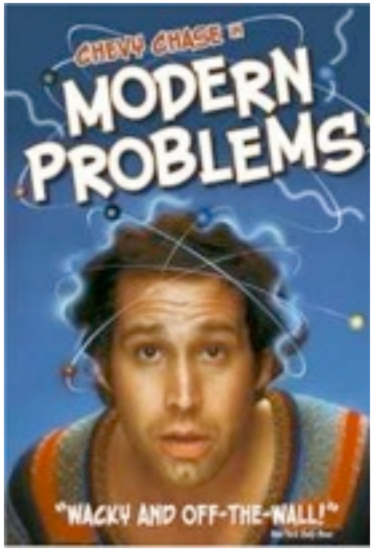
draft-peterson-modern-problems
draft-peterson-terq

MODERN BoF

IETF 92 (Texas)



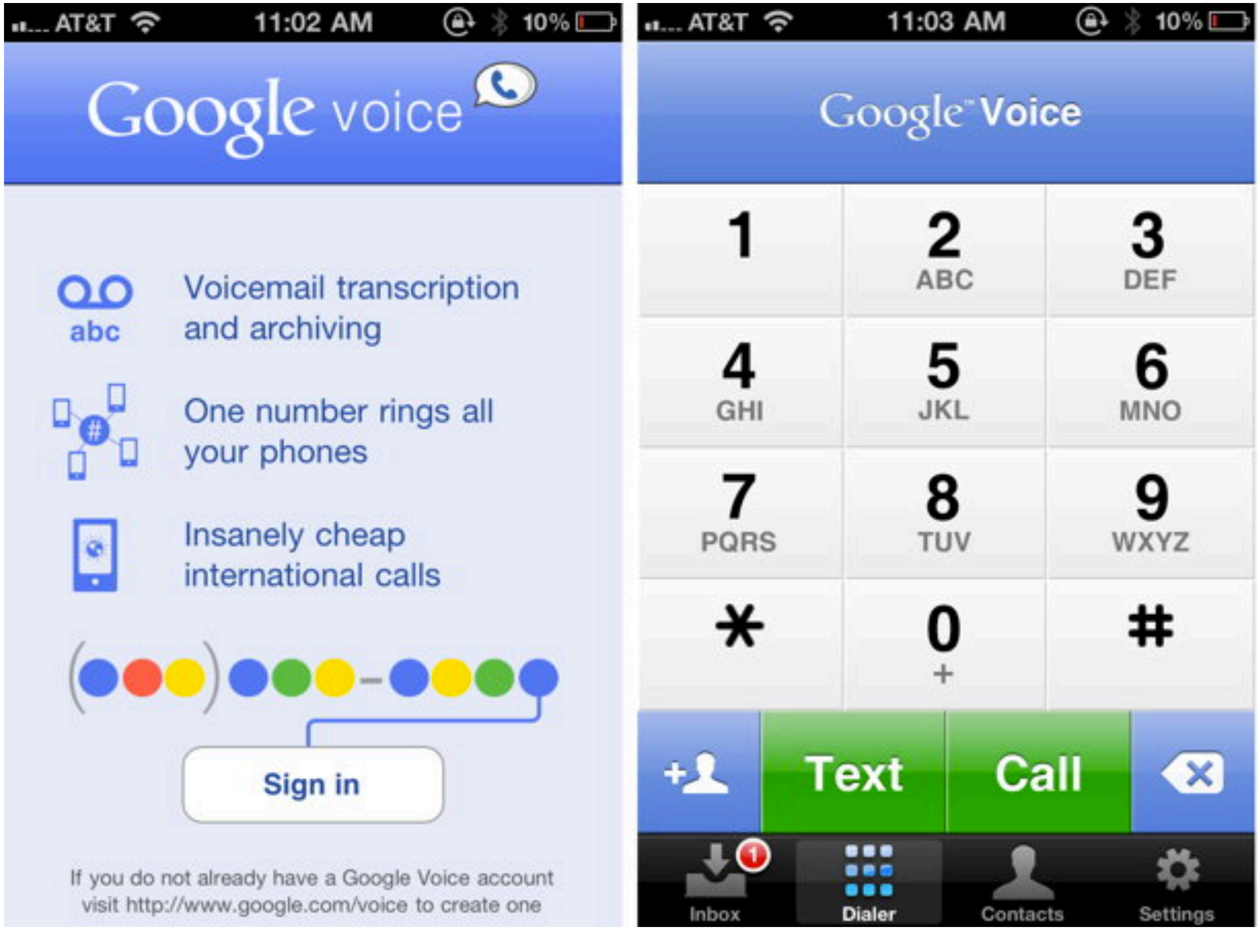
Telephones and the Internet

- Our long-term goal: migrate telephone routing and directory services to the Internet
  - Subject of much work over the past 20 years
- Telephone numbers have never gone away
  - Mobility now sustains their relevance
  - Non-linguistic, international, opaque
- But they have eluded the Internet's grasp
  - Still anchored in the PSTN

What if…

- … we treated numbers like Internet identifiers
  - What if you could get numbers the way you get domain names?
  - Or what if you could get numbers like you get IP addresses?
- This was the subject of an FCC workshop last year
  - Informed by accelerating PSTN transition plans
- In short, there's not just *one way* to get Internet IDs
  - You can buy individual domain names from registrars
  - You can get names bundled with a product or service
  - You can buy IP address blocks, or get dynamic allocations

Not just what if



AT&T 11:02 AM 10%

Google voice

Voicemail transcription and archiving

One number rings all your phones

Insanely cheap international calls

Sign in

If you do not already have a Google Voice account visit http://www.google.com/voice to create one

AT&T 11:03 AM 10%

Google™ Voice

“Search for a Google Voice number in our database by area code or zip code!”

