

Efficient RDAP Referrals

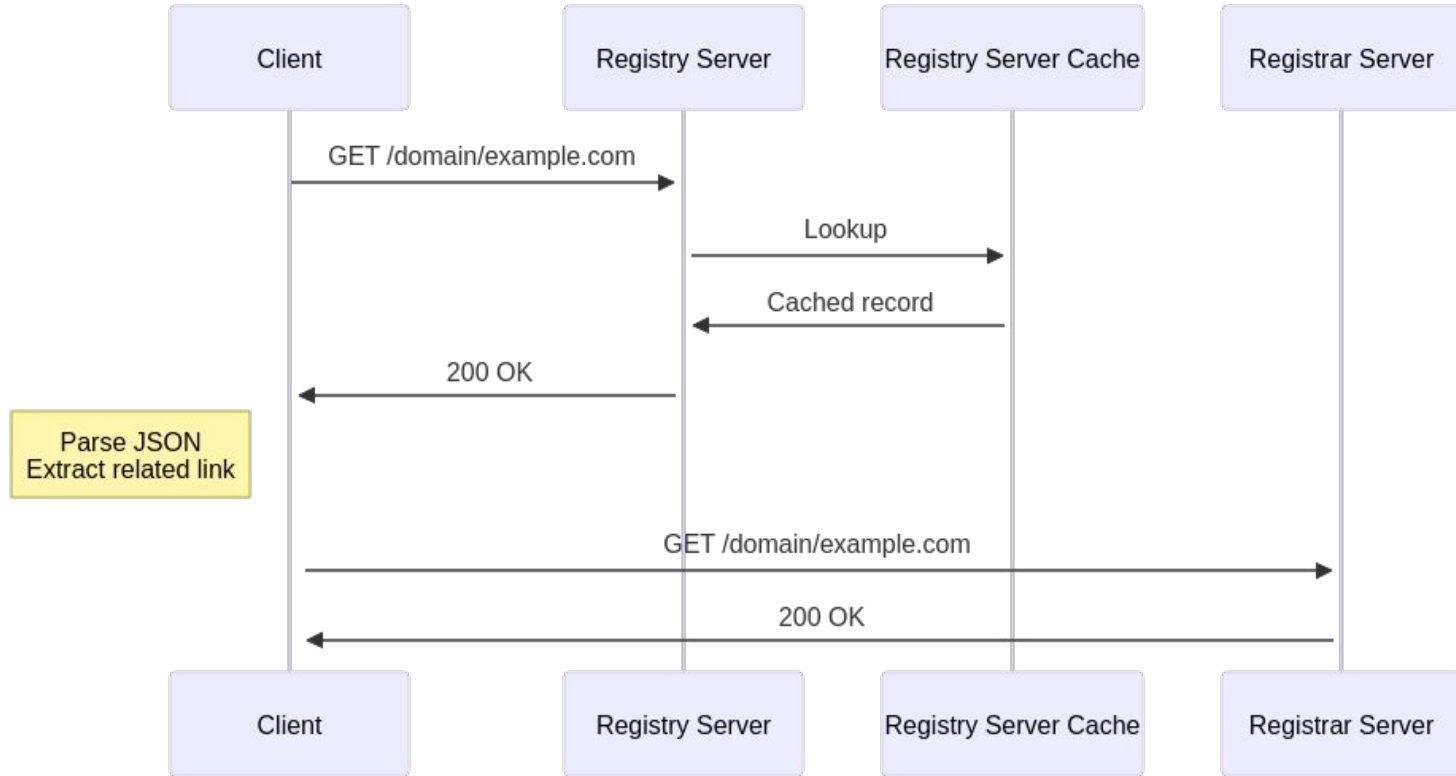
Gavin Brown
Andy Newton } ICANN

IETF 122, Bangkok, March 2025

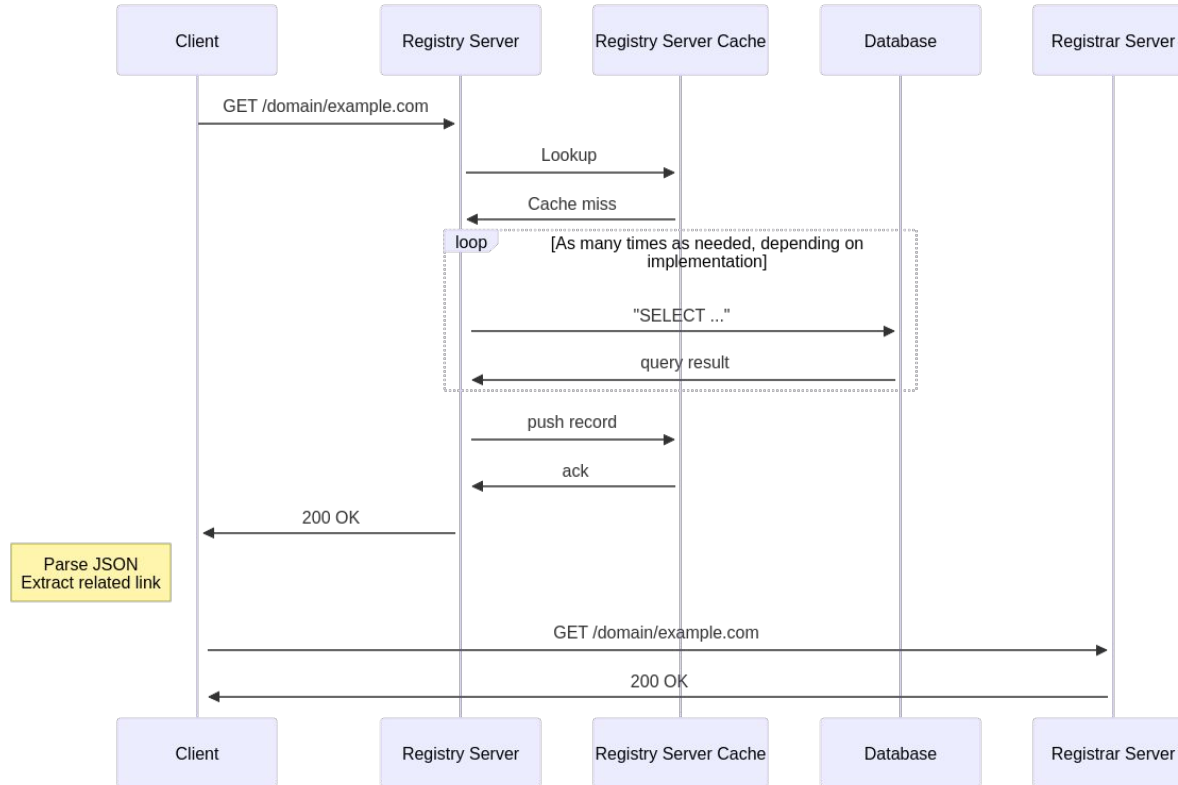
RDAP client behaviour

- Most RDAP usage is automated bulk requests, not human-initiated ad-hoc requests.
- For gTLD domain names, RDAP consumers are usually more interested in the RDAP record of the registrar than of the registry.
- To obtain the URL of the registrar record, the client has to query the registry server and extract the “related” RDAP link.
- What does that look like?

RDAP referrals: happy path (cache hit)



RDAP referrals: default path (cache miss)



