IETF 123
RPP Requirements

Maarten Wullink and Pawel Kowalik (SIDN & DENIC)

Requirements

The requirements document needs to be stable and there needs to be consensus, before continuing with the design phase.

- Interested registries and registrars should take notice
  - The requirements document is planned to be ready for starting design phase by IETF 124 in Montreal
  - Changing requirements is going to become more difficult during the next design and development phases

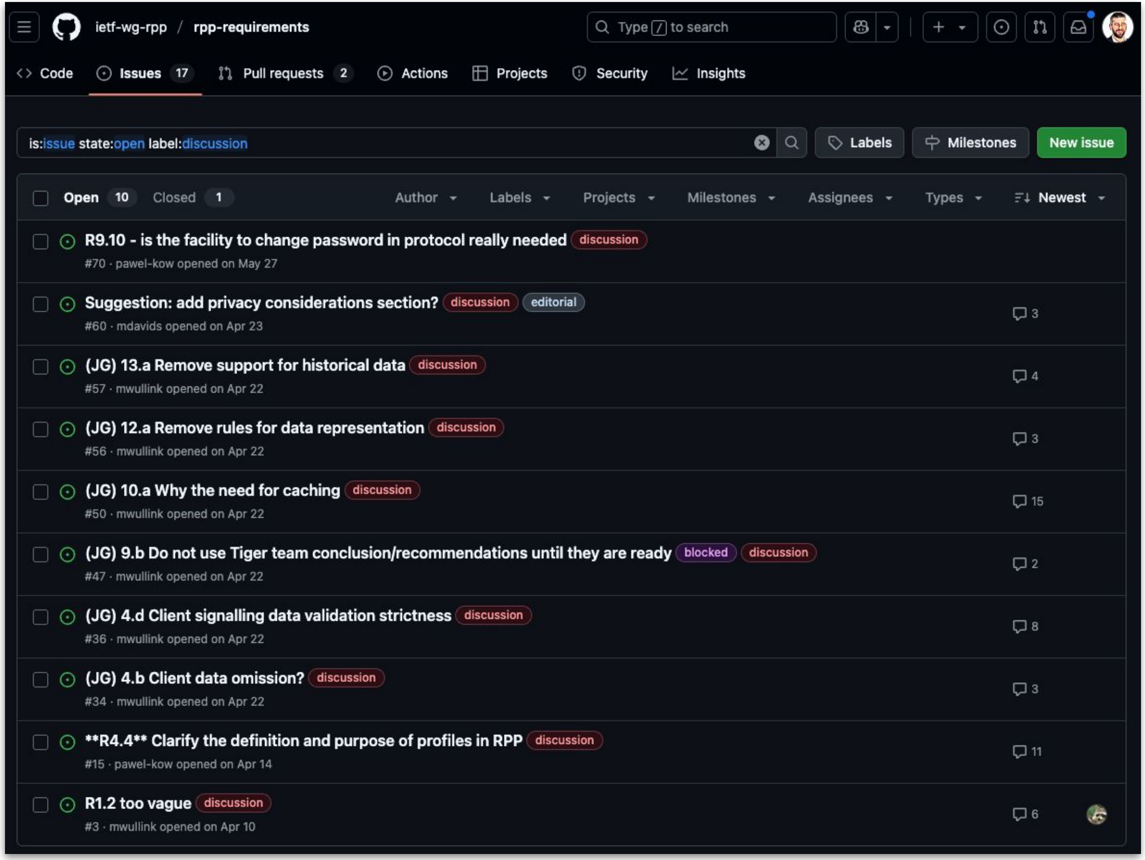
Requirements Highlights

- Based on HTTP and REST architectural style
- Developer friendly API, easy integration of existing systems
- Focus on performance and scalability
- Functionally equivalent to EPP (mostly)
- Extensible and support for existing EPP extensions
- Common EPP extensions (DNSSEC) are included in the core protocol
- Support for multiple data formats, the default will be JSON
- Discovery mechanism for server location and functionality
- Authentication framework allows more advanced use cases (transfer, NS-set updates)
- Future proof (DELEG)

Open Issues

- Multiple issues require wg input
  - Poll question for each issue
  - CENTR Jamboree RPP session results
- Use feedback for -02 version and as start for design phase