Sensitivity training

- MODERN absolutely will not set TN policy
  - – The IETF does not control TN policy anywhere
    - And we're not looking to change that
- Numbering policy is a sensitive topic
  - – Numbering inventory is a scarce asset, not like the DNS
- It's a *polarizing* topic
  - – Some feel their architectures are threatened if number allocation is opened up to new parties
  - – Others feel their architectures are threatened by current restrictions on number allocation
- Not our decision – today is not a referendum on that
  - – Even if policies stay the same, these tools will have value

Moving Parts

- **Acquisition mechanism**
  - How do I request and receive numbers?
- **Provisioning mechanism**
  - How do I provision a number with a CSP?
- **Query mechanism**
  - How do I get information about a number?

- **Not** new protocol design: just data models

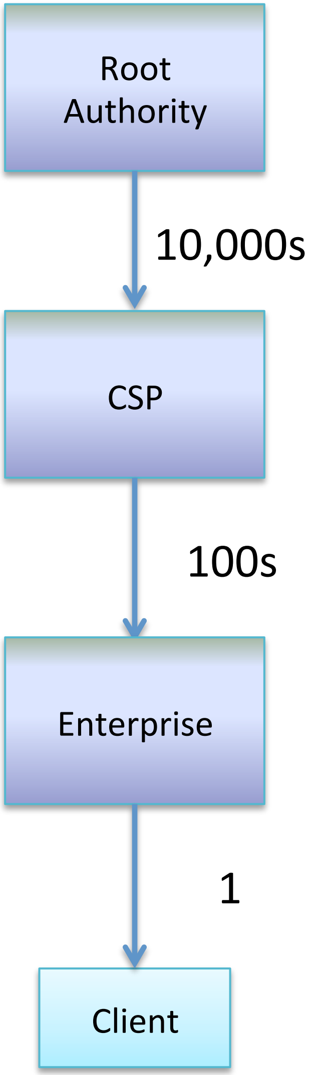
Taxonomy

- **Number Authorities**
  - Includes delegates, receiving and distributing blocks
  - Includes root authorities (never acquire blocks)
    - No “golden” root, all is relative
- **Number Users**
  - Includes enterprises, some resellers
    - May themselves act as number authorities for others
  - Operate clients, black phones, etc.
- **CSPs**
  - Provide communications services
  - May also act as authorities or users, or may not
- **Government entities**

