Distributed Delegated Mappings

draft-watson-dinrg-delmap-00

Sydney Li[1], Colin Man[2], Jean-Luc Watson[2]

DINRG - IETF 102

[1]Electronic Frontier Foundation, [2]Stanford University

Motivation

State of Delegated Mappings

Current delegated mapping systems have:

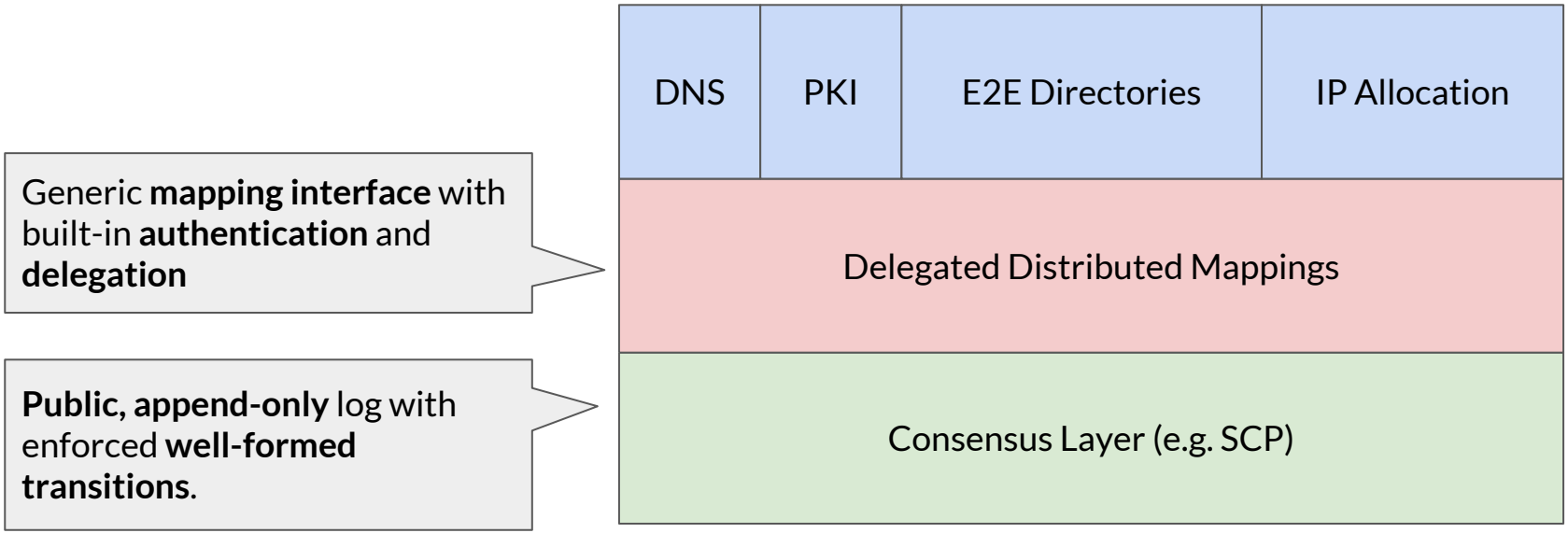
- Centralized trust
  - Often, with many trusted authorities that can maliciously rebind mappings.
- Unilateral revocation
  - Mappings can be revoked or overwritten by authorities without consent of the delegee.
- No common interface
  - Multiple systems exist to solve the problem of having **authenticated** delegated mappings in domain-specific ways.

Delegated Mapping Use Cases

System	Mapping	Problems
Domain Mappings (DNS/DNSSEC)	Domain → Zone	**Domain roots are trusted** and can remap existing entries.
Public Key Infrastructure (CA trust chains)	Domain → Certificate	**CAs are trusted** and can issue malicious certificates.
Web Security Policy (HSTS preload list)	Domain → Policy	**Not scalable**. Large lists. Policy tied to browser versions.
E2E encryption (Privately managed, non-transparent directories)	User → Key	**Provider is trusted** and can remap public key directories. Offline verification required to be secure.
IP Address Allocation	IP Address → Key	**AS trust each other**. Malicious AS can announce route that it does not own.

