Copenhagen, 14.12.2006
v0.2

**Attn.: Trilok Dabeesing/ICTA**

# Technical overview for the future Registry of the .MU TLD (MUNIC)

## Table of Contents

## Overview

This document describes in general terms the architecture of the .MU NIC Registry platform. The intent of this document is to define the scope of a proper operational/technical requirement specification and API for the new .MU NIC platform, in the form of an RFP for a new platform development project, or of existing platforms.

As agreed in the Terms of Reference, this document covers:

I. Review of .MU charters to extract requirements that may have an impact on the technical infrastructure

II. Technical proposal detailing the infrastructure required to implement the Registry with a 3-tier model, including
   1. Information and data flows between 3R
   2. Design of data model based on charter requirements, 3R model and associated organization logic, including registration/ administrative/ renewal constraints, taking into accounts such aspects as privacy issues ;
   3. Overall design of platform itself, including software protocols, location and operational aspects (registry and NS placement)

# I. Review of the .MU charter

## Requirements that may have an impact on the technical infrastructure

Technical review of charter

As the .MU charter (draft policy / 2005) is not finalized, in particular are missing: the technical parts of the policy, the technical specifications for registrars and the fee policy. However, a number of core ideas are clearly defined in the documents available, and in subsequent work, that allow us to establish a draft design of the technical platform.

Consequently, it is necessary to identify the requirements of the policy that may have an impact on the design.

These are:

1. Three tier model (Registry, Registrar, Registrant) ;
2. Respect of existing current practice and legislation on issues such as privacy (WHOIS service) ;
3. Admissible domain names / reserved terms ;
4. IDN support ;
5. Differentiated pricing models

Other points, while also included in the policy, do not have a direct impact on the technical aspects of the registry. For instance: copyright/ trademark issues, name conflicts etc... which are all covered by conflict resolution protocols (UDRP), though it may for example be necessary to record the reason for a domain being suspended.

### 1. Three tier model (3R)

This point is the most significant, as it dictates the architecture, not only of the infrastructure, but of the organization itself. A three tier system implies that there will be no direct contact with end users, which simplifies the requirements for staffing and support, as normally only professional, accredited registrars will be in daily contact with the registry.

While it is likely that certain points of the .MU charter may evolve over time, reflecting the evolution of legislation and of the Internet community, it is not expected that the 3 tier model (Registry - Registrar - Registrant) will be replaced by a 2 tier model (Registry, Registrant). Accordingly, the design described below assumes a 3 tier model, and does not make provisions for other models.

*Impact on the technical infrastructure:*

- no end-user interface requirements, other than WHOIS and possibly contact update ;
- provide customer provisioning interface to Registrars (Mail/HTTPS/SOAP and possibly EPP)

*Affected areas*:

data model and API.

## 2. Respect of existing current practice and legislation on issues such as privacy (WHOIS service)

This aspect is closely tied to the data model, but it should be taken into account as privacy laws in Mauritius may require personal information to be protected (cf. harvesting by spammers of WHOIS data), while other legislation may make it mandatory for commercial entities to be clearly identifiable by consumers purchasing on the Internet.

*Impact on the technical infrastructure:*

A mechanism will need to be implemented, by which registering entities will be able to choose whether or not their data should be made public, within the limitations of the law.

*Affected areas:*

data model, WHOIS service

## 3. Admissible domain names / reserved terms

Certain names and expressions might not be permissible for registration, for various reasons:

- offensive / obscene names
- names of towns or localities (Port Louis, Plaisance)
- generic names (car, house, ...)

This list needs to be maintained and updated.  Additionally, it will be necessary to determine in what cases it might be acceptable or necessary to register a name even though it appears on this reserver list (for instance, a business can document that they in fact own the trademark Port Louis, or a person is indeed named "Plaisance"), and in which cases this is not possible (definition of a concept of priority).

*Impact on the technical infrastructure:*

Support in the data model for the reserved list.

It may be required to implement registration processing in such a way as to offer immediate feedback on whether a domain name can be reserved or not, so as to avoid unnecessary processing of a domain registration request.

*Affected areas*:

data model, registration API.