Delegation & Authority

- Today, a common chain of delegation looks like this

- Number blocks trickle down from a root of authority through CSPs
  - Eventually single allocations to users



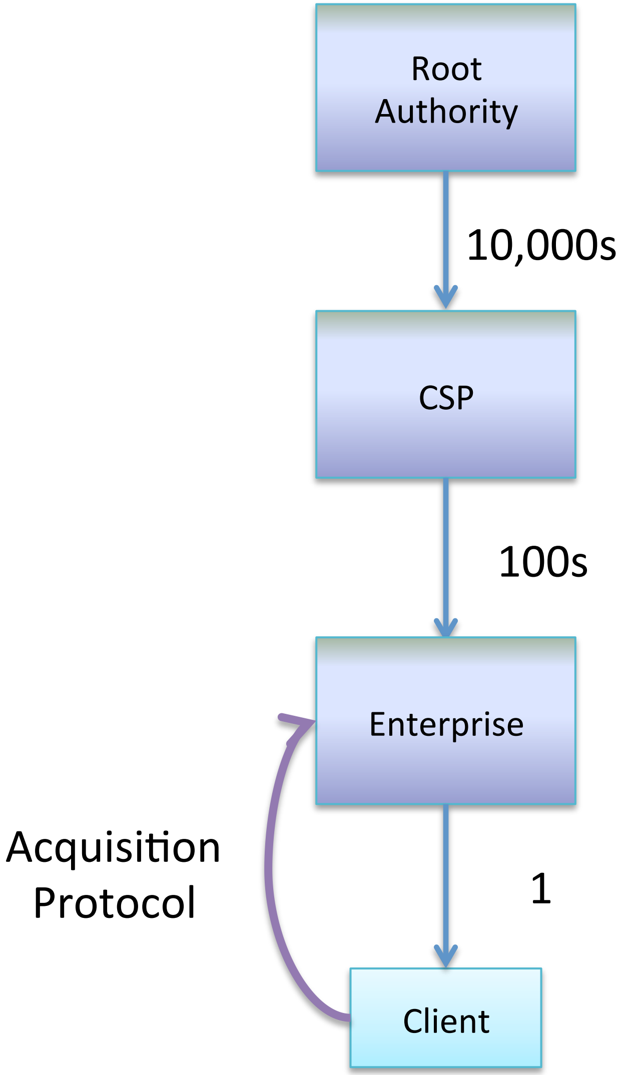
Acquisition

Manages number inventory, tracks who gets assigned what, assigns credentials (STIR)

Acquiring a number

- The acquisition tool could let a new phone get a number from its enterprise
  - Phone gives some information, gets a number and a credential

