RPP - Requirements

Maarten Wullink - SIDN Labs

IETF 122, Bangkok, 2025

Context

- This is an overview of the set of potential requirements from;
  - Mailing list
  - Existing draft documents
  - Discussions during meetings
  - Hackathon

- These are not the final requirements, just for inspiration and discussion
- What process to use to develop actual set of requirements?

- Requirements overview on WG wiki: https://wiki.ietf.org/en/group/rpp/requirements

Groups

Requirements are divided into groups;

- API
- Data Model
- Data Representation
- Discoverability
- EPP compatibility
- Security
- Extensibility
- Scalability & Performance
- Design Process

API

- Based on HTTP
- Using the REST architectural style
- Stateless (no state on server)
- Specification and documentation using OpenAPI (TypeSpec?)
- Leverage existing web standards

Data Model

- Based on data format agnostic data structures (TypeSpec?)
- Existing EPP extensions may be added to core data model (DNSSEC, DELEG)
- Support for internationalization

Data Representation

- Support for multiple data formats (e.g. JSON, YAML, XML)
- JSON as the default data format
- Support for data validation
- A server may choose to include support for multiple media types
- A client can signal to the server what media type is preferred

Discoverability

- Service location: Enable client to discover available RPP services
  - IANA (RDAP)
  - DNS RR

- Functionality
  - Discovery document (well-known URI of URI-template)
  - REST service discovery, allow for loose coupling between clients and servers (HATEOAS)
  - A service may choose to only support a subset of EPP functionality, this should be discoverable
  - Extensions supported by the server should be discoverable

SIDN LABS

EPP Compatibility

- Provide equivalents for core EPP functions and objects (domains, hosts, and contacts)
- Mapping of data structures, unidirectional or bidirectional? (RPP <-> EPP)
- Support for (all/most/some?) existing EPP extensions
- Compatibility profile for signalling level of EPP compatibility
- Reconsider some EPP design decisions

SIDN LABS

Security

- Support for scalable modern authentication standards (OAuth2.0, OIDC)
- Support for digital identities (Verifiable Credentials)
- Granular authorisation model
- Improve the current auth-code based authorisation for transfer request

- Allow for potential additional use cases:
  - DNS providers (NS-set, DNSEC keys)
  - Registry Lock
  - Renewals

Extensibility

- Design for extensibility, enable protocol evolution, e.g. new;
  - Resources
  - Data structures
  - Data formats (mediatype)
  - Authentication and authorization methods

Scalability & Performance

- No (or optional) request payload when not required (Check/Info)
- Load balancing at request level
- Response caching

Design Process

- Learn from work on existing extension frameworks (EPP, RDAP)
- Try to go fast without breaking things
- Lightweight process:
  - Designing requirements (not an RFC)
  - Developing documents
  - Publishing extensions

- Implementer-friendly documentation (OpenAPI)


SIDN.nl


@SIDN


SIDN

Thank You

www.sidnlabs.nl | stats.sidnlabs.nl