## 4. IDN support

This point is of importance, as its scope is not obvious. Should IDN[1] support be implemented so that only those non-ASCII characters that appear in the French language be supported ? Or should other languages be supported, so that domain names in Chinese can be registered under .MU ?

Impact on the technical infrastructure:

Determine scope of IDN support, and decide what level will be provided, if any, in .MU domain names.

Affected areas: data model, API, policies.

5. Differentiated pricing models

As part of the charter, a minimum fixed price is determined, paid by registrars to the registry for registering domains. The actual breakdown of this fee is to be determined, with suggestions that part of the revenue derived be contributed to the USF (Universal Service Fund), but it may be desirable to offer volume discount for large registrars, as long as this does not adversely affect the finances of the Registry Service (RS).

*Impact on technical infrastructure*:

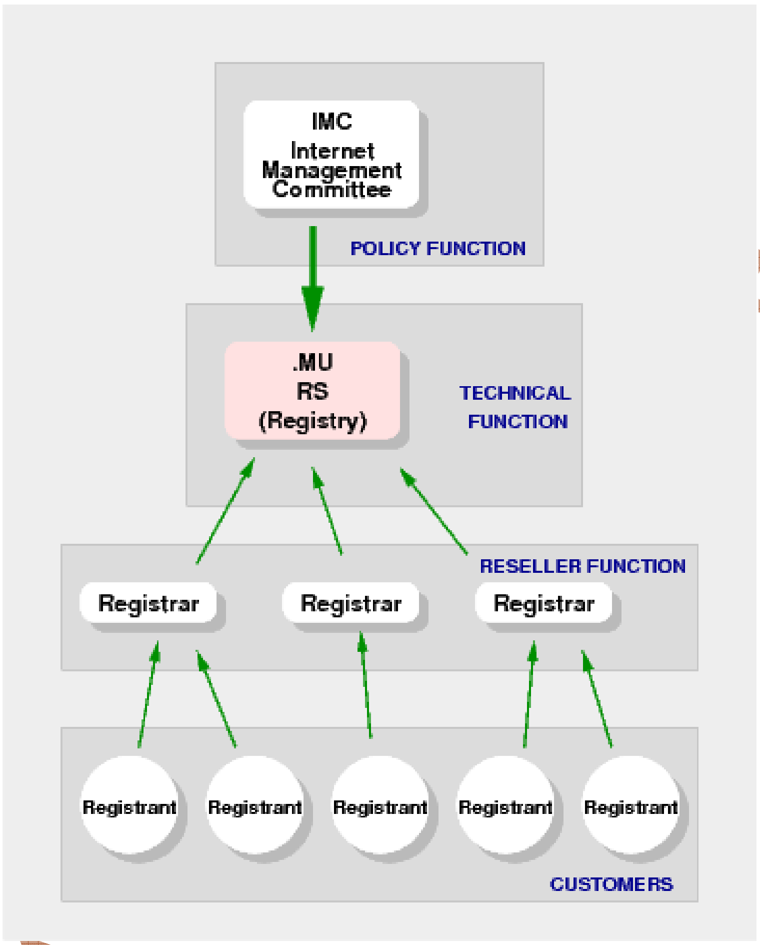
Take into account that pricing is not a fixed value at a given time, but a result of registration history (volume and rate) for each registrar.

