# Towards a Distributed Multi-Signer Design

Steve Crocker

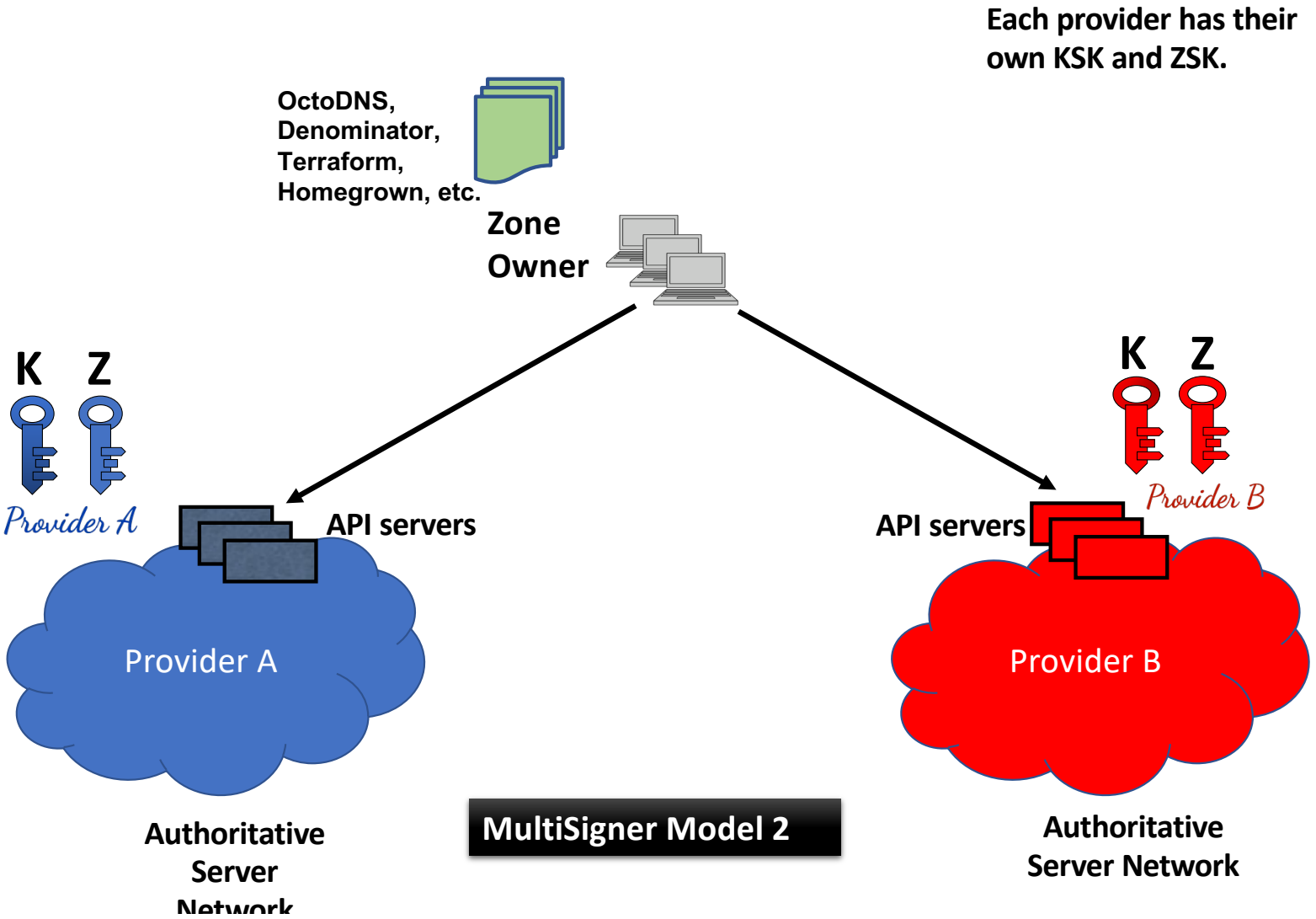
