

ccTLD Migration: Lessons Learned

Iliya Bazlyankov

Namingo Registry – *open source registry platform*

ICANN ccNSO Tech Day . 9 March 2026

The Reality of Some Small ccTLDs

Some small ccTLDs still operate in a very manual way.

- Zone files are edited directly in BIND.
- Changes arrive by email.
- Authorizations are checked manually.
- There is limited or no automation.

This model works at small scale.

But as volume grows or policies evolve, it becomes fragile.

Migration Goals

When planning a transition, the goals are usually practical:

- Introduce EPP for structured provisioning
- Implement RDAP for modern data access
- Maintain stable DNS and introduce DNSSEC where needed
- Preserve existing data
- Avoid disruption to registrars or end users
- Keep infrastructure costs reasonable

The key is evolution, not revolution.

Migration Pattern

A typical migration follows these steps:

1. Data extraction from zone files and existing records
2. Normalization of domain and contact data
3. Import into structured registry database
4. Activation of DNS automation and EPP access
5. Introduction of RDAP

This can be done incrementally without switching everything overnight.

Zone Files to Object Model

In the manual model, the zone file or a spreadsheet often serves as the source of truth.

In a structured registry model:

- Domain objects are stored in a database
- Nameservers are linked to domain objects
- Changes are validated before publication
- Zone files are generated automatically

This reduces human error and makes policy enforcement consistent.

Namingo Registry Overview

Namingo Registry is an open-source registry platform designed for small and medium registries.

It provides:

- Full EPP server
- RDAP server
- Web-based control panel
- Structured domain, contact, and host objects
- Modern database-backed architecture

The focus is simplicity, transparency, and operational resilience.

Operational Resilience

Structured registry workflow improves resilience:

- Role-based access control
- Full logging of all changes
- Automated DNS generation
- Easier backup and replication
- Cleaner separation between registry and registrar functions

It also reduces the “single person dependency” problem.

Infrastructure Requirements

One of the design goals was to avoid complex infrastructure requirements.

- Namingo Registry runs on Standard VPS with Ubuntu 24.04, PHP 8, MariaDB and Caddy.
- There is no requirement for Kubernetes, container orchestration, or proprietary dependencies.
- A typical small ccTLD deployment can run reliably on a modest VPS with proper backups and replication.
- This makes it accessible to registries with limited budgets or limited DevOps resources.

Design Principles

Namingo Registry was designed with several practical principles in mind:

- Runs on standard Linux environments such as Ubuntu 24.04 or Debian
- Deployable on any VPS without dependence on specific cloud providers
- Uses standard EPP protocol without mandatory proprietary extensions
- Avoids vendor lock-in and proprietary infrastructure components
- Built using widely known technologies such as PHP and MariaDB

This approach allows registries to maintain full control over their infrastructure while remaining compatible with standard registry workflows.

Namingo Registrar (1)

Namingo Registrar complements the registry side.

It allows:

- Acting as an accredited registrar
- EPP connectivity to registries
- Domain lifecycle management
- Registrar-level compliance workflows

This is useful for registries that also operate a retail channel or need a reference implementation.

Namingo Registrar (2)

For smaller ccTLDs, this can be helpful in two scenarios:

- Operating an internal registrar during transition
- Providing a lightweight registrar interface to local partners

It also helps test and validate EPP behavior from a real registrar perspective.

Partial Adoption

Not every registry wants a full replacement. In some cases, only specific components are needed:

- EPP server connected to existing database
- RDAP server layered on current infrastructure
- DNS management separated from provisioning
- Gradual replacement of legacy tools

The architecture allows adapting to existing registry environments rather than forcing a full rebuild.

Real-World Lessons Learned

From real deployments, several lessons stand out:

- Data cleanup takes longer than expected
- Policy documentation must match technical reality
- Registrar onboarding needs clear documentation
- Automation reduces long-term operational stress

Most importantly, small registries can modernize without enterprise-scale budgets.

Suitability

This approach is particularly suitable when:

- Registry size is modest
- Engineering team is small
- Budget constraints exist
- There is a need for ICANN-aligned standards
- There is desire to reduce operational risk

The focus is pragmatic modernization.

Conclusion

Migration from manual zone management to structured registry operations is achievable.

- It does not require a massive rebuild.
- It requires planning, phased migration, and careful data normalization.

Open-source tooling such as Namingo Registry can provide:

- EPP
- RDAP
- Structured workflows
- Operational resilience

The goal is stability, transparency, and long-term sustainability for ccTLD operations.

help@namingo.org

<https://namingo.org>

Discussion

- Questions
- Migration challenges
- Operational considerations