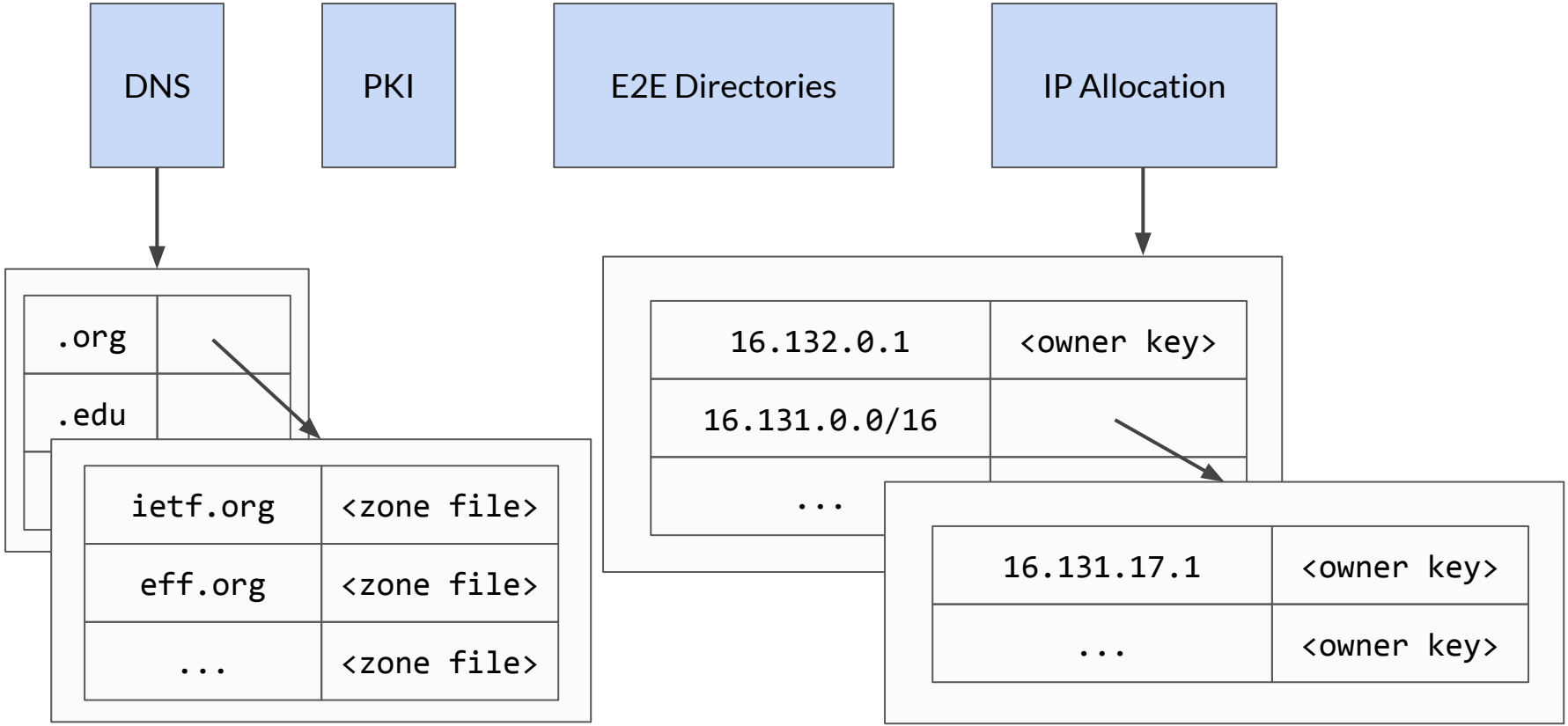
Generalized Mappings

Can we derive a scalable solution that will work for any mapping?



