


# DNS Automation: Catalog Zones

Aleksi Suhonen  
March 2025

A large, abstract red graphic is located in the bottom right corner of the slide. It consists of several rounded, overlapping shapes, with a white curved line segment visible within one of the red areas.

# What's This All About?

---

- Adding and removing zones on secondaries has to be done out-of-band
  - Especially if it's somebody else's server
- RFC9432 introduces a way to do this in-band
- Needs integration into existing processes and automation
- I'm offering secondaries on `ns.axu.fi` or `ns.trex.fi` as rewards :)

# Basic Concepts

---

- Catalog Zone: lists zones and related configuration
- Member Zone: one of the zones in the Catalog
- Member Label: unique deterministic placeholder for per-zone configuration

```
example-catz.invalid.  IN SOA ...  
version                IN TXT "2"  
masters               IN A 192.0.2.53  
MEMBERLABEL1.zones    IN PTR example.com.  
MEMBERLABEL2.zones    IN PTR example.net.  
masters.MASTERLABEL2.zones IN A 192.0.2.54
```

# Label Generation Example

---

```
#!/usr/bin/env python3
# printf "\7example\3com\0" | openssl sha1
import dns.name, hashlib, sys
print(hashlib.sha1(dns.name.from_text(sys.argv[1]).to_wire()).hexdigest() + \
      ".zones\tIN PTR\t" + sys.argv[1] + ".")
```

# Security Considerations

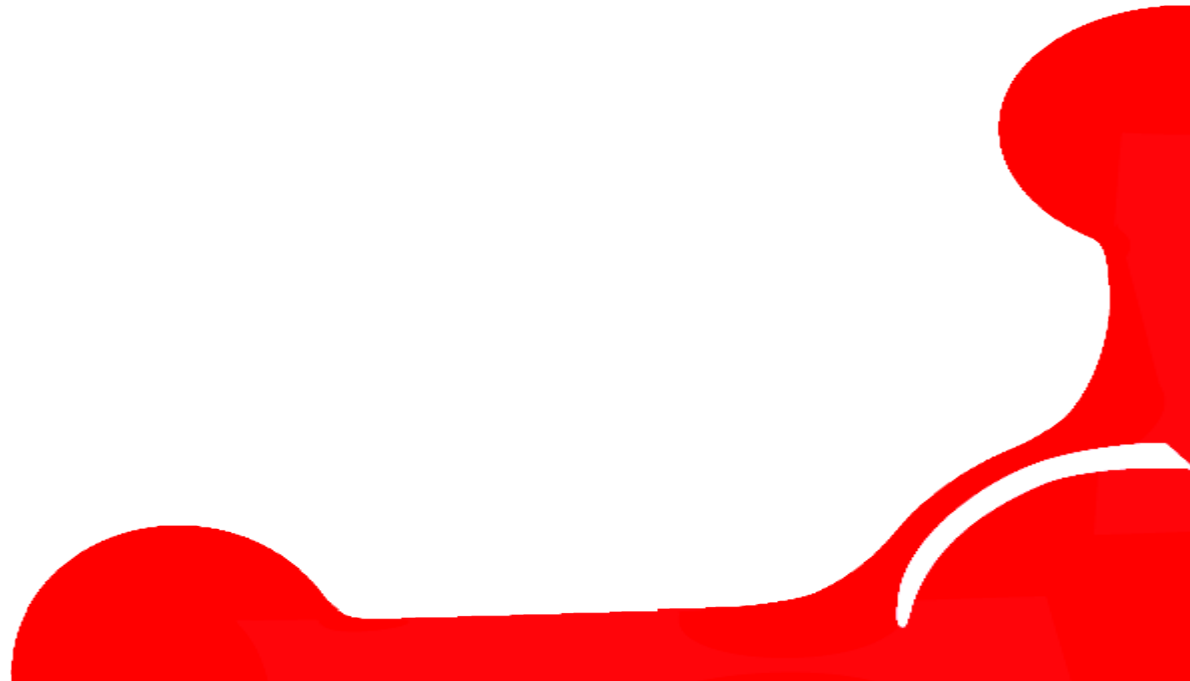
---

- TSIG or TLS for transfers
- Member Zone collisions: first come first served
- Catalog Zone should be “inaccessible”
  - Empty `allow-query` and `allow-transfer` on the secondaries
  - Under some non-global TLD such as `local.` or `invalid.`
  - Unique name to avoid collisions with other Catalog Zones

# Thank you!

---

Questions?



# Plan B

---

- Write Internet-Draft for extensions to RFC9432?
- I'm also interested in sharing experiences on and further developing abuse-prevention-filtering of open resolvers