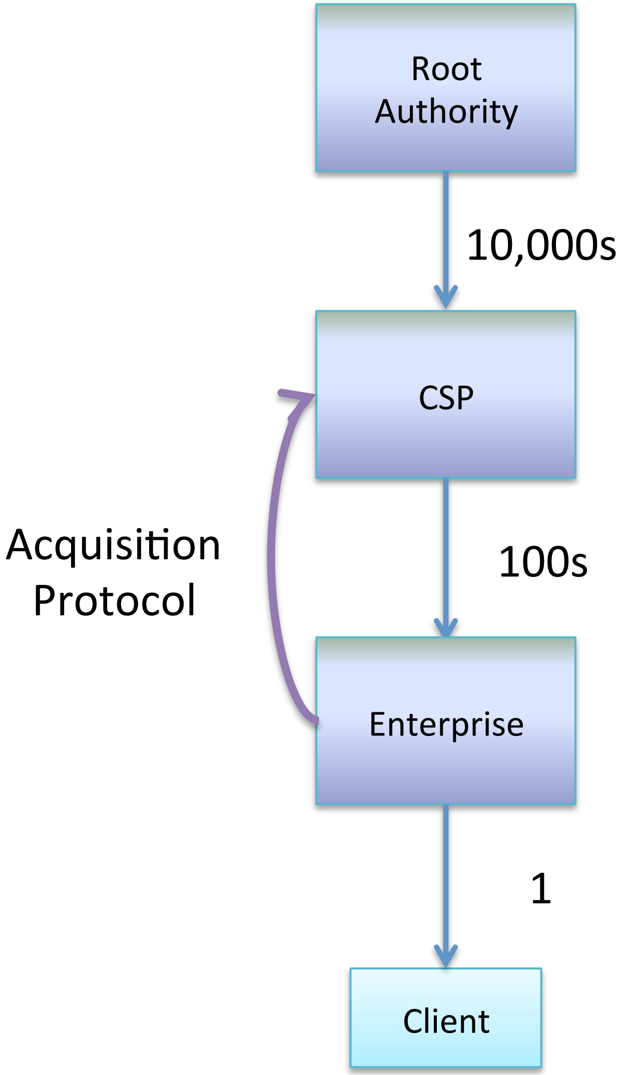
- *Similar use case for Google, or Skype, or whatever*



Acquiring a range

- Possible to request whole blocks of numbers
  - STIR certs indicate a range, or you might get multiple certs

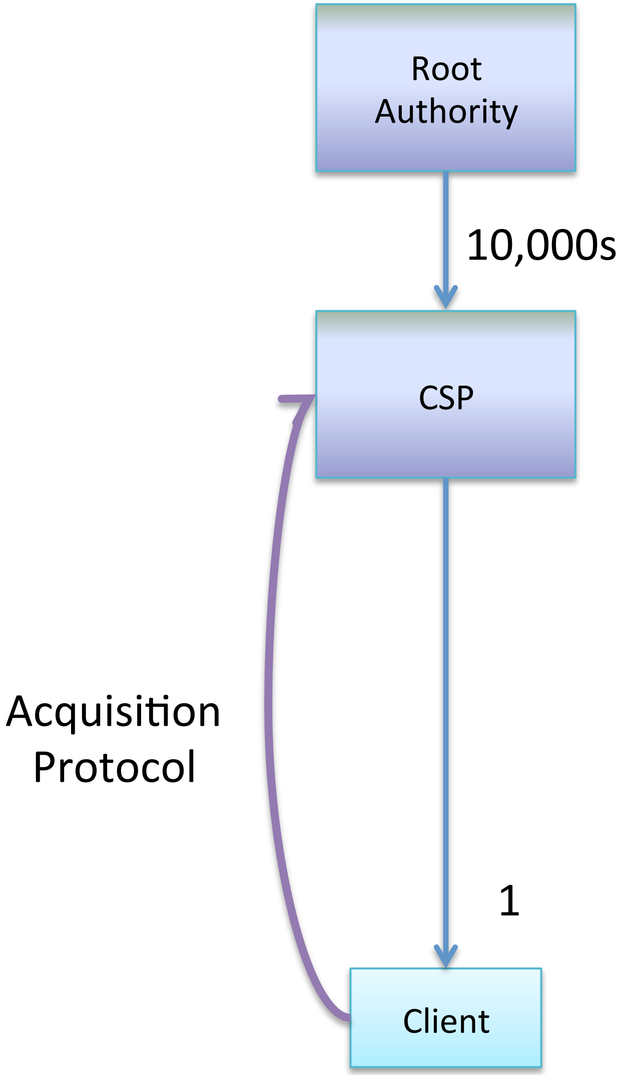
- *Enterprise turns up a new PBX, needs to activate 100 new seats*



