draft-harrison-regext-rdap-mirroring

●●●

Tom Harrison (tomh@apnic.net)
George Michaelson (ggm@apnic.net)
Andy Newton (andy@arin.net)
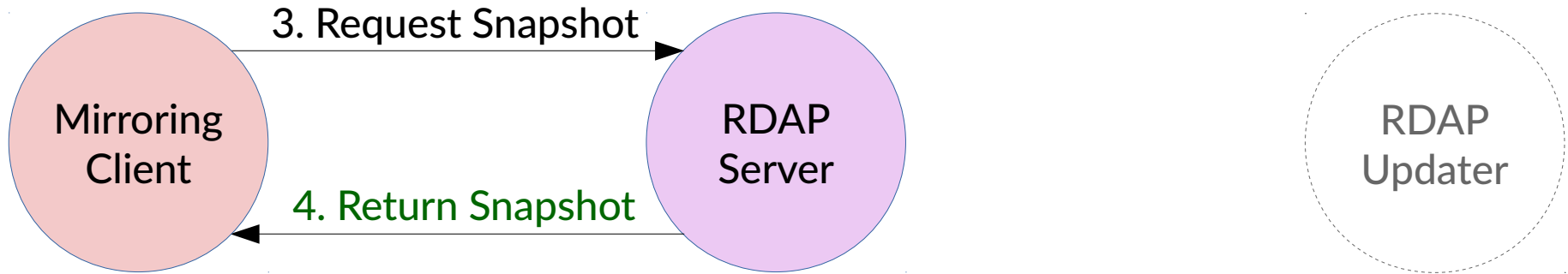IETF 104 REGEXT Working Group

What is RDAP mirroring?

- A protocol for retrieving all of the data from an RDAP server's data store, and for keeping a local copy of that data up to date over time

- Use cases:
    - When the overhead of querying the remote RDAP server is too high
    - When there's a need to analyse the RDAP data set as a whole
    - When a client wants to provide access in its own right to the remote RDAP server's data

How does it work?

- The server hosts an **update notification file (UNF)**, which contains links to a **snapshot** and a series of **deltas**

- A **snapshot** contains all of the server's RDAP data as at the time of generation

- A **delta** is used to record changes that have happened since the snapshot was generated
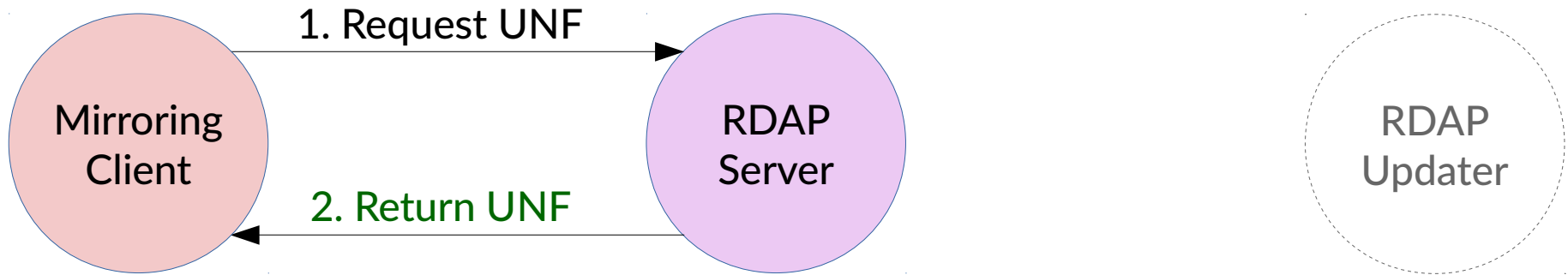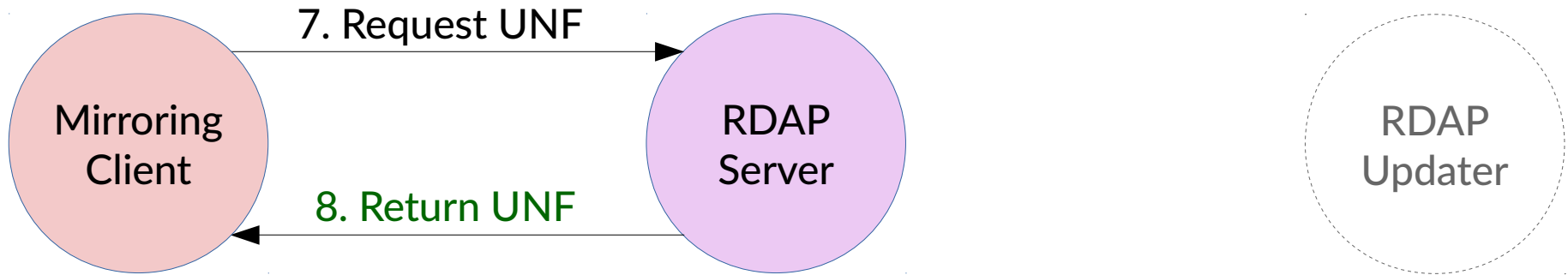
```
{ "version": 1,
  "refresh": 3600,
  "snapshot": {
    "uri": "https://example.com/snapshot-1.json",
    "serial": 1,
  },
  "deltas": [] }
```

