Telephone Related Queries (TeRQ)

draft-peterson-terq-00
The Changing Focus of ENUM

• RFC2916 (2000), most recently RFC6116 (2011)
• The original vision: user-driven, public records
  – Now: carrier-driven, records often contain internal network data
• Change in applicability places different requirements on ENUM
  – Authentication/authorization of query sources
  – Resolution of compound queries
    • Not just the domain name, but relational elements
  – Returning data that isn’t really a resource indicator
    • e.g., CNAM – storing the resource itself in a NAPTR record

• We saw these requirements in the E2MD proposals a few years ago
  – SPEERMINT/DRINKS have looked at this from the provisioning side
Time to Reconsider?

• The standard DNS protocols don’t do those things well today
  – In deployments, non-standard solutions are prevalent
  – Path to getting those solutions standardized is not clear
    • Not here to rehash E2MD arguments

• Requirements in the field aren’t going away, though
  – Need a way to ask rich questions about telephone routing and get rich answers

• The question: would it help to approach these problems without factoring in the constraints of any underlying protocol? Without:
  – A legacy public “golden root” anchor
  – Semantics of DNS queries (exact match on label)
  – Exclusive focus on TN
  – Requirement to return a URI (limiting syntactically)
The TeRQ Strawman

• Proposal: Establish a data model first, then worry about underlying transports and encoding

• Query Elements:
  – Source (Query Source, Query Intermediary, Route Source)
  – Subject (Telephone Number, SPID, etc.)
  – Attributes (constrains query: e.g., “voip” if only looking for VoIP)

• Response Elements:
  – Response Code
  – Subject (Optional)
  – Records
    • Authority (Source of the data)
    • Attributes (Name/Value pairs)
    • Priority
    • Expiration
The TeRQ Architecture

Client -> Inter-Mediary -> Server

Client -> Inter-Mediary -> Authorities

Server -> Records

Server -> DRINKS, etc.
Transporting TeRQ

• Once we agree on semantics, work on bindings
  – A binding is defined as an encoding and a transport
  – Could build on JSON/HTTP, could build on ASN.1/UDP
    • DNS? Never say never…
  – 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.
  – Also must be possible to transcode between bindings without losing data (at an Intermediary)

• Aim for maximum applicability
  – Not just a telco protocol, a web protocol
  – Something to work for Verizon and Google
Thoughts?

• Today, just floating the idea
  – There’s a -00 to read
• If people think this has potential, could come back with a charter in Vancouver
• Some good list discussions already
• Welcome any further feedback here