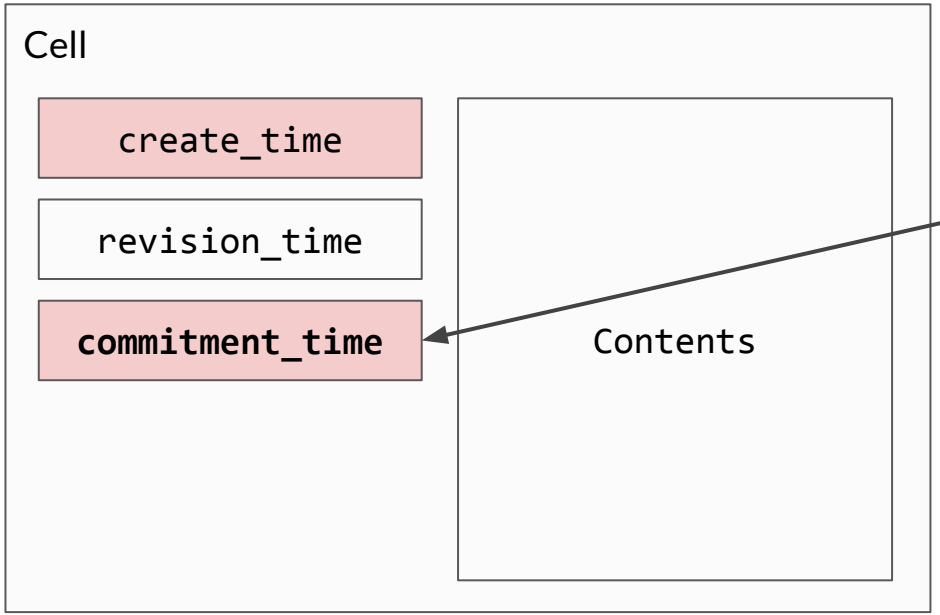
Delegated Mappings

Overview


DNS
PKI
E2E Directories
IP Allocation
.org
.edu
ietf.org
<zone file>
eff.org
<zone file>
...
<zone file>
16.132.0.1
<owner key>
16.131.0.0/16
...
16.131.17.1
<owner key>
...
<owner key>

Building Blocks

- `Cell` - basic record of delegation

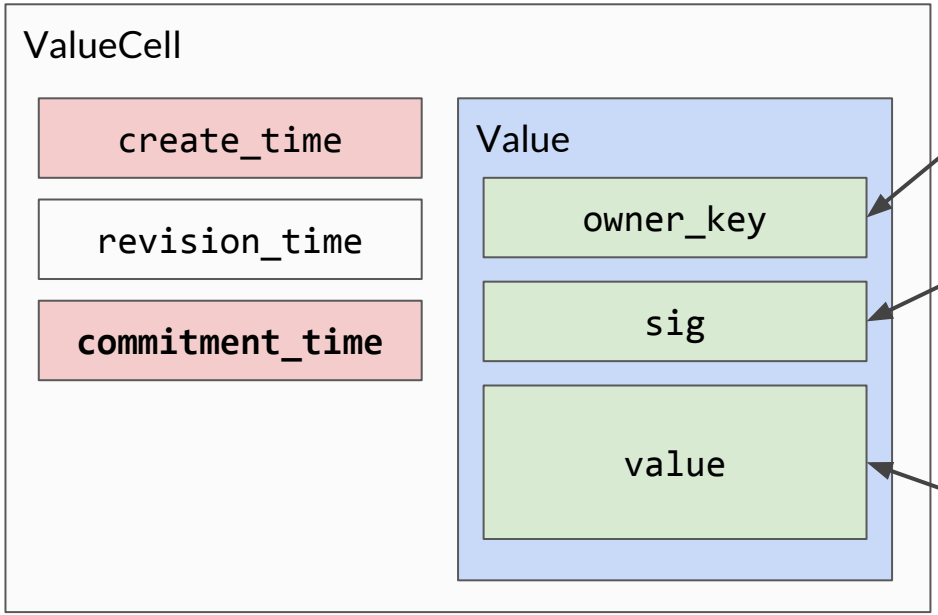


Guarantee from *delegator* to *delegee* that the mapping will remain valid until the `commitment_time`.

Protection from arbitrary delegator actions.

