.br RDAP
Implementation experience and deployment plans
ROW - IETF 93 - 20150719
Frederico Neves
Implementation Experience

- 4 months of development (RDAP server + REST backend)

- In general straightforward

- Public RDAP test suite is very important

- Use of VCard makes the protocol debugging a little more difficult as we get arrays of JSON values instead of a "well defined" structure – probably due to the lack of good libraries

- Extensions created for the Entity type are instance type dependent to be meaningful
Implementation Experience

Behaviour

.br equivalency name policy (hyphen/IDN) will return a 303 HTTP status when the requested domain doesn’t exist but there’s an equivalent domain.

→ GET /domain/xn--rfael-xqa.net.br

← HTTP/1.1 303 See Other
Location: /domain/rafael.net.br
## Protocol Overview

**NIC.br extension**

<table>
<thead>
<tr>
<th>Extension Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nicbr_status, nicbr_lastCheck, nicbr_lastOK</td>
<td>Nameserver and DS checks</td>
</tr>
<tr>
<td>nicbr_arbitration</td>
<td>NIC.br domain special policy</td>
</tr>
<tr>
<td>nicbr_responsible</td>
<td>Entity Responsible</td>
</tr>
<tr>
<td>nicbr_customerSupportService (object)</td>
<td>Entity Customer Service Support</td>
</tr>
<tr>
<td>nicbr_domainCount, nicbr_inetCount, nicbr_autnumCount</td>
<td>Entity resources counters</td>
</tr>
<tr>
<td>nicbr_autnum</td>
<td>IP network Autonomous System relation</td>
</tr>
<tr>
<td>nicbr_routingPolicy (object)</td>
<td>Autonomous System routing policy</td>
</tr>
</tbody>
</table>
Test and conformance

Comparing Viagenie test results

Lack of support for registries that do not have host objects (Viagenie expects nameserver handle and links in responses).

<table>
<thead>
<tr>
<th></th>
<th>Viagenie expected</th>
<th>.br returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ip/1.2.3.4//32</td>
<td>400 (Bad Request)</td>
<td>404 (Not Found) ¹</td>
</tr>
<tr>
<td>/autnum/0666</td>
<td>400 (Bad Request)</td>
<td>200 (OK)</td>
</tr>
</tbody>
</table>

¹ In RFC 3986, appendix A, empty path segments are allowed, but Apache and nginx ignore them.

Robustness principle: “Be conservative in what you send, be liberal in what you accept"
Implementation Architecture

Legacy C++ Whois works as an asynchronous proxy in a pre-fork mode that talks directly with the database.

Go RDAP daemon works using a web service architecture and exchanges data with a REST backend.
Implementation Architecture

Rate Limit

- Token bucket strategy
- Centralised whitelist, blacklist, etc.
- 4 different content levels
Deployment Plan

Architecture

C++
Whois

Python
Web Whois

rdap.beta.registro.br
July 2015

RDAP
Go

REST
Go

Database
Deployment Plan

Architecture

Go
Whois

Python
Web Whois

RDAP

REST

Database

Q3 2015
Deployment Plan
Architecture

- Whois
- RDAP
- Web RDDS
- REST
- Database

Go

Q1 2016
Authentication
Search