*Affected areas*: data model

1 International Domain Names

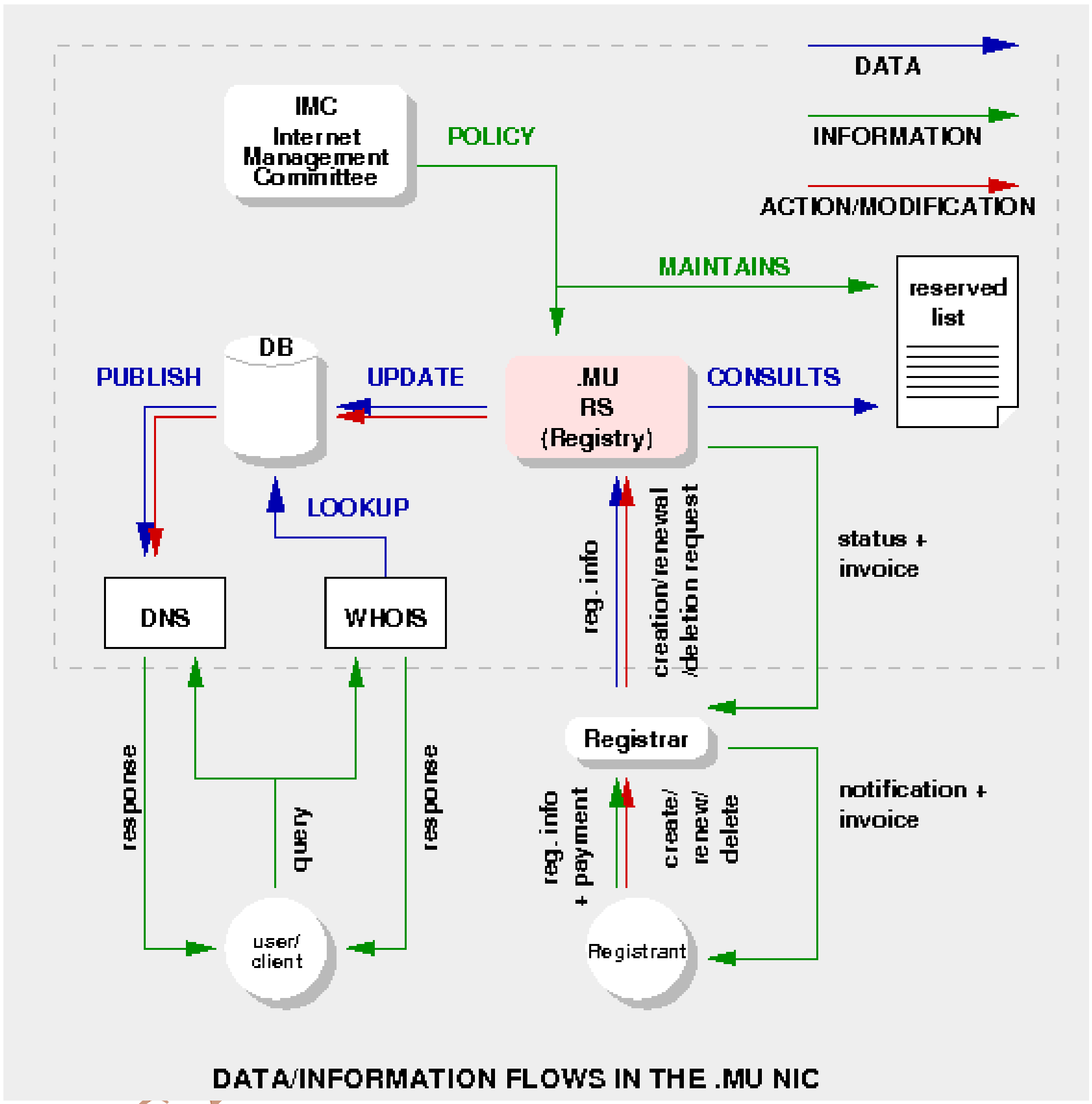
## II. Technical proposal

To put in perspective the technical proposal outlined below, it is necessary to illustrate the operational model of a 3-tier Registry:



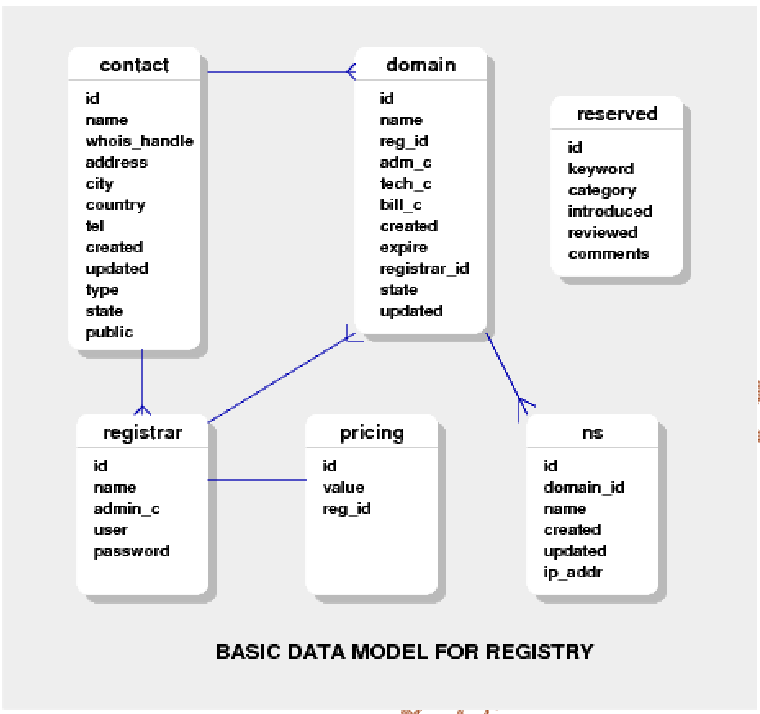
In such a model, no registrants (end-customers) are normally in direct communication/contact with the registry.

