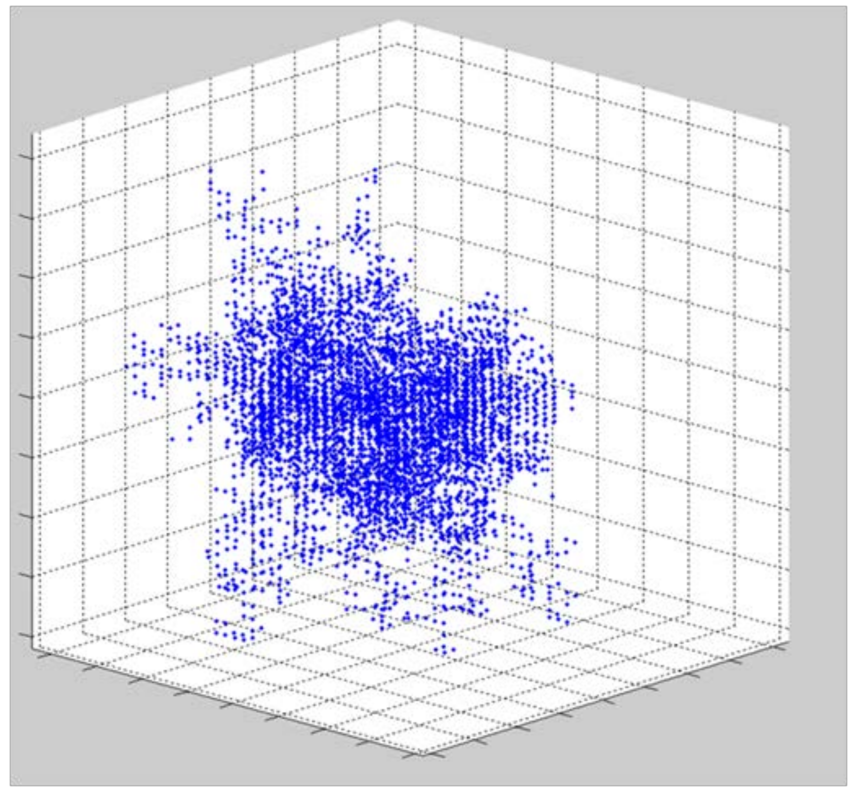
Internet Safety with KERI

Invasion Percolation Discovery
OOBIs (Out-Of-Band-Introductions)
Spanning Trust Layer

Samuel M. Smith Ph.D.
IIW 20201 B
sam@keri.one
https://keri.one

User Permissioned (web-of-trust) Percolated Discovery



Invasion-Percolation Graph Theory for attack resistance

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

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

http://www.physics.purdue.edu/flow/MMproject/Wilkinson1983.pdf

https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.103.018701

The Square and the Tower: Networks and Power. Niall Ferguson 2018

Percolation Theory uses graph theory to model the rate and extent of information flow by

pair-wise or n-wise sharing of information. No global lookup. Weak and Strong Links etc.

If network enables percolation and is spanned then all information is eventually available everywhere

Primary Result (Invasion-Percolation):

Eventually information fills (invades) all honest nodes in the graph whenever "capillary force" (authenticity) is greater for good information over bad information.

User permissioning means honest nodes self-isolate dishonest-nodes.

Each honest user forms identity graph of other honest nodes it interacts with that forms web-of-trust anchoring percolation discovery network.

User Permission Percolated Discovery

Insight: Need-to-know just-in-time discovery (NTK-JIT)

Issuer may provide upon demand at issuance all information an Issuee (*Holder*) *needs to verify the issuance. Now Holder has discovered by percolation what it needs-to-know* (NTK) *just-in-time* (JTK) *to verify.*

Holder now may provide upon demand at presentation all information any verifier needs to to verify the presentation. Now verifier has discovered by percolation what it needs-to-know (NTK) *just-in-time* (JTK) *to verify. This includes all the percolated discovery from Issuer to Holder.*

Likewise the Verifier may imbue on a NTK-JIT *basis any subsequent use of that information with all the percolated discovery information it already received from the Holder plus any other information the Verifier needs to contribute.*

KERI End-Verifiability means zero-trust in the percolation path.

Discovery becomes an availability not a security problem.

User Permission Percolated Discovery

SPED (Speedy Percolated Endpoint Discovery)

Privacy preserving or public discovery as needed

User permissioned & totally decentralized

Replaces or Augments User Permissioned DHT

Watcher Network may provide super Nodes for aggregated discovery if desirable

End-to-end verifiability means any discovery source is as good as any other.

End verifiable "truth" is still true from whatever source it may have come.

This enables secure bootstrap of discovery from any source on a NTK JIT basis.

No need for a globally trusted discovery bootstrap resolver

Zero Trust Percolated Discovery

Primary Discovery Data are Endpoints of KERI Components:

- *Controllers, Agents, Backers (Witness, Registrar), Watchers, Jurors, Judges, Forwarders*
- *Endpoint is URL: IP Scheme, Host, Port, Path etc*

Data Model for Securely Managing EndPoint Data

- *Controller (Principal AID)*
- *Authorizes a Component to act as Player in Role*
- *Player is AID of Component Controller*
- *Role is purpose or function such as Watcher*

Zero Trust Data as Authorization in context of KERI KeyState

- *ACDC Issue Revoke Reissue model*
- *RUN model (Read, Update, Nullify)*
  - *Anchored or Signed with replay and deletion attack protection*

Safe Internet Use

Minimally Sufficient Means

Leverage existing internet but safely, with end-verifiability

Internet DNS/CA is out-of-band w.r.t. KERI security

Use DSN/CA for out-of-band introductions w.r.t. KERI only, not authentication

Use IP addresses (128.187.16.184) for communication

OOBI (Out-Of-Band-Introduction)

How to use DNS safely!

Vaccuous discovery of service endpoints

Basic

```
https://hackmd.io/MxTAIBQTRkWU4-w140tNuA
```

OOBI = Url and AID   Simple enough for QR Code

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Variant: Use query string to label endpoint to be discovered.

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=watcher&name=eve

https://example.com/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=witness
```

Well-Known Variant:

```
/.well-known/keri/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Result of well-known request is target URL or redirection

```
https://example.com/witness/witmer  (redirection)

http://8.8.5.5:8080/witness/witmer  (public IP)

http://10.0.5.15:8088/witness/witmer  (private IP)
```

OOBI (Out-Of-Band-Introduction)

How to use DNS safely! Vacuuous discovery of service endpoints.

Basic

```
https://hackmd.io/MxTAIBQTRkWU4-w140tNuA
```

OOBI = Url and AID   Simple enough for QR Code

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Variant: Use query string to label endpoint to be discovered.

```
http://8.8.5.6:8080/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=watcher&name=eve

https://example.com/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM?role=witness
```

Well-Known Variant:

```
/.well-known/keri/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM
```

Result of well-known request is target URL or redirection

```
https://example.com/witness/witmer  (redirection)

http://8.8.5.5:8080/witness/witmer  (public IP)

http://10.0.5.15:8088/witness/witmer  (private IP)
```

Any OOBI may forward to another OOBI.

This is safe because the eventual endpoint is end-verifiable (authenticated).

OOBI (Out-Of-Band-Introduction)

Verbose OOBI Multi-OOBI

```
{
          "v" : "KERI10JSON00011c_",
          "t" : "rpy",
          "d": "EZ-i0d8JZAoTNZH3ULaU6JR2nmwyvYAfSVPzhzS6b5CM",
          "dt": "2020-08-22T17:50:12.988921+00:00",
          "r" : "/oobi/witness",
          "a" :
          {
             "urls":  ["http://example.com/watcher/watson", "http://example.com/witness/wilma"]
             "aid":  "EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM"
          }
}
```

Special Route Path

```
{
          "v" : "KERI10JSON00011c_",
          "t" : "rpy",
          "d": "EZ-i0d8JZAoTNZH3ULaU6JR2nmwyvYAfSVPzhzS6b5CM",
          "dt": "2020-08-22T17:50:12.988921+00:00",
          "r" : "/oobi/EaU6JR2nmwyZ-i0d8JZAoTNZH3ULvYAfSVPzhzS6b5CM/watcher",
          "a" :
          {
             "eid": "BrHLayDN-mXKv62DAjFLX1_Y5yEUe0vA9YPe_ihiKYHE",
             "scheme": "http",
             "url":  "http://example.com/watcher/wilma",
          }
}
```

Bare URL as Self or Blind OOBI

A bare URL but no AID may be used as a bare OOBI for blind or self introductions.

Querying that bare URL (OOBI) may return or result in a default target OOBI or default target endpoint reply.

This provides a mechanism for self-introduction, self OOBI (SOOBI) or blind-introduction, blind OOBI (BOOBI) .

http://8.8.5.7:8080/oobi

http://localhost:8080/oobi```

http://8.8.5.7:8080/oobi?role=controller&name=eve

http://localhost:8080/oobi?role=controller&name=eve

By default the result of get request to this OOBI URL could be another OOBI with an AID that is the `self` AID of the node providing the bare OOBI endpoint or the actual authenticatable `self` endpoint with its AID or a default set of authenticatable endpoints.

Useful to bootstrap components in an infrastructure where the target URLs do not use a public DNS address but use instead something more secure like an explicit public IP address or a private IP or private DNS address.

A self introduction provides a bootstrap mechanism similar to a hostname configuration file with the exception that in the OOBI case the AID is not in the configuration file just the bare OOBI URL and the given node queries that bare OOBI to get the target endpoint AID. This allows bootstrap using bare IP addresses in systems where the IP infrastructure is more securely managed than public DNS or where some other Out-Of-Band-Authentication (OOBA) mechanism is used in concert.

Blind OOBI

Because the OOBI does not expose an AID, the the resultant response when querying the OOBI may depend on other factors such as the source IP of the querier (requester) and/or another out-of-band-authentication (OOBA) mechanism. This supports private bootstrap of infrastructure.

Of course one could argue that this is just kicking the can down the road but IP addresses are correlatable and a blind OOBI can leverage IP infrastructure for discovery when useful in combination with some other OOBA mechanism without unnecessary correlation.

Onion Routing with Blind OOBI

did-comm with Blind OOBI

Attack Protection

Replay Attack: Replay *of Authenticated (signed) Data*

TEL (ACDC) VDR Issue Revoke (kel anchored tel events) Heavyweight

Non TEL based: Best Available Data Model (BADA)

KEL anchored ordered data

KeyState-DateTime of signature ordered data.

