Whois History & RDAP
History **OF** whois

- Its so old, it has a low RFC number (predates almost any other active protocol in use)
- The protocol specification is basic:
  - Connect on TCP
  - Whatever the client sends is the “key”
  - Whatever the server sends back is the “value” in a key->Value lookup
    - The format is unspecified.
- Its had revisions
  - They mostly just say
    - “its very old” and “gosh, its really bad with UTF8” and “we could do better”

- Hence what we call “the WHOIS problem” –which is on ICANN’s docket
  - No global commonality, consistency of WHOIS services at this time.
WHOIS for Internet Numbers is (mostly) RPSL

- RIPE NCC specified format for type:value records
  - Implemented at RIPE NCC, AfriNIC and APNIC
  - ARIN and LACNIC run locally defined WHOIS formats

- INR WHOIS is mostly about reverse-DNS, and Routing permissions
  - Except the overwhelming majority of queries are firewall rule checks, abuse checks
  - Highly structured data

- Lack of consistency in the RIR services of concern
  - We were very motivated to look into RDAP as a common platform for future service
  - It’s a work in progress, but we ARE now all providing RDAP services which are entirely consistent across the RIR
    - Much NIR data (APNIC Region) still to be done.
History **IN** WHOIS

- History mechanism added to the RIPE NCC WHOIS code ~2012
  - ADD of new object instantiates object serial in DB schema
  - On UPD of WHOIS record, prior object moves to history table
  - On DEL, head object moves to history table.

- Provides basic ‘AUDIT/LOG’ of changes to record state

- ADD of ‘same’ object creates new object serial in DB
  - No linkage from prior object serial to current object serial
  - Sequence (ADD-or-UPD, UPD, UPD, **DEL**, ADD) of ‘identical’ object truncates visible history
  - Not deleted, but invisible: Its inside the system but not linked.
Consequence: loss of visible history

• We know what happened to the object, we can’t show it
• RIPE view this as ‘by design’ to permit history truncation
  – European Data privacy law, related motivations
• LEA, IPR, other searches of object history now hard
  – Fraud detection has become harder
  – History Serials not exposed to search mechanism. Requires DB access

• What can we do about his?
  – Move to RDAP!
  – We solved this problem elegantly, in a platform-neutral way.
RDAP is the future

• Consistent data format, common across Numbers & Names WHOIS
• JSON structured, easy to incorporate into modern coding languages with native support of “hash” or “dict” data structures
  – Structure amenable to modification based on ‘extensions’ in syntactically consistent manner
• Limitations:
  – Does not currently represent all RPSL data so not suitable for IRR
  – Not yet widely adopted in names WHOIS but actively under consideration
  – (recent ICANN announcements of note)
• APNIC offered RDAP as a fully supported service May 2015
RDAP highlights

• Leverages “vcard” specification for whois ”contacts”
• Lots of sub-types possible
  – Abuse-C, Admin-C, Tech-C …
• Well understood format for systems integration
  – “who you gonna call” -> Vcard -> mail/phone/<other>
• Handles i8n gracefully
  – RDAP specifies JSON == UTF-8
  – Vcard handles combined Latin/ASCII and native script records well
• “Top down steerage” –ICANN registry to glue all clients to sources
URL and content well defined
By a set of RFC’s which reflect current coding practices
RDAP: JSON encoded resource records

• Same data as in WHOIS, but re-encoded into JSON
• Carried over HTTP(S) using REST query syntax for objects
• Structure used is IETF defined, common with names WHOIS
• Easily (trivially) fetched and parsed by commandline tools eg curl/jq
• Automatically self-steering:
  – Directory from the top down maintained by IANA
  – Redirect between RIR via HTTP 302 redirection
• We added history as an outer From/To reference to make WHOWAS
  – WHOWAS service started December 2016
$ curl \nhttps://rdap.apnic.net/history/ip/203.133.248.0 | \\njq

```
{
  "applicableFrom": "2016-09-19T08:17:49Z",
  "applicableUntil": "2017-02-08T05:39:21Z",
  "content": {
    "handle": "203.133.248.0 - 203.133.251.255",
    "startAddress": "203.133.248.0",
    "endAddress": "203.133.251.255",
    "ipVersion": "v4",
    "name": "APNIC-AU-RD",
    "type": "ALLOCATED PORTABLE",
    "country": "AU",
    "objectClassName": "ip network",
    "entities": [
      {
        "handle": "IRT-APNICRANDNET-AU",
        "vcardArray": ["vcard"
```
We’re testing clients to use this

- Uses this underlying http://rdap.apnic.net/history URL form
- Client then uses JS code to display the returned JSON as a history view
- Timeline of change points of the resource
- Can walk through prior states of the data, see what changed (diff colourized)
- You can write your own code. Its HTTPS + REST + JSON
network name
APNIC-JP-RD

network
203.133.248.0 - 203.133.251.255

country
JP

type
ASSIGNED PORTABLE

remarks

/22 for APNIC R&D node in Japan and AU.
This object can only be updated by APNIC hostmasters. To update this object, please contact APNIC hostmasters and include your organisation’s account name in the subject line.

/22 for APNIC R&D node in Japan and AU.
To report network abuse, please contact tech-c and admin-c.
For assistance, please contact the APNIC Helpdesk.

description
APNIC R&D Centre JP

handle
IRT-APNICRANDBNET-AU

name
IRT-APNICRANDBNET-AU

kind
RDAP History extension

• Developed in common with the NRO engineering coordination group
• Taken into IETF standards in REGEXT working group

• Fully documented on github, UI code examples
• Constructs history by stepping over the new history-table serials in WHOIS DB, to compute path back in time beyond DEL operations
• Encompasses block splits, merges, can make for a large search space
  – We limit this in the UI and the back-end REST api
Got code!

- APNIC WHOWAS server is in GITHUB
  - Modular design, interfaces to WHOIS data source
  - As deployed, linked to RIPE NCC WHOIS database (java/MySQL)

- Experimental client(s) in GITHUB
  - Using ELM (functional abstraction targetting JavaScript)

- It’s fast
  - …At the cost of memory (we preload *everything*)
  - But memory is cheap: move to the cloud

- Browse https://github.com/apnic-net
Thank You!

Feedback, questions to ggm@apnic.net