Open Issues in GitHub



is:issue state:open label:discussion

Open 10 Closed 1

- R9.10 - is the facility to change password in protocol really needed `discussion`
  #70 · pawel-kow opened on May 27
- Suggestion: add privacy considerations section? `discussion` `editorial` — 3
  #60 · mdavids opened on Apr 23
- (JG) 13.a Remove support for historical data `discussion` — 4
  #57 · mwullink opened on Apr 22
- (JG) 12.a Remove rules for data representation `discussion` — 3
  #56 · mwullink opened on Apr 22
- (JG) 10.a Why the need for caching `discussion` — 15
  #50 · mwullink opened on Apr 22
- (JG) 9.b Do not use Tiger team conclusion/recommendations until they are ready `blocked` `discussion` — 2
  #47 · mwullink opened on Apr 22
- (JG) 4.d Client signalling data validation strictness `discussion` — 8
  #36 · mwullink opened on Apr 22
- (JG) 4.b Client data omission? `discussion` — 3
  #34 · mwullink opened on Apr 22
- **R4.4** Clarify the definition and purpose of profiles in RPP `discussion` — 11
  #15 · pawel-kow opened on Apr 14
- R1.2 too vague `discussion` — 6
  #3 · mwullink opened on Apr 10

Q1: Strict vs. lenient data validation (1)

R4.5 The RPP architecture MUST include loose coupling between the server and the client, *allowing for non-coordinated introduction of non-breaking version changes on both sides*.

R4.6 A RPP MUST have either a lenient validation mode, where *unknown properties are ignored, or a strict validation mode, where unknown properties are treated as an error.* The mode is up to client and server policy with mode signalling.

Q1: Strict vs. lenient data validation (2)

Strict

- Will to lead to serious protocol evolution issues
- The server must fully understand the request or reject it as whole

Lenient

- Very common for RESTful APIs and JSON representation to ignore unknown parameters or properties
- Default handling in OpenAPI is to ignore unknown properties