Deletion Attack

Total erasure a security problem (GDPR flaw)

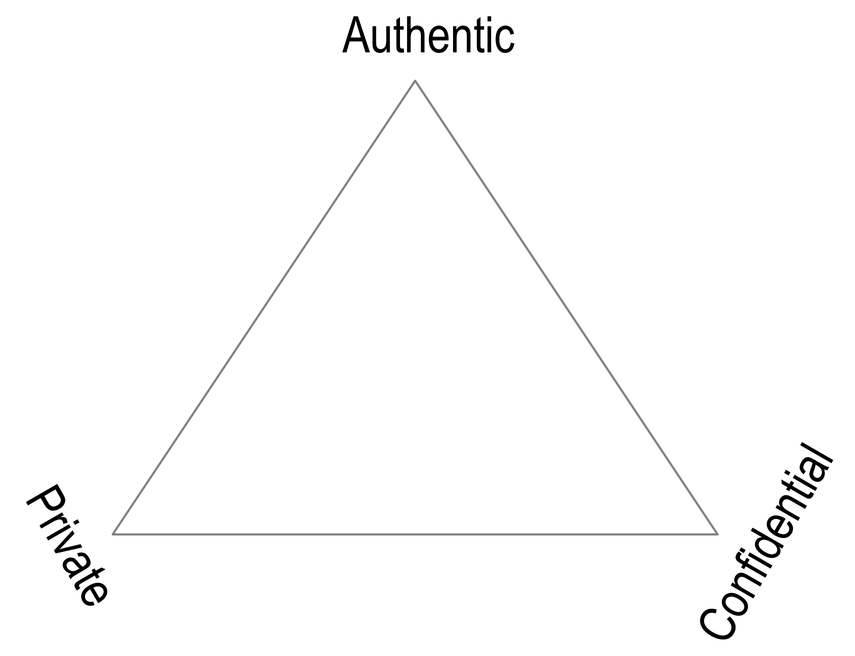
Once erased any stale authenticated data acting as authorization may be replayed without detection.

Mitigation for Deletion attack are redundant signed copies (eventually consistent DB)

BADA (Best Available Data Acceptance) Policy

Authentic Data:
- Two primary attacks:
  - Replay attack:
    - Mitigation: Monotonicity
  - Deletion attack:
    - Mitigation: Redundancy

Replay Monotonicity:
- Interactive:
  - Nonce
- Non-interactive:
  - Memory (sequence number, date-time stamp, nullification)
  - More scalable



BADA Rules

Rules for Update : (anchored to key state in KEL)

Accept if no prior record.

Accept if anchor is later than prior record.

Rules for Update: (signed by keys given by key state in KEL, ephemeral identifiers have constant key state)

Accept if no prior record.

Accept if key state is later than prior record.

Accept if key state is the same and date-time stamp is later than prior record.

RUN off the CRUD

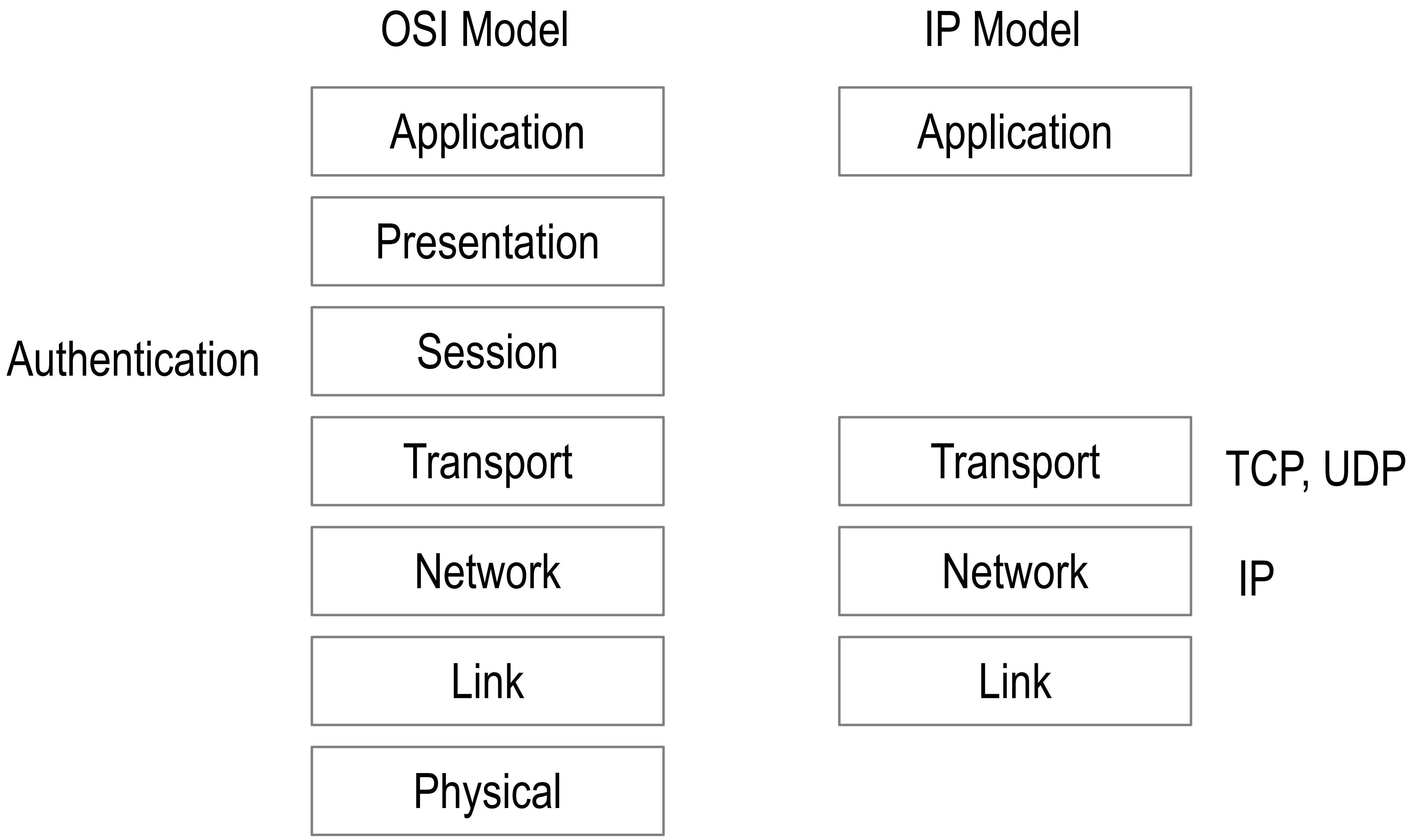
Client-Server API or Peer-to-Peer.
Create, Read, Update, Delete (CRUD)
Read, Update, Nullify (RUN)
Decentralized control means server never creates only client. Client (Peer) updates server (other Peer) always for data sourced by Client (Peer). So no Create.
Non-interactive monotonicity means we can’t ever delete.
So no Delete. We must Nullify instead. Nullify is a special type of Update.
Ways to Nullify:

- null value
- flag indicating nullified

EndPoint Disclosue

Datetime stamped BADA authorization by CID of EID in Role (Update)
Datetime stamped BADA deauthorization by CID of EID in Role (Nullify)
Datetime stamped BADA authorization by EID of URL for scheme (Update).
Datetime stamped BADA deauthorization by EID of URL for scheme (Nullify)

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



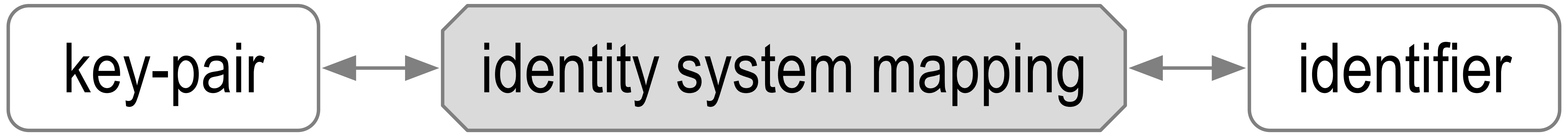
	OSI Model	IP Model	
	Application	Application	
	Presentation		
Authentication	Session		
	Transport	Transport	TCP, UDP
	Network	Network	IP
	Link	Link	
	Physical		

