ENUM validation architecture & friends

Bernie Höneisen
SWITCH / 1.4.e164.arpa
hoeneisen@switch.ch

Robert Schischka
enum.at / 3.4.e164.arpa
robert.schischka@enum.at

draft-ietf-enum-validation-arch
draft-ietf-enum-validation-epp
draft-lendl-enum-validation-token
Introduction

• enum.at provides commercial ENUM registry services for Austria (+43)
• SWITCH provides Registry for the Swiss ENUM Trial (+41)
• Cooperate on standardization of ENUM validation in the IETF
• Others are invited to join!
Motivation & Goals

Motivation:

• Solving validation is crucial for ENUM deployment
  – Major reason why trials precede production
  – Validation is the major difference between ENUM
    and "ordinary" domain registration

• Validation definition and requirements currently vague
  – Common view considered useful

Major Goals:

• Common understanding
  – Terminology, Processes, Roles, …

• Keeping entropy low
  – Minimize number of solutions addressing same problem
  – Prevent reinventing the wheel – foster deployment instead
validation draft orientation map

Requirements  Role model  Process & trust assumptions  validation data transport  validation data format

draft-ietf-enum-validation-arch

draft-ietf-enum-validation-epp

draft-lendl-enum-validation-token

EPP

SOAP?  IRIS?  Other?

XML  E.115  alternative formats?
Disclaimer

• It is out of scope of these documents how an actual validation is performed ("validation method")

• The documents just attempt to provide a generic framework to base validation processes and communication on.
ENUM Provisioning Model & Roles

Legend
VE: Validation Entity
NAE: Number Assignment Entity

- ENUM Registry
- ENUM Registrar
- Registrant / Assignee
- VE
- NAE

registration
trust relation
validation
E.164 number assignment verification
number assignment

ENUM management
Validation Requirements

1. The ENUM domain name corresponds to an assigned E.164 number

2. The corresponding E.164 number is within a number area approved to be used with ENUM

3. The registration of the ENUM domain name is authorized by the Assignee of the corresponding E.164 number

4. The Registrant of the ENUM domain name is identical to the Assignee of the corresponding E.164 number
Initial vs. recurring validation

• Initial Validation
  – Verify requirements before registration of the ENUM Domain takes place

• Recurring Validation (Re-Validation)
  – Verify that requirements are still satisfied
    • Usually making use of data acquired during initial validation
    • ENUM domain is to be removed when corresponding E.164 number is e.g. revoked
Registration process assumption

**Legend**
- **VE**: Validation Entity
- **NAE**: Number Assignment Entity

1. Number assignment
2. ENUM management
3. Validation
4. E.164 number assignment verification
5. Registration
Transport / data format extension framework

- existing RFCs
- common (policy independent)
- policy dependent

EPP → EPP domain → EPP validation framework

Some other suitable XML transport mechanism (e.g. SOAP)

- example
- validation token enum.at
- other variant?

Scott's EPP RFCs

draft-ietf-enum-validation-epp

draft-lendl-enum-validation-token
EPP transport

- Framework for **Transport** of validation information along with the EPP Domain object

- Elements for validation information itself are out-of-scope of this document
  - Example for better readability included

▶ enables usage of different locally adjusted validation information elements or "tokens"
Validation Token

- Conveys information about a validation
  - E.164 Number (obviously)
  - Contact information (in the style of EPP and E.115)
  - Serial, validation method, validation entity, expiration, …

- XML schema

- Optional cryptographic signature
  - Non-repudiation
  - Authenticity
  - Supports trust relation between VE and Registry

- To be embedded in transport protocols
  - EPP (draft-ietf-enum-validation-epp, enum.at implementation)
  - SOAP? Email? HTTPS?

- In productive use for 3.4.e164.arpa.

- Probably useful for other purposes (number porting?)
Feedback request!

• How are you going to do validation?
• How will you implement it?
• What prevents you from using the architecture just presented?
• What would you like to see in those validation documents?