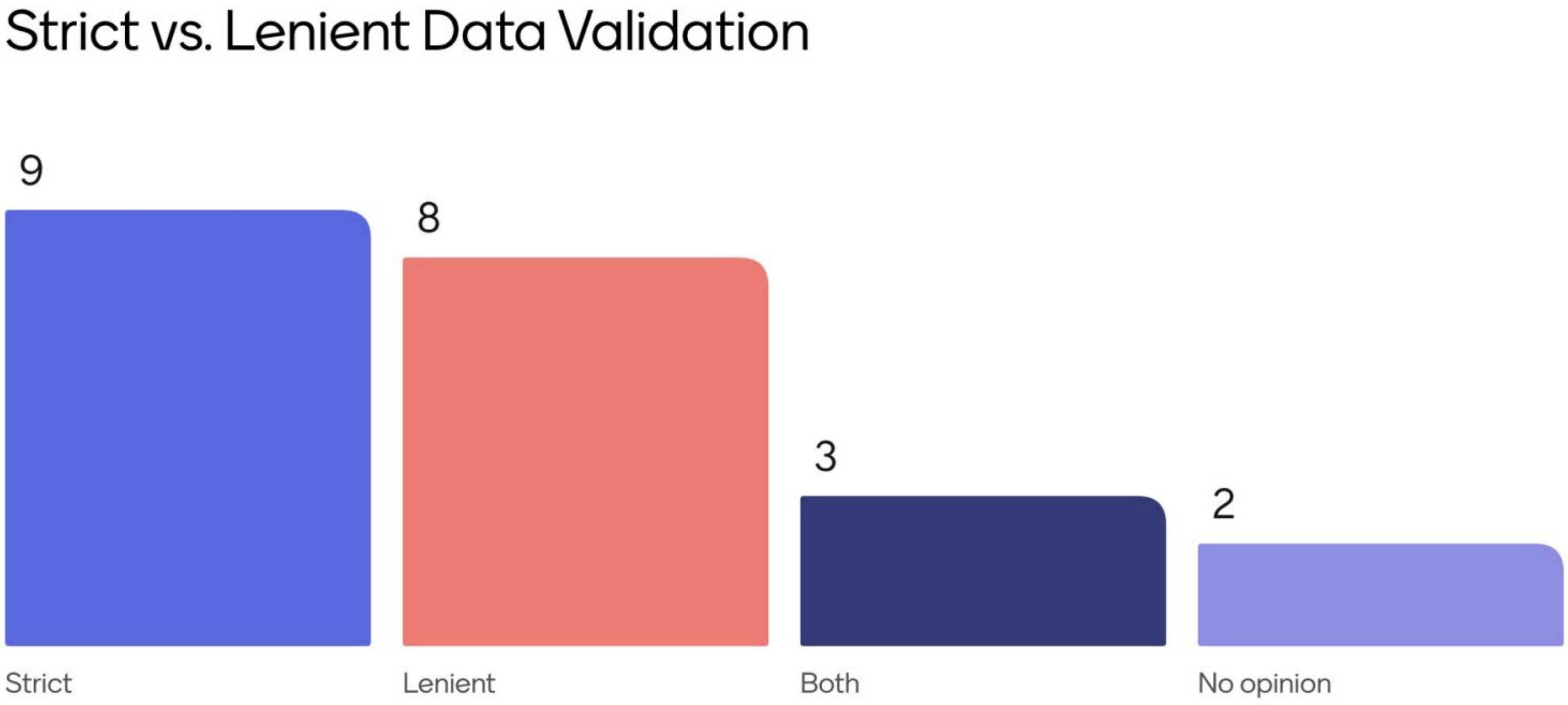
🖐 Show of Hands 🖐

How should RPP support data validation?

- Both
- Lenient
- Strict

Split up into 3 poll questions, due to MeetEcho limitations

CENTR Jamboree - Results



Q2: Response Caching (1)

R11.2 If applicable to the operation semantics (e.g. read operations) RPP MUST support cacheability of the response body using standard HTTP mechanisms.

- RPP MUST support != every server or client MUST
- HTTP has built-in support for caching

Q2: Response Caching (2)

- RFC9205 (**Building Protocols with HTTP**) Section: 4.9:

  “HTTP caching is one of the primary benefits of using HTTP for applications; it **provides scalability, reduces latency, and improves reliability** …..

  Even when an application using HTTP isn't designed to take advantage of caching, it needs to ***consider how caches will handle its responses to preserve correct behaviour*** ….”

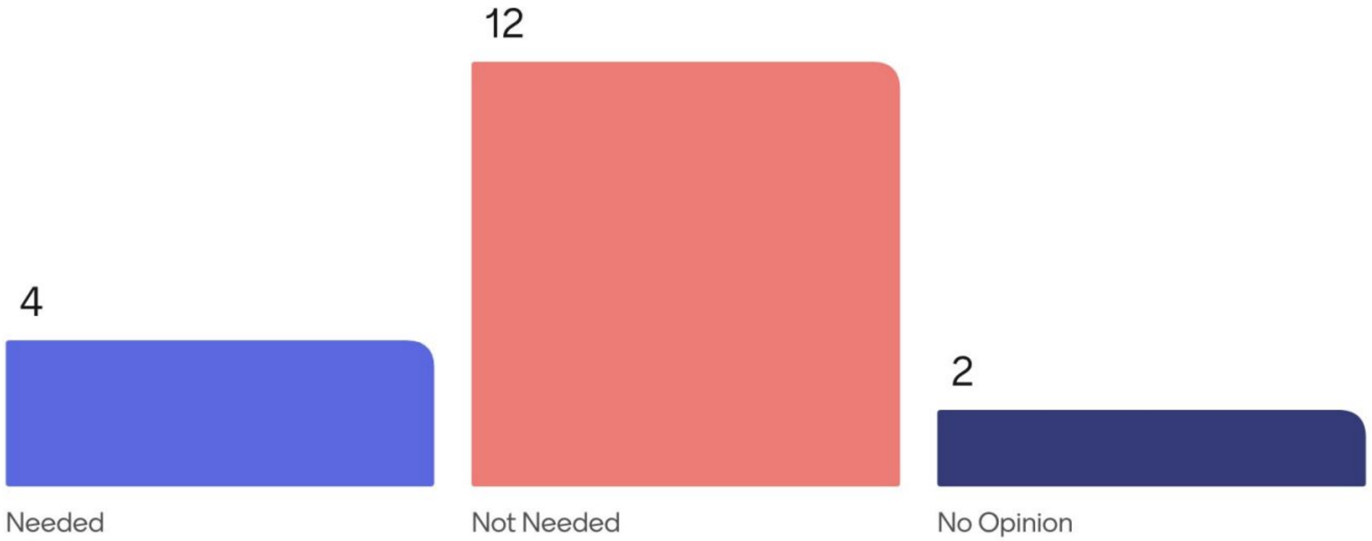
🖐 Show of Hands 🖐

Should RPP consider caching in the requirements?

- Yes
- No

CENTR Jamboree - Results

Response Caching



Q3: Historical Data (extension)

A.3 An extension that allows generating a historical overview for an object, e.g. show all events linked to the object (create, update ...). The historical time window is determined by server policy and is included in the discovery service document.

