draft-ietf-modern-problem-framework
&
draft-peterson-modern-teri

Jon Peterson
MODERN WG
IETF 96 (Berlin)
Now a WG item!

Issued a minor revision based on two sets of nit reviews

This went through several pre-WG item revs

Probably little to change here before WGLC

Also, Pierce still thinks we’ve got it all wrong
  – Noted, we discussed on the list
• Now a -01
• Main focus of the revision was alignment with the framework document
• Added an overview of operations
• Made a few small alignment tweaks to the model
• No list discussion (like, really)
What is TeRI?

• A model for telephone-related information
  – Based on modern-problem-framework
• Successor to my old TeRQ proposal
  – Generalized to acquisition, retrieval, management
• Like TeRQ, this is an information model
  – Trying to find the right semantics for records and operations
  – We’ll worry later about the proper encoding and transports
• We decided in Prague to do this in one spec
Moving Parts

• Acquisition operation
  – How do I request and receive numbers?

• Management operation
  – How do I provision information about a number?

• Retrieval operation
  – How do I get information about a number?

• These protocols access overlapping data
  – If you can provision it, you should be able to query for it

• Surely this is a common information model
The TeRI Interfaces

Client → Service

Acquisition

Service → Authorities

TeRI Records

Authorities

Management

Client → Inter-Mediary

Retrieval

Client

Client

Client
Operations and Records

• Each Operation consists of a Request and a Response
  – All operate our core building block: TeRI Records
• Requests will have a Source, Subject, and Attributes
  – Source indicates the originator of the Operation
  – Subject would typically be a TN itself (or a range)
• Responses will have a Response Code
• TeRI Records contain information about TNs
  – Some Records might cover a range of TNs
TeRI Records

- TeRI Records would be available at Services
  - Services could be public, centralized and monolithic
    - Distributed, or private
    - The Operations and Info Model will be the same
  - Each TN might be associated with multiple Records
  - Records are trusted based on the Authority that generated them
    - Usually not based on the Service that shared them

- Entities from the MODERN framework act as
  - Clients
    - Users, CSP, Government Entities
  - Services
    - Registries, Registrars, CSPs
Roles of MODERN Actors

• **Numbering Authorities**
  – Regulators, etc.
  – Roots of authorities (never acquire blocks)
    • No “golden” root, all is relative

• **Registries**
  – Manages core number allocation functions, handles requests for numbers from registrars

• **Registrars**
  – *Has relationships with CSPs/Users to provide numbers – often is also a CSP*

• **Communication Service Provider (CSP)**
  – Provides services to users
  – May include traditional carriers, enterprises, OTT providers, etc.

• **Users**
  – Individuals reachable through a communications service
  – Operate clients, black phones, etc.

• **Government entities**
  – Privileged access to data
Records: Think SCRUD

- Search, Create, Read, Update, Delete
- Creation begins the lifecycle
  - A Registry always creates the first Record
    - Registrars then acquire Authority from Registries
  - Bootstrap administration record designating the Registry itself
- Should Records be partially updated, or wholly replaced?
  - Currently, only wholly replaced
  - Any Authority can update or delete its own records
    - In hierarchical assignment models, Authorities above the chain can delete the records of their delegates
The Acquisition Operation

• Query:
  – Source (Query Source, Query Intermediary)
  – Subject (Telephone Number/Range)
    • Used to have SPID, currently removed per MODERN scope
  – Attributes (constrains query, say, to finding a particular number in a range)

• Response:
  – Response Code
  – TeRI Record (newly generated assignment granting authority for this TN/Range)
    **Result:** This makes the Client an Authority for that TN/range
The Management Operation

• Query:
  – Source (Query Source, Query Intermediary)
  – Subject (Telephone Number/Range)
    • Used to have SPID, currently removed per MODERN scope
  – TeRI Records (including Record ID)

• Response:
  – Response Code
    **Result:** This replaces/deletes a previous TeRI Record, or creates a new one
The Retrieval Operation

• Query:
  – Source (Query Source, Query Intermediary)
  – Subject (Telephone Number/Range)
    • Used to have SPID, currently removed per MODERN scope
  – Attributes (constrains query: e.g., “voip” if only looking for VoIP, or Route Source, or Record ID)

• Response:
  – Response Code
  – TeRI Record

  **Result:** Retrieves Record if successful
TeRI Record Contents

• TeRI Records would contain
  – **Subject** (the TN or TN range of the record)
  – **Authority** (Source of the data, usually the provisioner)
  – Contact (administrative contact, WHOIS/WEIRDS)
  – Service (a service associated with the TN)
  – **Identifier** (unique ID for the Record)
  – Signature (typically a crypto assurance of the Authority)

• Divided into Service and Administrative Information
  – Services records always have a Service
  – Administrative records always have a Contact

• Obviously different actors would set/get different Record elements
TeRI Record Element Types

- Telephone Number (RFC3966 – but should we revisit?)
  - Ranges – need some work here
- Domain Name
- URI
- IP Address
  - IPv4/IPv6
- Contact
  - Per jCard
- SPID
  - Currently specified as four-digits, other SPID types possible
    - GSPID, ITAD, etc.
- Trunk Group
  - Currently points to the Gurbani/Jennings RFC
- Display Name
  - Support for CNAM as well as a SIP “From” header field
- Extension
  - Reserved for further use
Telephone-Related Information

TeRI

Records

Admin Info

Service Info

Retrieval

Acquisition

Management

Just a logical picture
Transport and Encoding?

• Agree on semantics first, then define bindings and profiles
  – A binding is defined as an encoding and a transport
    • We want at least one binding per protocol, maybe allow more
  – Could build on JSON/HTTP, could build on ASN.1/UDP
  – Bindings need to detail how the elements of the data model are mapped to the encoding
    • Other low-level details like chunking, representation of cryptographic security, etc.
  – Requirement: to transcode between bindings without losing data (at an intermediary)

• Aim for maximum applicability
  – While not overcomplicating the model
Next Steps

• Energy needed, and discussion
• Need more input on Record elements
  – Varies by the use case
• Aligning with use cases
  – e.g. DRIP
  – STIR is another
• Define necessary profiles and bindings
  – Probably JSON
• When we have something more concrete, and with some energy, look toward adoption