Edgemoor Research Institute

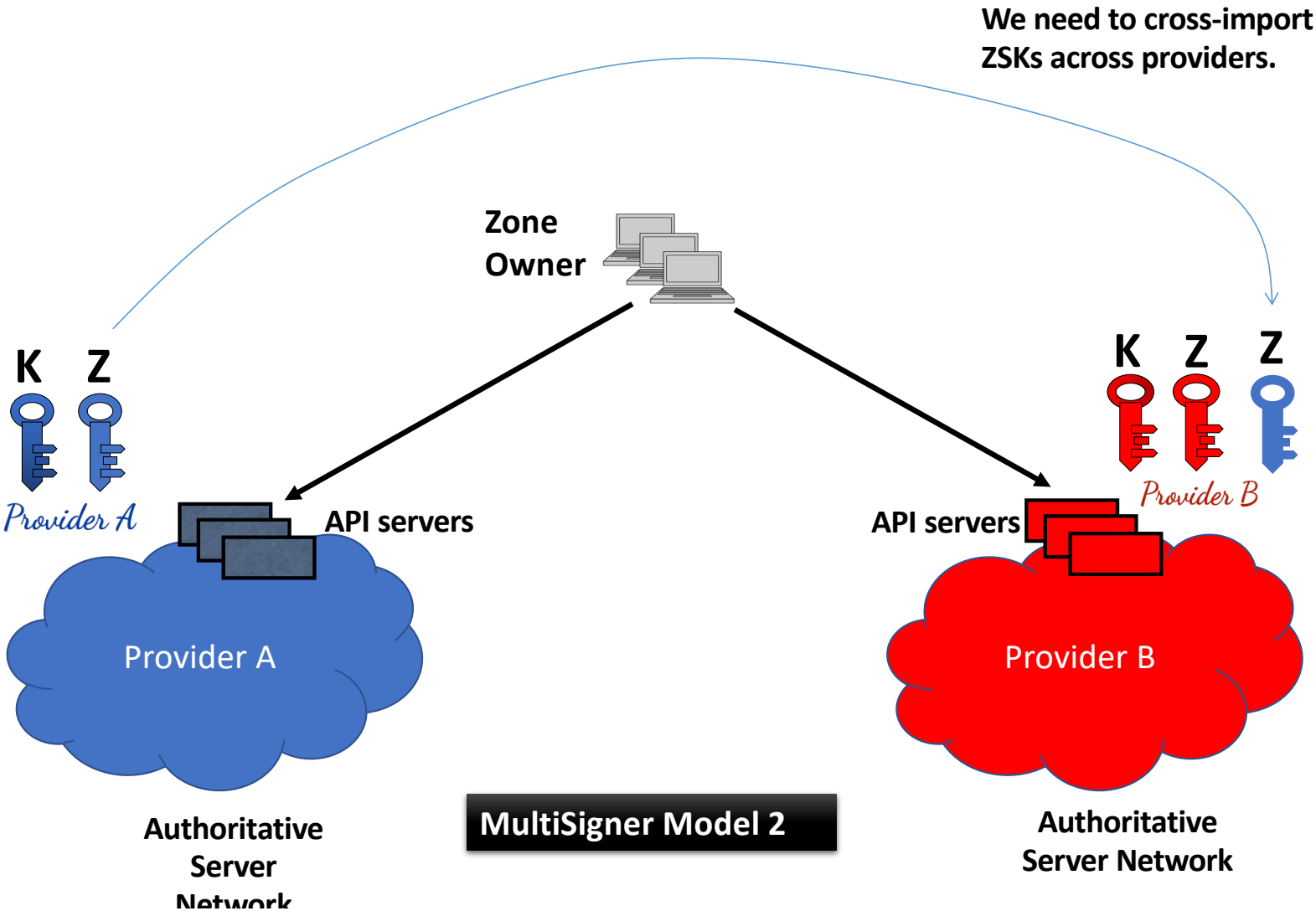
10 June 2024

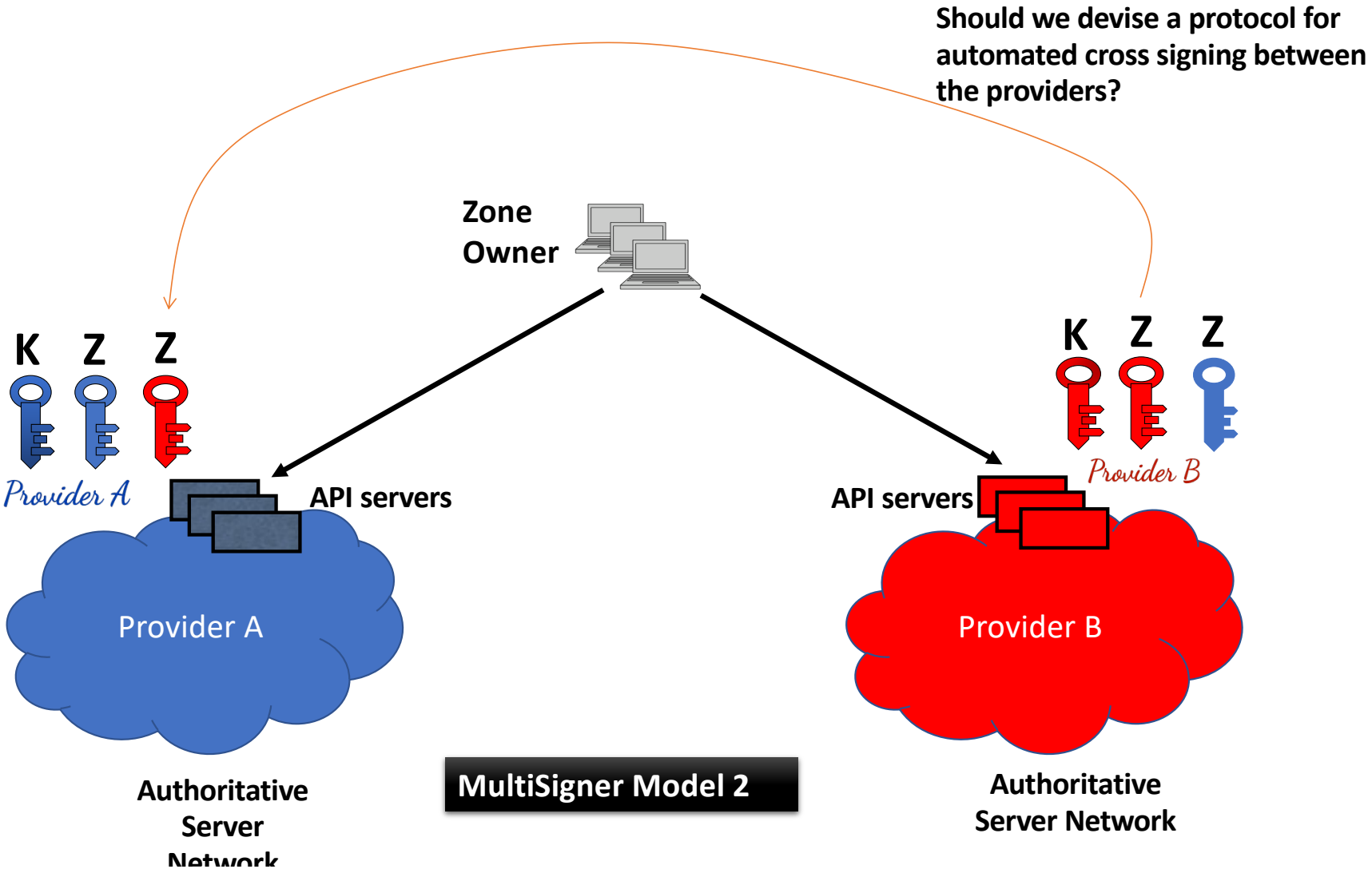