Customer to CSP

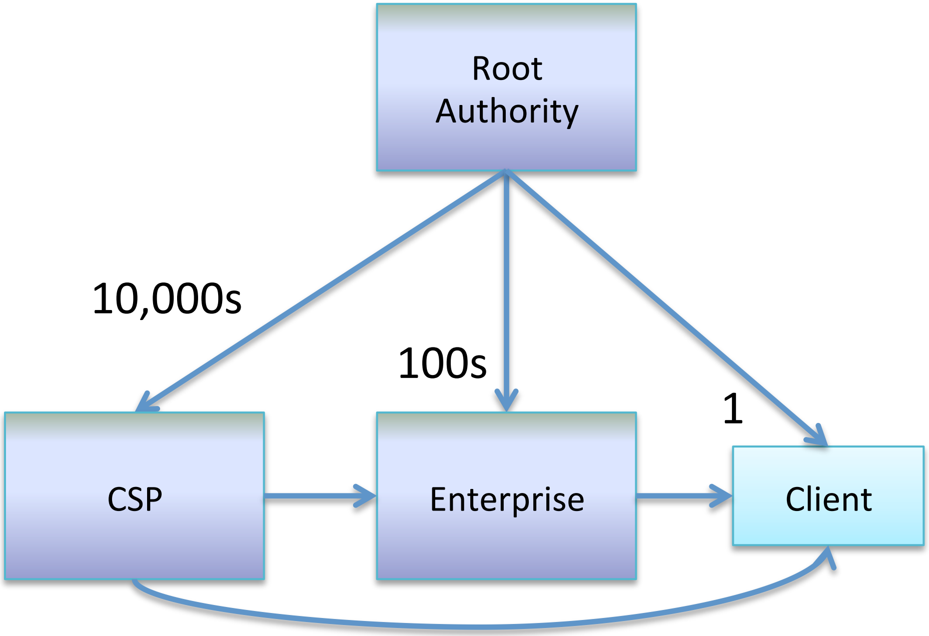
- The acquisition tool could let a new phone get a number from a CSP
  - Get a new number and a single certificate

- *You just bought a new phone at the Apple store, and want to replace an existing account*



Acquisition mechanism scope

- We propose a tool that would *enable* allocation to anyone
  - **Not setting any policy!**
- Allocation is just the first step
  - How a client connects to a CSP is provisioning