Building Blocks

- `ValueCell` - resolves lookups for individual values



Ties namespace value to owner public key.

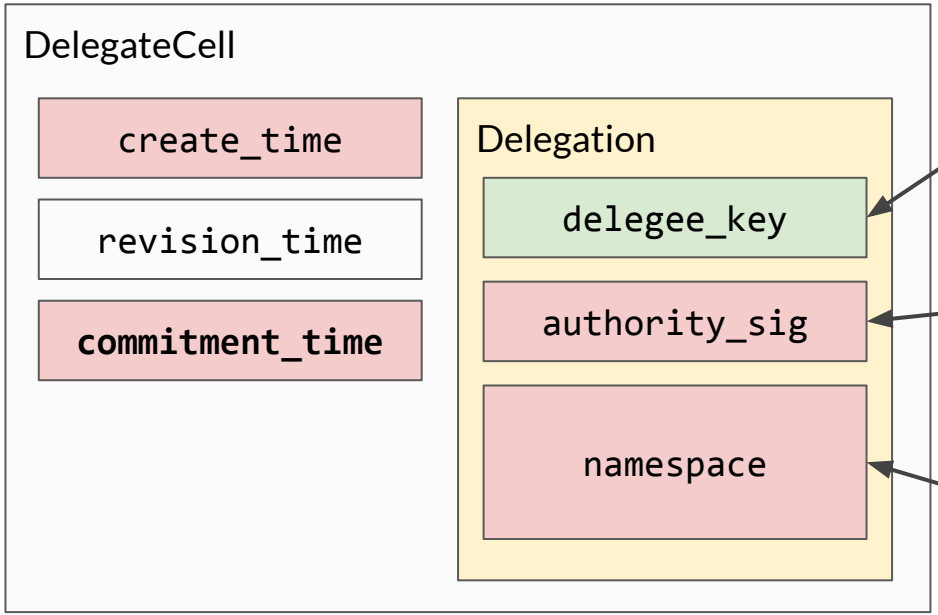
Signature of cell owner

Application-specific content:

- email encryption key
- domain zone file
- binary hash

Building Blocks

- `DelegateCell` - authorizes delegee to manage namespace



Links management of namespace *range* to delegee key.

Delegator signature

Specific namespace range:

- 192.0.2.0/24
- .com

Building Blocks

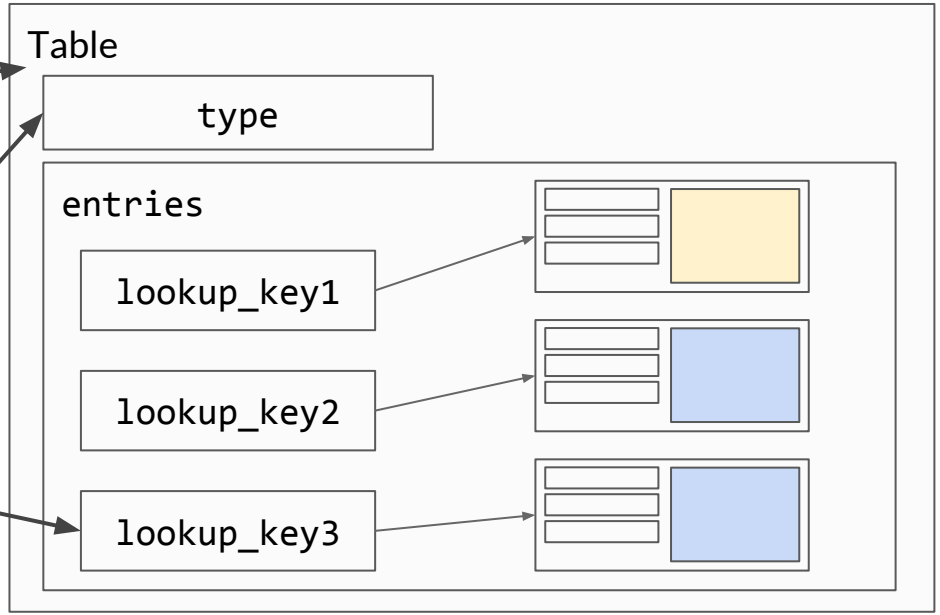
- `Table` - maps lookup keys in a namespace to individual cells

