



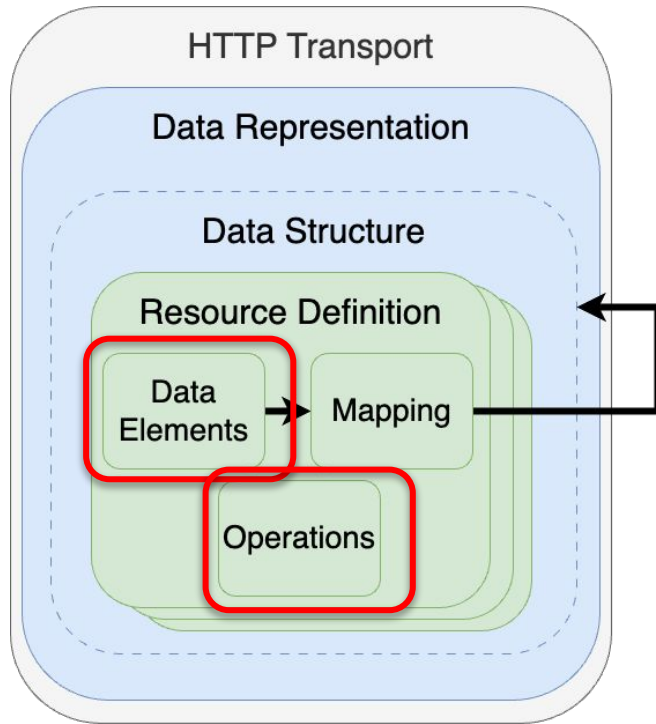
# IETF 125

## RPP Data Objects Update

Pawel Kowalik (DENIC)  
Maarten Wullink (SIDN)



# You are here



## From draft-ietf-rpp-architecture: 4.2.3. Resource Definition Layer

This layer defines the **structure and operations** for each **resource type**, independent of media type or representation.

- **Data elements** (and operations)
- **Resource type definitions**
- **IANA registry definitions**



## -03 version published

- DNS data model aligned with draft-simmen-rpp-dns-data-01
  - including DNSSEC support based on RFC 5910
- Fully integrated RFC 3915 for RGP support and all added statuses
- Restore Operations as 1 Step process, 2 Steps optional for EPP compatibility
- Extensibility
  - standardised/private extension mechanisms
  - extension points
  - operation extensibility
- New object type: Process Objects
- New primitives: Dictionary[Value Type], Object
- Common Provisioning Metadata component object
- Objects and operations now complete for core set: domain, host, contact



# DNS data model

Fully aligned with draft-simmen-rpp-dns-data-01

## DNS Data Object

- 0+ Records (Composition[DNS Resource Record Object])
- 0-1 Controls (DNS Operational Controls Object)

Part of Domain (vis a vis Host Attribute, also DNSSEC), Host (Host Object)

## DNS Resource Record Object

- 1 Name
- 0-1 Class
- 1 Type - NS/A/AAAA/DS/DNSKEY (which ones depends on the usage)
- 1 RDATA (Object)

## DNS Operational Controls Object

- 0-1 TTL
- 0-1 Maximum Signature Lifetime



# Process Objects

- Can contain data - represent process input/state/output
- Can contain own operations - interactions with a long running process
- Are created and operate in context of “owner data object”. Lifecycle is bound to the owning object.

Usage now defined in the document:

- Transfer process - generic one and extension for domain names (with renewal period and new expiration date)
- Restore process



# Common Provisioning Metadata

These elements are common for all data objects and read-only from client perspective:

- Repository ID
- Sponsoring Client ID
- Creating Client ID
- Creation Date
- Updating Client ID
- Update Date
- Transfer Date

Therefore these are now defined as component object Provisioning Metadata and referenced to.



# Open questions



## “Urgent” flag

RFC 5910 5.2.5. - Update command:

OPTIONAL "urgent" attribute that a client can use to ask the server operator to complete and implement the update request with high priority.

- support in EPP is optional for servers
- does it have any operational significance anymore, when zones are published in a matter of minutes, not hours or days?
- shall RPP carry over this functionality, or skip it?



