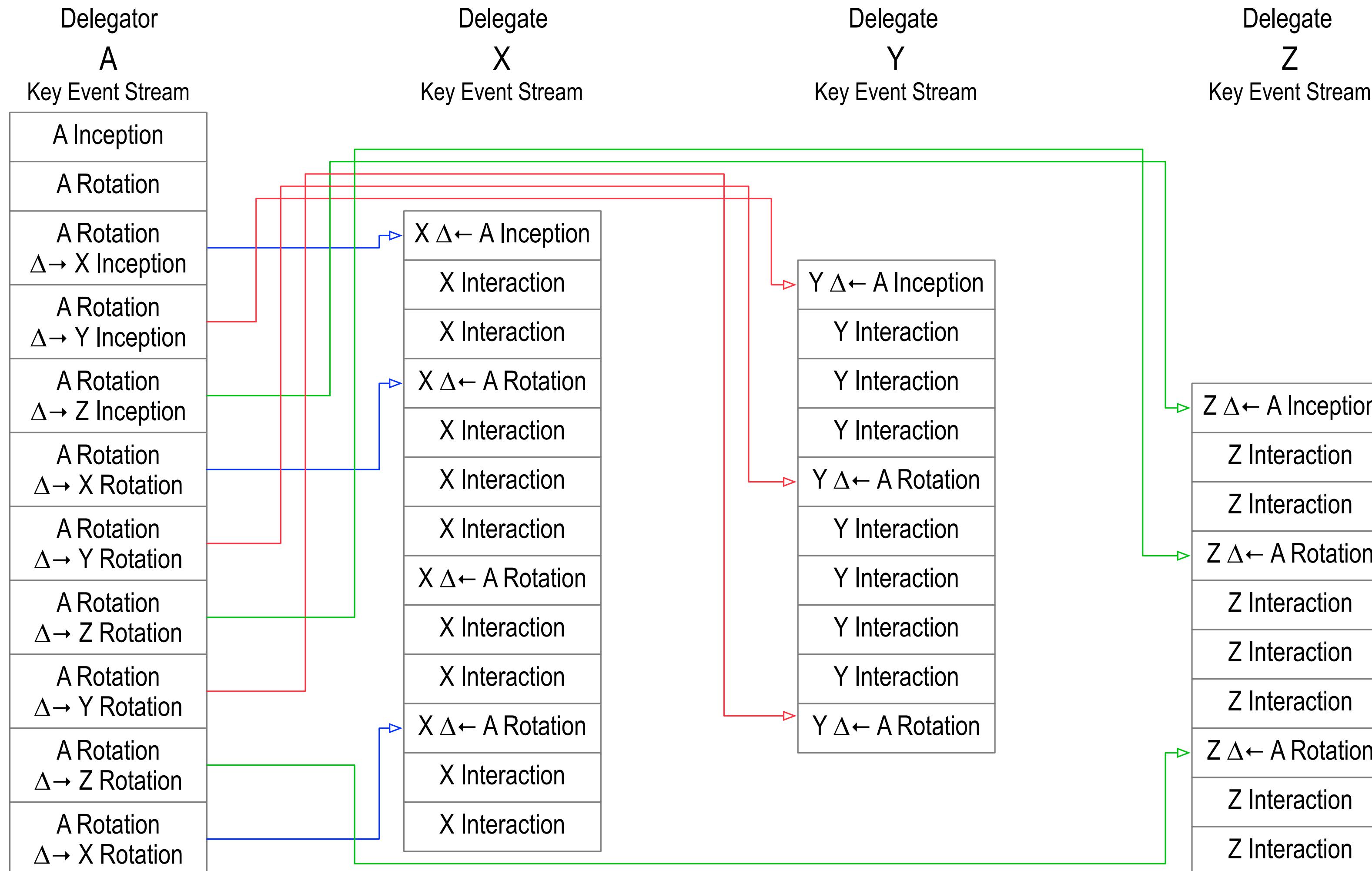
# Scaling Delegation via Interaction



$\Delta \rightarrow X$  : Delegation to X

$\Delta \leftarrow A$  : Delegation from A

# Scaling Delegation via Rotation



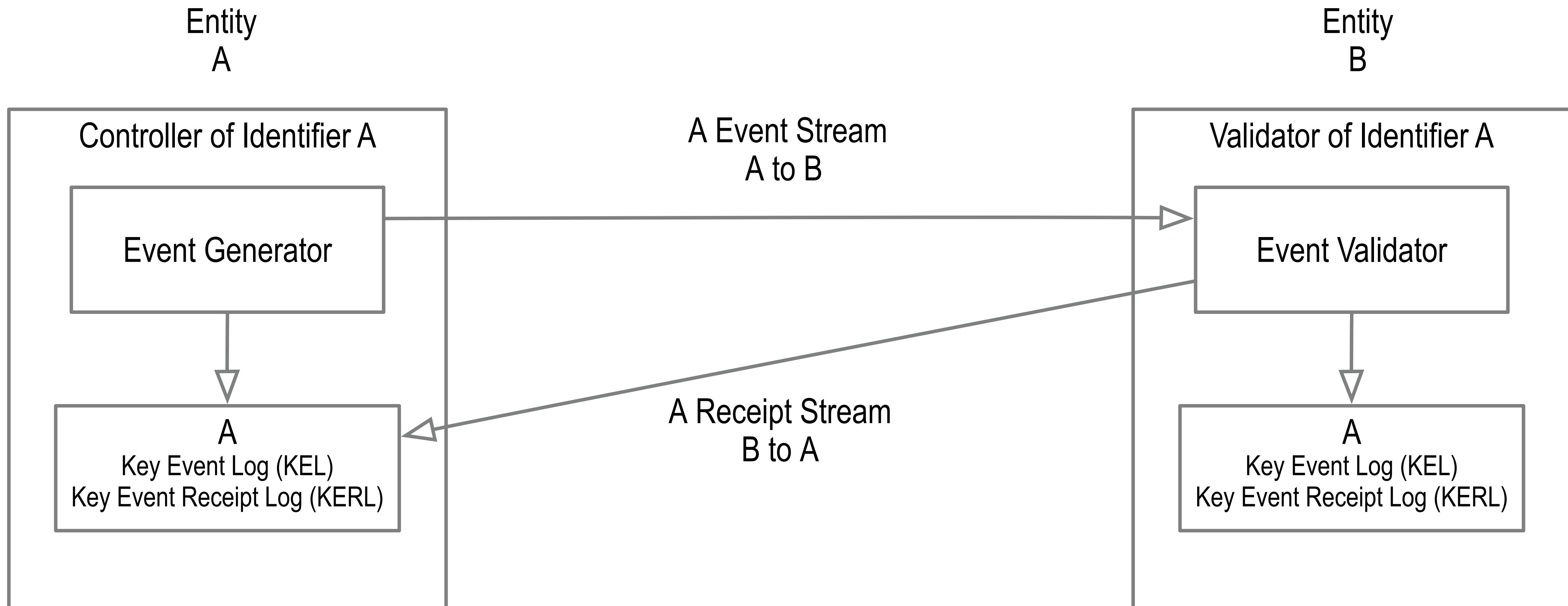
$\Delta \rightarrow X$ : Delegation to X  
 $\Delta \leftarrow A$ : Delegation from A

# Protocol Operational Modes

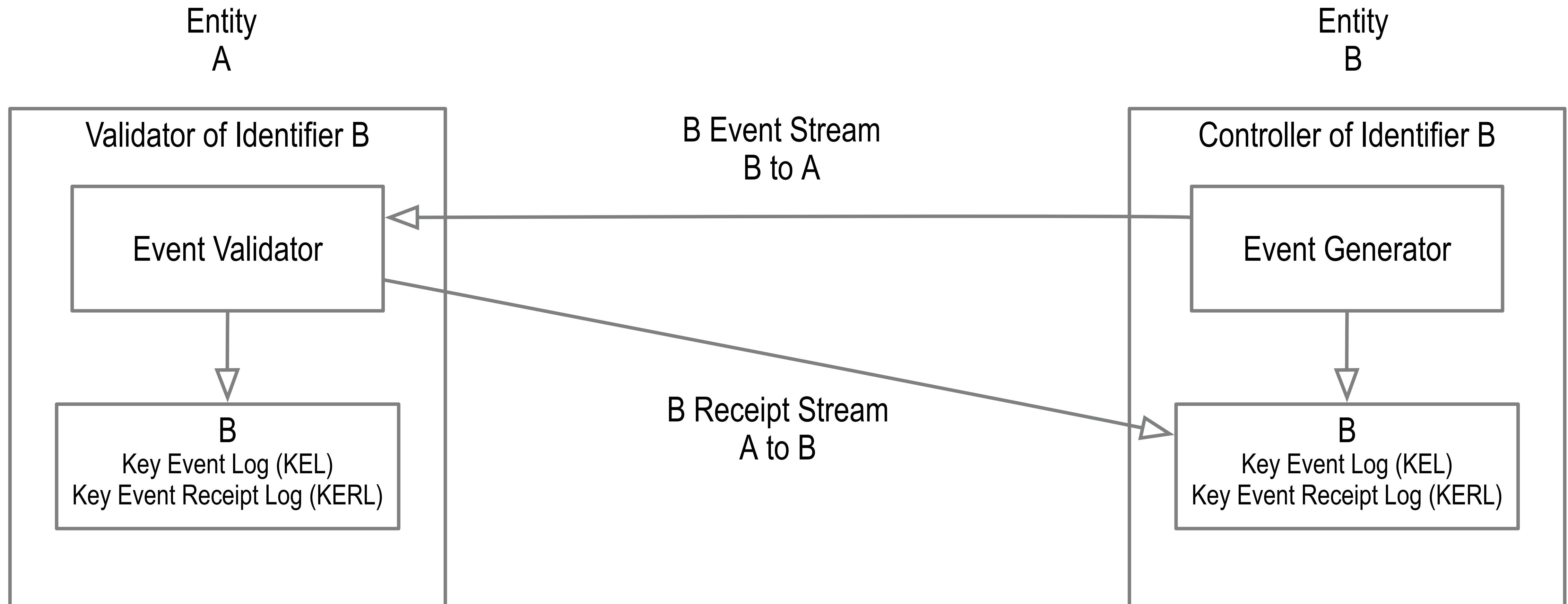
Direct Event Replay Mode (one-to-one)

Indirect Event Replay Mode (one-to-any)

