RPP – Core Update

Maarten Wullink - SIDN Labs

Pawel Kowalik - DENIC

IETF 125, Shenzen, 2026

Updates – Version 04 & 05

- Bootstrapping
- Discoverability
- Versioning
- Profiles
- IANA Registrations

Bootstrapping

- Bootstrapping is the process for locating a RPP discovery endpoint

- Two mechanisms defined for locating an RPP server

  1) IANA registry for RPP servers

  - TLD (e.g. example)
  - URL (discovery endpoint, e.g. https://rpp.example/.well-known/rpp)
  - Description

Bootstrapping

2) DNS lookup using an SRV (RFC2782)

- DNS SRV record MUST be **_rpp._tcp.<zone>**

  “_rpp._tcp.example. 3600 IN SRV 0 0 443 rpp.example.”

- Target point to well-known endpoint

  “https://rpp.example/.well-known/rpp”

- Support for multiple locations using the SRV priority option

- Absence indication using single dot notation

Discoverability

- RPP server capabilities MUST be discoverable by clients.
- The server provides a well-known endpoint (/.well-known/rpp)
- The Discovery document includes the information required by the client:
  - Base URL
  - RPP Version
  - Tlds
  - Extensions
  - Profiles
  - Objects
  - Authentication Type
  - Endpoints?
  - Maintenance notices

Discoverability

Discovery document

Example


```
{
  "base_url": "https://rpp.example/rpp/v1",
  "version": "1.0",
  "tlds": ["example", "org"],
  "extensions": [
    {
      "name": "RPP example extension",
      "id": "urn:ietf:params:rpp:extension:example-extension",
      "version": "1.0",
      "url": "https://www.iana.org/assignments/rpp-extensions/rpp-example-extension-1.0"
    }
  ],
  "profiles": [
    {
      "name": "EPP compatibility profile",
      "id": "urn:ietf:params:rpp:profile:epp-compatibility",
      "version": "1.0",
      "url": "https://www.iana.org/assignments/rpp-profiles/epp-compatibility-provisioning-profile-1.0"
    }
  ],
  "objects": ["domains", "hosts", "entities"],
  "endpoints": [
    {
      "name": "availability",
      "url_template": "/{collection}/{id}/availability"
    },
    {
      "name": "info",
      "url_template": "/{collection}/{id}"
    },
    {
      "name": "poll",
      "url_template": "/messages"
    },
    {
      "name": "create",
      "url_template": "/{collection}"
    },
  ],
  "authentication": ["Bearer"],
  "maintenance": [
    {
      "start_time": "2026-06-01T00:00:00Z",
      "end_time": "2026-06-01T06:00:00Z",
      "description": "Planned maintenance for server upgrades"
    }
  ]

}
```


SIDN LABS

Workflow

- Typical client workflow when using bootstrapping and discoverability :

1) Bootstrap (optional): Lookup the location of the RPP server by using one of two defined mechanisms

2) Discover capabilities (optional): Retrieve the capabilities of the RPP server from the well-known endpoint at /.well-known/rpp

3) Extract RPP URLs (optional): Extract the base URL and endpoint URL templates from the discovery response

4) Perform provisioning operations

Versioning

- Semantic Versioning (x.y.z) for extensibility and backwards compatibility of these elements:
    - URL Endpoints (encoded in URL)
    - Request and Response Messages (version property)
    - Extensions (Profile, Discovery document)
    - Profiles (Profile, Discovery document)
    - ~~Result Codes~~

Versioning – Examples

Discovery document versioning examples

- Extensions

```
"extensions": [
    {
      "name": "RPP example extension",
      "id": "urn:ietf:params:rpp:extension:example-extension",
      "version": "1.0",
      "url": "https://www.iana.org/assignments/rpp-extensions/rpp-example-extension-1.0"
    }
  ],
```

- Profiles

```
"profiles": [
    {
      "name": "EPP compatibility profile",
      "id": "urn:ietf:params:rpp:profile:epp-compatibility",
      "version": "1.0",
      "url": "https://www.iana.org/assignments/rpp-profiles/epp-compatibility-provisioning-profile-1.0"
    }
  ]
```

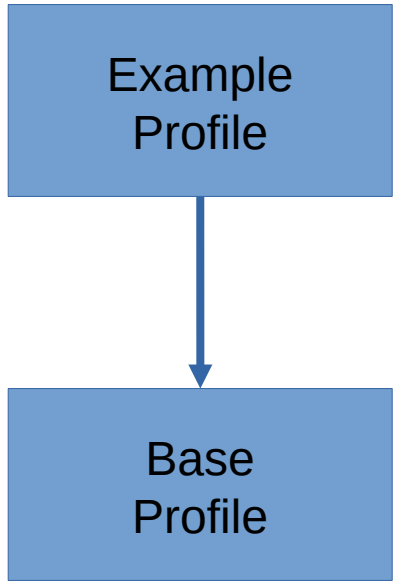
SIDN LABS

Profiles

- A profile is a named set of protocol features and versions that are used to define the compatibility and capabilities of an RPP server
- Promote interoperability and standardization across implementations
- Profile data:
  - Name
  - Description
  - Profile version
  - Minimum RPP version
  - Objects supported
  - Extensions supported
  - Base Profile

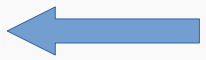
Profiles - Inheritance

- A profile can extend another registered profile
  - The new profile inherits all properties from the base profile
  - May add new properties or override properties from base profile



Profiles - Example

```
{
  "name": "urn:ietf:params:rpp:profile:example-profile",
  "description": "An example profile for provisioning objects using the RPP protocol.",
  "version": "1.0",
  "rpp_version": "1.0",
  "objects": ["domains", "hosts", "entities"],
  "extensions": [
    {
      "rppExample": {
        "version": "1.0"
      },
      "rppOther": {
        "version": "1.1"
      }
    }
  ],
  "profile": {
    "name": "urn:ietf:params:rpp:profile:base-profile",
    "version": "1.0"
  }
}
```



Updates rppOther version from base profile

Profiles - Signalling

- Two different mechanisms for client to signal what profile is used
  - HTTP Header (Rpp-Profile)

```
RPP-Profile: profile=urn:ietf:params:rpp:profile:example-profile;version=1.0
```

  - Media Type parameter (profile)

```
Accept: application/rpp+json; profile="urn:ietf:params:rpp:profile:example-profile"; version="1.0"
Content-Type: application/rpp+json; profile="urn:ietf:params:rpp:profile:example-profile"; version="1.0"
```

IANA - Requests

- Request to IANA for creating a RPP registry group with the following sub-registries:
  - Discovery registry
  - Extension registry
  - Profile registry
  - Result Codes registry

Additional requests:

- URN namespace: urn:ietf:params:rpp (do we need it?)
- Media Type: application/rpp+json

Bootstrapping

- Two bootstrap mechanisms is probably not a good idea
  - Possibility for out-of-sync data

- Possible solutions include:

  1) Drop 1 of the mechanisms

  2) Only allow single mechanism to be used per deployment

Question: Profile Signalling

- What is best mechanism for client to signal what profile is used?
  - HTTP Header (Rpp-Profile)
  - Media Type parameter (profile)

Next steps

- Sync with latest versions of Data Objects and JSON documents
- Ask to adopt as wg document


SIDN.nl


@SIDN


SIDN

Thank You

www.sidnlabs.nl | stats.sidnlabs.nl