# AuthInfo as part of standard data model

- Authorisation Information is now part of any transferable object (Domain, Contact) - read/write
- Not a best practice to send credentials (clear text) with every response
- Shall we move Authorisation Information to a separate sub-object, which has to be requested separately or only part of full dereferenced representation?
  - We can model it as separate sub-resource with its own lifecycle - expiration for AuthInfo



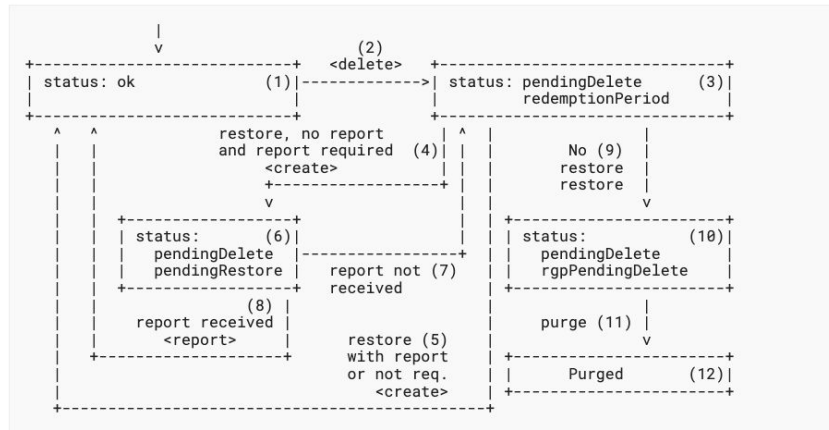
# Restore Report Object

- Optional Restore Report is part of a generic Restore Process object
- Elements imported from EPP - all optional
  - Pre-Delete Data
  - Post-Restore Data
  - Delete Time
  - Restore Time
  - Restore Reason
  - Statements
  - Other
- Shall this model be carried over? Would it fit for all object types or it's just fit tight to current ICANN requirements?
- Would a generic model like Object make a better fit for a generic process?



# Business logic in the data objects draft

- When integrating RFC 3915 this part appeared:



- This is obviously beyond “data elements” and “operations”
- Business rules are also defined in “Constraints” of data elements
- Shall data objects cover in the scope also business logic? Is there added value of splitting it to a different draft?



# Associations guaranteed ordering

- basic (not labelled) Aggregations/Compositions with more than one element are defined as “ordered array”.
  - same unchanged data is always inserted in the same order
  - allow stable reference by position to data elements
  - aligned with JSON definition of “array”, allows for comparison of JSON objects by content
  - allows using JSON pointers to array elements
- this however a constraint EPP does not define - i.e. hostObj elements within ns element of domain INFO response may appear in any order
- on the other hand it can be likely easily assured, even by RPP/EPP proxy implementation (order by hostname, id or sth. else stable)
- keep it or any reasons why not?





# Localisation of contact data

- Postal Information
  - Cardinality: 1-2
  - Mutability: read-write
  - Data Type: Dictionary Aggregation[Postal Info Object]
    - Label Description: type of contact data localisation
    - Label Constraints: Allowed values: "int" for "internationalised" **all-ASCII** version of an address and "loc" for localised forms with possible **non-ASCII** character sets.
- Shall RPP be designed as Universal Acceptance first, where all-ASCII limitation shall be considered obsolete?
- If no “all-ASCII” limitation there, what would be the right definition of "internationalised", if any? Or shall the data only have (one or more) localisation?



## Next steps

- Add “primary key” property to data elements - needed for referencing and defining canonical addressing
  - Split constraints and business rules
  - Define data elements in formal way: formats, ABNF, limits
  - Add organisation RFC8543 -> or “park” for a new draft?
  - Organise IANA registry better, add registration rules - as per IANA early feedback
- 
- More eyeballs needed !!
  - Help welcome for any of the above 
  - WG adoption 



# Thank you

- ▷ Email: [pawel.kowalik@denic.de](mailto:pawel.kowalik@denic.de)
- ▷ LinkedIn: <https://www.linkedin.com/in/pawelk/>
- ▷ Mastodon: <https://mastodon.social/@paulok>
- ▷ Bluesky: <https://bsky.app/profile/pawel.paulonet.eu>