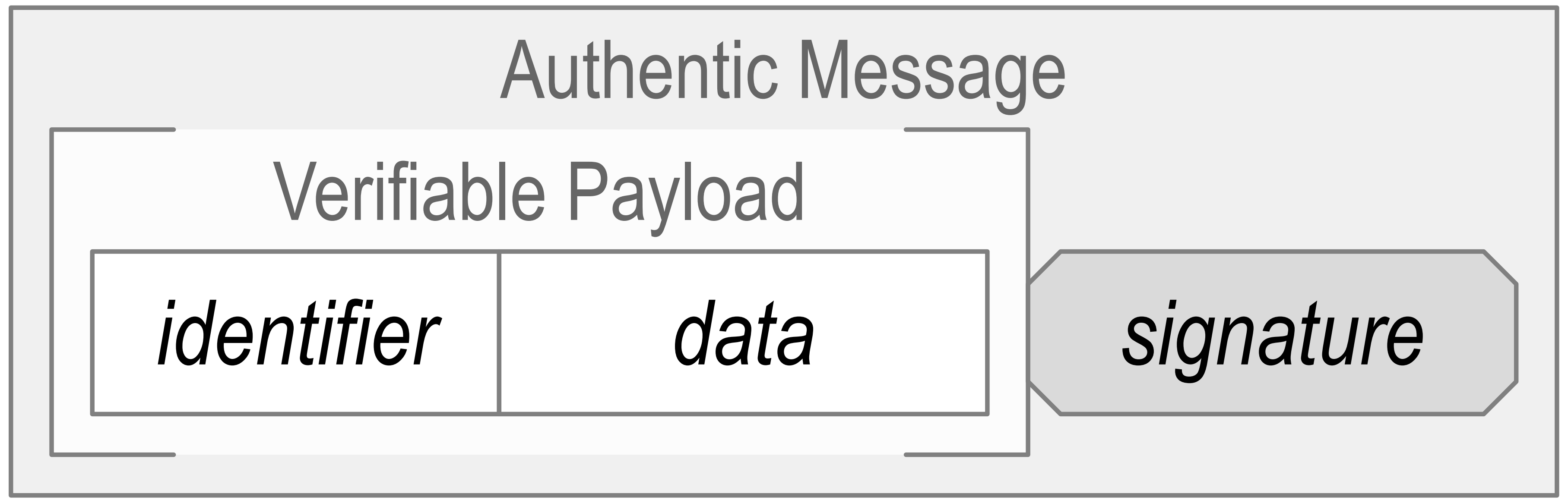
Instead …

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

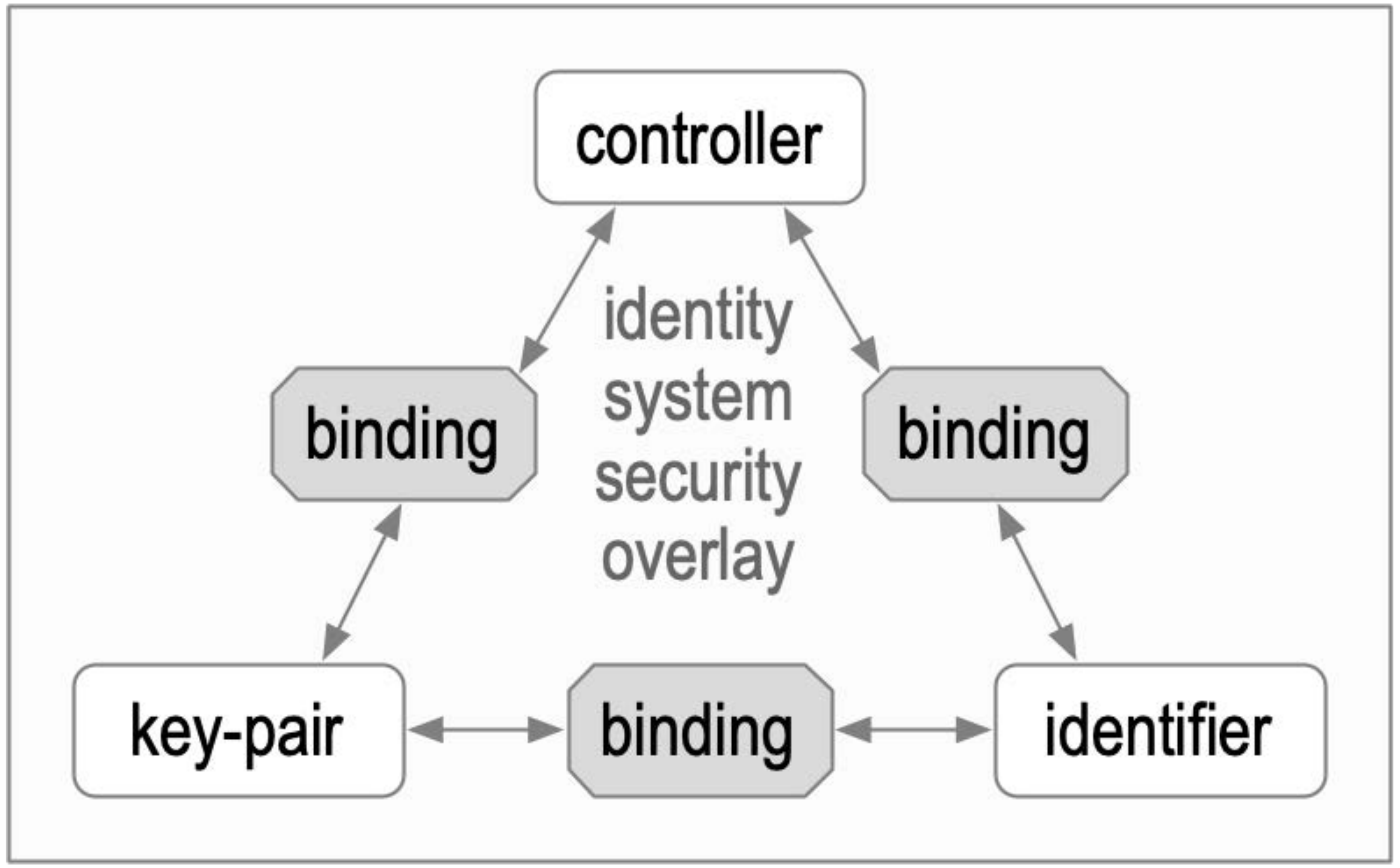
Identity System Security Overlay

Establish authenticity of IP packet's message payload.



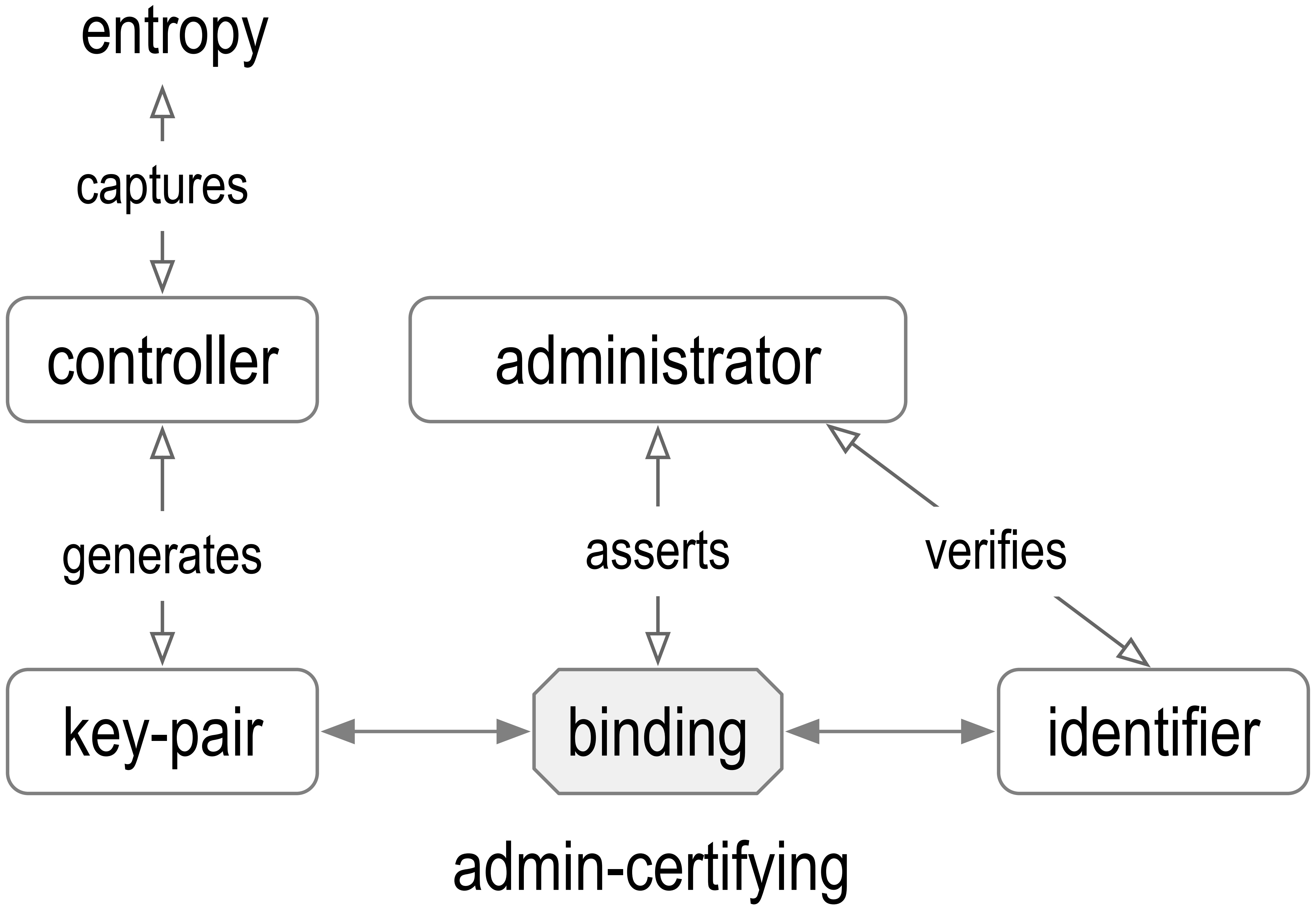


The overlay's security is contingent on the mapping's security.