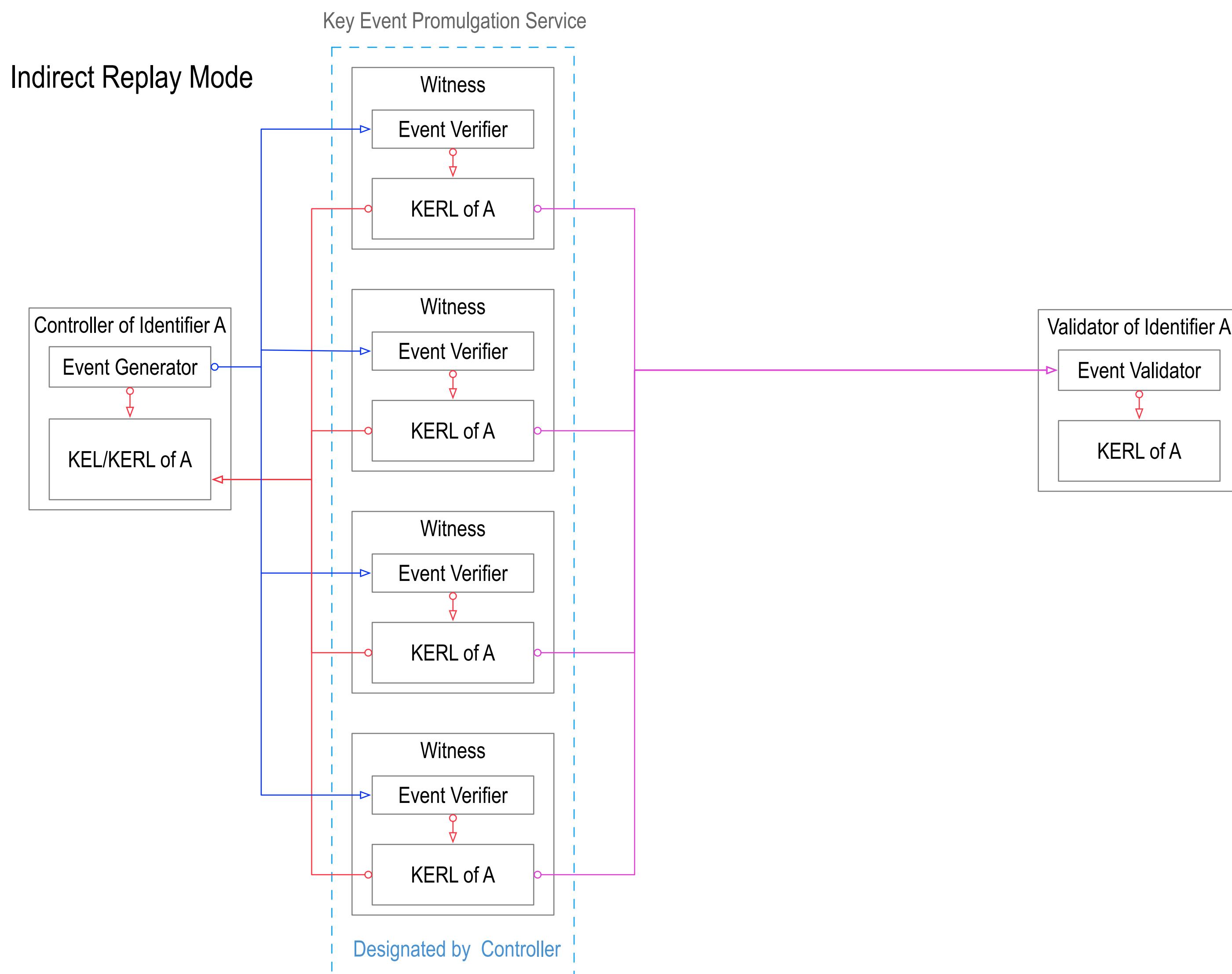
# Direct Mode: A to B



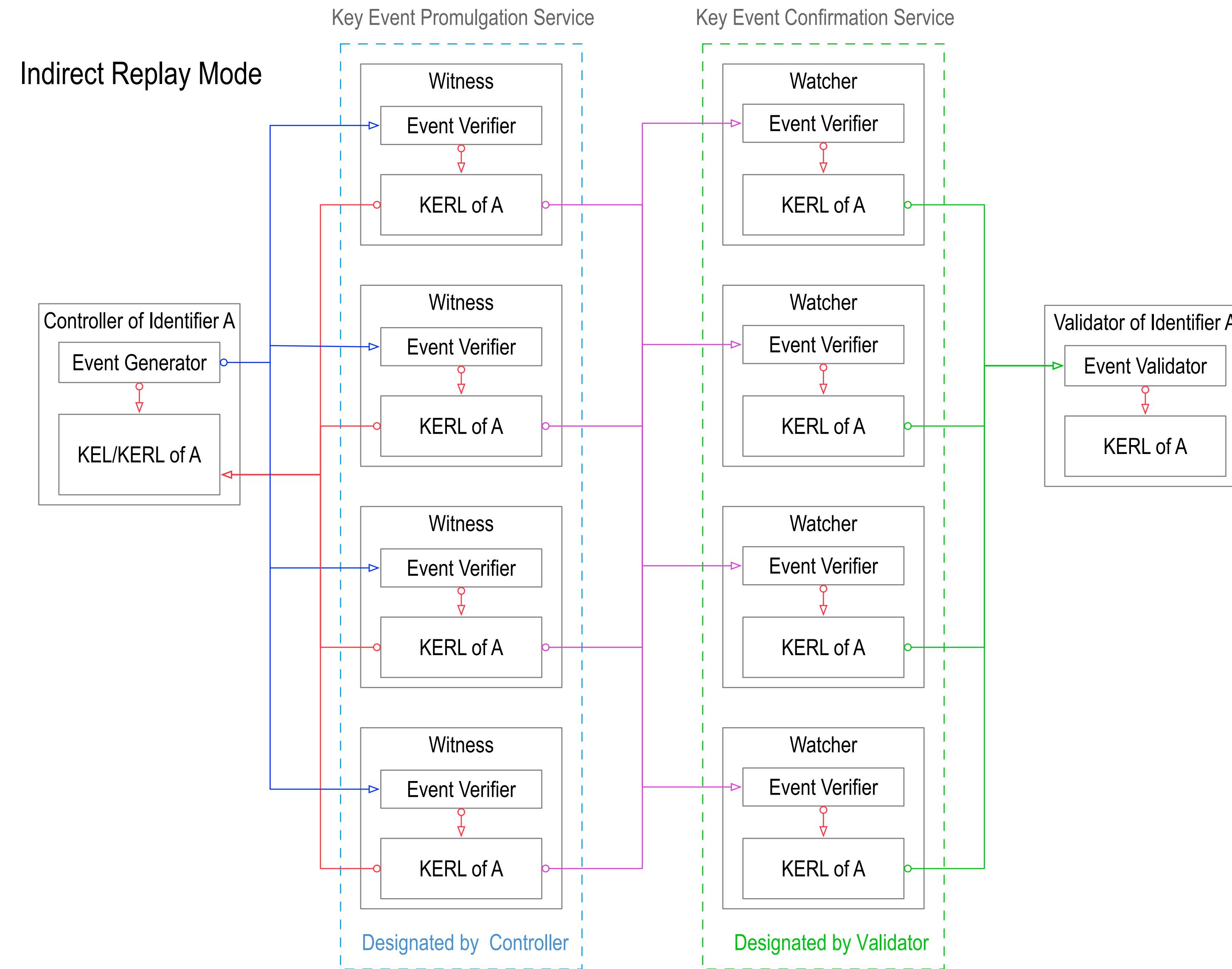
# Direct Mode: B to A



# Indirect Mode Promulgation Service

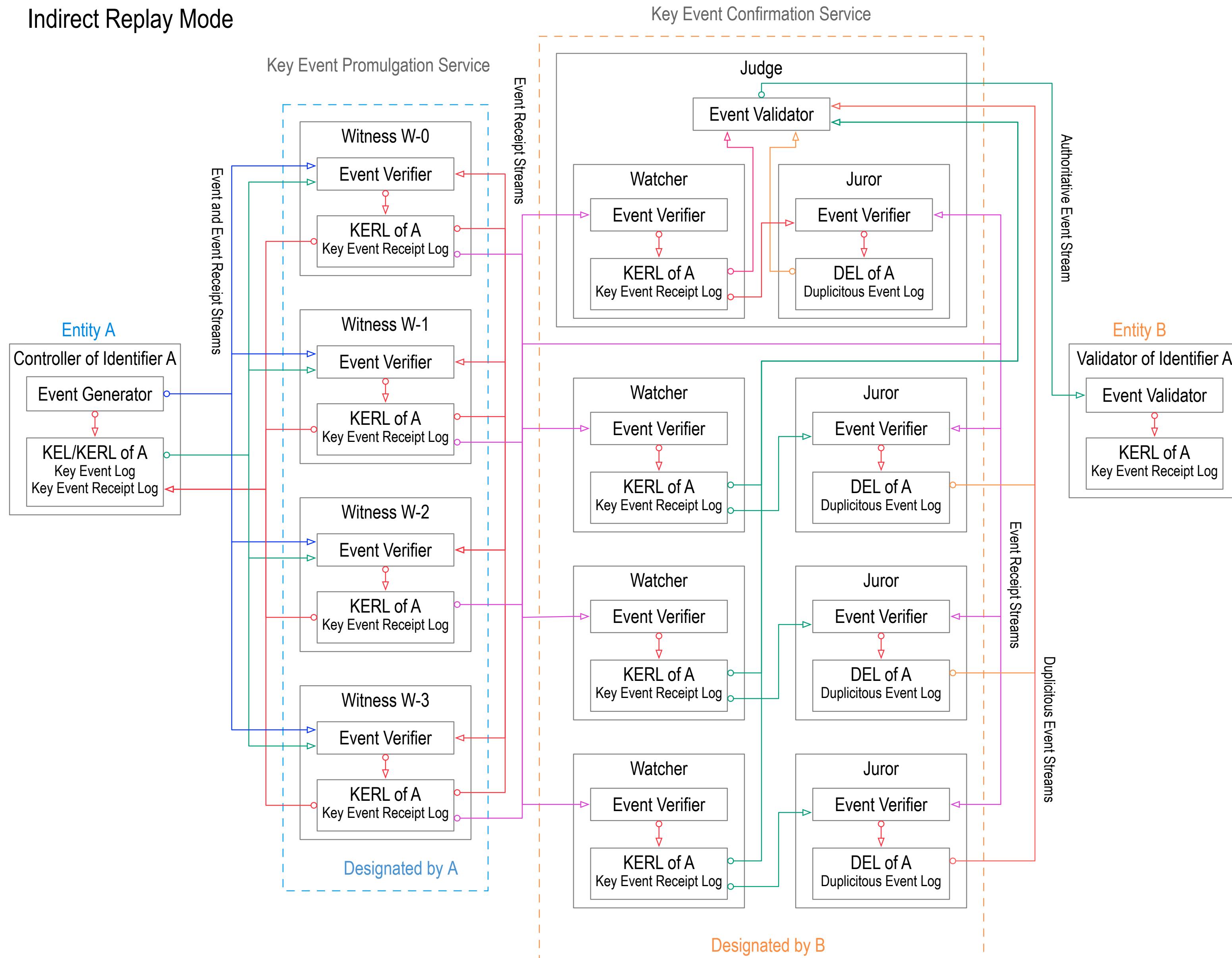


# Indirect Mode Promulgation and Confirmation Services

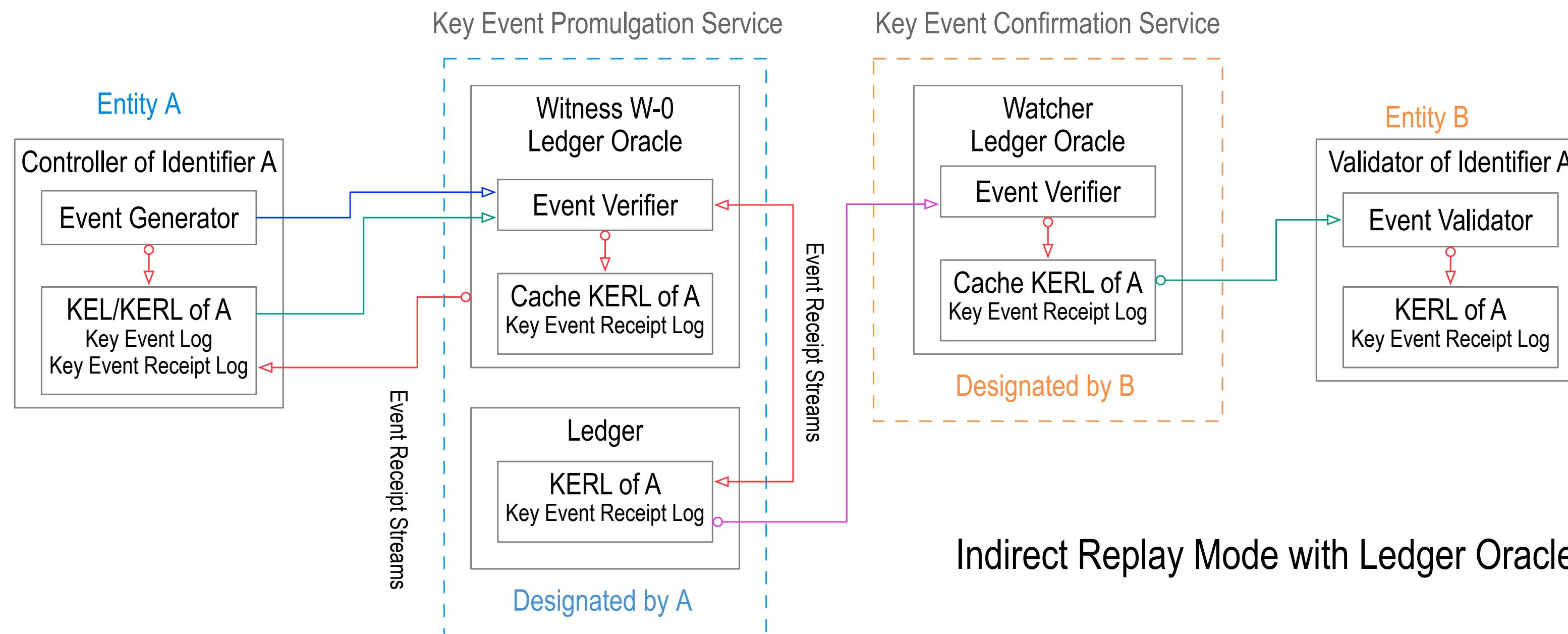


# Indirect Mode Full

Indirect Replay Mode



# Indirect Mode with Ledger Oracles



# Separation of Control

Shared (permissioned) ledger = *shared control* over *shared data*.

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

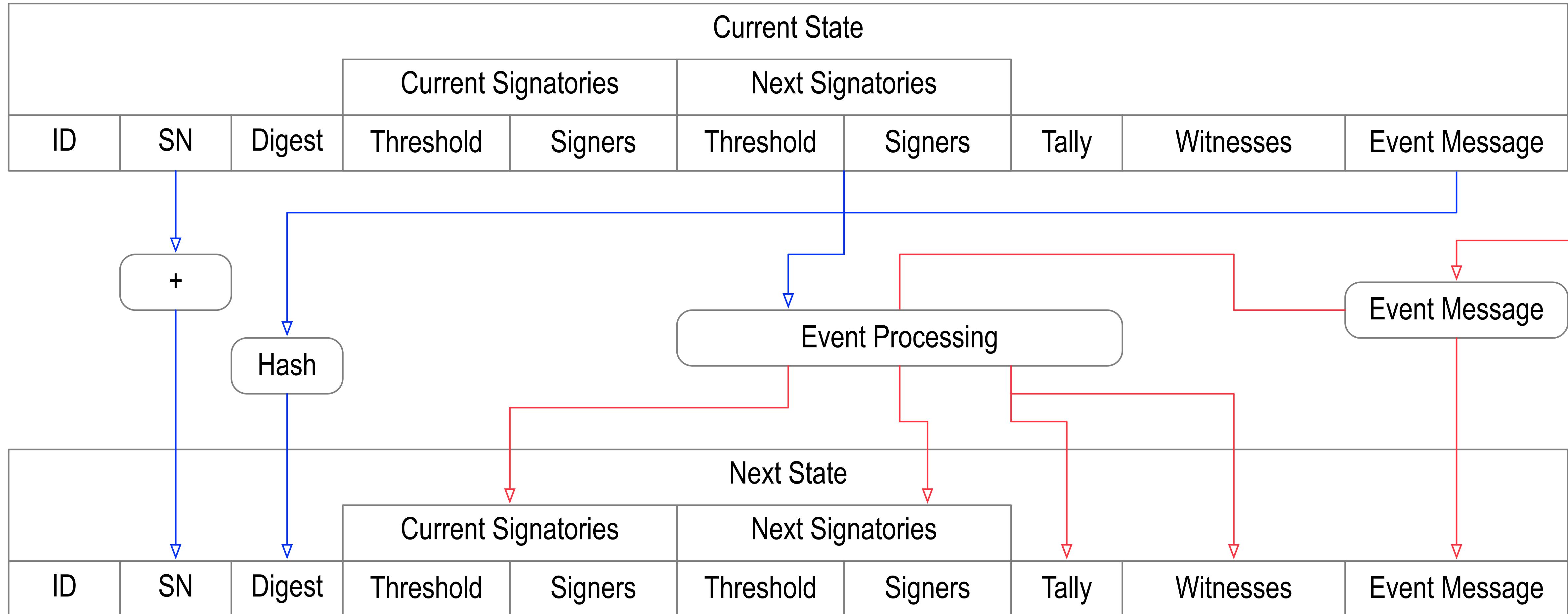
Shared control between controller and validator may be problematic for governance, scalability, and performance.