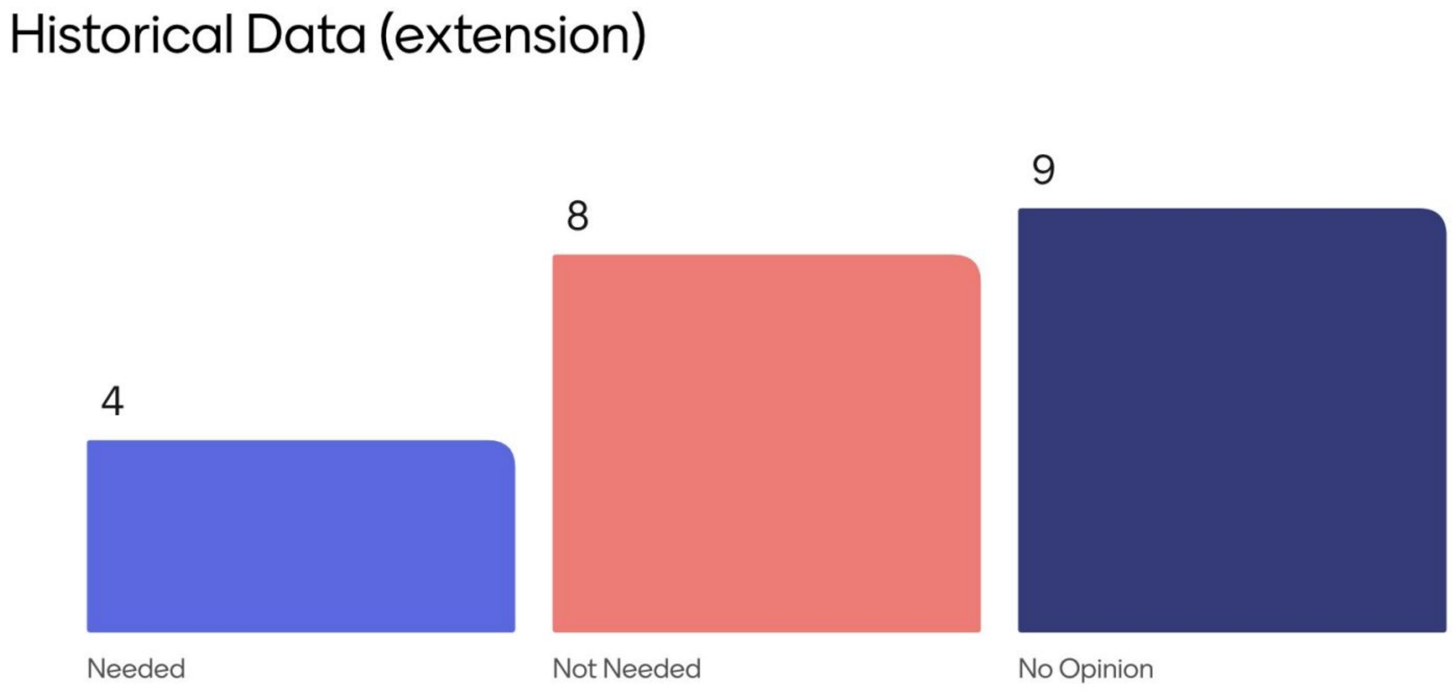
- WhoWas strikes back.
- Also include historical overview for processes?

🖐 Show of Hands 🖐

Should RPP consider extensibility to include historical data?

- Yes
- No

CENTR Jamboree - Results



Q4: Bodiless requests and responses

The RESTful architectural style does not require the use or HTTP request and response message body.

- HTTP HEAD/GET/DELETE no need for request message body
- HEAD/DELETE no need for response message body (HTTP headers only)

Complications:

- EPP CHECK or DELETE command may use extension and return data
- CHECK with a list of object is a scalability issue

This means part of the protocol must be covered by HTTP headers

Q4: Bodiless issue updated in -01

Before:

R12.1 RPP MUST NOT include a HTTP message body in the request or response when this is not necessary, for example when the required data can be transmitted using the URL and/or HTTP headers.