Identifier Issuance

Administrative Identifier Issuance and Binding



entropy

captures

controller

generates

key-pair

administrator

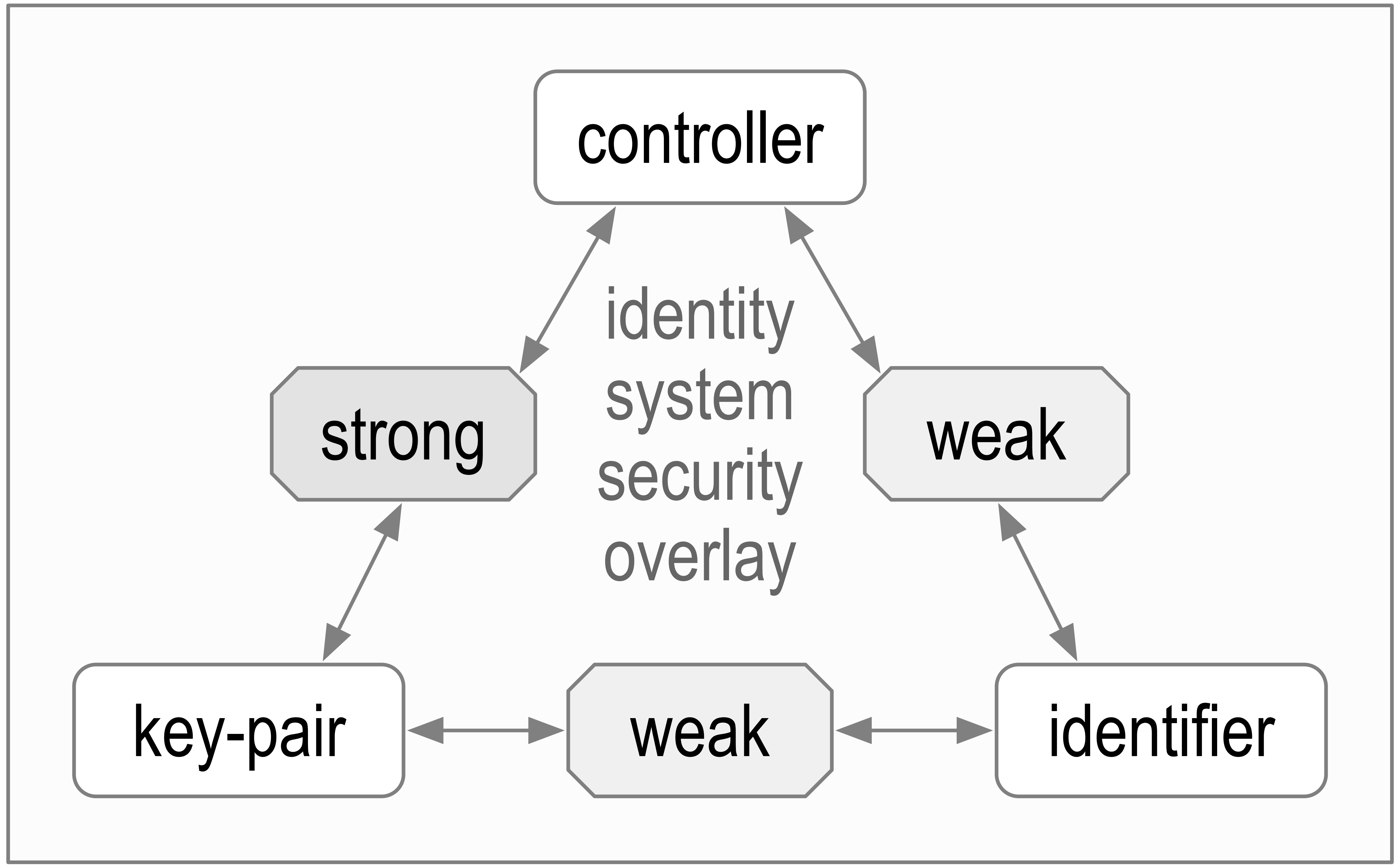
asserts

verifies

binding

identifier

admin-certifying



controller

identity system security overlay

strong

weak

key-pair

weak

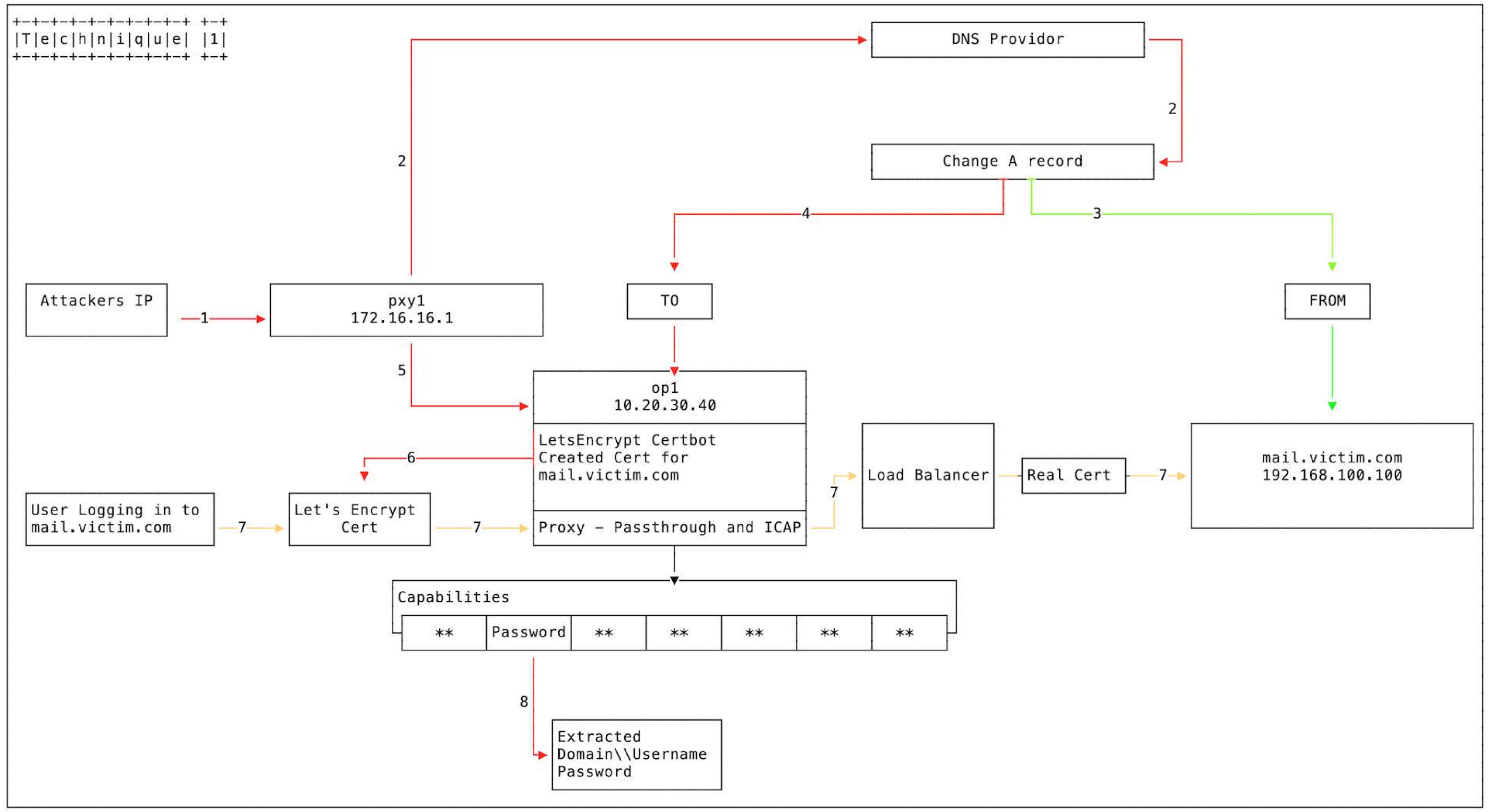
identifier

Admin-Certifying Identifier Issuance

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/



Technique 1

DNS Providor

Change A record

Attackers IP

pxy1
172.16.16.1

TO

FROM

op1
10.20.30.40

LetsEncrypt Certbot
Created Cert for
mail.victim.com

Proxy - Passthrough and ICAP

Load Balancer

Real Cert

mail.victim.com
192.168.100.100

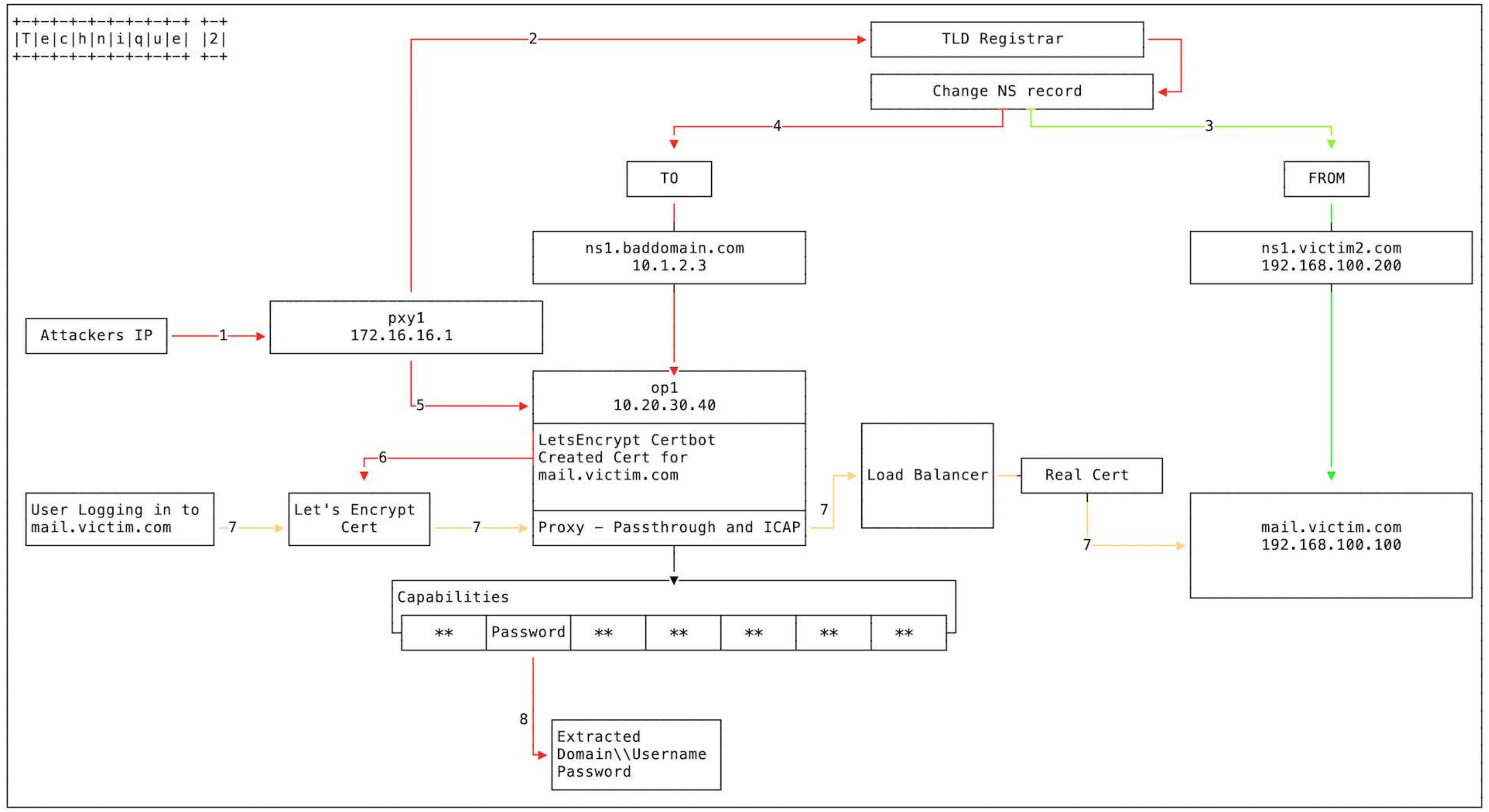
User Logging in to
mail.victim.com

Let's Encrypt
Cert

Capabilities

**	Password	**	**	**	**	**

Extracted
Domain\\Username
Password



Technique 2

TLD Registrar

Change NS record

TO

FROM

ns1.baddomain.com
10.1.2.3

ns1.victim2.com
192.168.100.200

Attackers IP

pxy1
172.16.16.1

op1
10.20.30.40

LetsEncrypt Certbot
Created Cert for
mail.victim.com

Proxy - Passthrough and ICAP

Load Balancer

Real Cert

mail.victim.com
192.168.100.100

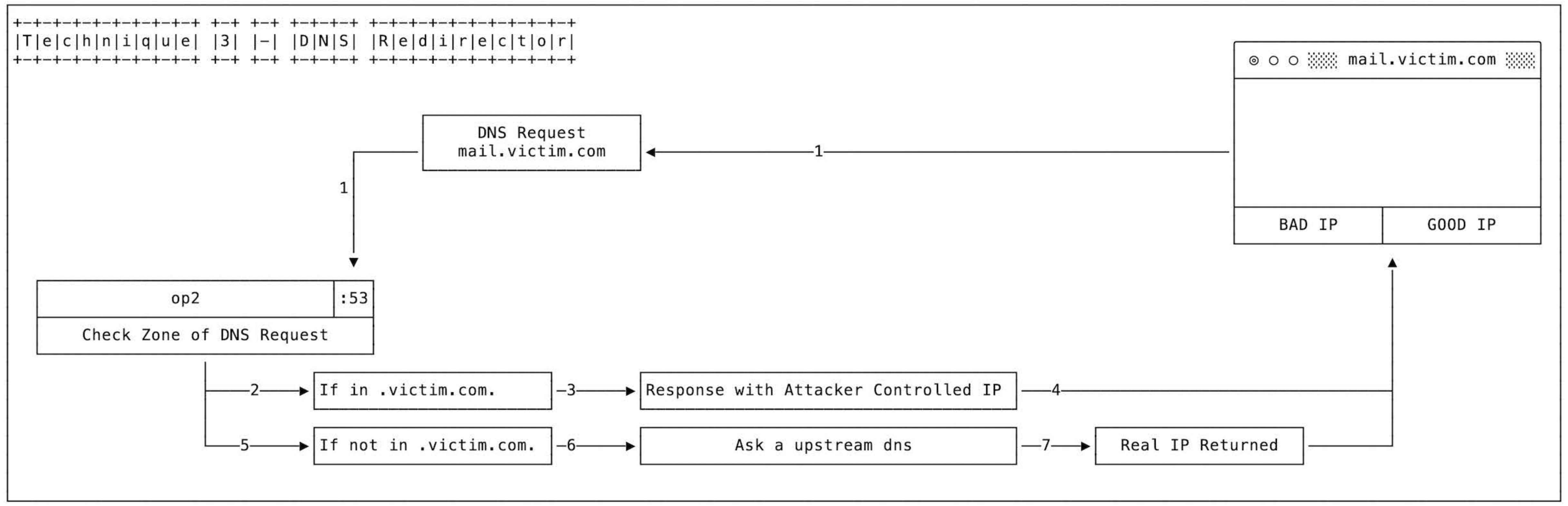
User Logging in to
mail.victim.com

Let's Encrypt
Cert

Capabilities

**	Password	**	**	**	**	**

Extracted
Domain\\Username
Password



Technique 3 - DNS Redirector

mail.victim.com

BAD IP | GOOD IP

DNS Request
mail.victim.com

op2 :53
Check Zone of DNS Request

If in .victim.com.

Response with Attacker Controlled IP

If not in .victim.com.