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

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

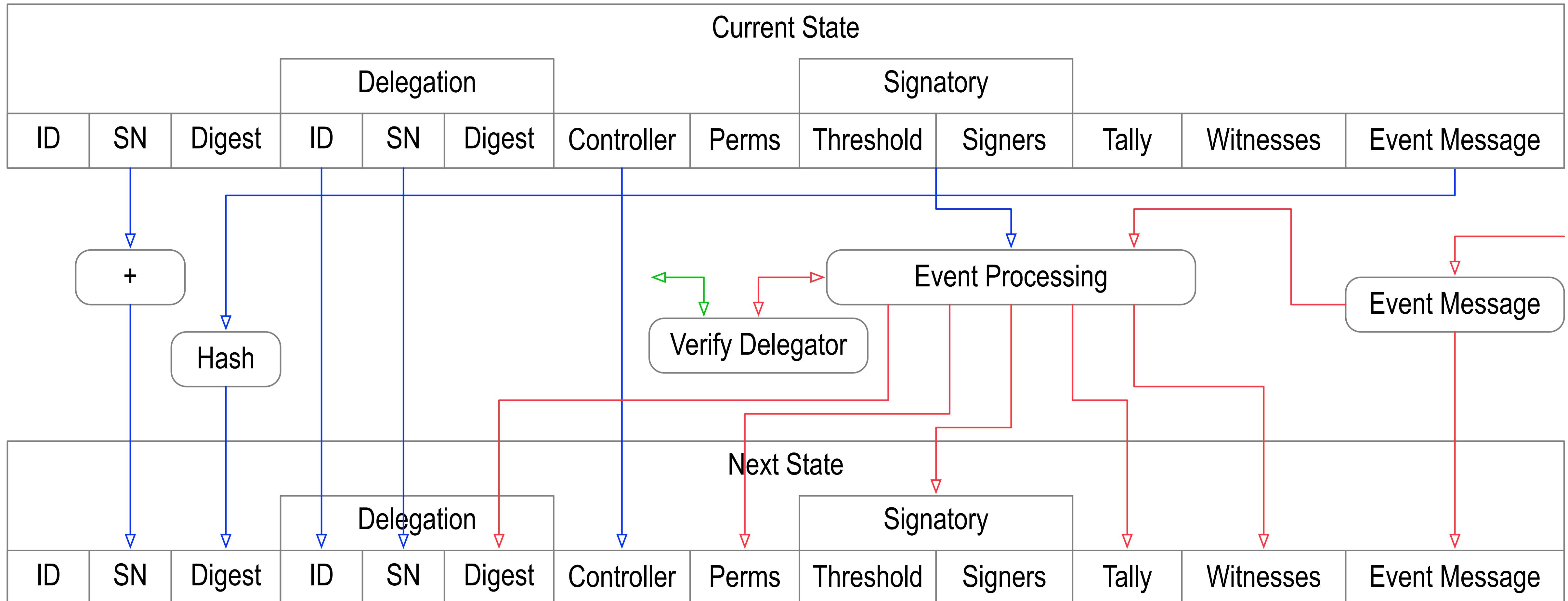
# State Verifier Engine

## KERI Core — State Verifier Engine

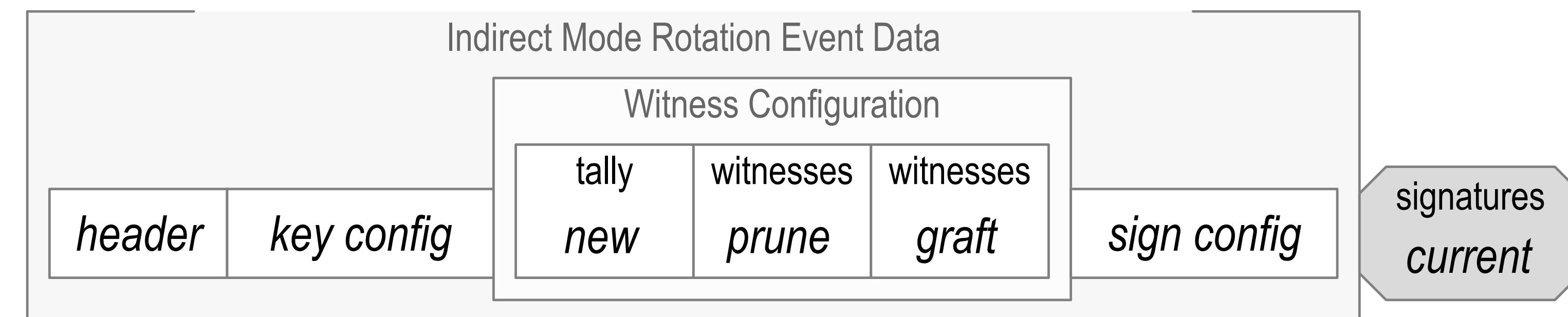
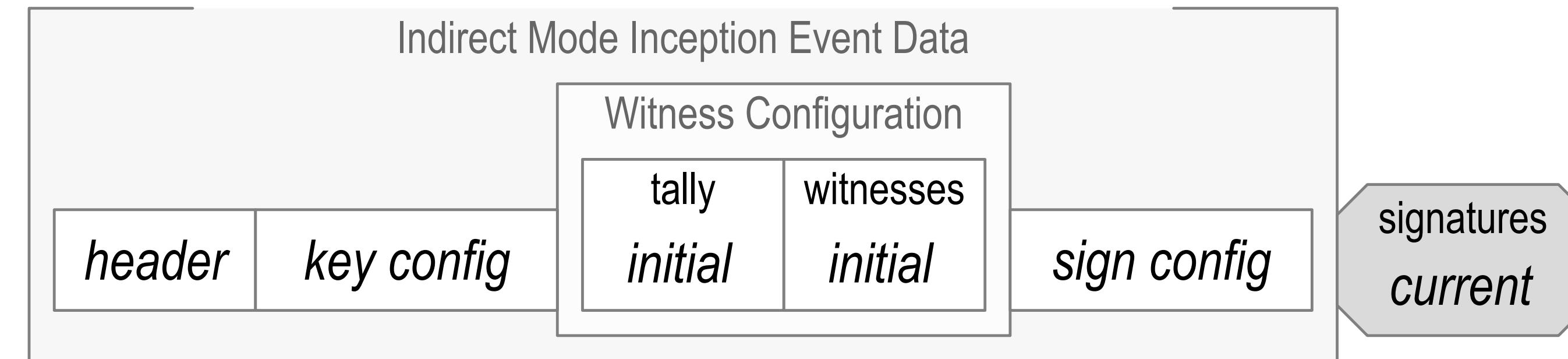


# Delegated State Verifier Engine

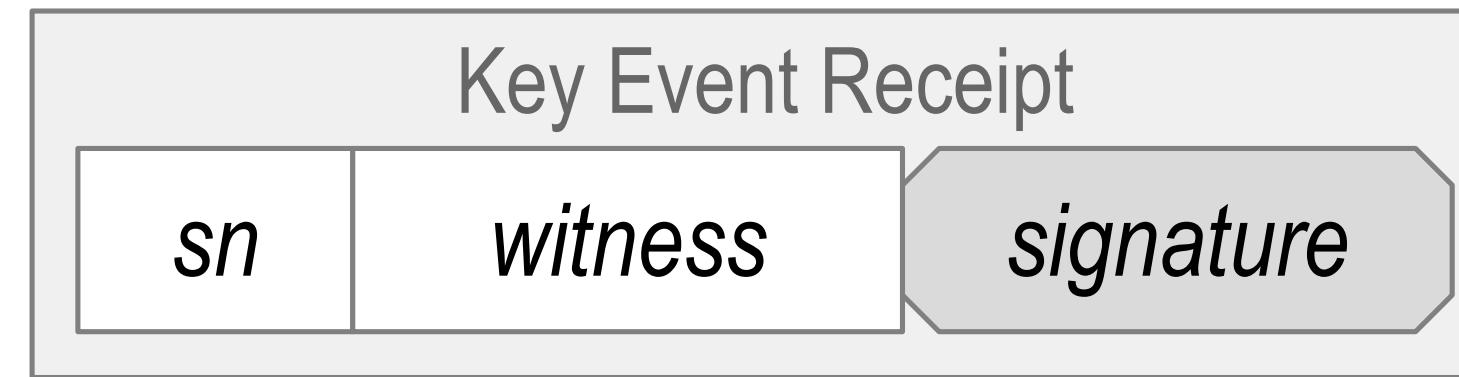
## KERI Delegated Core — State Verifier Engine



# Witness Designation



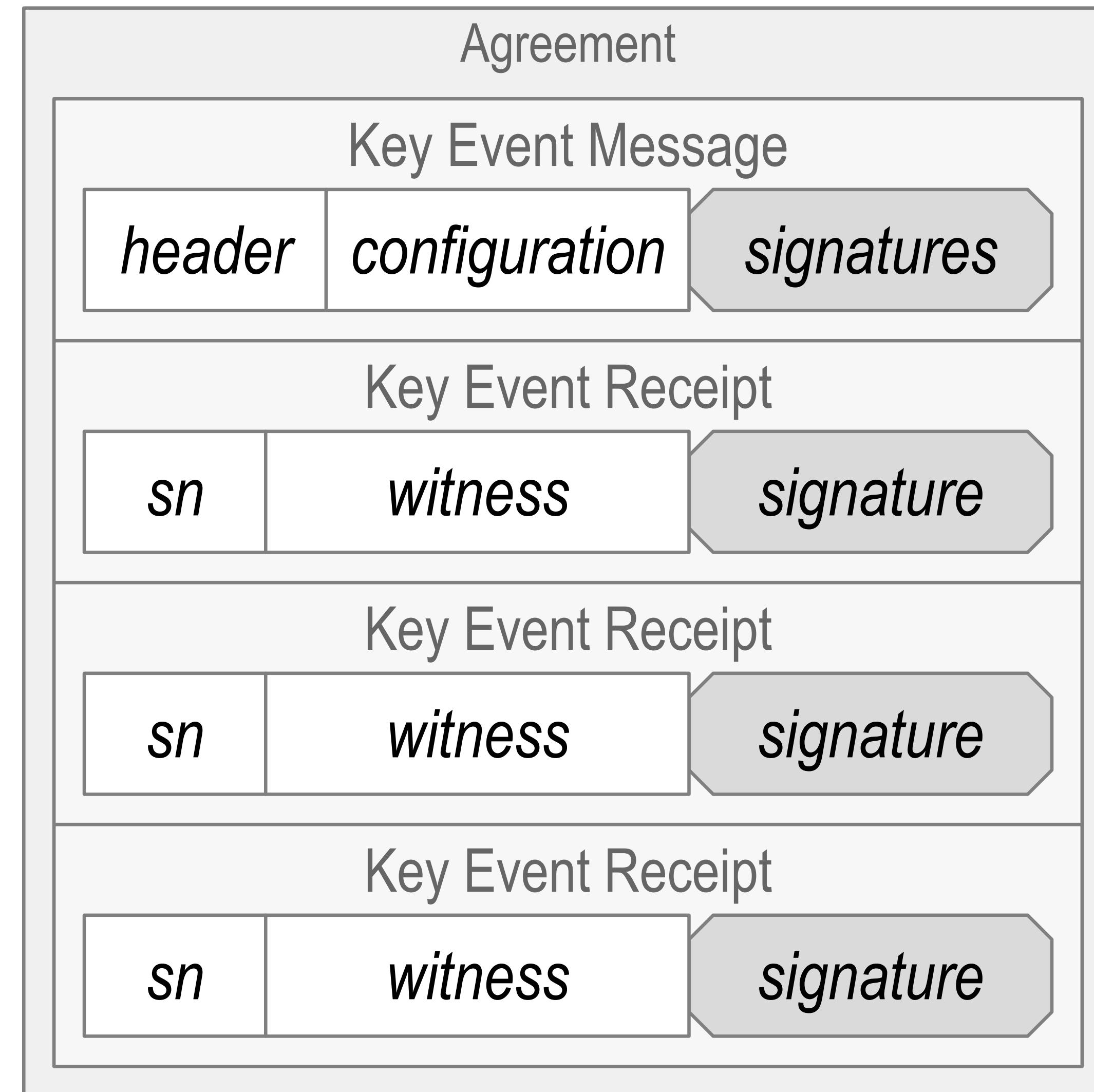
# Witnessed Key Event Receipt



(KA<sup>2</sup>CE)