After:

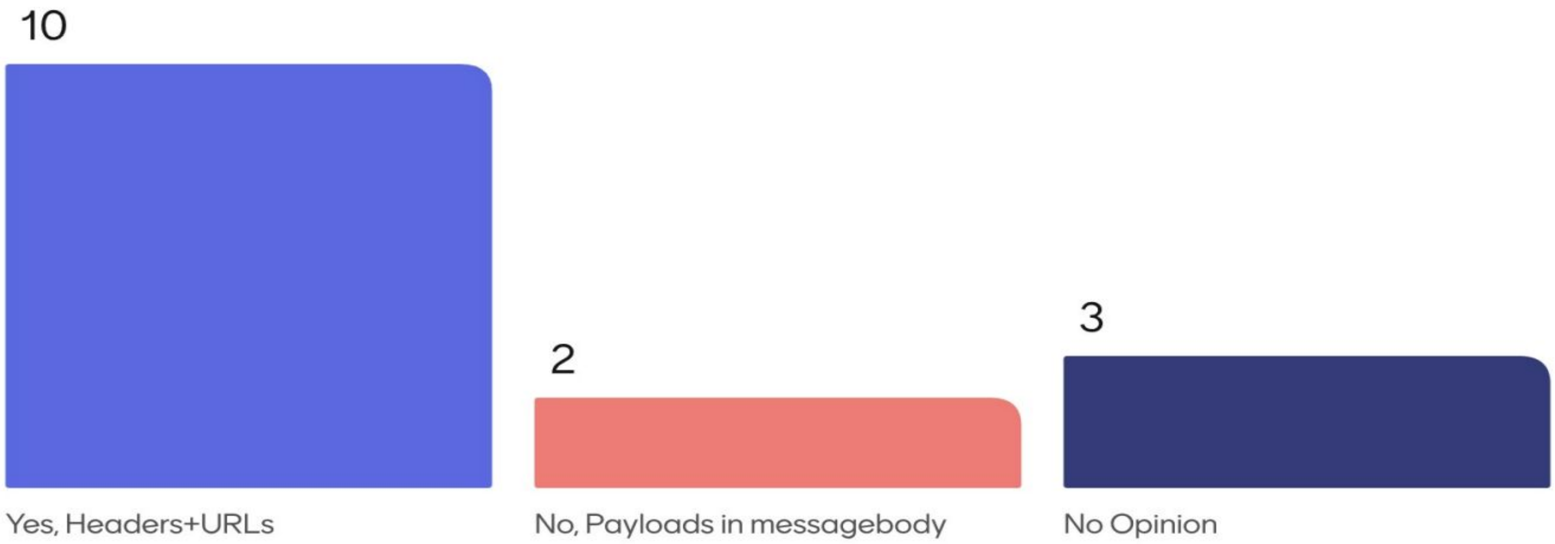
R12.1 In order to minimise message sizes and needed processing ***RPP SHOULD be designed not to include a HTTP message body in the request or response when this is not necessary,*** for example when the required data can be transmitted using the URL and/or HTTP headers.

🖐 Show of Hands 🖐 (only if needed)

Should RPP use Bodiless requests and responses?

- Yes, headers + URLs are good
- No, the data needs to be in the message body

CENTR Jamboree - Results



Body-less Requests and Responses

Q5: Facility for changing password in protocol

R9.10 RPP MUST allow users to update their credentials and enforce strong passwords and limited lifetime for passwords and other tokens.

In EPP there is a <newPW> element of <login> command for password change.

However:

- May conflict with commonly used standards:
  - OAuth2 / OpenID Connect
  - Token-based auth (e.g. JWTs)
- It violates separation of concerns, protocol should focus on data transport
- Increases implementation complexity
- Generally authentication and credential management should be outside of protocol

🖐 Show of Hands 🖐

Is a facility for changing password in protocol needed?

- Yes
- No

Q6: Transaction information in the headers?

J. Gould would like to remove:

R6.3 The data representation in a RPP response MUST only contain data related to the object, transactional information MUST be represented as one or more separate HTTP headers.”

Argument for keeping R6.3:

- Enables caching of object representations, which are independent of transactional information
- Bodiless responses, there is no payload to carry transaction information
- Processing, status information in the headers allows clients to make choices about processing flow without processing response body

🖐 Show of Hands 🖐

Should RPP have transactional information (e.g. clTRID) in the headers vs. payload?

- Yes, headers
- No, payload

Q7: Client data omission

R4.3 RPP MUST allow for an extension mechanism that allows clients to signal data omission or redaction, indicating data collected but not transmitted to the registry or redacted.