Ask a upstream dns

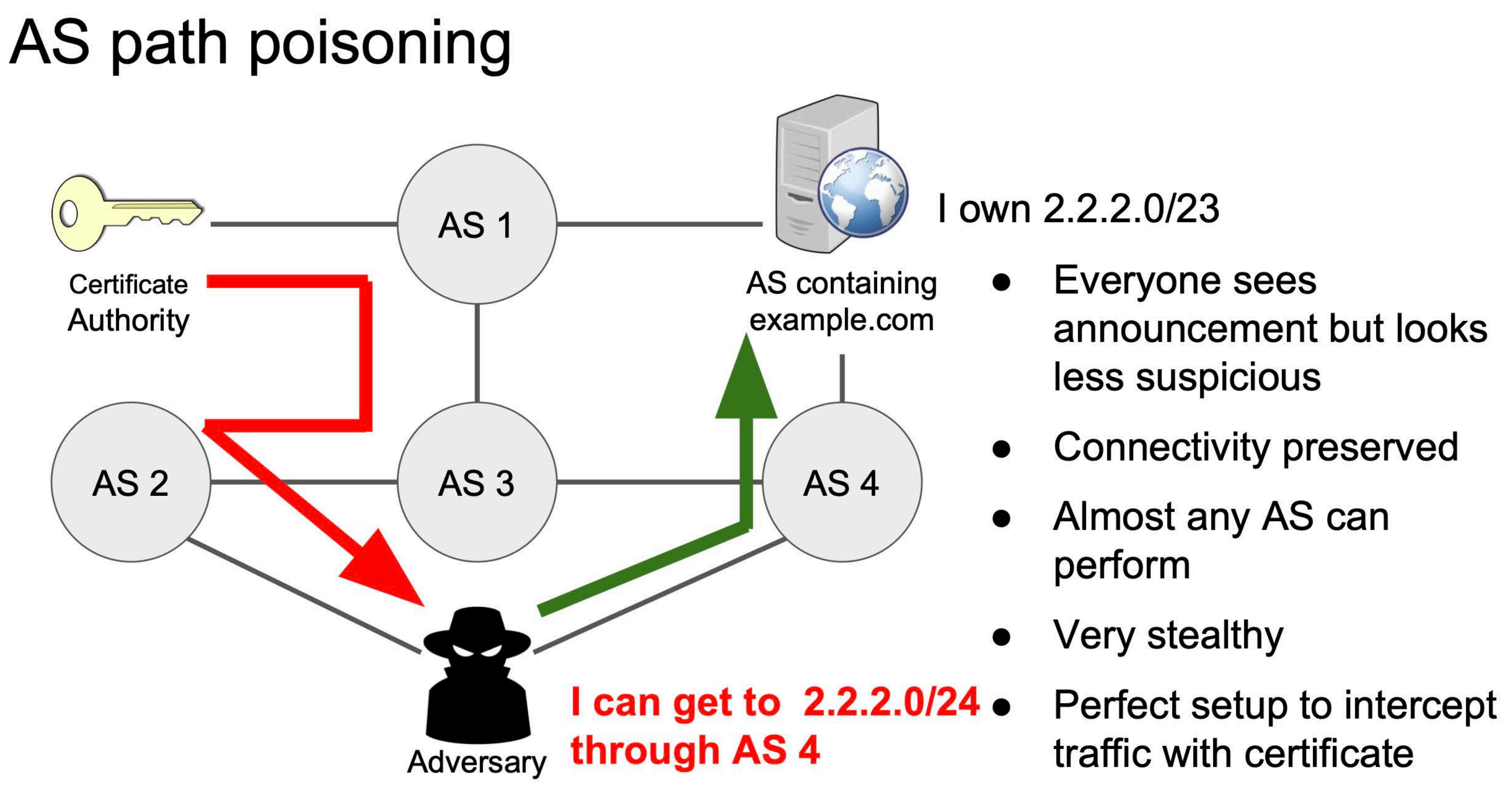
Real IP Returned

BGP Hijacking: AS Path Poisoning

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

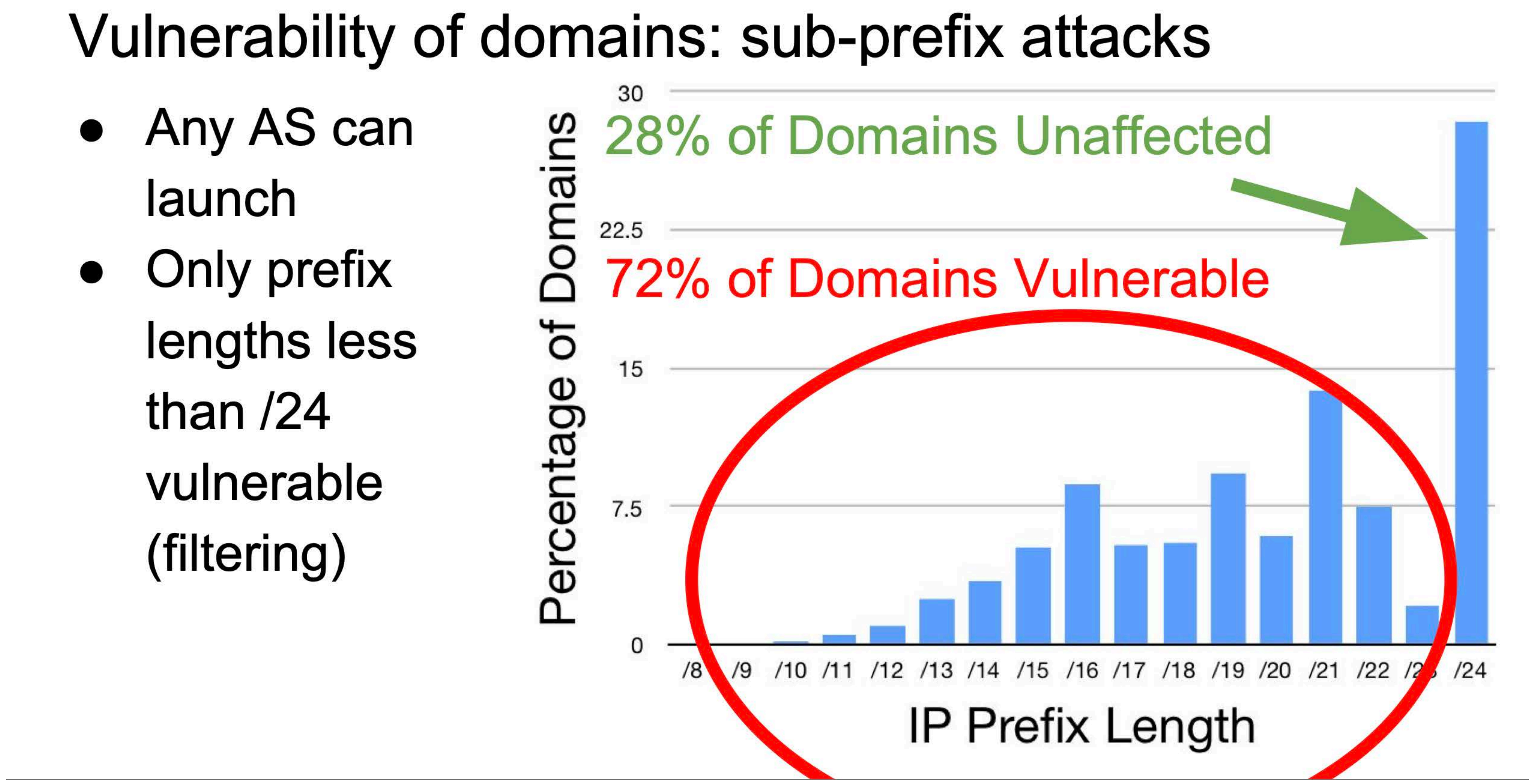
Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J. and Mittal, P., “Bamboozling certificate authorities with {BGP},” vol. 27th {USENIX} Security Symposium, no. {USENIX} Security 18, pp. 833-849, 2018 https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee

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

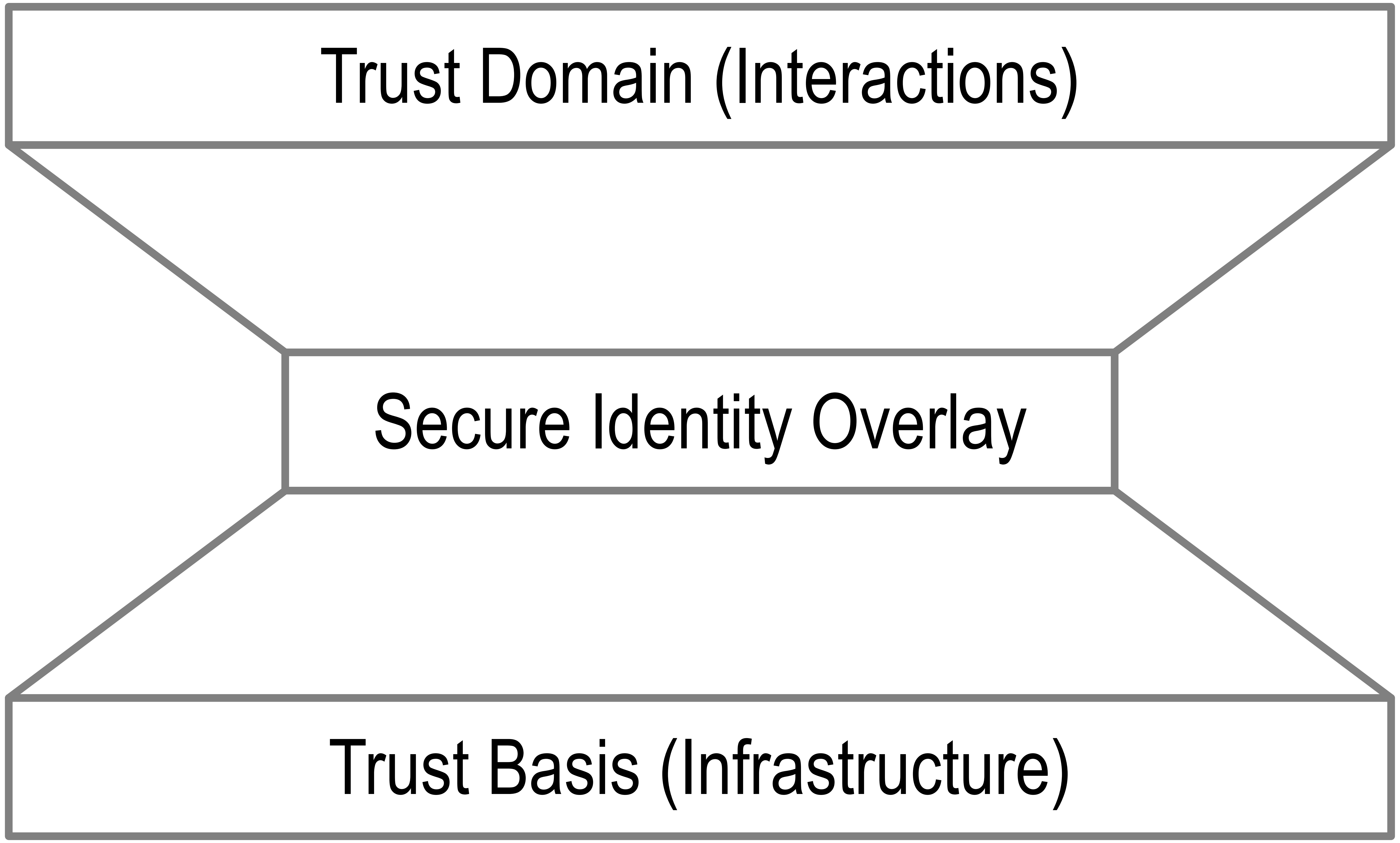
- Everyone sees announcement but looks less suspicious
- Connectivity preserved
- Almost any AS can perform
- Very stealthy
- Perfect setup to intercept traffic with certificate