# Keri's Agreement Algorithm for Control Establishment

Produce Agreements  
with Guarantees



# Agreement Constraints

Proper Agreement

$$F + 1$$

Sufficient Agreement

$$M > F$$

$$M \leq N - F$$

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

Intact Agreement

$$N \geq 2F + 1$$

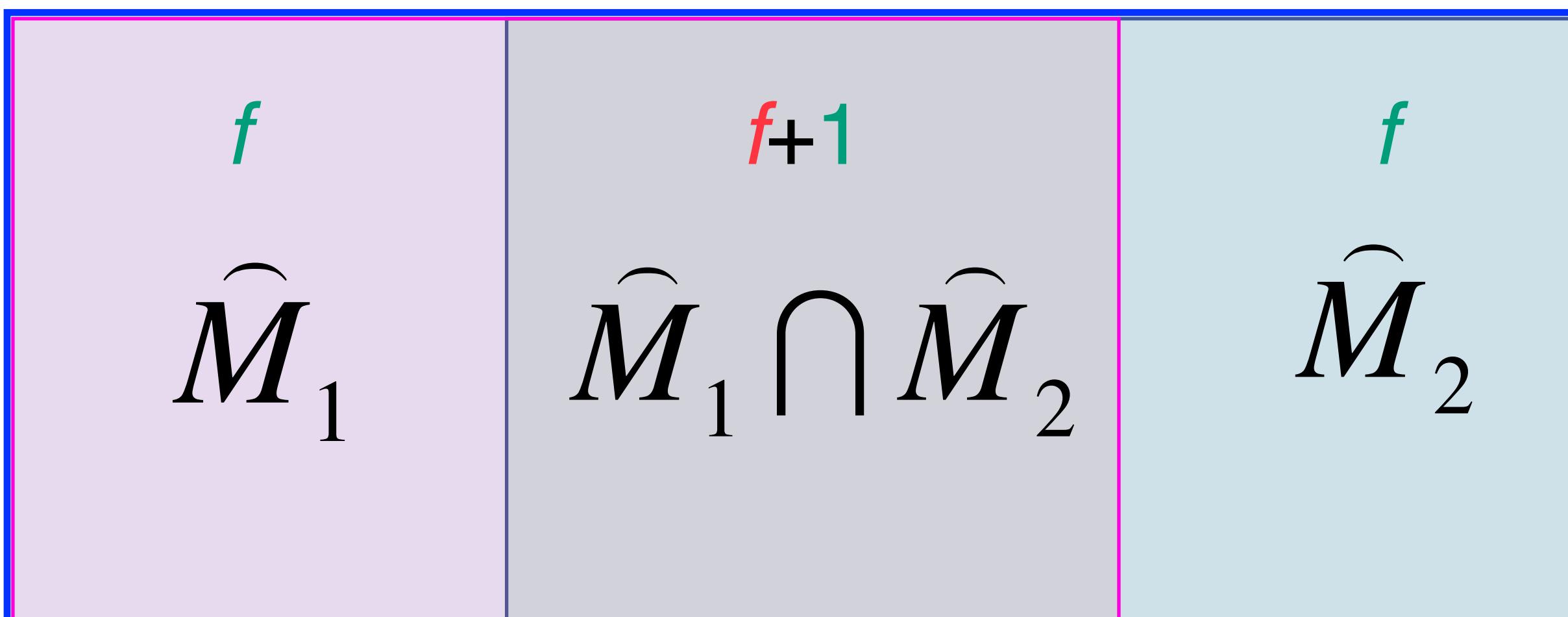
# One Agreement or None at All

$$|\hat{N}| = N \quad |\hat{M}_1| = |\hat{M}_2| = M$$

$$|\hat{M}_1 \cup \hat{M}_2| = |\hat{N}| = N$$

Overlapping Sets

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



One honest witness if:

$$|\hat{M}_1 \cap \hat{M}_2| \geq F + 1$$

$$|\hat{M}_1| + |\hat{M}_2| = |\hat{M}_1 \cup \hat{M}_2| + |\hat{M}_1 \cap \hat{M}_2|$$

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

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

$$M \leq N - F$$

Immune Agreement

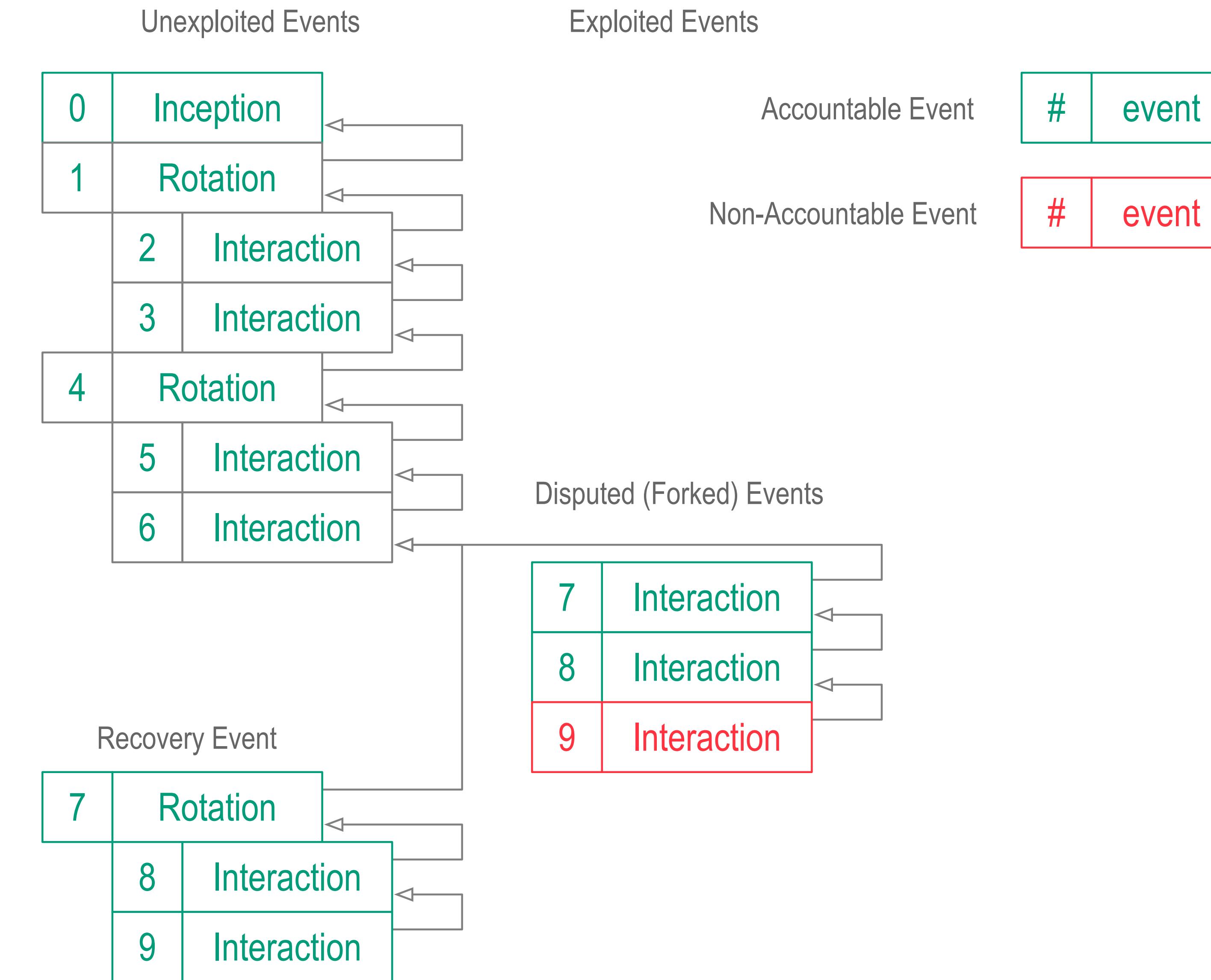
$$\frac{N + F + 1}{2} \leq M \leq N - F$$

# Example Values

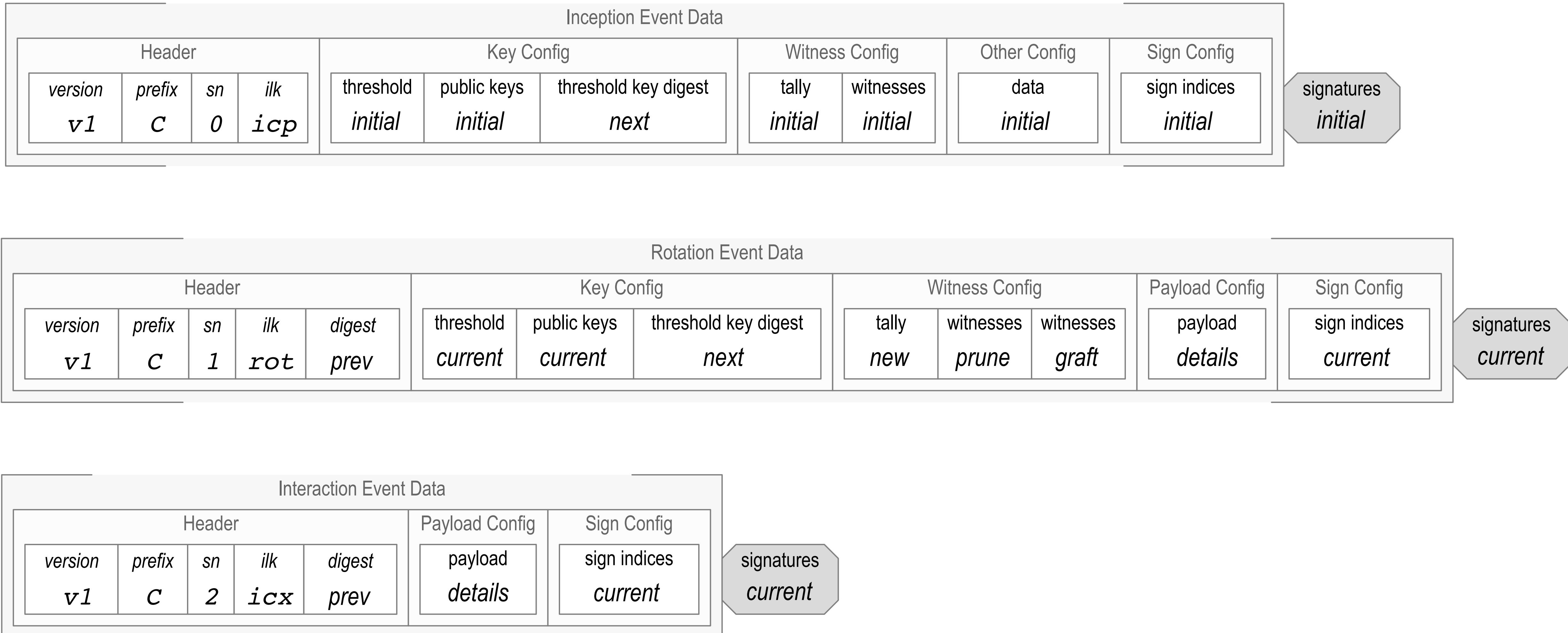
		Immunity			
F	N	3F+1	$\left\lceil \frac{N+F+1}{2} \right\rceil$	N-F	M
1	4	4	3	3	3
1	5	4	4	4	4
1	6	4	4	5	4, 5
1	7	4	5	6	5, 6
1	8	4	5	7	5, 6, 7
1	9	4	6	8	6, 7, 8
2	7	7	5	5	5
2	8	7	6	6	6
2	9	7	6	7	6, 7
2	10	7	7	8	7, 8
2	11	7	7	9	7, 8, 9
2	12	7	8	10	8, 9, 10
3	10	10	7	7	7
3	11	10	8	8	8
3	12	10	8	9	8, 9
3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

## Recovery from Live Exploit



# Generic Event Formats



# Generic Inception

$$\varepsilon_0^C = \left\langle v_0^C, C, t_0^C, \text{icp}, K_0^C, \hat{C}_0^C, \eta_0^C \left( \langle K_1^C, \hat{C}_1^C \rangle \right), M_0^C, \hat{W}_0^C, [data], \hat{s}_0^C \right\rangle \hat{\sigma}_0^C$$

$$\hat{C}_0^C = \left[ C^0, \dots, C^{L_0^C - 1} \right]_0^C$$

$$\hat{C}_1^C = \left[ C^{r_1}, \dots, C^{r_1 + L_1^C - 1} \right]_1^C$$

$$\hat{W}_0^C = \left[ W_0^C, \dots, W_{N_0^C - 1}^C \right]_0^C$$

$$\hat{s}_0^C = \left[ s_0, \dots, s_{S_0^C - 1} \right]_0^C$$

$$\hat{\sigma}_0^C = \sigma_{C^{s_0}} \dots \sigma_{C^{s_{S_0^C - 1}}}$$

# Generic Rotation

$$\varepsilon_k^C = \left\langle v_k^C, C, t_k^C, \eta_k^C(\varepsilon_{k-1}^C), \text{rot}, K_l^C, \hat{C}_l^C, \eta_l^C(\langle K_{l+1}^C, \hat{C}_{l+1}^C \rangle), M_l^C, \hat{X}_l^C, \hat{Y}_l^C, [data], \hat{s}_{kl}^C \right\rangle \hat{\sigma}_{kl}^C$$

$$\hat{C}_l^C = \left[ C^{r_l^C}, \dots, C^{r_l^C + L_l^C - 1} \right]_l^C$$

$$\hat{C}_{l+1}^C = \left[ C^{r_{l+1}^C}, \dots, C^{r_{l+1}^C + L_{l+1}^C - 1} \right]_{l+1}^C$$

$$\hat{X}_l^C = \left[ X_0^C, \dots, X_{O_l^C - 1}^C \right]_l^C$$

$$\hat{Y}_l^C = \left[ Y_0^C, \dots, Y_{P_l^C - 1}^C \right]_l^C$$

$$\hat{s}_{kl}^C = \left[ s_0, \dots, s_{S_{kl}^C - 1} \right]_{kl}^C$$

$$\hat{\sigma}_{kl}^C = \sigma_{C^{r_l^C + s_0}} \dots \sigma_{C^{r_l^C + s_{S_{kl}^C - 1}}}$$