# Background

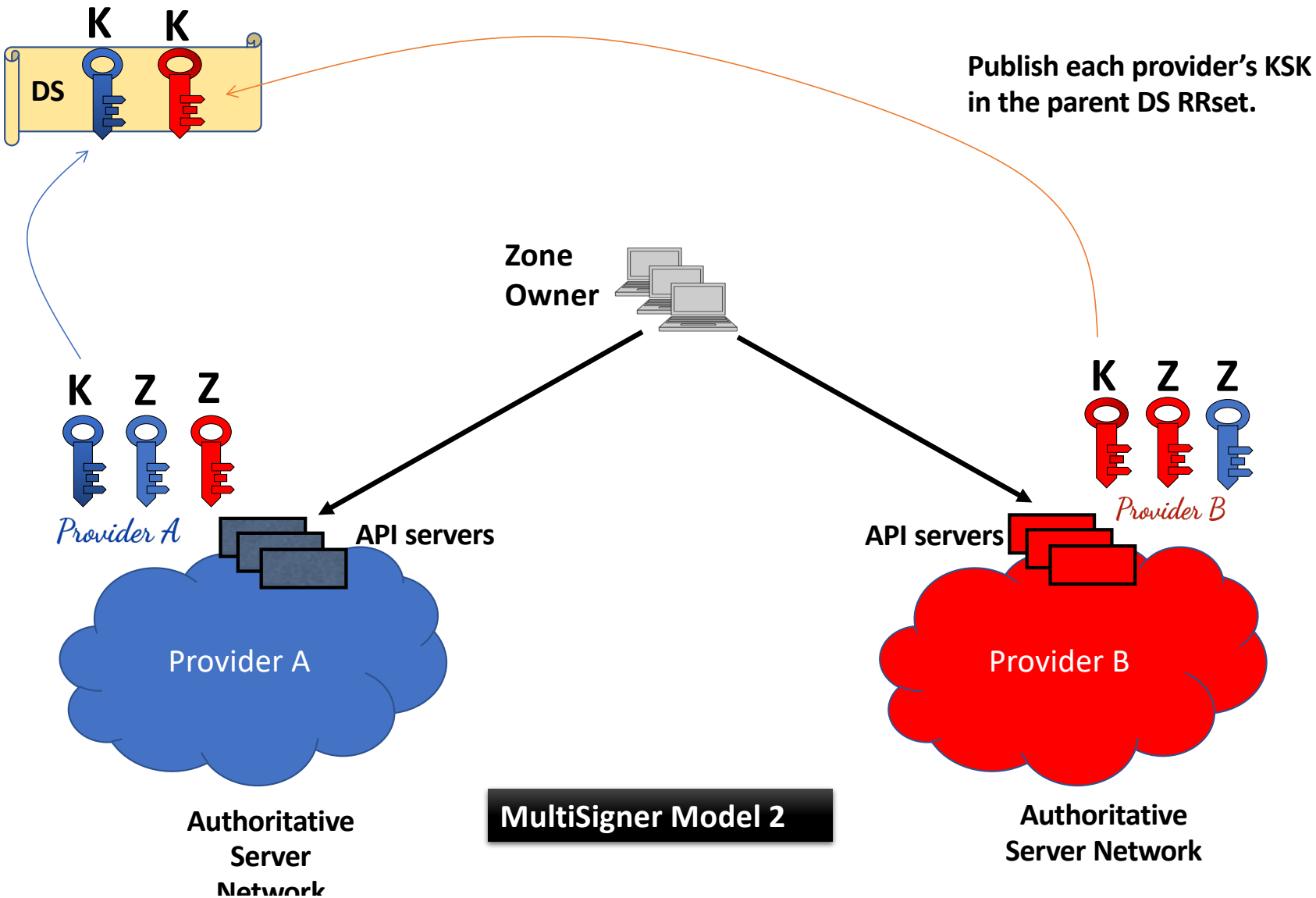
- Multi-signer => Two or more independent signers/servers of a zone
  - RFC 8901, model 2
- Lots of work toward implementation using a central controller
  - MUSIC