Vulnerability of domains: sub-prefix attacks

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

Identity System Security Overlay

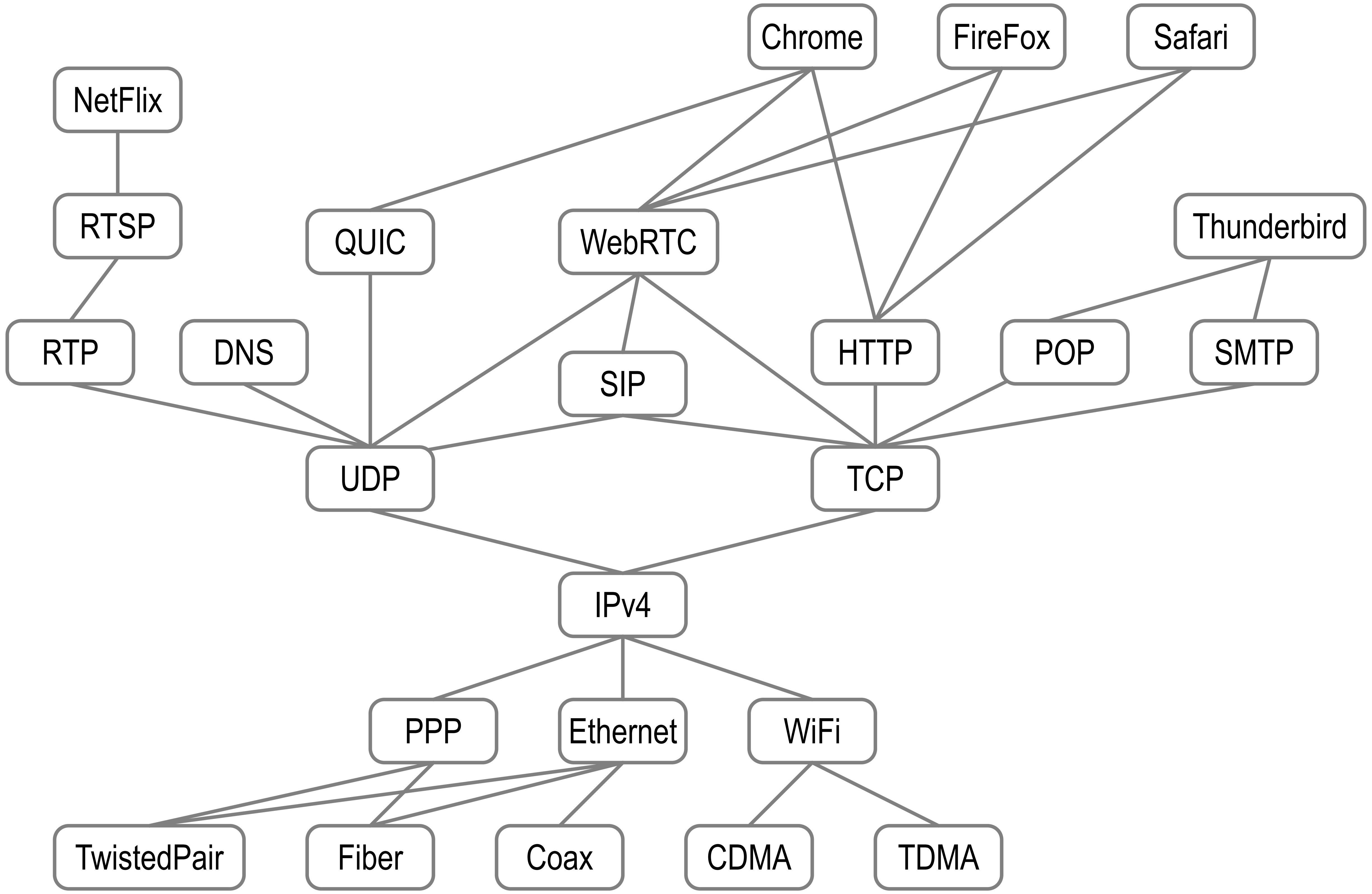


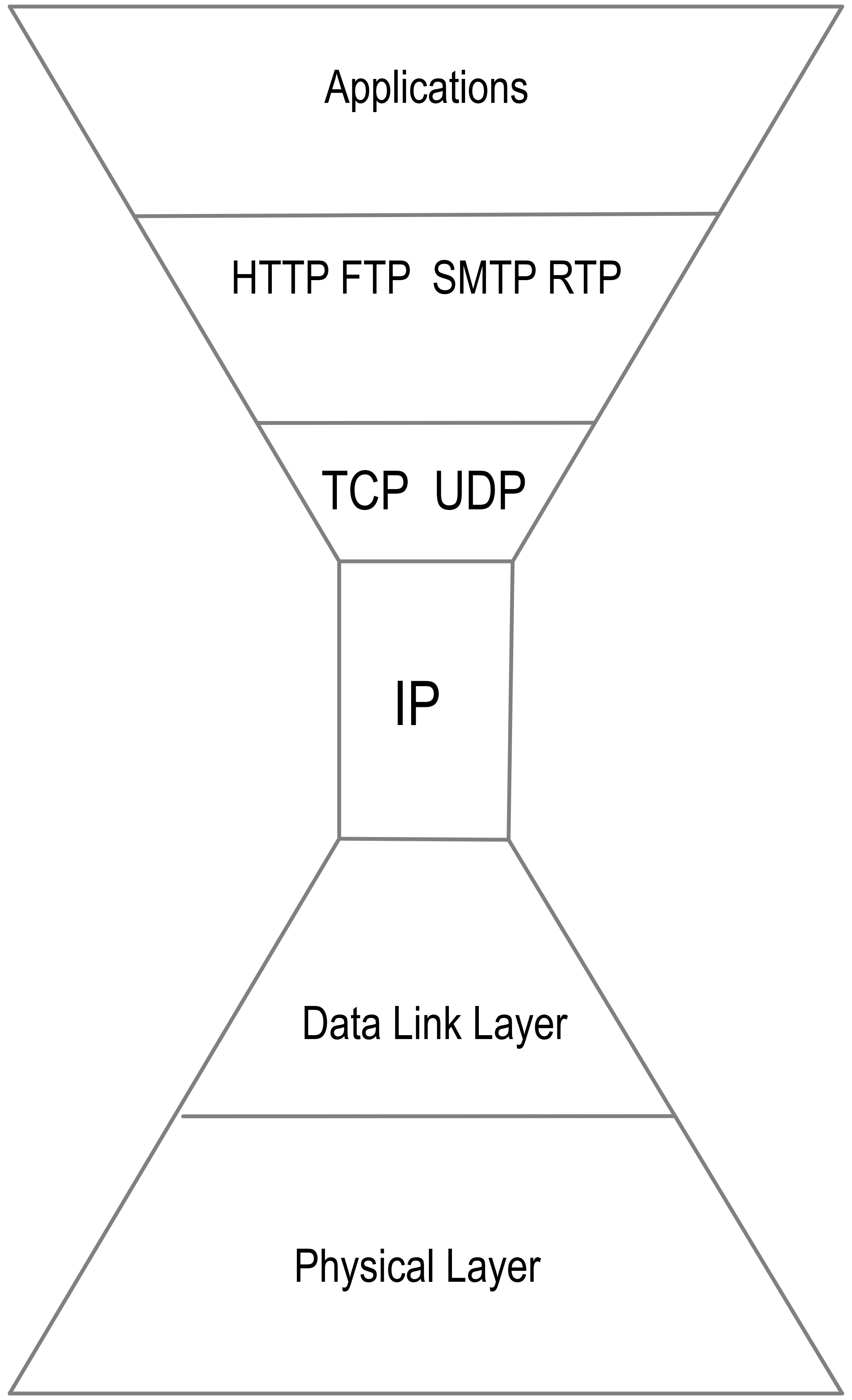
Trust Domain (Interactions)

Secure Identity Overlay

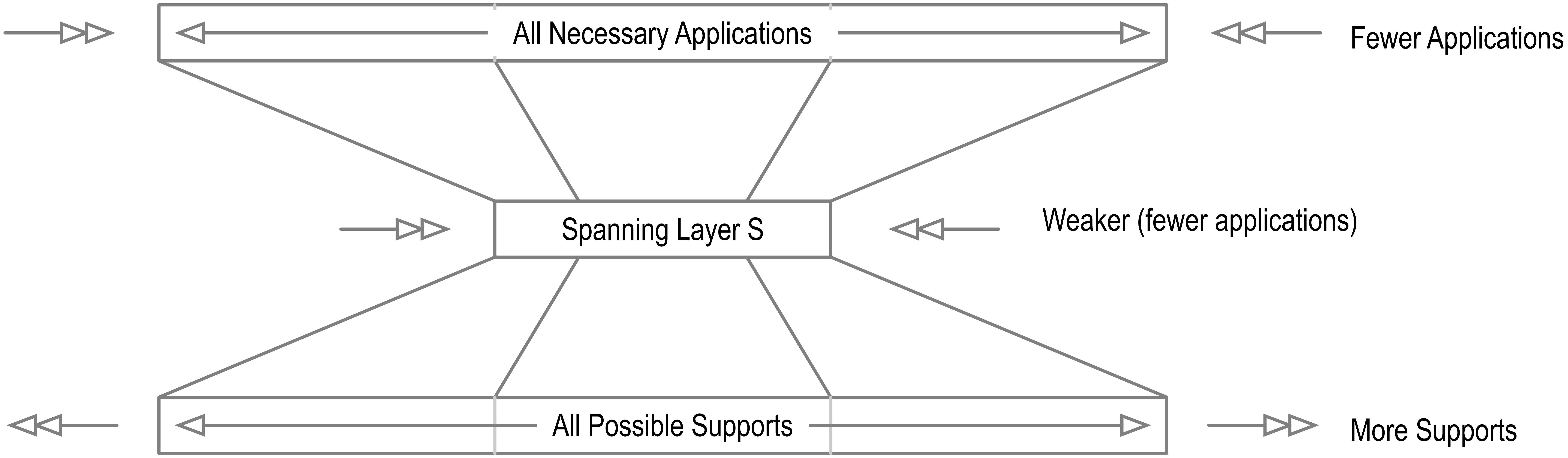
Trust Basis (Infrastructure)