# Generic Interaction

$$\varepsilon_k^C = \left\langle v_k^C, C, t_k^C, \eta_k^C(\varepsilon_{k-1}^C), \text{ixn}, [data], \hat{s}_{kl}^C \right\rangle \hat{\sigma}_{kl}^C$$

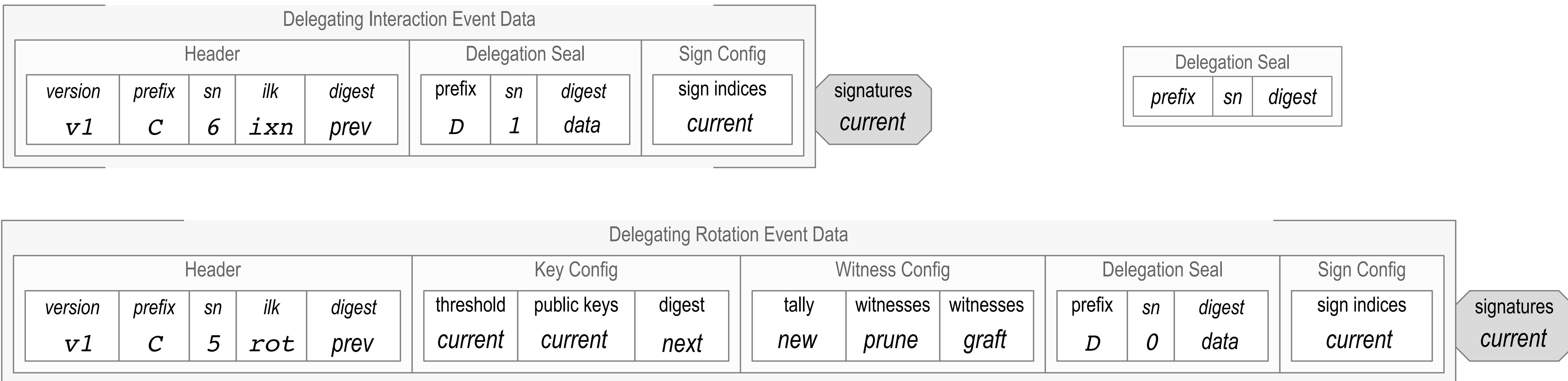
$$K_l^C$$

$$\hat{C}_l^C = \left[ C^{r_l^C}, \dots, C^{r_l^C + L_l^C - 1} \right]_l^C$$

$$\hat{s}_{kl}^C = \left[ s_0, \dots, s_{S_{kl}^C - 1} \right]_{kl}^C$$

$$\hat{\sigma}_{kl}^C = \sigma_{C^{r_l^C + s_0}} \dots \sigma_{C^{r_l^C + s_{S_{kl}^C - 1}}}$$

# Generic Delegating Event Formats



# Generic Delegated Event Formats

Inception Delegation Data									
version	prefix	sn	ilk	perms	initial threshold	initial keys	next digest	tally	initial witnesses

Delegation Seal		
prefix	sn	digest

Delegated Inception Event Data													
Header				Perms Config			Key Config			Witness Config		Delegation Seal	Sign Config
version	prefix	sn	ilk	perms	data	threshold	public keys	digest	tally	witnesses	sign indices	signatures initial	
v1	D	0	dip	initial	initial	initial	initial	next	initial	initial	initial	initial	

Rotation Delegation Data											
version	prefix	sn	ilk	digest	perms	rotated threshold	rotated keys	next digest	tally	pruned witnesses	grafted witnesses

Delegated Rotation Event Data														
Header					Perms Config			Key Config			Witness Config		Delegation Seal	Sign Config
version	prefix	sn	ilk	digest	perms	data	threshold	public keys	digest	tally	witnesses	witnesses	prefix	sign indices current
v1	D	1	drt	prev	current	current	current	next	new	prune	graft	6	C	signatures current

Delegated Interaction Event Data									
Header					Payload Config		Sign Config		
version	prefix	sn	ilk	digest	payload	details	sign indices	signatures current	prev
v1	D	2	isn	prev	details	details	current	current	

# Inception Delegation

$$\widehat{\Delta}_0^D = \left\{ D, t_0^D, \eta_k^C \left( \widehat{\delta}_0^D \right) \right\}$$

$$\widehat{\delta}_0^D = \left\langle v_0^D, D, t_0^D, \mathbf{dip}, perms, K_0^D, \widehat{D}_0^D, M_0^D, \widehat{W}_0^D \right\rangle$$

$$\widehat{D}_0^D = \left[ D^0, \dots, D^{L_0^D - 1} \right]_0^D$$

$$\widehat{W}_0^C = \left[ W_0^C, \dots, W_{N_0^C - 1}^C \right]_0^C$$

$$\mathcal{E}_0^D = \left\langle v_0^D, D, t_0^D, \mathbf{dip}, perms, K_0^D, \widehat{D}_0^D, M_0^D, \widehat{W}_0^D, \widehat{\Delta}_k^C, \widehat{s}_0^D \right\rangle \widehat{\sigma}_0^D$$

$$\widehat{\Delta}_k^C = \left\{ C, t_k^C, \eta_0^D \left( \mathcal{E}_k^C \right) \right\}$$

$$\widehat{s}_0^D = \left[ s_0, \dots, s_{S_0^D - 1} \right]_0^D$$

$$\widehat{\sigma}_0^D = \sigma_{D^{s_0}} \dots \sigma_{D^{s_{S_0^D - 1}}}$$

# Rotation Delegation

$$\widehat{\Delta}_k^D = \left\{ D, t_k^D, \eta_k^C \left( \widehat{\delta}_k^D \right) \right\}$$

$$\widehat{\delta}_k^D = \left\langle v_k^D, D, t_k^D, \eta_k^D \left( \varepsilon_{k-1}^D \right), \text{drt}, \text{perms}, K_l^D, \widehat{D}_l^D, M_l^D, \widehat{X}_l^D, \widehat{Y}_l^D \right\rangle$$

$$\widehat{D}_l^D = \left[ D^{r_l^D}, \dots, D^{r_l^D + L_l^D - 1} \right]_l^D$$

$$\widehat{X}_l^D = \left[ X_0^D, \dots, X_{O_l^D - 1}^D \right]_l^D$$

$$\widehat{Y}_l^D = \left[ Y_0^D, \dots, Y_{P_l^D - 1}^D \right]_l^D$$

$$\varepsilon_k^D = \left\langle v_k^D, D, t_k^D, \eta_k^D \left( \varepsilon_{k-1}^D \right), \text{drt}, \text{perms}, K_l^D, \widehat{D}_l^D, M_l^D, \widehat{X}_l^D, \widehat{Y}_l^D, \widehat{\Delta}_k^C, \widehat{s}_{kl}^D \right\rangle \widehat{\sigma}_{kl}^D$$

$$\widehat{\Delta}_k^C = \left\{ C, t_k^C, \eta_k^D \left( \varepsilon_k^C \right) \right\}$$

$$\widehat{s}_{kl}^D = \left[ s_0, \dots, s_{S_{kl}^D - 1} \right]_{kl}^D$$

$$\widehat{\sigma}_{kl} = \sigma_{C + r_l^D + s_0} \dots \sigma_{C + r_l^D + s_{kl}^D - 1}$$

# Delegated Interaction

$$\mathcal{E}_k^D = \left\langle v_k^D, D, t_k^D, \eta_k^D(\mathcal{E}_{k-1}^D), \text{ixn}, [data], \hat{s}_{kl}^D \right\rangle \hat{\sigma}_{kl}^D$$

# Witness Rotations

$$\hat{W}_0 = \left[ W_0 \ , W_1 \ , \cdots , W_{N-1} \right]$$

$$\hat{W}_l = \left( \hat{W}_{l-1} - \hat{X}_l \right) \cap \hat{Y}_l$$

$$\hat{X}_l \subseteq \hat{W}_{l-1} \quad \hat{Y}_l \not\subset \hat{W}_{l-1} \quad \hat{X}_l \not\subset \hat{W}_l$$

$$N_l = N_{l-1} - O_l + P_l$$

$$M_l \leq N_l$$

$$\left| \hat{X}_l \right| = O_l \quad \left| \hat{Y}_l \right| = P_l \quad \left| \hat{W}_l \right| = N_l$$

$$\hat{U}_{l-1} \subseteq \hat{W}_{l-1} \quad \left| \hat{U}_{l-1} \right| \geq M_{l-1}$$

$$\hat{U}_l \subseteq \hat{W}_l \quad \left| \hat{U}_l \right| \geq M_l$$

$$\left| \hat{U}_{l-1} \cup \hat{U}_l \right| \leq M_{l-1} + M_l$$

# Complex Weighted Signing Thresholds

$$\hat{C}_l = [C_l^1, \dots, C_l^{L_l}]_l$$

$$\hat{K}_l = [U_l^1, \dots, U_l^{L_1}]_l \quad \hat{C} = [C^1, C^2, C^3]$$

$$0 < U_l^j \leq 1$$

$$U_l^j = \frac{1}{K_l}$$

$$\hat{s}_k^l = [s_0, \dots, s_{S_k^l - 1}]_k^l$$

$$\hat{K} = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$$

$$\bar{U}_l = \sum\nolimits_{i=s_0}^{s_{S_k-1}} U_l^i \geq 1$$

$$\hat{K}_l = [\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]_l$$

$$\hat{K}_l = [[\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}], [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}], [1, 1, 1, 1]]$$

# BACKGROUND

# Derivation Code Tables

Length of crypt material determines number of pad characters. One character table for one pad char. Two character table for two pad char.

One Character KERI Base64 Prefix Derivation Code

Derivation Code	Prefix Description
-	Skip. Start code with next character.
-	
0	Two character derivation code. Use next character from two character table.
1	Four character derivation code. Use next three characters from four character table.
2	Five character derivation code. Use next four characters from five character table.
3	Six character derivation code. Use next five characters from six character table.
4	Eight character derivation code. Use next seven characters from eight character table.
5	Nine character derivation code. Use next eight characters from nine character table.
6	Ten character derivation code. Use next nine characters from ten character table.

Two Character KERI Base64 Prefix Derivation Code

Derivation Code	Prefix Description	Data Length Bytes	Pad Length h	Derivation Code Length h	Prefix Length Base64	Prefix Length h Bytes
0A	Ed25519 signature. Self-signing derivation.	64	2	2	88	66
0B	ECDSA secp256k1 signature. Self-signing derivation.	64	2	2	88	66
0C	Blake3-512 Digest. Self-addressing derivation.	64	2	2	88	66
0D	SHA3-512 Digest. Self-addressing derivation.	64	2	2	88	66
0E	Blake2b-512 Digest. Self-addressing derivation.	64	2	2	88	66
0F	SHA2-512 Digest. Self-addressing derivation.	64	2	2	88	66

One Character KERI Base64 Prefix Derivation Code-1

Derivation Code	Prefix Description	Data Length Bytes	Pad Length h	Derivation Code Length h	Prefix Length Base64	Prefix Length h Bytes
A	Ed25519 public key. Basic derivation.	32	1	1	44	33
B	Non-transferable prefix using Ed25519 public key. Basic derivation.	32	1	1	44	33
C	ECDSA secp256k1 public key. Basic derivation.	32	1	1	44	33
D	Non-transferable prefix using ECDSA secp256k1 public key. Basic derivation.	32	1	1	44	33
E	Blake3-256 Digest. Self-addressing derivation.	32	1	1	44	33
F	SHA3-256 Digest. Self-addressing derivation.	32	1	1	44	33
G	Blake2b-256 Digest. Self-addressing derivation.	32	1	1	44	33
H	Blake2s-256 Digest. Self-addressing derivation.	32	1	1	44	33
I	SHA2-256 Digest. Self-addressing derivation.	32	1	1	44	33