- Motivation is similar to RDAP redaction
- Client would want to skip some “required” data and signal it instead of sending “dummy” data
- One may say, that this data shall not be “required” in the first place
- In practice, for RPP data models it might mean, that in the base protocol all data elements must be optional so that they can be omitted

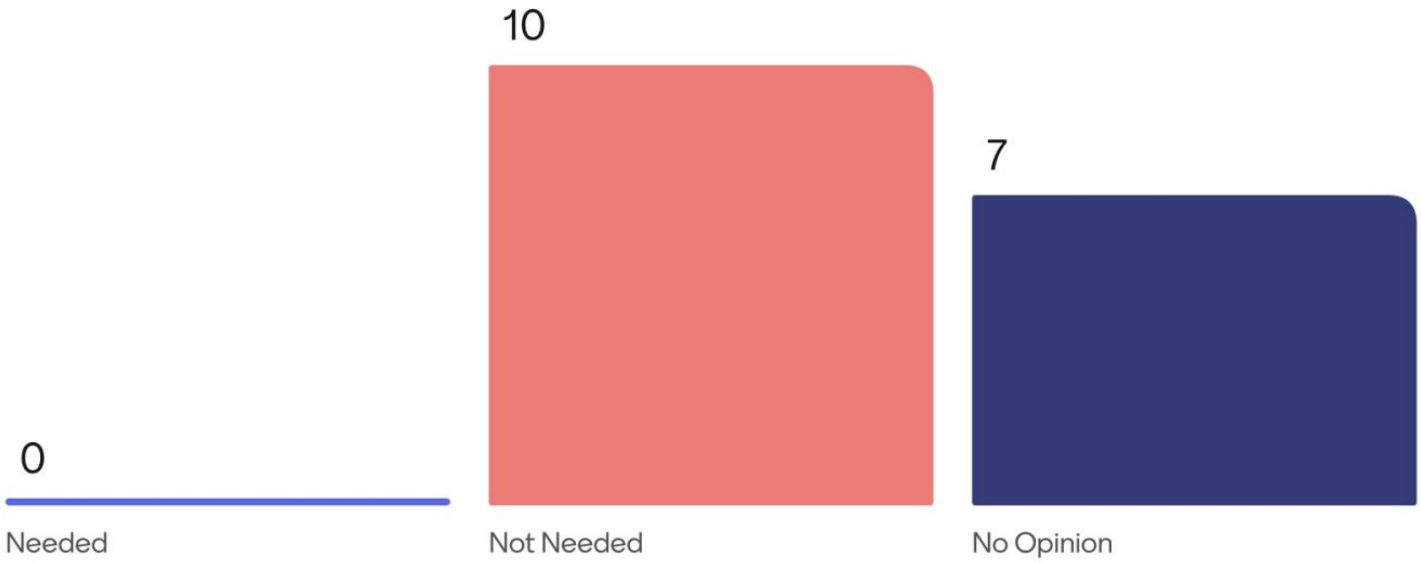
🖐 Show of Hands 🖐

Should RPP consider data omission by the client as an use-case and requirement?

- Yes
- No

CENTR Jamboree - Results

Client Data Omission



Q8: R1.2 too vague

R1.2. RPP MUST provide a clear, clean, easy to use and self-explanatory interface that can easily be integrated into existing software systems.

A. Newton:

Who will be the judge of something being "clear" and "clean" and "easy to use"? All of that is very subjective.

- This was meant to be a meta requirement
- Yes, it is vague and subjective, meant to reflect “**easier integration between registries and registrars**” from the charter
- Is removing the requirement same as saying “RPP MAY be clunky, difficult to use, integrate and understand”?

🖐 Show of Hands 🖐

R1.2. is needed (potentially change text)?

- Yes
- No

Q9: Purpose of profiles

R4.4 RPP MUST have mechanisms to define profiles to indicate:

- Required parts of the data model
- Mapping definitions
- Functional subsets for compatibility.

J. Gould raised the point that it is extremely complicated to map all possible server configurations, so it should rather be an extension - therefore not a MUST requirement.

- This is a question about how much discoverability RPP should be having
- Similar in scope to what profiles in RDAP are defining

🖐 Show of Hands 🖐

Should RPP support profiles as MUST requirement?

- Yes
- No

Are the requirements complete?

We have a limited number of reviews and discussions for the current set of requirements.

- Is the requirements set complete?
- Are relevant requirements missing?
    - For example: Do we need requirements related to privacy?

Please review and provide proposals for missing requirements.

Thank you

Additional Questions (Time permitting)

🖐 Show of Hands 🖐

Should RPP support Thick AND Thin registry models?

- Yes
- No

🖐 Show of Hands 🖐

RPP needs to support all EPP features, even rarely used (e.g. Contact Transfer)?

- Yes
- No