## Information and data flows



The arrows represent information and data flow between the different actors / components. This is not exhaustive, but gives an overview of the expected interaction between the different parties. It also gives a general overview of the technical components within the Registry.

## Data model



The schema above is a relational representation of the entities in the data model. Note that the schema is representational, and not fully normalized (i.e.: intermediate entities are not modeled).

The points listed below dictate operational aspects of the registry (work flow), as well as data constraints within the model (database). Together, these are labeled "business logic". As these points change over time, so will the business logic, to reflect the changes in policy, additional constraints, etc...

The model is based on the current following assumptions:

– all entities (private persons, companies, technical operators) are represented as contacts in the system. For each contact, a unique identified (the "handle") is attributed ;

– a domain belongs (is assigned) to a registrant, identified by a contact handle ;

– for all domains, there exists: a technical contact, an administrative contact, and a billing contact ;

– a domain is associated with one and only one registrar at a time ;

– to each domain, at least 2 name servers are associated ;

– registrar staff also exist as contacts ;

– pricing is a fixed base, with a possibility of discount for large volumes ;

– changes to the contact information can be performed by the contacts themselves, but through the registrar's interface ;

– contacts are created and maintained by registrars

## Journalization and auditing

Additionally, all events are recorded so that at any point in time it is possible to consult the history for a domain, and more generally facilitate audit of the platform. Events include:

– zone (domain) creation/suspension/deletion
– contact creation/deletion/modification
– change of ownership (change of registrant)
– transfer (from one registrar to the other)
– redelegation (change of NS)

## Constraints on registration

Registration of domains is submitted to the following constraints:

– AUP (Accepted Use Policy) is reported by registrar to have been accepted by registrant ;

– registrar had ensured that registrant has provided valid information ;

– domain name is not on list of reserved terms ;

– registrant contact is not blocked (outstanding issues) ;

– registrar has a valid (positive) credit status with the registry (does not owe money) ;

– delegation functions (name servers listed provide authoritative name service for the concerned domain) ;

## Transfer of domains

Domain transfers are submitted to the following constraints:

– fee has been paid by registrar for domain ;

– no disputes exist on the domain that require resolution (intellectual property or otherwise)