# Base64

Base64 to ASCII

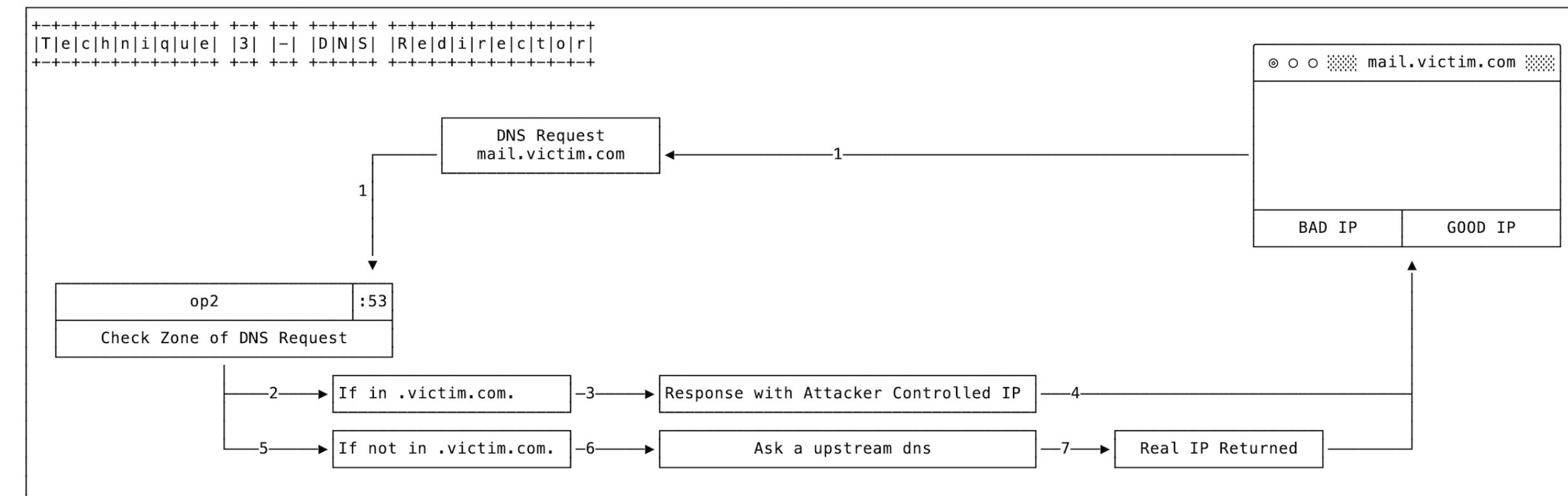
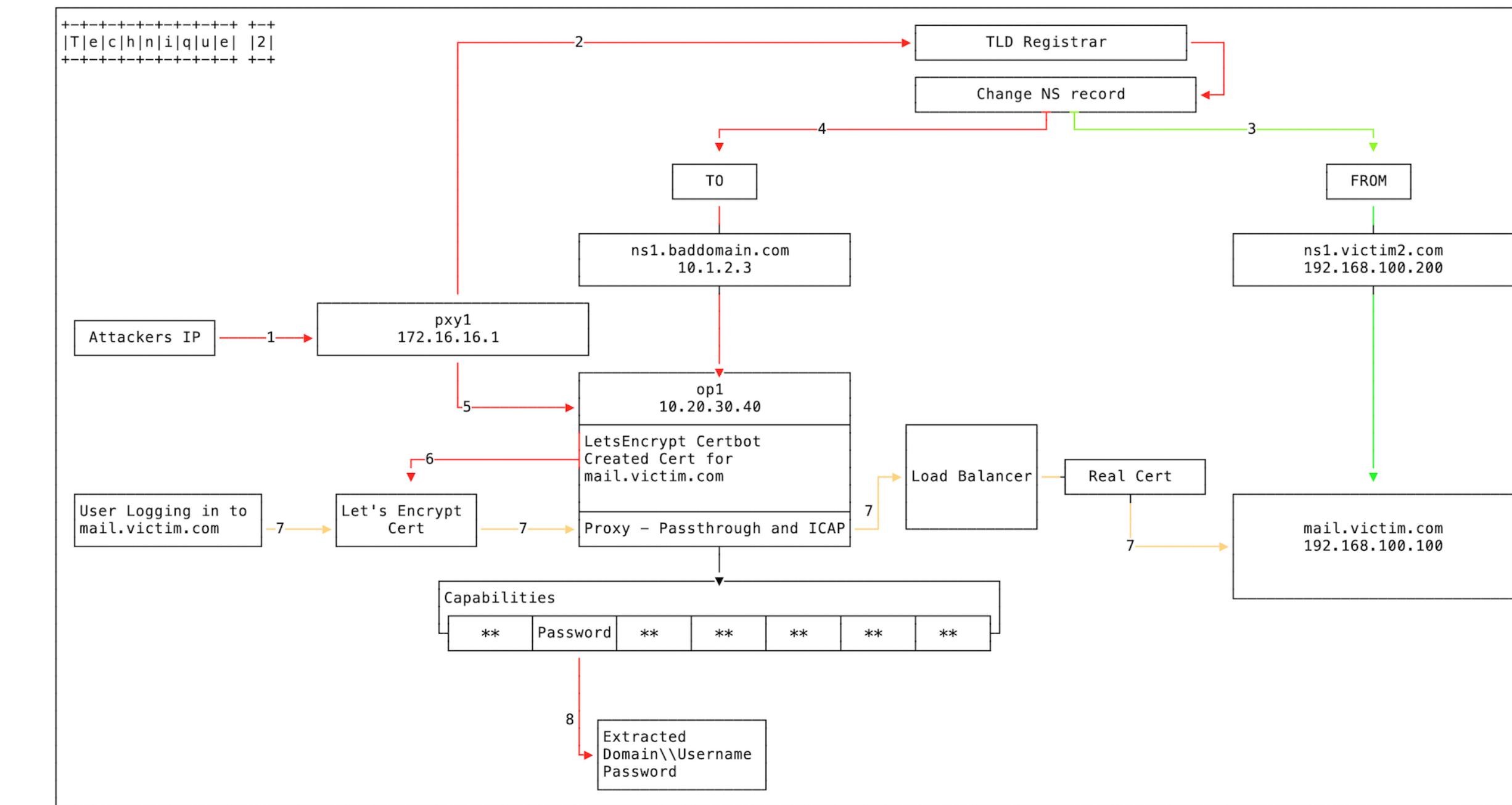
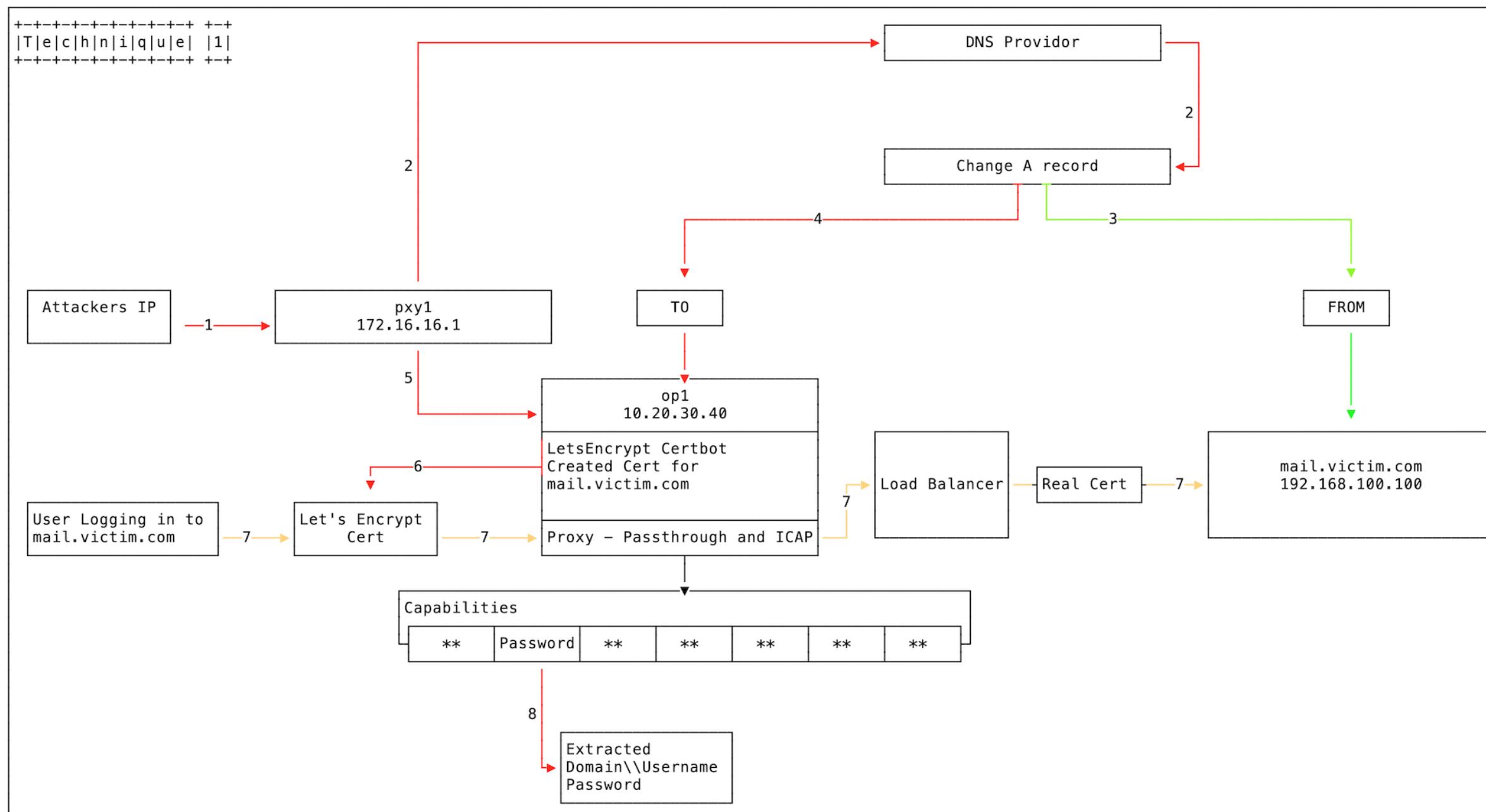
Base64 Index	ASCII Char	ASCII Decimal	ASCII Hex	ASCII Binary
0	A	65	41	01000001
1	B	66	42	01000010
2	C	67	43	01000011
3	D	68	44	01000100
4	E	69	45	01000101
5	F	70	46	01000110
6	G	71	47	01000111
7	H	72	48	01001000
8	I	73	49	01001001
9	J	74	4A	01001010
10	K	75	4B	01001011
11	L	76	4C	01001100
12	M	77	4D	01001101
13	N	78	4E	01001110
14	O	79	4F	01001111
15	P	80	50	01010000
16	Q	81	51	01010001
17	R	82	52	01010010
18	S	83	53	01010011
19	T	84	54	01010100
20	U	85	55	01010101
21	V	86	56	01010110
22	W	87	57	01010111
23	X	88	58	01011000
24	Y	89	59	01011001
25	Z	90	5A	01011010
26	a	97	61	01100001
27	b	98	62	01100010
28	c	99	63	01100011
29	d	100	64	01100100
30	e	101	65	01100101
31	f	102	66	01100110
32	g	103	67	01100111
33	h	104	68	01101000
34	i	105	69	01101001
35	j	106	6A	01101010
36	k	107	6B	01101011
37	l	108	6C	01101100
38	m	109	6D	01101101
39	n	110	6E	01101110
40	o	111	6F	01101111
41	p	112	70	01110000
42	q	113	71	01110001
43	r	114	72	01110010
44	s	115	73	01110011
45	t	116	74	01110100
46	u	117	75	01110101
47	v	118	76	01110110
48	w	119	77	01110111