Delegee key controls delegations (i.e. table modifications) within the namespace range it controls

Determines delegation rules:

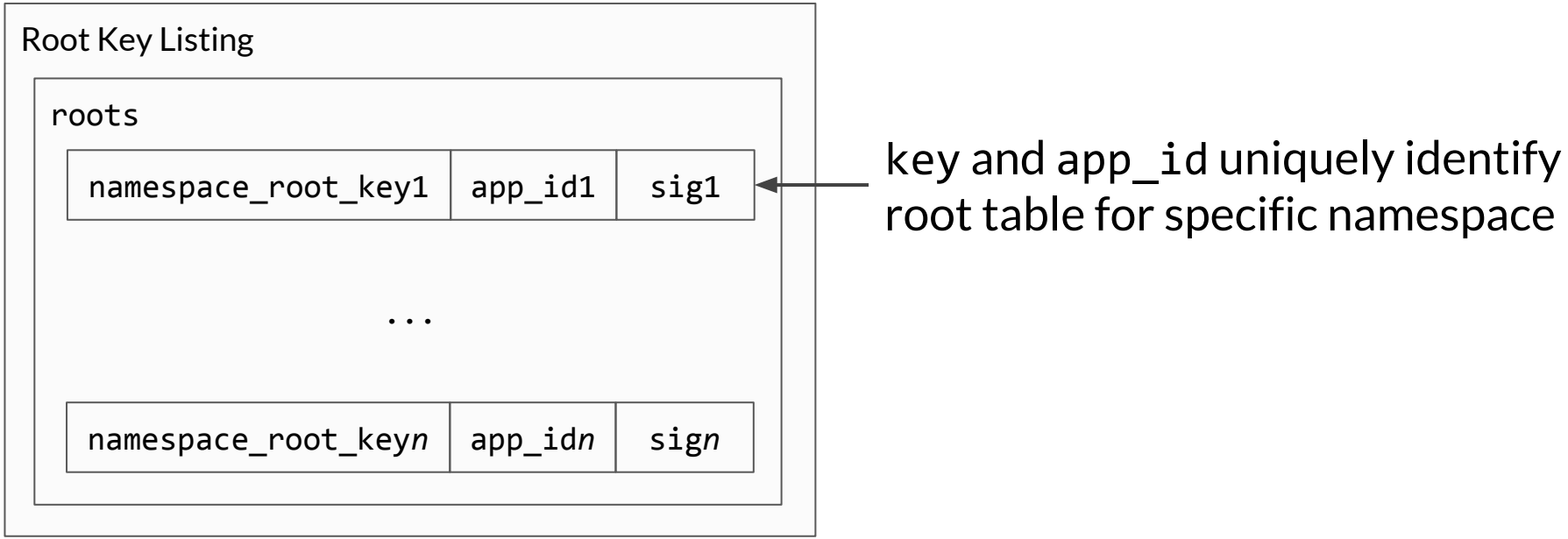
`PREFIX,SUFFIX`

Keys must be part of the table namespace



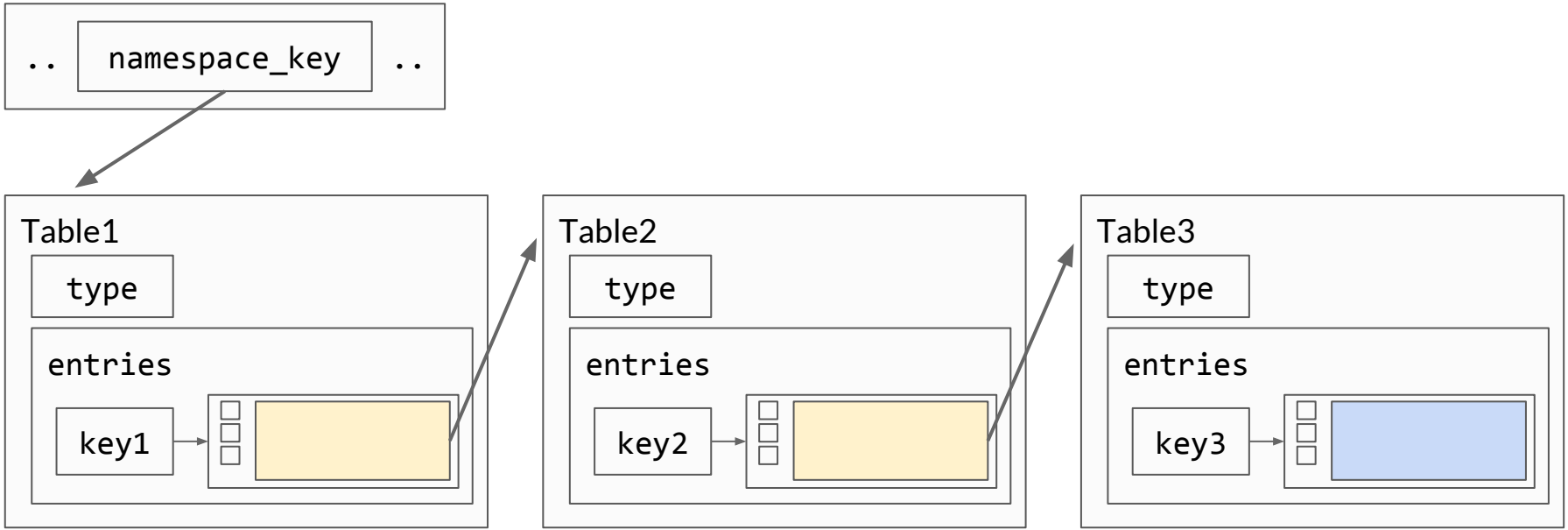
Building Blocks

- `Root Key Listing` - stores root key for each namespace



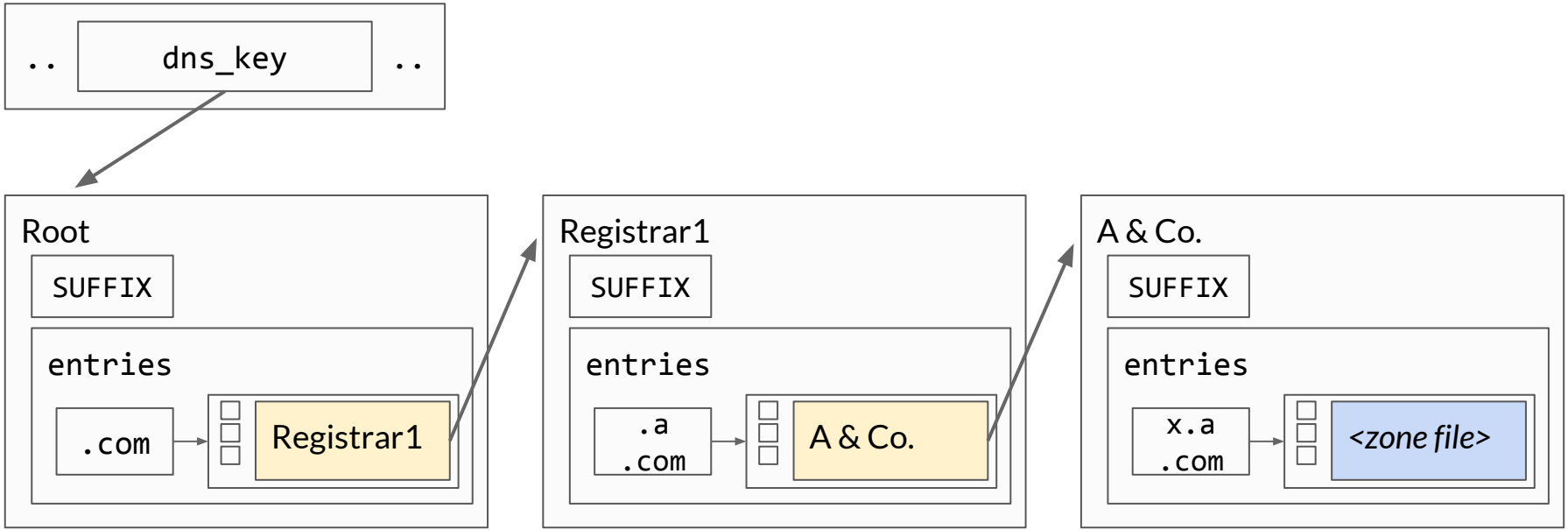
Structure

- The chain of delegated keys for a namespace links each table



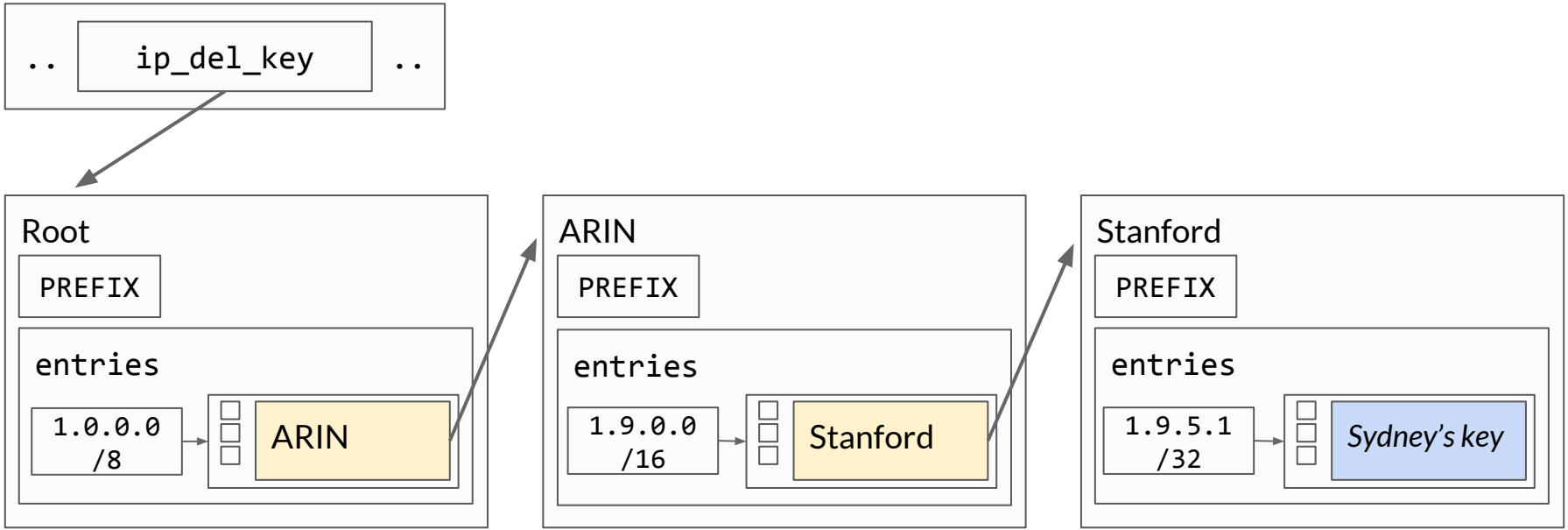
Structure

- **Domain Delegation**



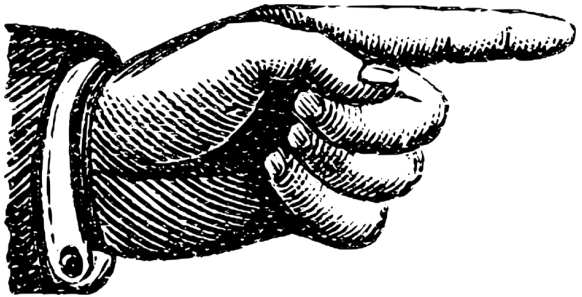