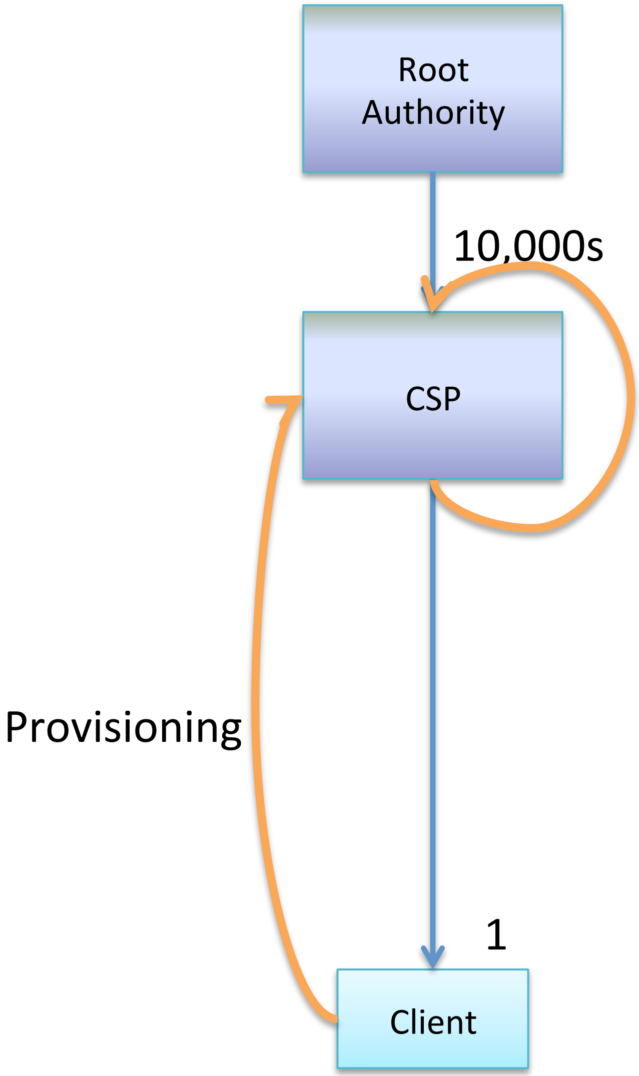
Provisioning

Associates numbers with CSPs, manages related user data

Provisioning today

- CSPs take care of everything

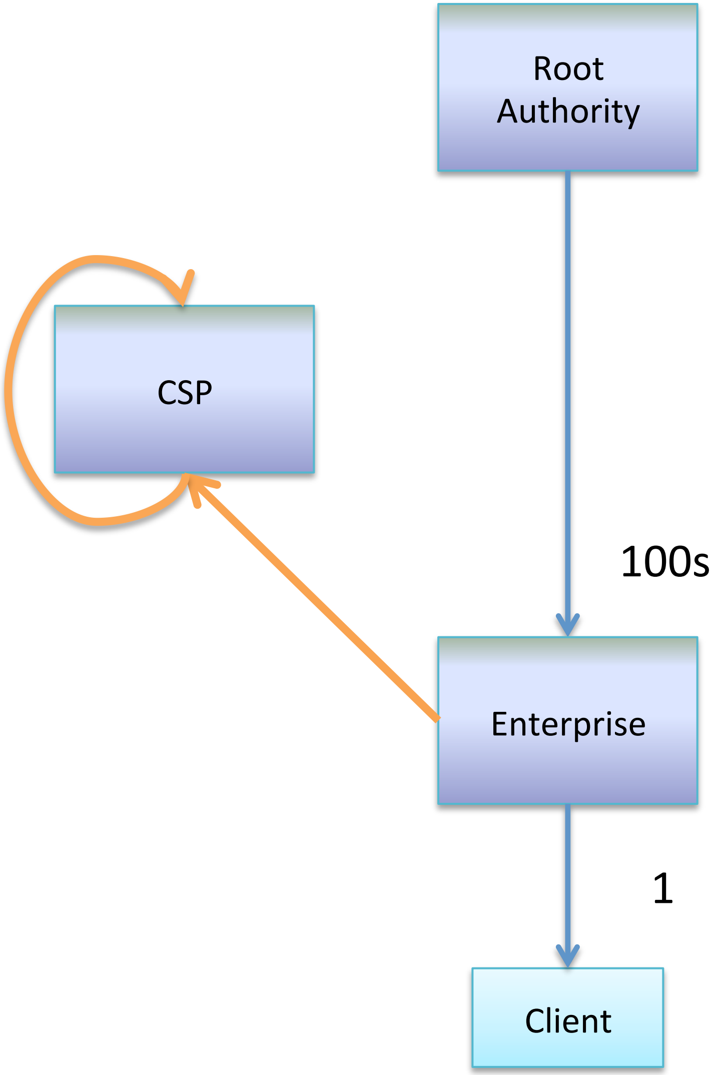
- Mostly CSP internal provisioning to itself
- A small amount of user info