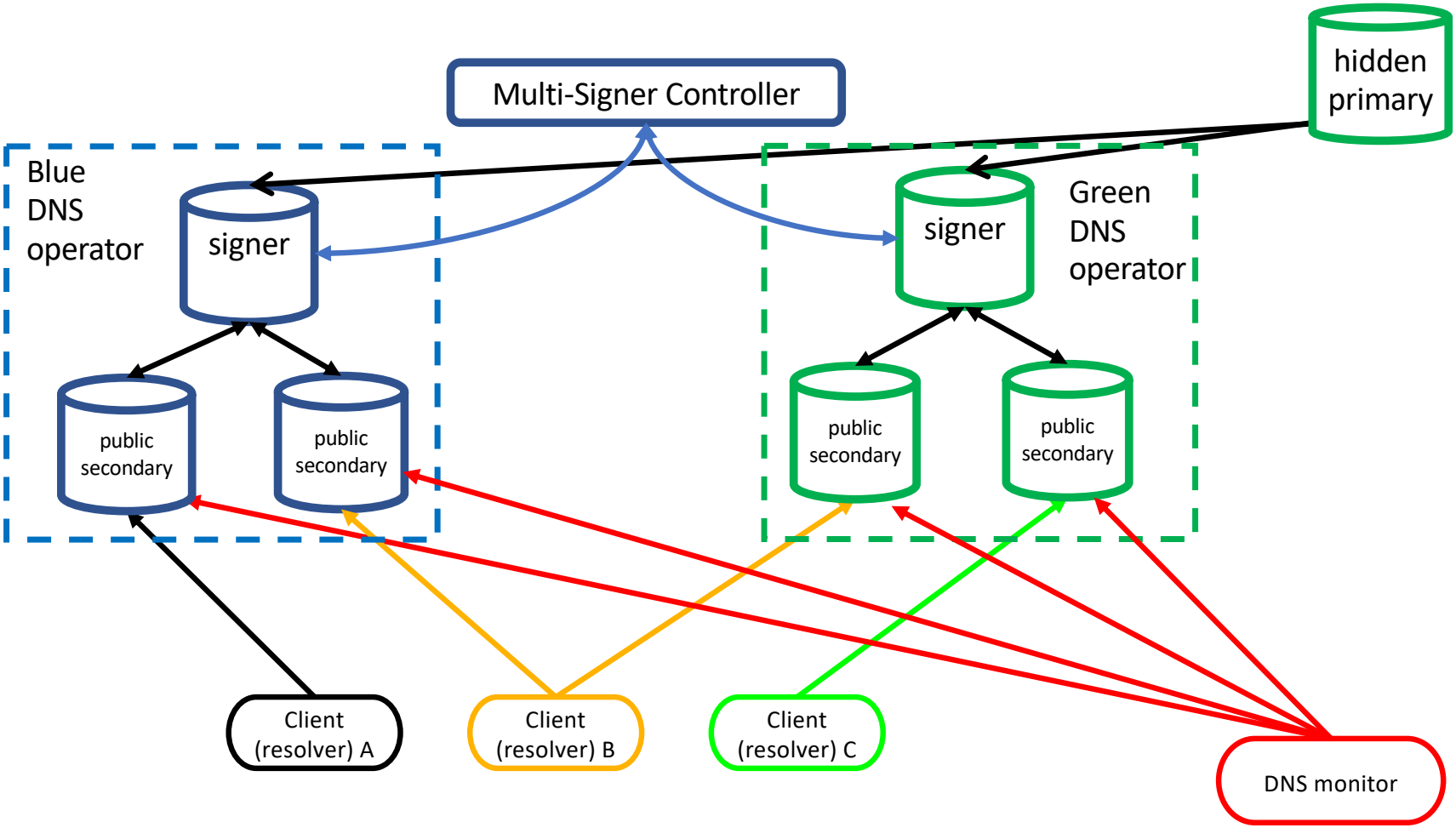

salesforce
Multi-Signer Model 2
• Unique KSK & ZSK per Provider


Each provider has their own KSK and ZSK.
OctoDNS, Denominator, Terraform, Homegrown, etc.
Zone Owner
K Z
Provider A
API servers
Provider A
K Z
Provider B
API servers
Provider B
Authoritative Server Network
MultiSigner Model 2
Authoritative Server Network


We need to cross-import
ZSKs across providers.
Zone
Owner
K
Z
Provider A
API servers
Provider A
Authoritative
Server
Network
K
Z
Z
Provider B
API servers
Provider B
Authoritative
Server Network
MultiSigner Model 2


Should we devise a protocol for automated cross signing between the providers?
Zone Owner
K Z Z
Provider A
API servers
Provider A
Authoritative Server Network
K Z Z
Provider B
API servers
Provider B
Authoritative Server Network
MultiSigner Model 2


K
K
DS
Publish each provider's KSK
in the parent DS RRset.
Zone
Owner
K
Z
Z
Provider A
API servers
Provider A
K
Z
Z
Provider B
API servers
Provider B
Authoritative
Server
Network
MultiSigner Model 2
Authoritative
Server Network


Multi-Signer Controller
hidden primary
Blue DNS operator
signer
public secondary
public secondary
Green DNS operator
signer
public secondary
public secondary
Client (resolver) A
Client (resolver) B
Client (resolver) C
DNS monitor

# The Insertion Problem

- Multi-signer protocol requires inserting keys from Signer A into Signer B and vice versa.
- Software packages and systems have evolved to permit this, but only if the “inserter” has permission.