Spanning Layer





Hourglass



All Necessary Applications

Fewer Applications

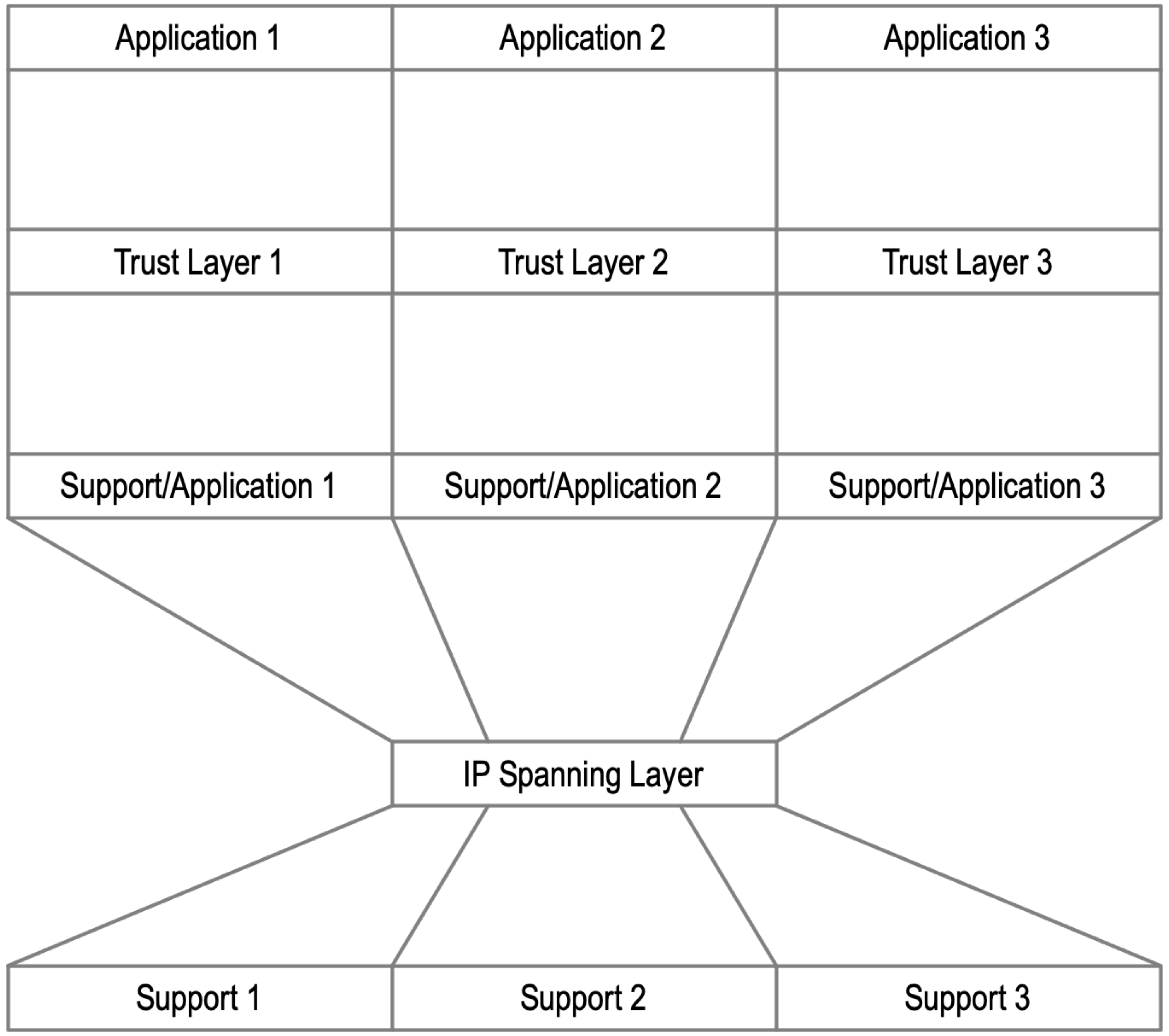
Spanning Layer S

Weaker (fewer applications)