Provisioning

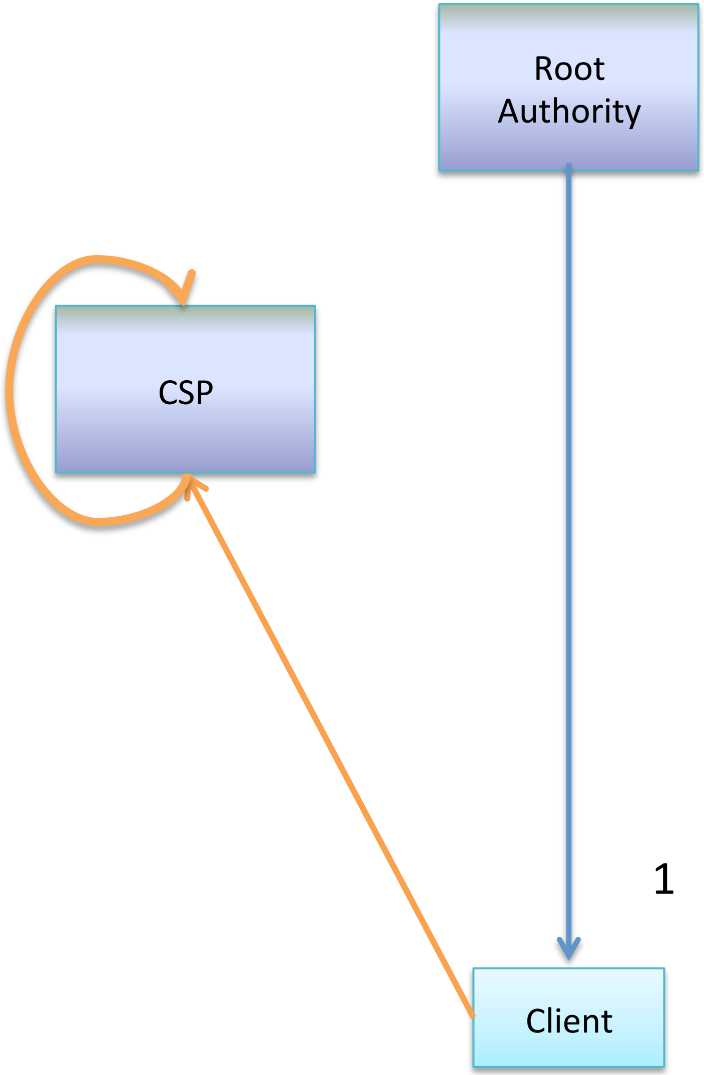
- New acquisition models create new provisioning flows
  - "I own TNs, please manage them for me!"

- Enterprise acquires numbers for a new PBX itself, provisions these at the CSP
  - Proves ownership with STIR credential



Provisioning

- … or, end user buys a number and then provisions a CSP
- Like SIP phone registration
  - Now proves number with credential
  - Maybe it's as simple as registering a new Contact
- Much like the domain name model today