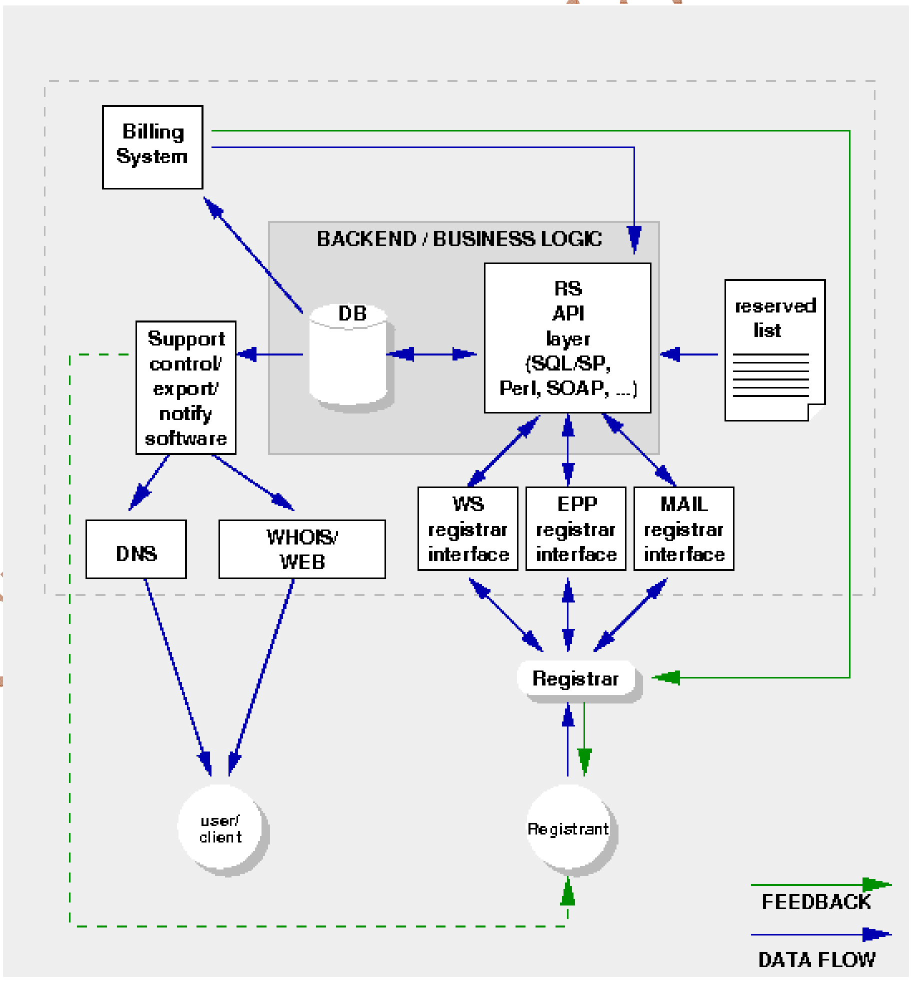
## SLD management

As of today, a number of SLD (Second Level Domain) are reserved under .MU for specific categories of domains or public service. For instance, .GOV.MU exists for naming government specific resources and institutions on the Internet (passport.gov.mu, mof.gov.mu), and .AC.MU for domains belonging to the academic sector. Other SLDs may be reserved in the future.

Typically, the administration of such SLDs will be delegated to a single organization, which will be responsible for ensuring that entities applying for domain names within the SLDs meet the criteria. Such SLDs do not differ from any other registered domain, but they appear as "special" domains in the registry system, in that they cannot expire, for example.

## Overall design

Taking into consideration the above points, below is a “big picture” design of the registry from a technical point of view:



## Functions, software and protocols (API)

I. Functional modules

The registry functionality is logically split into a number of functional modules, which constitute as a whole the “Registry Services”. Implementation wise, these modules need not necessarily be distinct, but are typically so in most registry systems.

These modules are:

- Storage and business logic (BACKEND)

  The Storage and business logic module, or back end:

  - stores the registry data (zones, contacts, registrars, states, ...) in a secure and easily retrievable fashion
  - ensures that changes to the data are done atomically (transactions), in a consistent way (integrity constraints), and only by the authorized entities (authentication, authorization)

- DNS service (DNS)

  The DNS service provides the public with authoritative information on the zones registered, with goals of high availability and fast response

- WHOIS service (WHOIS)

  The WHOIS service provides the public with up-to-date information about the objects residing in the back end, including contacts, zones and name servers

- API / Provisioning service

  The provisioning service implements one or more APIs (sub-modules) which allow registrars and internal Registry services to provision the TLD with objects (zones, contacts and name servers primarily). Possible API submodules are:

  - EPP interface (RFC 3730, XML)
  - Web interface (HTTP/S, SOAP)
  - Mail interface (PGP or S/MIME/X.509 authenticated email)

- Billing system

  The billing system interfaces with the back end, gathers transactions such as creations, transfers, renewals carried out by each Registrar, and facilitates invoicing, taking into account any discounts that have been agreed with a given registrar. Optionally, can manage Registrar accounts for prepaid registration of domains.

## II. Components

On a technical level, the functional modules are implemented by individual software components. Listed below are typical software component examples for providing implementing the functional modules defined above.

- RDBMS (PostgreSQL, Oracle, ...)