Structure

- **IP Prefix Delegation**



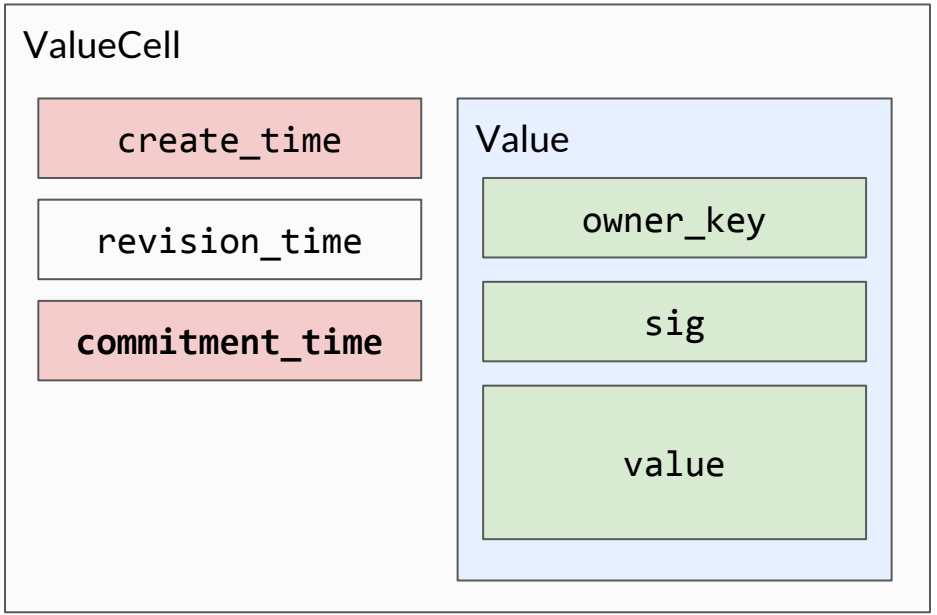
Common Operations

- **Creating a delegation**
  - The table authority inserts a new `DelegateCell` for the delegee in its table and creates a new table for the delegee
  - Proof of the additions is passed to the consensus algorithm
- Key rotation
- Lookup



Common Operations

- Recording a delegation
- **Key rotation**
  - The owner rotates their own key in the cell and signs with the old key
- Lookup



Common Operations

- Recording a delegation
- Key rotation
- **Lookup**