# Multi-Signer Score Card

| **3 Mar 2022** | Designed | In Progress | Done |
|---|---|---|---|
| **Specifications** | ✓ | **draft-wisser-dnssec-automation** | **RFC 8901**<br>**RFC 8078** |
| **Multi-Signer Controller** | ✓ | ✓ | |
| **Name Server Software Capabilities** | ✓ | **Knot** | **PowerDNS, BIND** |
| **DNS Service Provider Capabilities** | ✓ | **NS1, UltraDNS, Cloudflare** | **deSEC** |
| **Documents** | | | |
| **Observation & Analysis** | ✓ | ✓ | |
| **Demonstrations** | | | |

# Name Server Software Capabilities

| **28 Aug 2023** | BIND | | | Knot 3.2.0 | | | PowerDNS | | | (Others TBD) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | D | R | C | D | R | C | D | R | C | D | R | C | D | R |
| Add DNSKEY records | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| Remove DNSKEY records | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| Add CDS/CDNSKEY records | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| Remove CDS/CDNSKEY records | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| Add CSYNC record | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| Remove CSYNC record | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | | |
| DS Polling/Checking | ❑ | | | ✓* | | | | | | | | | | | |

C = Command Line Interface – not usable

D = Dynamic DNS

R = Rest API

✓ **Complete**

❑ **In progress**

o **Planned but not started**

**Not Planned**

# DNS Service Provider Capabilities

| **12 February 2024** | deSEC | | | NS1 | | | UltraDNS | | | Cloudflare | | | (Others, e.g. DigiCert, Constellix, Akamai) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | D | R | C | D | R | C | D | R | C | D | R | C | D | R | C | D | R |
| Add DNSKEY records | | | ✓ | | | ✓ | | | ❑ | | | ✓ | | | | | | |
| Remove DNSKEY records | | | ✓ | | | ✓ | | | ❑ | | | ✓ | | | | | | |
| Add CDS/CDNSKEY records | | | ✓ | | | ❑ | | | ❑ | | | ❑ | | | | | | |
| Remove CDS/CDNSKEY records | | | ✓ | | | ❑ | | | ❑ | | | ❑ | | | | | | |
| Add CSYNC record | | | ✓ | | | ❑ | | | ❑ | | | o | | | | | | |
| Remove CSYNC record | | | ✓ | | | ❑ | | | ❑ | | | o | | | | | | |

C = Command Line Interface – not usable

D = Dynamic DNS

R = API configuration

✓ **Complete**

❑ **In progress**

o **Planned but not started**

**Not Planned**

# The Access Control Problem

- A centralized multi-signer controller must be configured with the necessary access.
  - The result is another layer of manual configuration

# Meanwhile, DS Updates…

- A DS update is required whenever keys are changed
- Multi-signer operation also requires a DS update whenever a new signer is added

- DS updates from 3rd party DSN Providers, i.e. not part of the registrar, are problematic.  No direct access to the parent.
- Scanning for CDS and CDNSKEY records has emerged as the main solution.
  - Not yet widely implemented
  - Scanning poses scaling and performance issues

# “Notify” to the Rescue

- As John Levine has described, RFC 1996 formalizes an extension to the Notify protocol
- Dramatically collapses the parent workload
- Dramatically decreases the delay in updating DS records

- But wait!  There’s more.

# When A wants B to get data from A's zone

- (Push) A can directly insert the data into B's zone
  - But only if it has access.
- (Pull) B can scan A's zone to see if there is new data it should know about.
- (Tickle) With Notify, A can notify B, and B can pull the data from A's zone.

- Notify solves both performance and access control problems.

# Revisiting Multi-signer Coordination Actions

**Adding a new signer**

- Initiated by Zone Owner
- Currently requires Zone Owner to insert list of nameservers associated with the Signer

**Changing keys**

- Initiated by Signer
- May not be – perhaps usually not – coordinated with Zone Owner activities

# Two ideas

- The Signer as a first-class citizen
  - The Signer is explicitly identified
  - The Signer knows and manages its nameservers
  - No need for the Zone Owner to interact with or know the names of the nameservers

- Distributed operation of the multi-signer protocol
  - Each Signer has access to its nameservers
  - Each Signer watches for changes in its zone
  - Each Signer receives Notifies from the other Signers
  - Each Signer retrieves records from the other Signers' zones

# The Connective Tissue

- Zone Owner (ZO) inserts info about the Signers
  1. ZO inserts Signer names.  Signers insert nameservers names
  2. Zone Owner inserts nameservers names
- Notify Targets specified by Signer
  1. Target details visible in the zone
  2. Targer details in a side zone

# Error Control

- How is error control handled in a distributed operation?
- Each Signer watches to see if expected changes occur
- If a time threshold is exceeded, notify an operator.
  - Need to work out the appropriate threshold
  - With Notify, the threshold should be fairly short, perhaps a few minutes.

# What’s Next?

- More detailed design
- Compare design options
- Trial implementation
- Stay tuned