  A relational database management system is traditionally used to implement the storage and business logic modules of a registry. The DB is used to store information regarding zones, contacts, name servers, as well as state transitions and current state for each object. An SQL92 compliant database with ACID support (Atomicity, Consistency, Isolation, and Durability), including stored procedures, transactions, triggers, views, constraints and foreign keys. This is to ensure that consistency of the data is maintained at the closest level to the storage as possible, and avoid corruption or invalid data structures resulting from faulty programming or aborted procedures.

- DNS (BIND, NSD)

  For serving DNS, either BIND (Berkeley Internet Name Domain from ISC[1] or NSD from NLNet Labs[2]) can be deployed. Both name servers support all the required features for running a modern Registry, including Ipv4/IPv6, advanced access control, replication, support for zone and record signing with DNSSEC and TSIG for authenticated queries).

- WHOIS (custom or modified available)

  The server implementing the WHOIS service is closely tied to the back end, in that it must query the DB to obtain and show the data to the user. Most likely the WHOIS service will be implemented from zero. Eventually, code can be reused from existing WHOIS servers, such as GNU jwhois[3] or swhoisd[4].

- API / Provisioning service (custom developed)

  The provisioning service will be a set of programs/modules that implement different interfaces (EPP, SOAP/HTTP, Mail), allowing the Registrars and internal tools to communicate with the back end, for modification or query purposes.

- Billing system (existing system with import/export mechanism)

  The billing system can either be a part of the Registry Services, or administered outside, exchanging data with the Registry via CSV or XML formatted data on a scheduled basis (daily, weekly, monthly, ...)

- Misc. control / export software

  Other software components, linking the functions together are required:

1 http://www.isc.org/index.pl?/sw/bind/
2 http://www.nlnetlabs.nl/nsd/
3 http://www.gnu.org/software/jwhois/
4 http://dan.drydog.com/swhoisd/

- zone creator and exporter (Back end -> DNS)
- statistics gatherers and reporters (Back end -> management)
- notification and reminder generators (Back end -> Registrars and Registrants), for notifying and reminding of creation, renewal, impending deletion, etc...

These components are typically developed in high level languages, and use the predefined back end APIs for the task, for example Stored Procedures in the DB, or SOAP, Perl and similar.

III. APIs and UIs

The following APIs and Uis must be defined for each "boundary" between two modules or actors:

| Boundary | API / UI | Type |
|---|---|---|
| Registrant – Registrar | Human - System | Registrar defined (Web form, ...) |
| Registrar – Registry (FE) | System – System (EXT) | EPP, SOAP/HTTP, Mail |
| Registry – Back end | System – System (INT) | SQL Stored procedures, SOAP, Perl, PHP, ... |
| WHOIS – Back end | System – System (INT) | SQL Stored procedures, SOAP, Perl, PHP, ... |
| User - WHOIS (FE) | Human – System | Web, WHOIS (tcp/43) |
| Support tools – Back end | System – System | SQL, Stored procedures, SOAP, Perl, PHP, ... |
| Back end – Billing system | System – System | Data export (CSV/XML) |
| Back end – Contacts | System – Human | Email, RSS, ... |

FE = Front End

## Software candidates

As the management of ccTLDs is not a very large market, with wide differences in size (less than a hundred SLDs to several million) and deployment periods (end of 1980s to beginning of 2000s), most TLD registries ended up developing their own registry software. However, there exists today a number of freely available Registry management software, in various stages of maintenance:

- OpenReg, developed by the ISC: http://www.isc.org/index.pl?/sw/openreg/
- tinyReg, coordinated by NSRC: http://nsrc.org/tinyReg/
- CODEV-NIC, coordinated by AFNIC (no official page)
- Registro.BR (Brazilian NIC's software) (no official page)

### *OpenReg*

OpenReg, developed by the Internet Software Consortium, is freely available under a open source license. It is unknown which ccTLDs use OpenReg today, if any. OpenReg is primarily developed in Perl, and uses the Open Source PostgreSQL database as the back-end/logic component. Development status seems to be halted, but Uniforum SA (which administrates .co.za) have picked up and improved the code, though it has not been re-released at this stage.