Mirroring Client → 3. Request Snapshot → **RDAP Server**

4. Return Snapshot ← (RDAP Server → Mirroring Client)
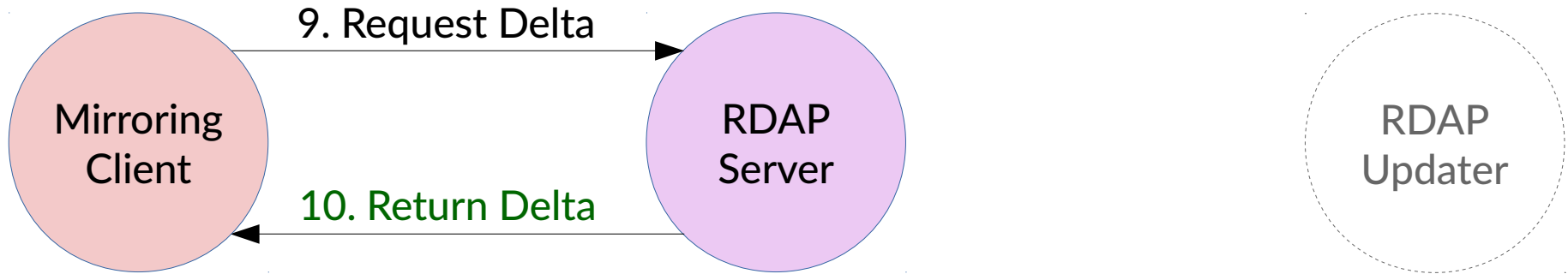
RDAP Updater

```
{ "version" : 1,
  "serial": 1,
  "objects": [
    { "id": "https://example.org/I1",
      "object": {
        "rdapConformance": [ "rdap_level_0" ],
        "objectClassName": "ip network",
        "startAddress": "192.0.2.0",
        ... } },
    ...
  ] }
```
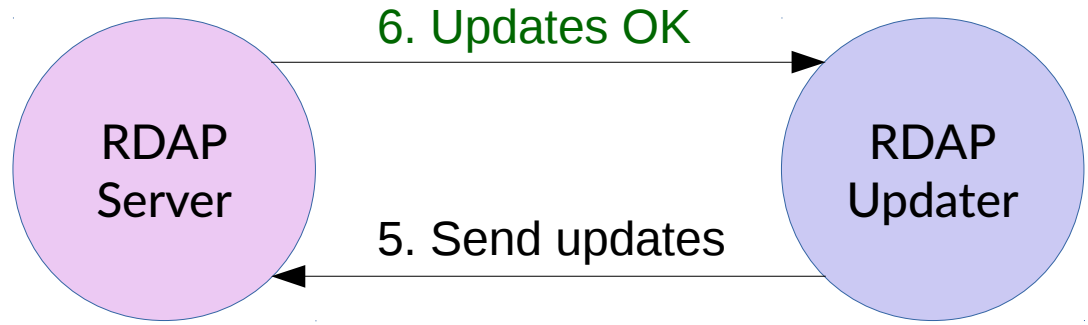
```
{ "rdapConformance": [ "rdap_level_0" ],
  "objectClassName": "ip network",
  "startAddress": "192.0.2.0",
  "name": "updated-name",
  ... }
```

```
{ "version": 1,
  "refresh": 3600,
  "snapshot": {
    "uri": "https://example.com/snapshot-1.json",
    "serial": 1
  },
  "deltas": [
    { "uri": "https://example.com/delta-2.json",
      "serial": 2 }
  ] }
```

```
        9. Request Delta
Mirroring  ──────────────▶   RDAP              RDAP
Client                       Server           Updater
        10. Return Delta
       ◀──────────────
```

```json
{ "version" : 1,
  "serial": 2,
  "removed_objects": [],
  "added_or_updated_objects": [
    { "id": "https://example.org/I1",
      "object": {
        "rdapConformance": [ "rdap_level_0" ],
        "objectClassName": "ip network",
        "startAddress": "192.0.2.0",
        "name": "updated-name",
        ... } },
  ] }
```

Miscellaneous features

- Define default set of response attributes
  - Terms and conditions, `port43`, etc.
- Refer to other RDAP objects by reference
- Server may specify requested refresh rate

Open topics

- Is this useful in the domain name context?

- Not looking for adoption just yet