CODEV-NIC free registry software

Stéphane Bortzmeyer
AFNIC (".fr" registry)
bortzmeyer@nic.fr

2 march 2006

The target

The small or medium DNS registry for a TLD (Top-Level Domain)

Managing a TLD is possible even in a small country, with few resources. A lot of value for a small investment.

A registry is . . .

- A database,
- Tools to update and query it, from the registry and from outside,
- A few applications like the whois server, a DNS zone file generator. . .

The state of ccTLD

Many ccTLD in the world are a in very poor state :

1. No real Information System,
2. Often, no written registration rules and procedures,
3. No whois server,
4. No real database (Excel spreadsheet, paper), registration often manual,
5. Only a few dozens of registrations,
6. Expensive,
7. Name servers not fully operational.

CODEV-NIC tries to address #1, #3 and #4 and may be indirectly #5 and #6.

No fatality

A ccTLD is manageable with low-tech systems

Small machines and simple software

Set of requirments

1. Multi-policy,
2. Free software,
3. Easily managed from the registry or from outside (registrars, public),
4. Entirely automatic,
5. Co-developed (no outsourcing, real co-development).

Mostly for small and medium registries ("com" was not a target).

Multi-policy

Our most important requirment.

Every TLD has a different registration policy. Our vision is not one of uniformity. We want to be multi-policy.

Examples of differences :

- Direct selling vs. registry/registrar. If registrars, are they the only authority (for instance with contacts) or not ?
- IDN or not,
- Synchronous registration or not (papers required),
- Automatic expiration or explicit deletion ?
- Personal data protection.

Existing software

- OpenReg, ISC `http://www.isc.org/index.pl?/sw/openreg/` Very good software but :
  - Only one registration policy (basically the one of "org"),
  - No interface for the registry staff or for the ordinary user (you have to develop it as an EPP client or as a message bus component),
  - The only interface for the registrars is EPP, which I regard as completely unrealistic for most countries.
- SRS-NZ `http://sourceforge.net/projects/dnrs/`
  - Only one registration policy
  - Not maintained for general use
- registro.br : non free

The project

Four teams (three actually) in different countries. This is co-development : the best way to be sure the software is suitable for the users.

- NIC-CI (Ivory Coast)
- NIC-MG (Madagascar)
- AFNIC (France), manager.

Most of the money came from the french government.

The process

First, a one-month workshop (feb. 2005) to discuss, prototype, brainstorm.

Participants came from seven NIC : Haiti, Algeria, Ivory Coast, Gabon, Madagascar, Mauritania, France.

Then, development, with various groupware tools (Subversion, Request Tracker, mailing lists, IRC session).

Poor Internet connectivity was a big problem in Madagascar.

The Ivory Coast developers worked during a civil war.

Technical choices

- Python programming language : simplicity and readability,
- DBMS PostgreSQL : triggers, integrity constraints, stored procedures,
- Cheetah templating system (not only for the Web site but also for making the system multi-policy),
- Apache, mod_python and Vampire for the Web site.
- Docbook and Python's reST for the documentation.

Continous integration testing automatically, every day `http://www.martinfowler.com/articles/continuousIntegration.html`.

The configuration file

Technical and policy choices are there.

Most parameters are static : you cannot change them afterwards.

```
tld=example
idn = false
have_registrars= true
registrar_manages_contacts = true
```

The database

- Ambition : to be the unique reference for the TLD (no state in the zone file, for instance),
- Integrity constraints, as far as SQL goes (triggers and stored procedures),
- Not really portable (impossible with SQL),
- Use of Cheetah templates to implement multi-policy.

The library

Every access to the database (except read-only accesses) goes through the library.

One Python class per type of object : Domains, Contacts, Nameservers. . .

Hooks for the local customization.

The XML-RPC server

The only external interface is a XML-RPC server (an email interface is planned).

People outside of the registry (registrars, authorized users) can read and write in the database.

Aynchronous operations

For instance, transfers between registrars.

CODEV-NIC relies on an existing tool : Request Tracker.

That way, we do not have to reinvent the wheel. Request Tracker is easily interfacable, thanks to scrips.

Deployment

Operational today in Ivory Coast (starting in february 2006).

In Madagascar may be this year.

Current state

- Not enough deployments to be too assertive,
- Installation and deployment still very rough (no tarball, no `configure` script),
- Some pieces are missing like IDN support.

But it is free software, you are welcome to help.