Base64 to ASCII

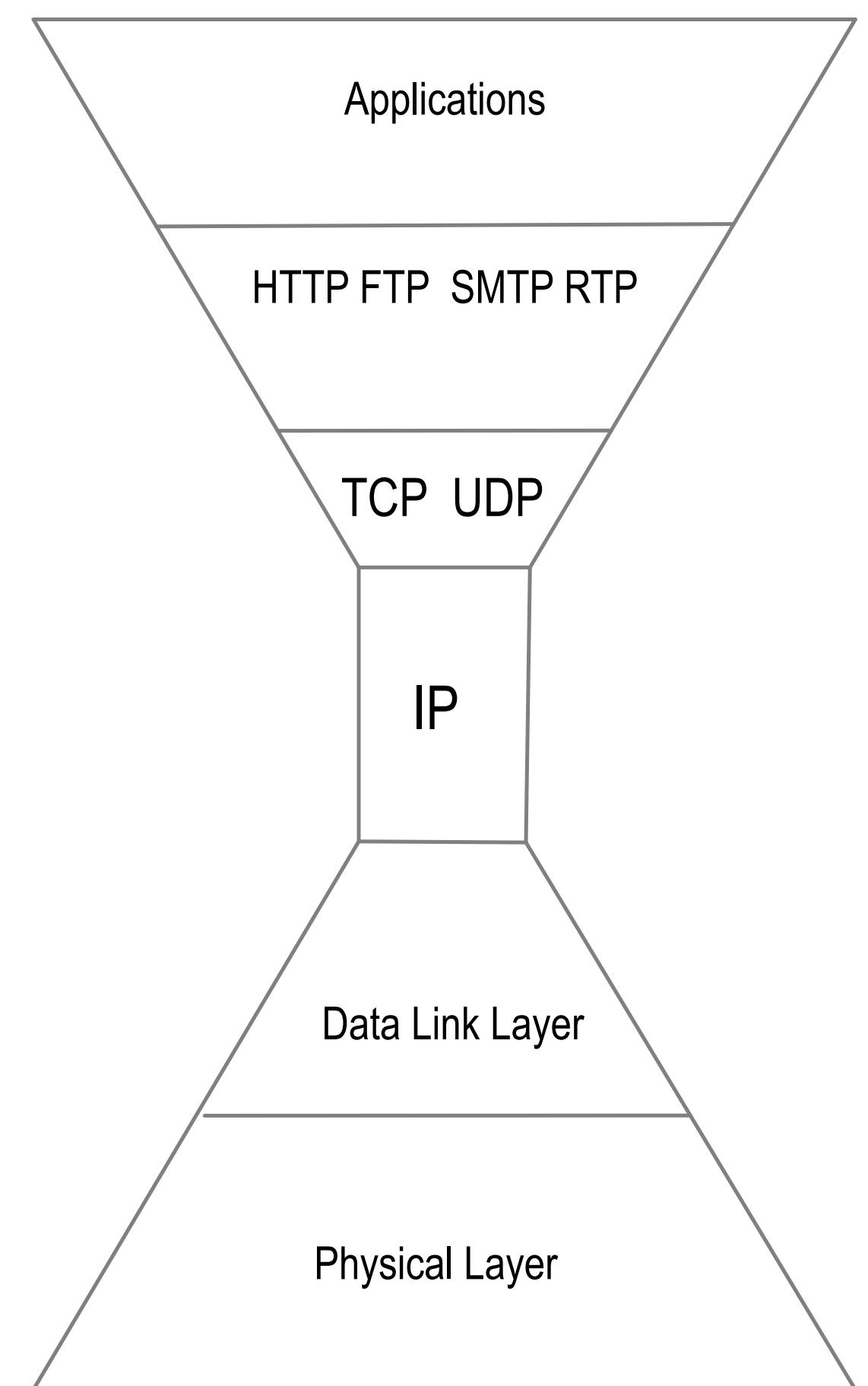
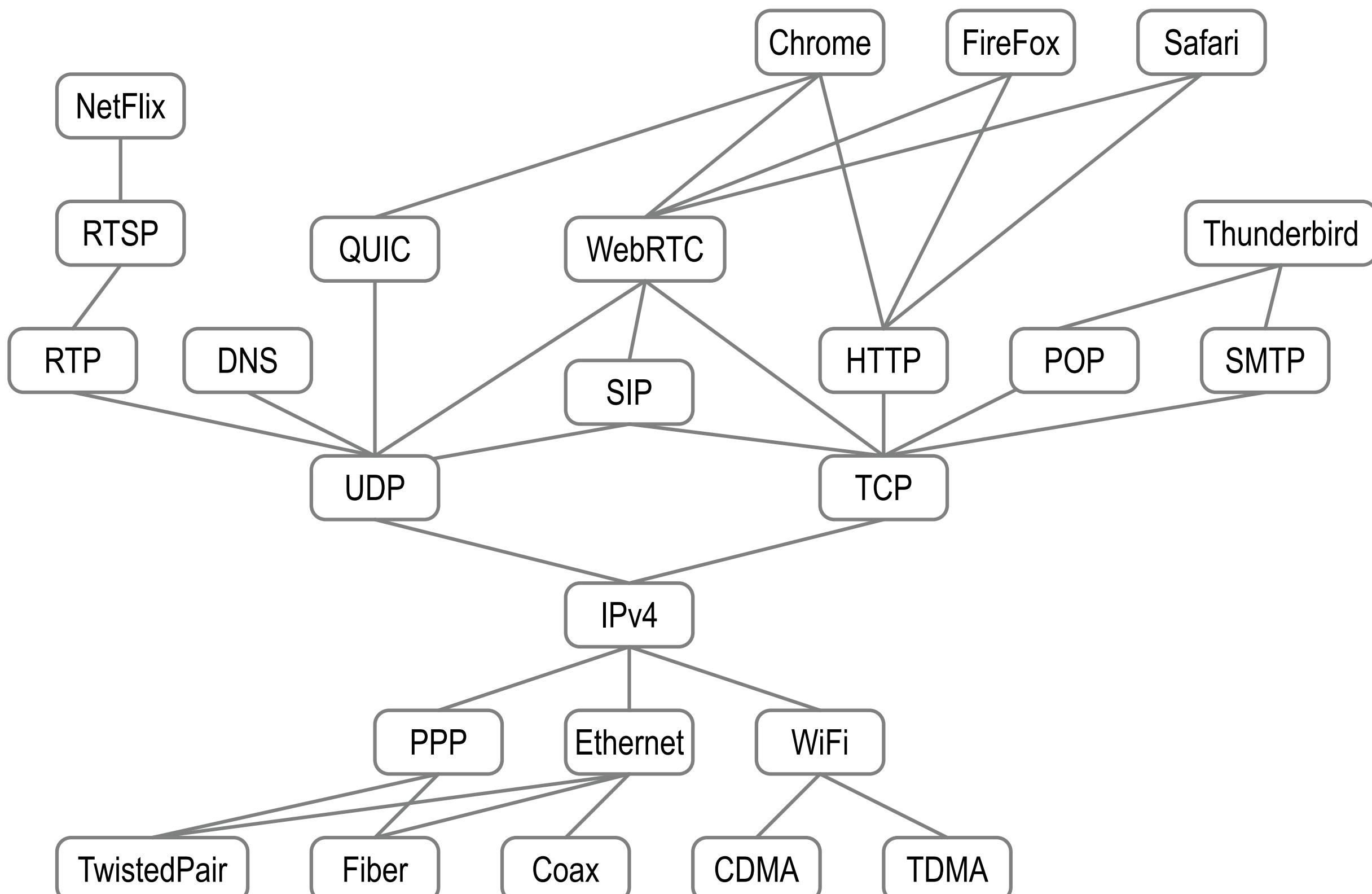
# DNS Hijacking

A DNS hijacking wave is targeting companies at an almost unprecedented scale. Clever trick allows attackers to obtain valid TLS certificate for hijacked domains.

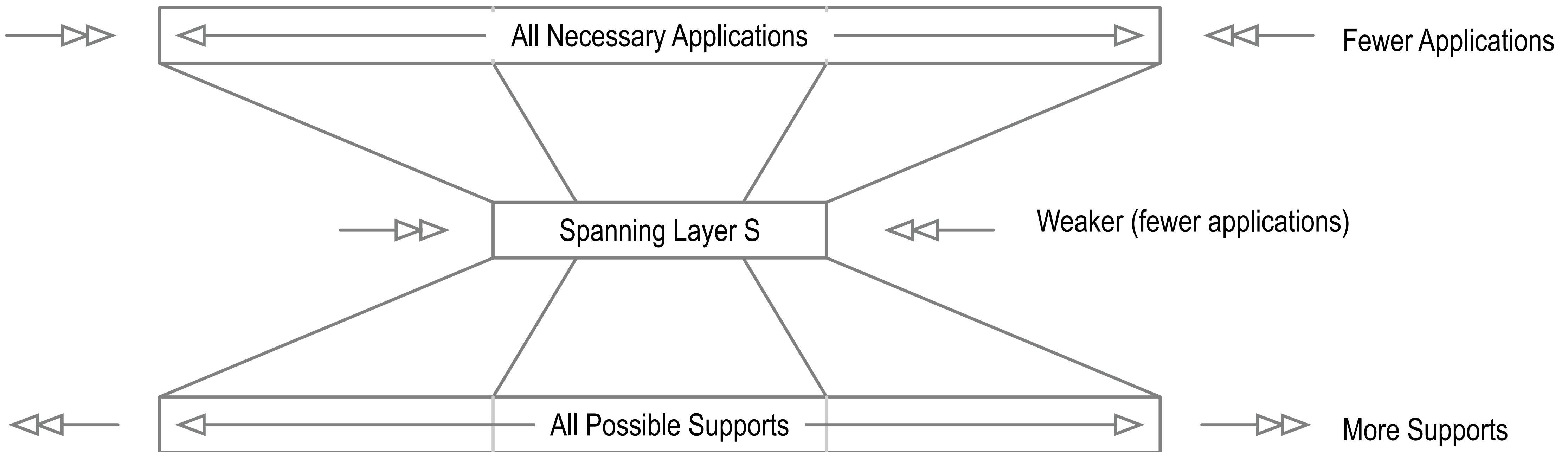
<https://arstechnica.com/information-technology/2019/01/a-dns-hijacking-wave-is-targeting-companies-at-an-almost-unprecedented-scale/>



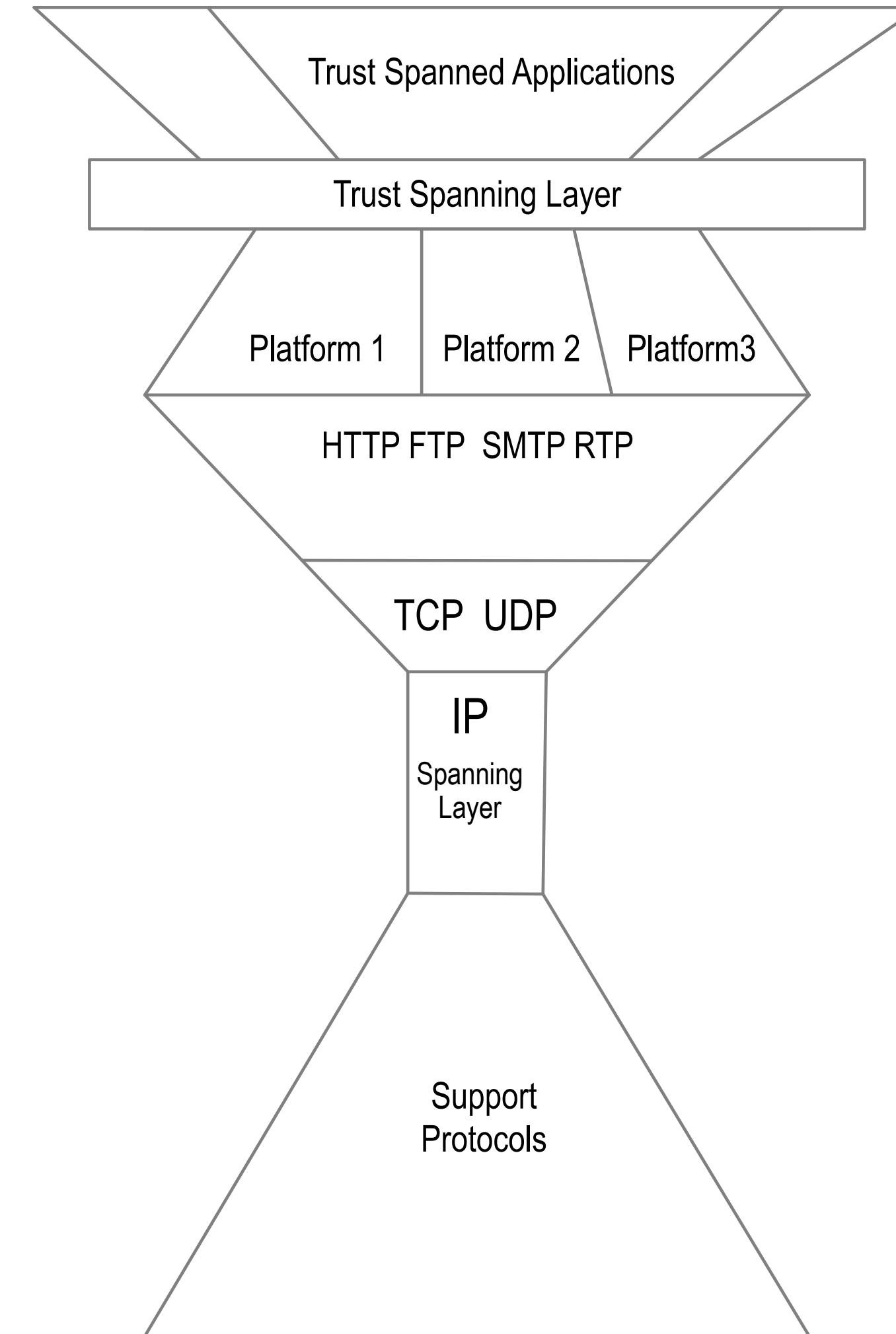
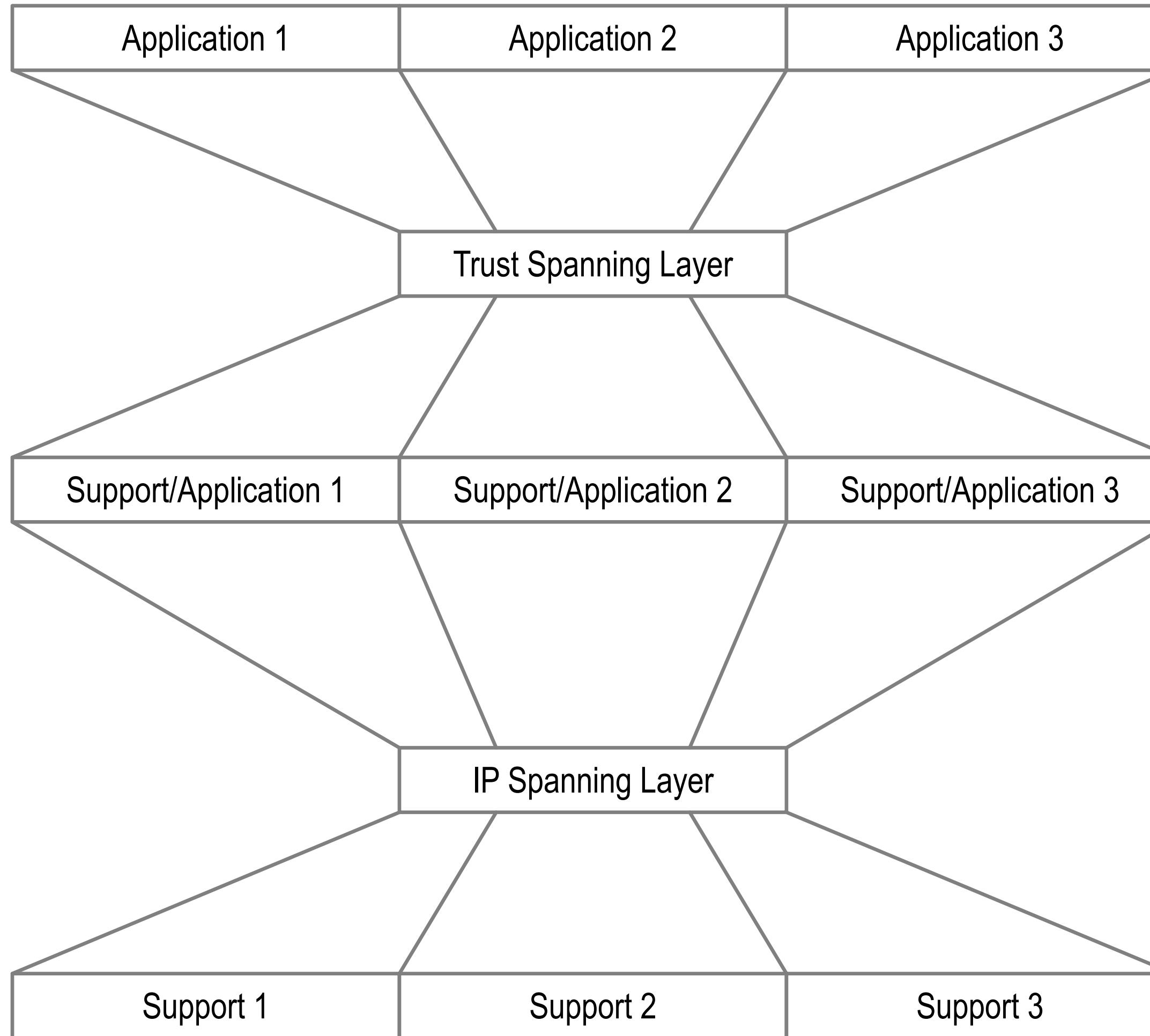
# Spanning Layer



# Hourglass



# Waist and Neck



# Certificate Transparency Problem

“The solution the computer world has relied on for many years is to introduce into the system trusted third parties (CAs) that vouch for the binding between the domain name and the private key. The problem is that we've managed to bless several hundred of these supposedly trusted parties, any of which can vouch for any domain name. Every now and then, one of them gets it wrong, sometimes spectacularly.”