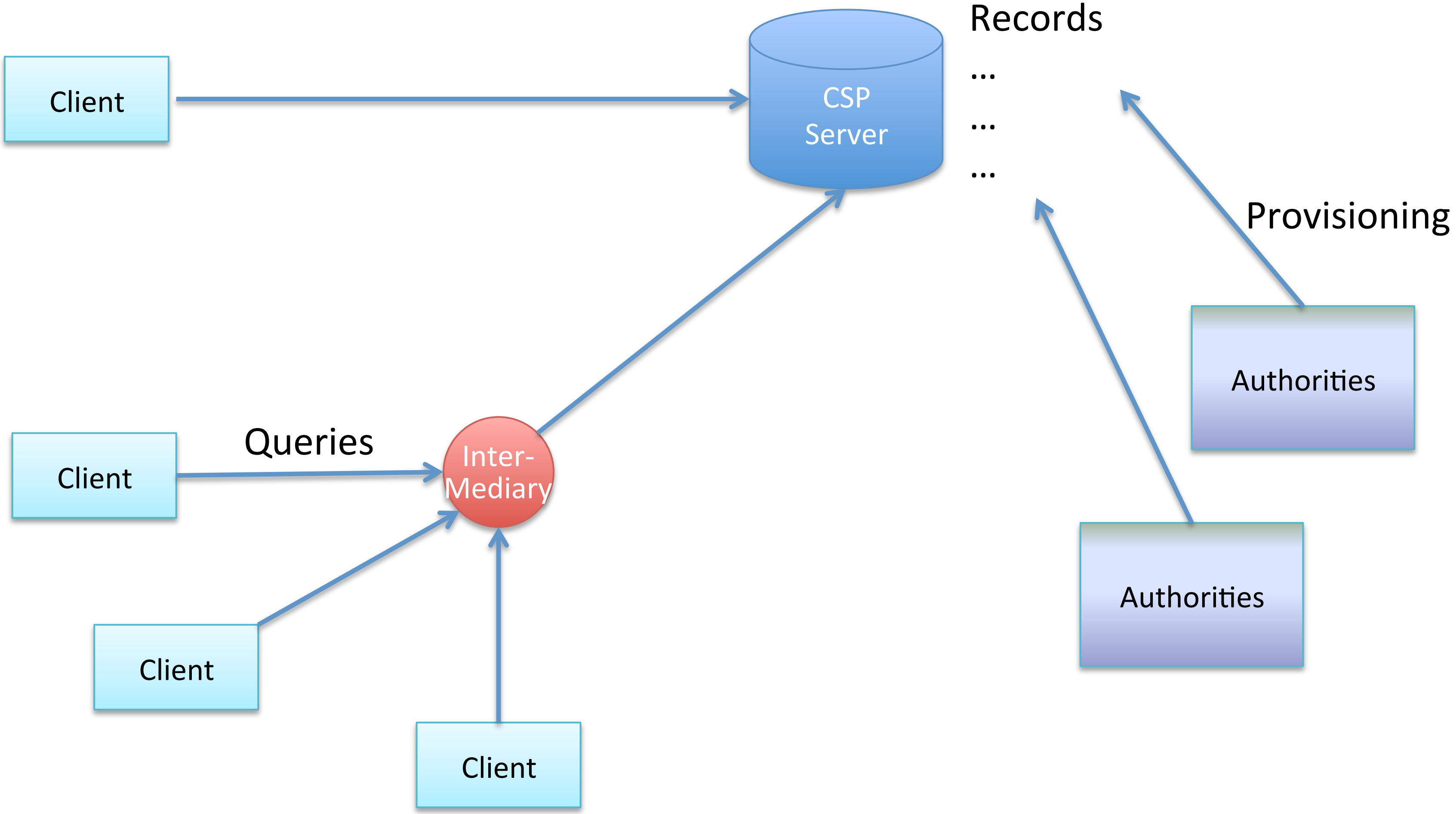
Querying

"ENUM Bashing?"

Okay, just a re-run of IETF 85,

Already DISPATCHed

The TeRQ Architecture



About the Data Models

- Again, not much protocol design proposed here
- TeRQ info model split into two parts
  - Routing Information
    - Registered endpoints and their disposition
    - Mostly would be used by the MODERN provisioning protocol
  - Administrative Information
    - Contacts, billing addresses, compliance information
- Beg, borrow and steal for other components
  - WEIRDS, DRINKS, SIP, STIR, others
- MODERN will deliver an architecture

Somebody special

- Government entities will want to ask questions about numbers
  - Very similar to situation of DNS names
- We plan to follow the WEIRDS lead
  - weirds-rdap-sec-12
    - "Full access may be granted to a special group of authenticated clients."
  - Or whatever they go with

Direction for modern-problems

- Ready for WGLC? (heh)

- Hopefully helped focus discussion

thanks!

BACK UP

TeRQ

- Method: 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)
  - 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

Transporting TeRQ

- Once we agree on semantics, work on bindings
  - A binding is defined as an encoding and a transport
    - We want to allow for multiple bindings for different environments
  - 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.
  - 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 both Verizon and Google

Element Types

- Data model current specifies:
  - Telephone Number (RFC3966 – but should we revisit?)
    - Ranges – need some work here
  - Domain Name
  - URI
  - IP Address
    - IPv4/IPv6
  - 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
  - Expiry
    - Absolute time
  - Priority
    - Value from 0 to 1
  - Extension
    - Reserved for further use