### ***TinyReg***

tinyReg, based on domreg, is a project of the NSRC and University of Oregon, funded by the American National Science Foundation. From tinyReg's description[1]:

```
"domreg is a very lightweight domain registration and management
system, to handle new registrations, modify existing or delete
existing domains. With a web interface for registrants, and
web/email interface for registrars, it provides a simple but
complete system for managing outstanding/in-process registrations.

The goal is a minimal but fully-functional, extremely robust, and
exhaustively error-checked system using all open source
components."
```

TinyReg is very lightweight, and simple to implement. At first glance it does not appear to be very modular, but is written in Perl and relatively straightforward to customize / extend.

### ***Codev-NIC***

CODEV-NIC has the advantage that it supports both 2-tier and 3-tier Registry architectures (with and without registrars). On the downside, it is not currently deployed in production today, and it is not clear how much work remains to be done for the software to be production ready. Codev-NIC is primarily developed in Python, and uses the PostgreSQL database as the back-end/logic component.

### ***Registro.BR***

Finally, the Brazilian software, Registro.BR, developed for managing the .BR ccTLD, is normally available for free, with full source code, under a non-redistribute license[2]. However, being written in C++, it demands a strong working knowledge of the language to be able to tailor the system to specific needs.

Once specific technical requirements are established in the form of a technical RFP, the platforms listed above should be evaluated to establish if one of them could qualify for the .MU NIC, and at which cost (customization).

1http://nsrc.org/tinyReg/docs/extras/domreg-what.txt
2 http://ws.edu.isoc.org/workshops/2004/ccTLD-bkk/day3/registro_br.pdf

## Operational aspects

As a starting point, it is recommended to base requirements on those listed in RFC2870[1], "*Root Name Server Operational Requirements*".

Key points to consider:

▪ *Security*

The physical security of the premises, as well as the data security of the systems implementing the Registry Services must be guaranteed, including:

- limited or no access for non-Registry personnel to Registry premises and hosting environment ;
- regular encrypted AND signed data backup, with off-site transport of backup media and/or replication to a remote site ;
- regular audit of database integrity to ensure consistency of the data representing the TLD.

▪ *Location: registry and name server placement*

The registry itself should be located in Mauritius, most likely in Port Louis, where the availability of reliable network connectivity, power, and secure building facilities are common. At most ½ of rack space (20U) will be necessary to host the registry hardware. While a dedicated physical location is not strictly necessary (i.e. The existing co location facilities of an ISP will be adequate), it might for reasons of neutrality and physical security be recommended to establish a dedicate hosting facility. Any premise of at least 4 x 4 meters with secured physical access, reliable power, the possibility to install tele-communications equipment and cabling within, and fire extinction is adequate.

At least one name server will be hosted in Mauritius, most likely two, possible in a failover model, so that one name server can take over if the primary fails.

Facilities for hosting the Registry and name servers do not necessarily need to be physically in the same location, but it is by far the most convenient model. If insufficient bandwidth is available where the RS servers will be hosted (see below under Infrastructure, Network Connectivity), then name servers will be hosted separately from the RS.

Additional secondary name servers of .MU in Mauritius can be placed at the facilities of various ISPs.

Alternatively, operation of the RS can be outsourced, though this is not covered at this stage.

▪ *Redundancy and resilience through geographical diversity*

As with any DNS infrastructure, reliability of the service, hereunder resilience to failure and fast response, are ensured through redundancy and geographical

1 http://www.rfc-editor.org/rfc/rfc2870.txt

diversity.  Additional name servers, at least two, at most four, will be located in different geographical locations around the world, preferably hosted by other ccTLD operators, through a partnership agreement.  Ideally, a name server should be present in North America, Europe, Oceania/Asia and Africa, for instance USA/Canada, UK/Netherlands/Sweden/France, Japan/Australia/New Zealand, Kenya/South Africa.

## infrastructure

This section is an overview of the requirement hardware and network connectivity to implement the RS (Registry Services), as well as the expected staffing for administrating the Registry and operating the technical services.  The listed hardware is including production and development/testing.