Pinning inadequate

Notaries inadequate

DNSSec inadequate

All require trust in 3rd party compute infrastructure that is inherently vulnerable

Certificate Transparency: (related EFF SSL Observatory)

Public end-verifiable append-only event log with consistency and inclusion proofs

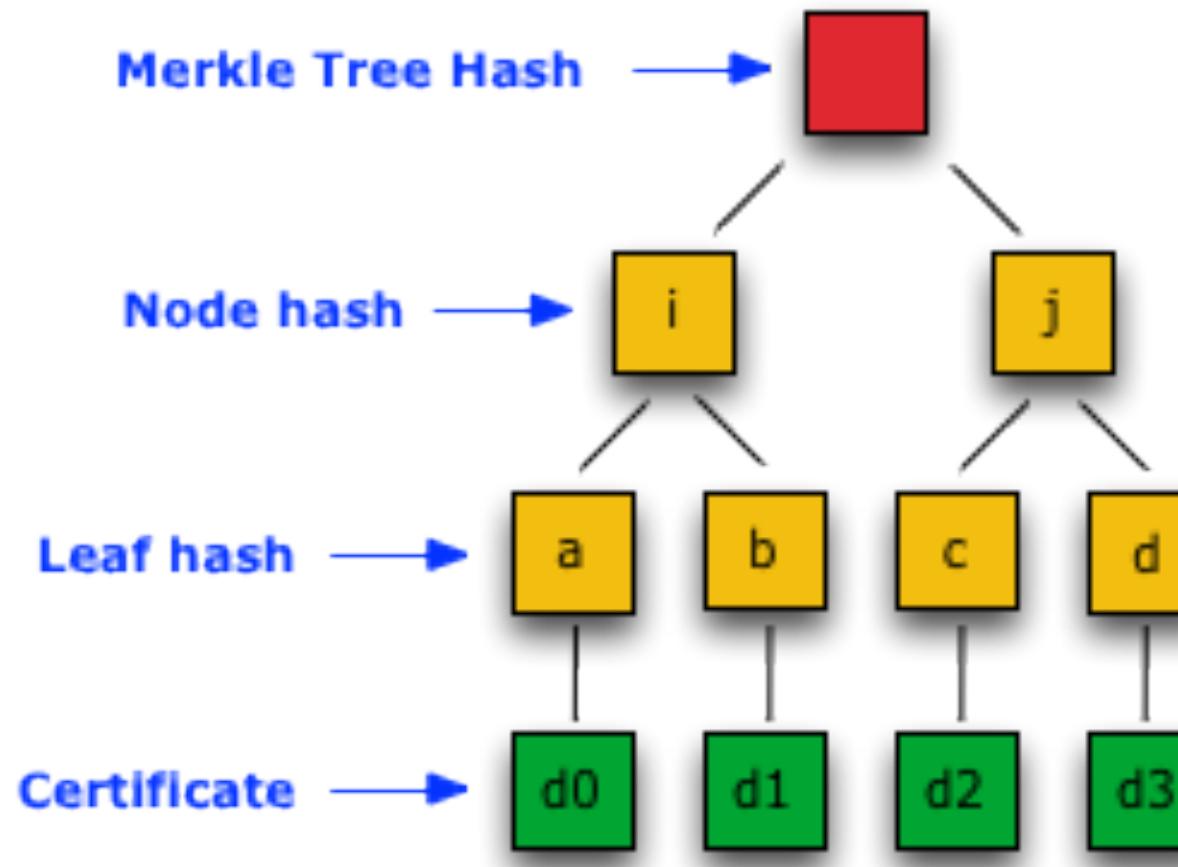
End-verifiable duplicity detection = Ambient verifiability of duplicity

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

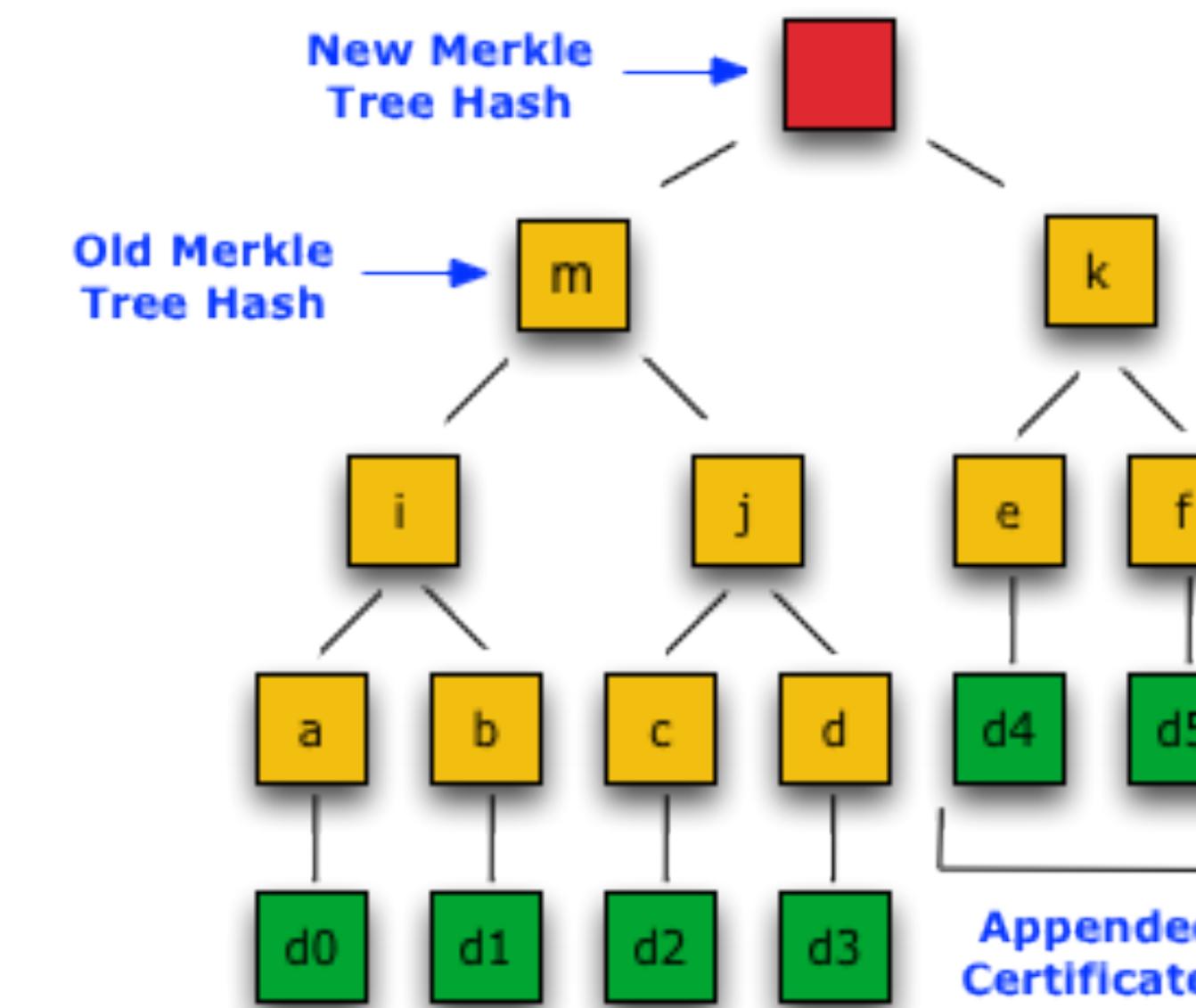
Sparse Merkle Trees for revocation of certificates

# Certificate Transparency Solution

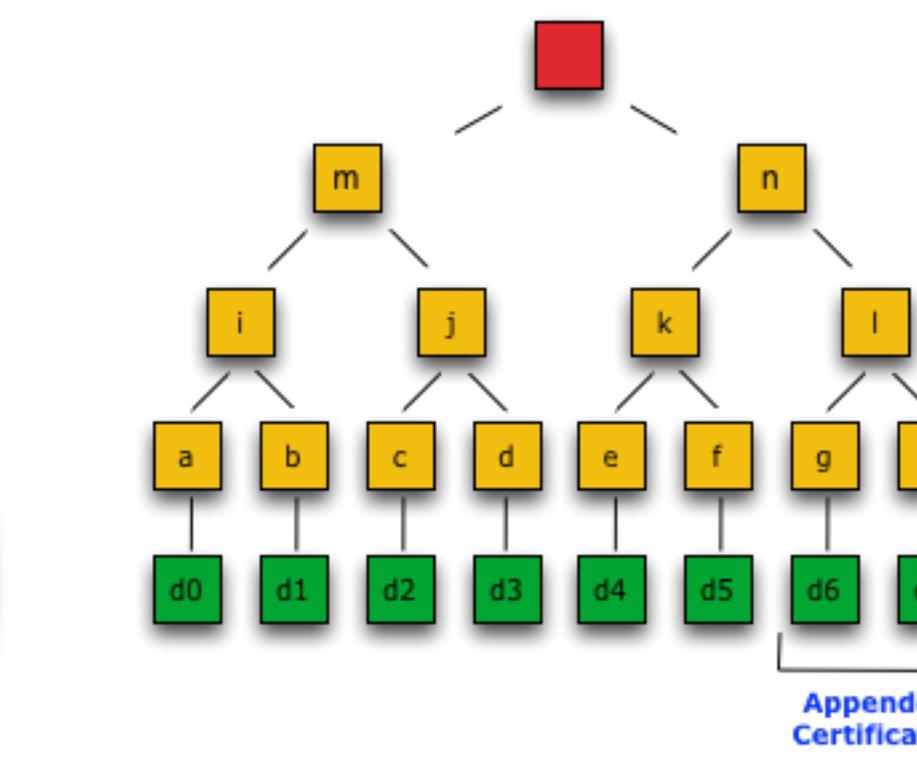
Public end-verifiable append-only event log with consistency and inclusion proofs  
End-verifiable duplicity detection = ambient verifiability of duplicity  
Event log is third party infrastructure but it is not trusted because logs are verifiable.  
Sparse Merkle trees for revocation of certificates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



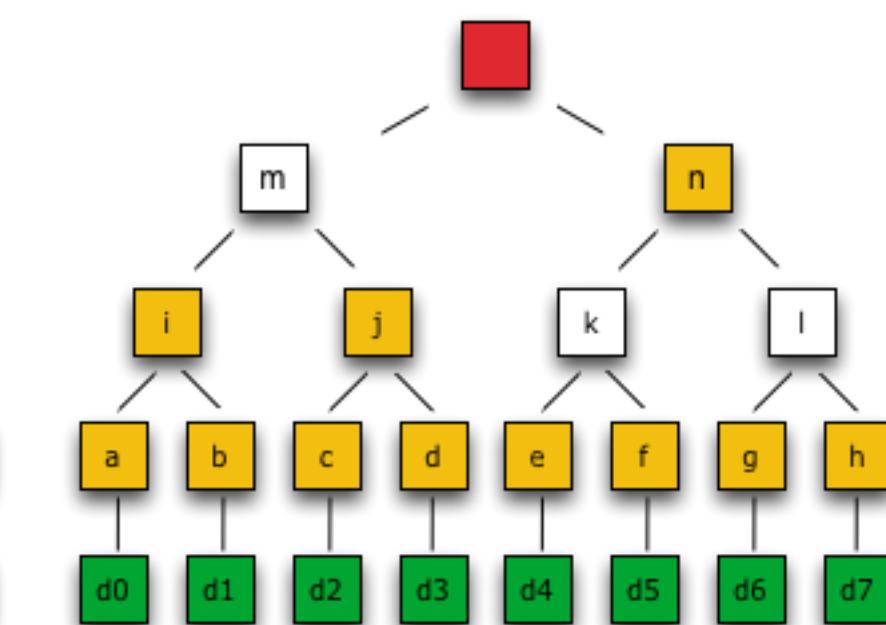
**Figure 1**



**Figure 2**



**Figure 3**



**Figure 4**