  - Read root key from Root Key Listing and find root table
  - Access table for delegation with the prefix/suffix match recursively
  - Return the contents of the desired `ValueCell` or `NULL`

Consensus

- Safety and consistency of the delegation tables must be provided by a consensus algorithm
  - Enforcement of table and delegation semantics
- Can use any consensus protocol (but not all algorithms are created equal)
- SCP
  - Proofs of table updates are batched and agreed to between SCP nodes
  - Authorized/correct modifications are determined by a distributed-mapping-specific *validity function* on each node.

Considerations

Considerations

Who controls the root namespace?

- DNSSEC?
  - IANA root keys
- A set of n authorities, or k of n authorities (threshold)?
- Vote amongst existing root authorities?

Considerations

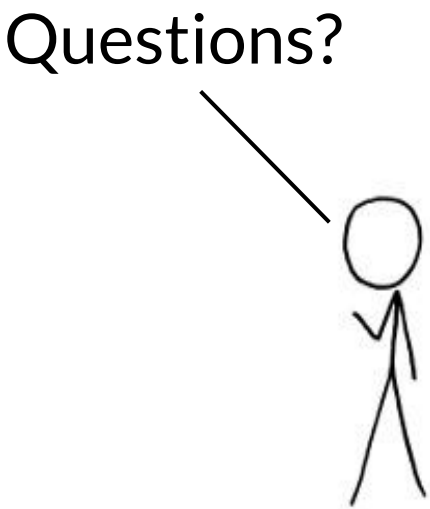
How do we prevent spam?

- Explicit delegation quotas in state machine
- Each delegator responsible for direct sub-delegees

Considerations

How do we scale?

- Who needs to run full consensus validation nodes?
- Scaling to per-user magnitude?



https://tools.ietf.org/id/draft-watson-dinrg-delmap-00.txt