▪ Servers

– 2 x production DB / API / back end application servers (1 master, 1 failover)
– 2 x production Whois / Web server (1 master, 1 slave)
– 1 x test/development server running the complete environment

If the DNS servers are hosted at the same facilities than the RS:

– 2 x DNS servers (1 master, 1 failover)

Additionally

– 1 backup server / storage device for taking backups (off site)

The billing system is outside the scope at this stage, but it will most likely require 2 servers.

Total: 5-7 servers + 1 off site server.  Using virtual machines, this can be reduced to 3 or 4 physical machines for production, and 1 for test/development.

▪ Network connectivity

The network connectivity required for hosting the Registry Services should be of a fixed circuit type, symmetric – reliable copper or fiber connection, eventually wireless if robust enough, with failover, and a bandwidth of at least 512 Kbit/s each way, ideally 1 Mbit/s or above.  Failing to secure this bandwidth, name servers should be hosted separately, located at a hosting facility with faster connectivity (see above).

▪ Staffing

Staffing for operating a registry requires at the minimum:

– 1 administrative function, half to full time

The role of this person will be to administrate non-technical requests from Registrars and users, and function as the daily manager for the Registry operations (non-technical). The person will also function as a liaison with the IMC and the ICTA. The person will handle creation requests for new Registrars, modification of their contact information, etc...

– 1 system administrator, half to full time

  The system administrator will have the responsibility of running the production and test/development environment servers of the RS, through reliable systems, network and security administration, as well as network and services monitoring.

  Additionally, the system administrator will ensure that remote name servers hosted in other geographical localities are operational, and will act as a liaison with system administrators at these facilities.

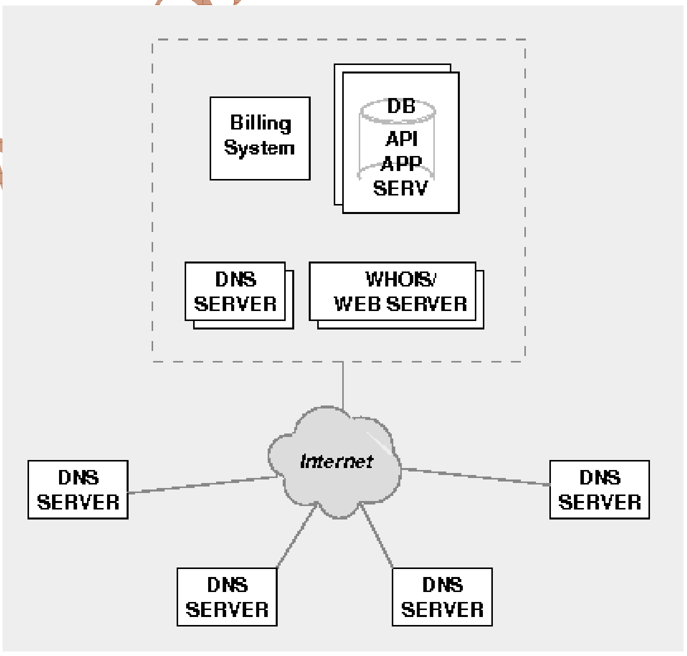
  The system administrator will have basic Database Administration (DBA) skills to maintain, backup and restore the DB if necessary.

  This function is not necessary if operations of the RS are outsourced.

– 1 developer, half time

  A developer profile, familiar with the design and implementation of the RS back end and front end, and the skills to correct bugs, implement new features, eventually with external assistance from contractors.

  Alternatively, in the case of an outsourcing scenario, this person will act as a liaison with external contractors and function as a project manager / functional architect for the platform, ensuring that contractors and external developers deliver the expected software and services.



## Glossary:

| | |
|---|---|
| Registrant | person or organization registering a domain name, i.e.: the legal owner of the domain name. |
| Registrar | organization that manages Internet domain names on behalf of registrants, within one ore more registries. |
| Registry | organization which manages the ownership of Domain names within the top-level domains for which it is responsible, controls the policies of domain name allocation, and operates DNS resolution for said domain name. |
| EPP | RFC3730 - Extensible Provisioning Protocol |
| IDN | Internationalized Domain Name (see http://en.wikipedia.org/wiki/Internationalized_domain_name) |
| Transfer | Transfer of a domain from one registrar to another |
| Redelegation | Modifications of the name server records for a domain |