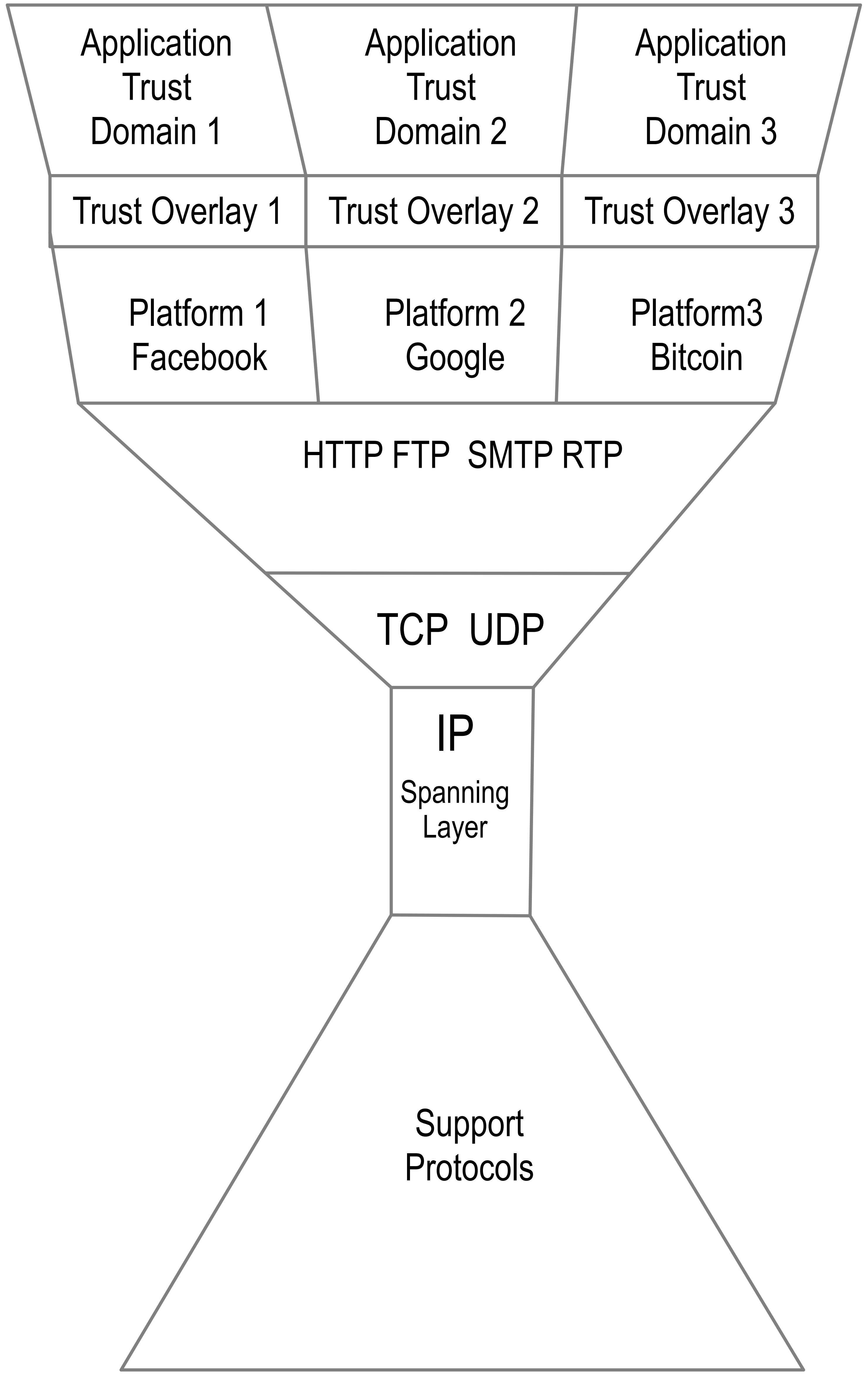
All Possible Supports

More Supports

Platform Locked Trust



Trust Domain Based Segmentation

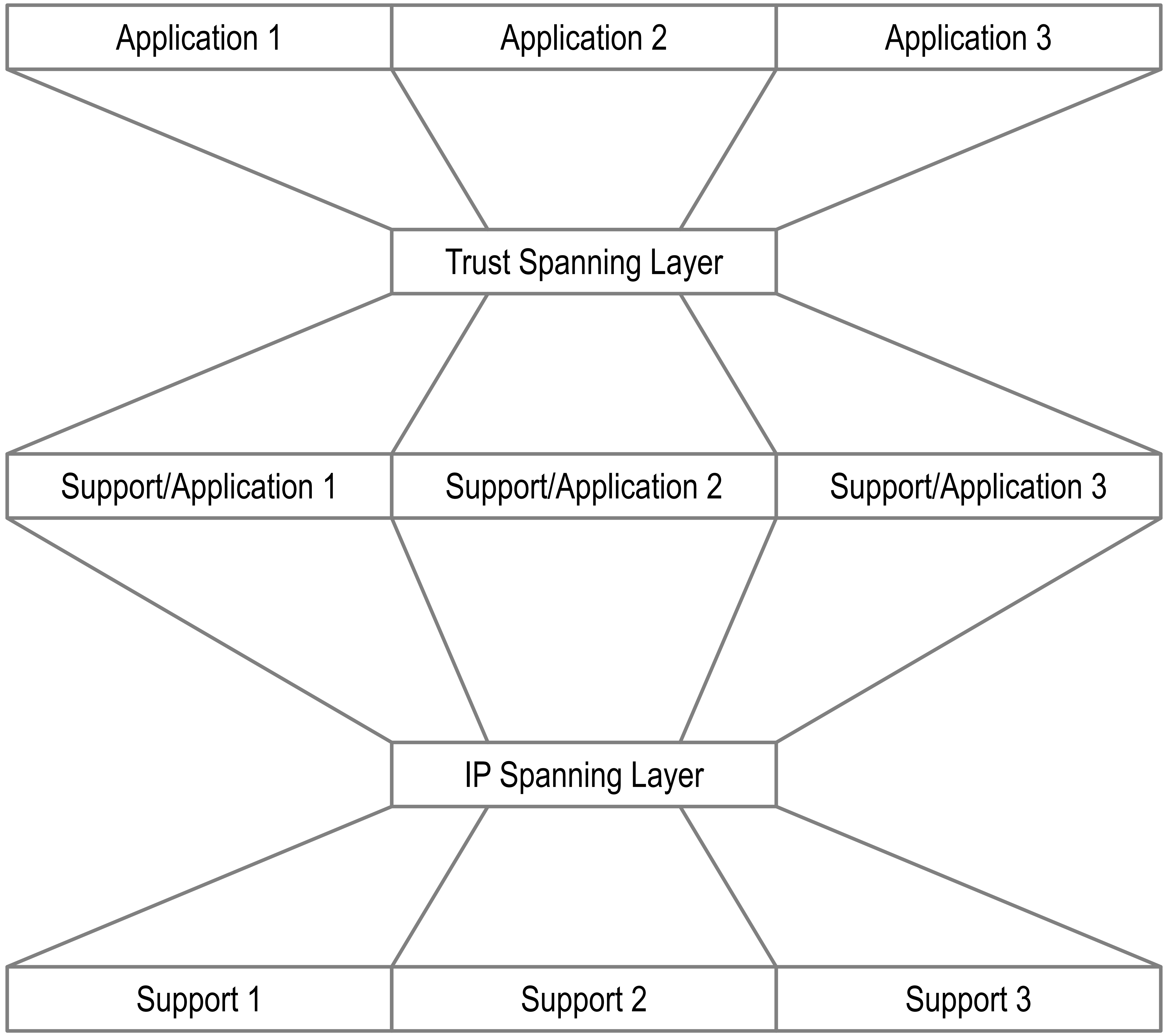


Each trust layer only spans platform specific applications

Bifurcates the internet trust map

No spanning trust layer

Waist and Neck



Application 1

Application 2

Application 3

Trust Spanning Layer

Support/Application 1

Support/Application 2

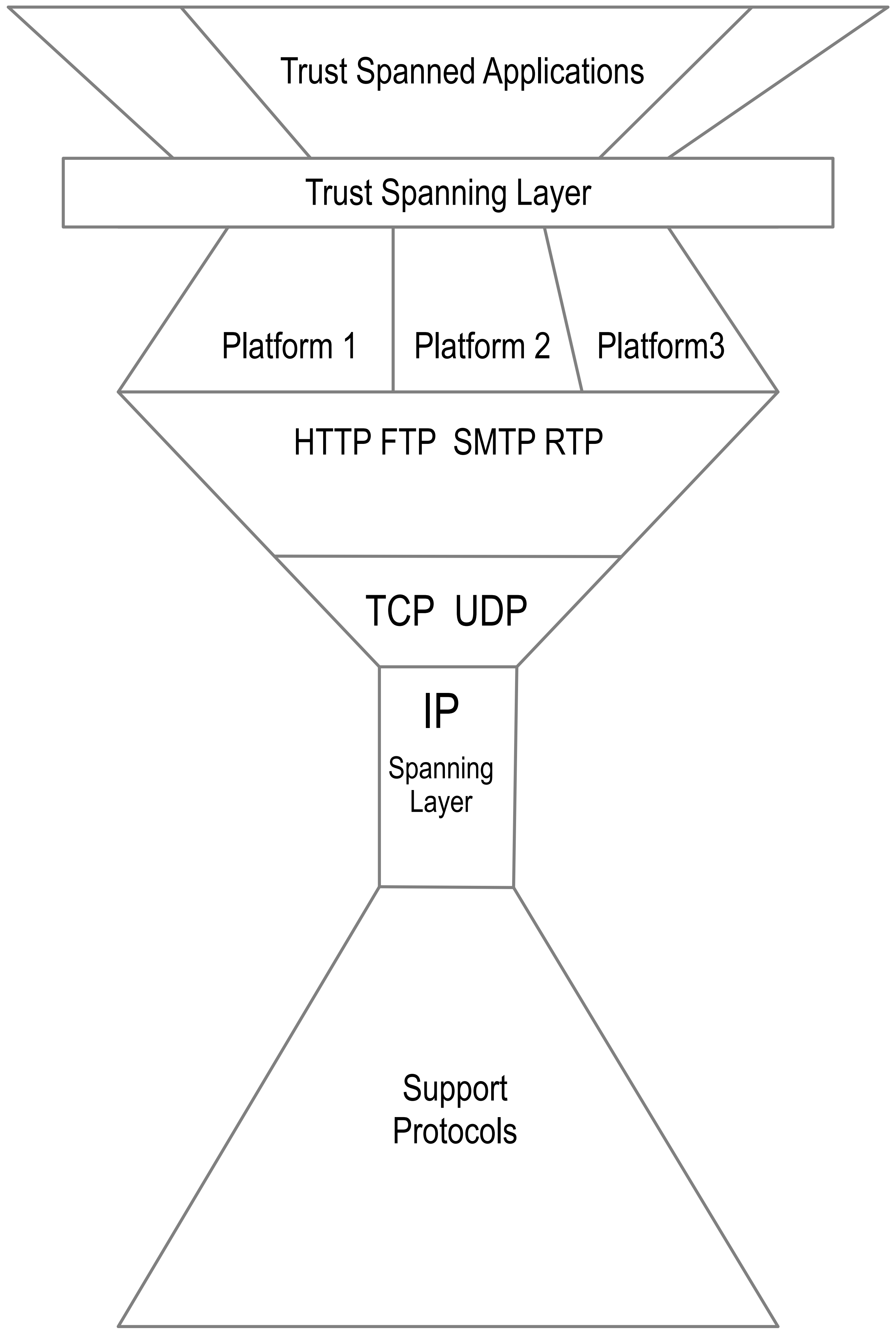
Support/Application 3

IP Spanning Layer

Support 1

Support 2

Support 3



Trust Spanned Applications

Trust Spanning Layer

Platform 1

Platform 2

Platform3

HTTP FTP SMTP RTP

TCP UDP

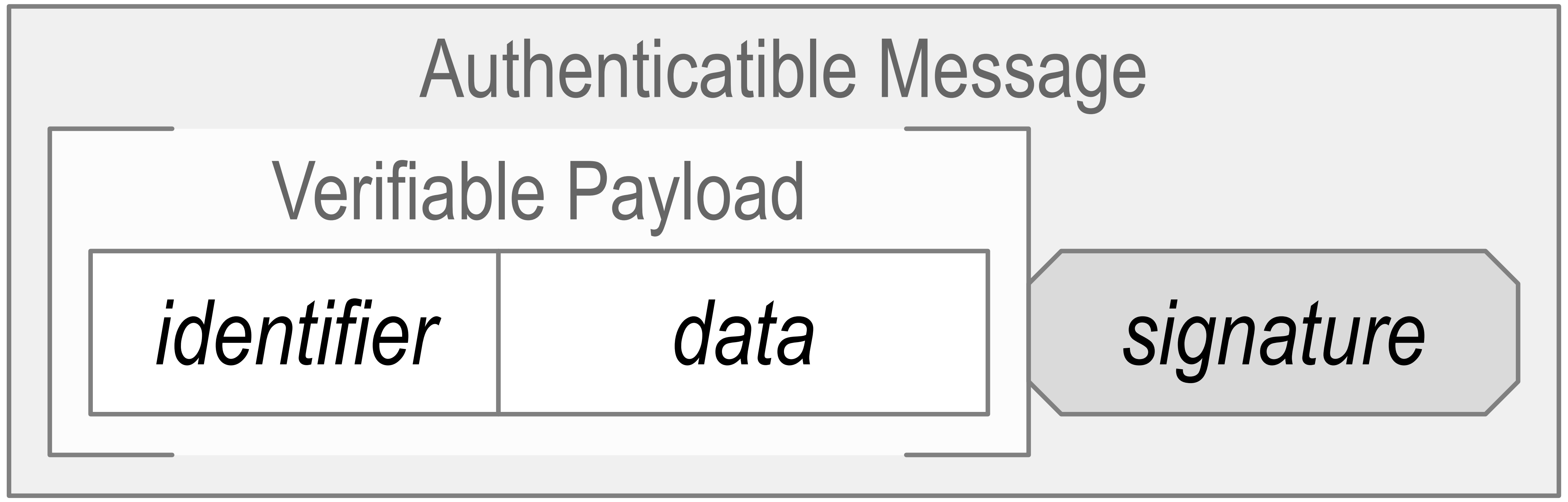
IP

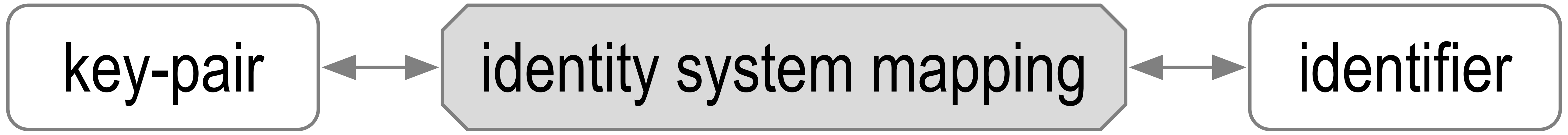
Spanning Layer

Support Protocols

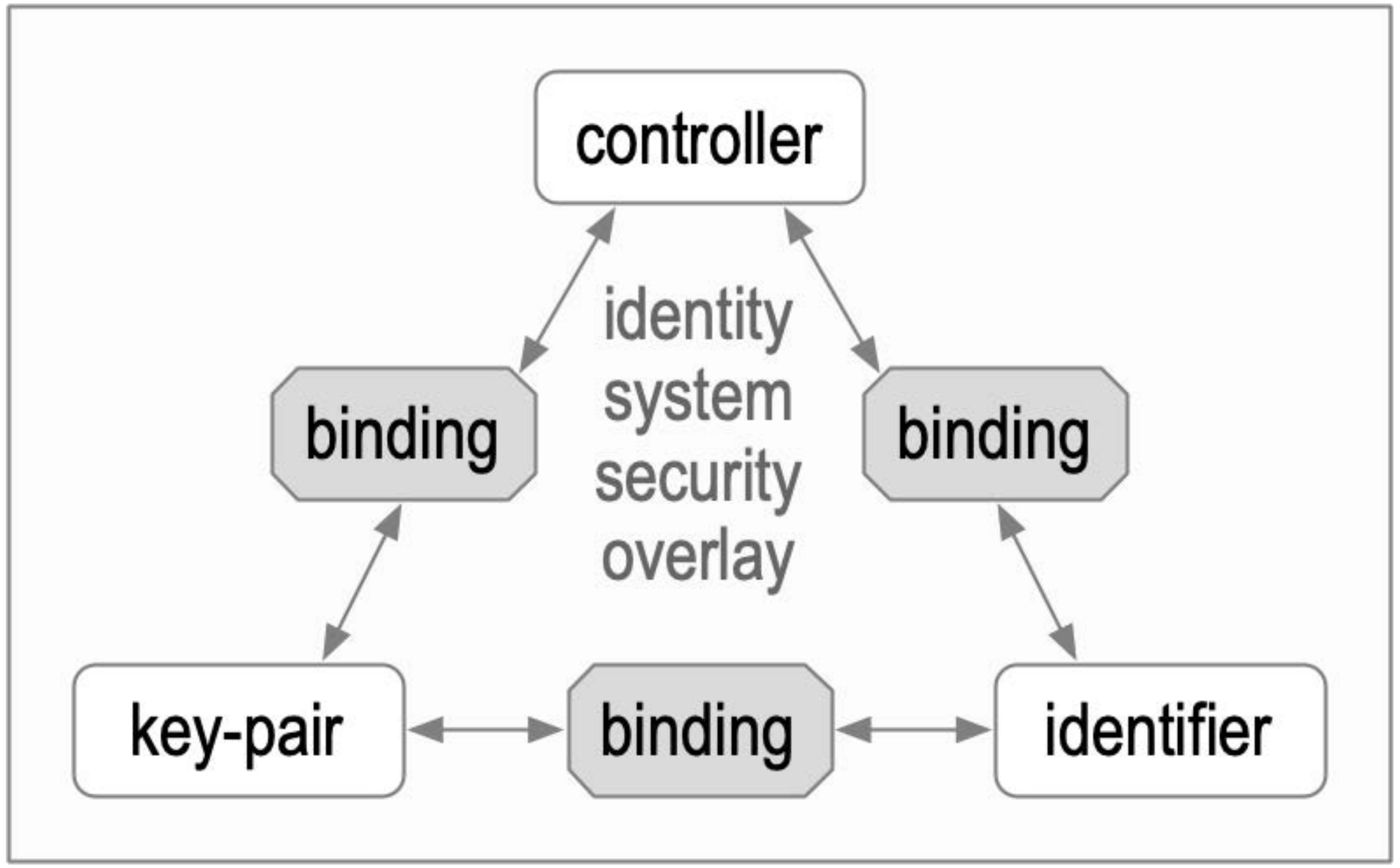
Identity System Security Overlay

Establish authenticity of IP packet's message payload.





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



Identifier Issuance

Questions