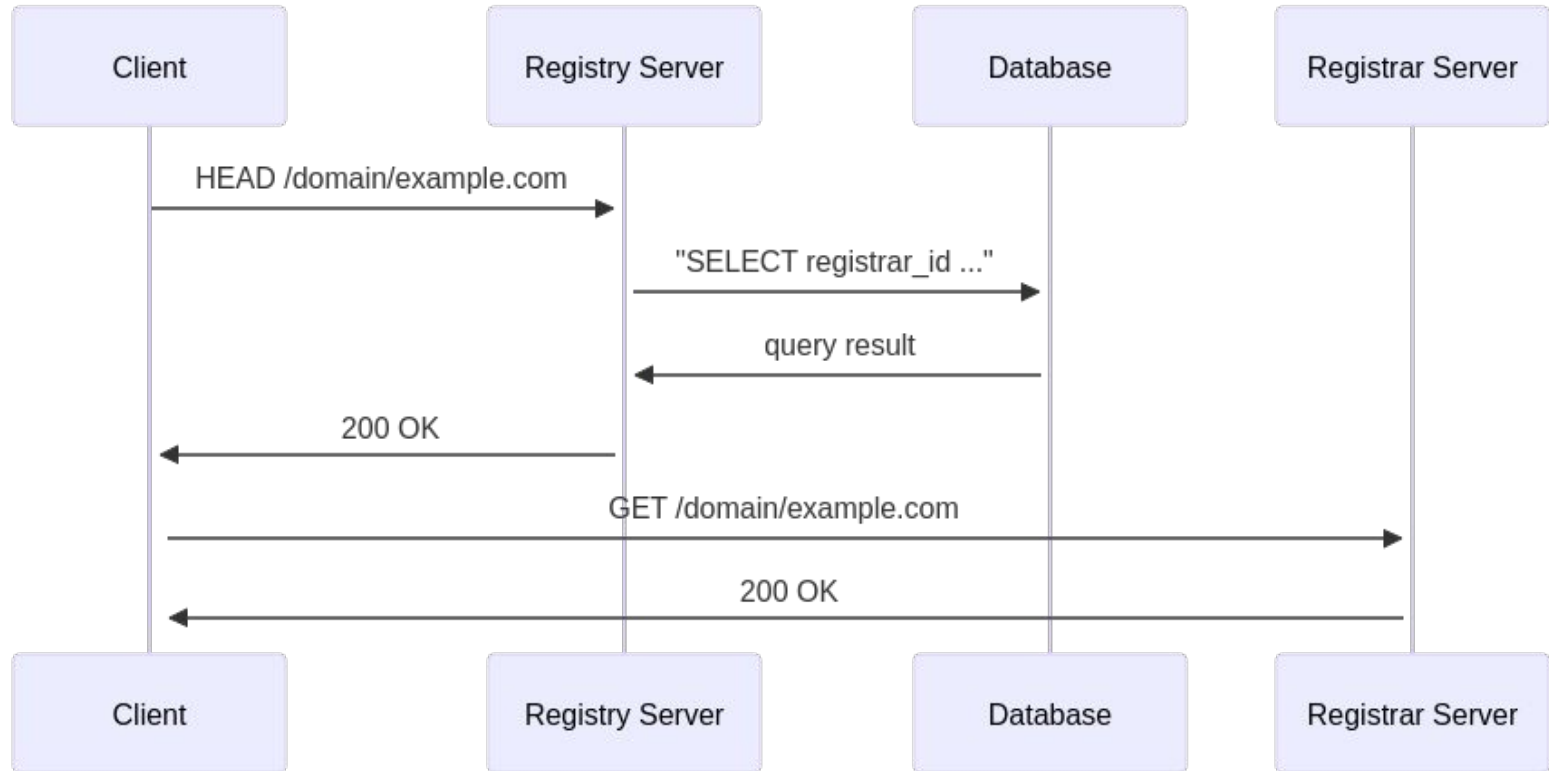
The problem with RDAP referrals

- Cache hit rates are low (20-30%) so the “happy” path is not the most likely one, except where a registry operates a hot cache.
- Most of the registry RDAP record is not useful to the client and is thrown away.
- A lot of the work being done by the server isn’t needed to satisfy the client.
- The client spends time waiting for the server to do things it’s not interested in.
- This results in wasted time, compute and bandwidth.

The problem with RDAP referrals (ctd)

- Assuming:
 - ~5Bn registry RDAP queries per month (source: monthly reports).
 - Average 1,071ms to receive an answer (394ms for a minimal answer).
 - Average 7,958 bytes for an answer (2,266 bytes for a minimal answer).
- Therefore:
 - 1,902 months generating/waiting for RDAP responses.
 - 32.4PB of data transferred, most of which is thrown away.
- As RDAP query volumes grow, this will only get worse.

Efficient RDAP referrals



Efficient RDAP referrals

- Client sends a **HEAD** request to the registry server.
- Server includes **Link:** header(s) in the response.
- Client can then immediately initiate the request to the registrar server.
- This is functionally equivalent to a 30X redirect.
- Performance-wise, better than the happy path because the client doesn't need to parse the JSON response.
- If **Link:** header(s) not present in response, fall back to HEAD.
- Or: default to **GET** for the first request, check **rdapConformance**, and use **HEAD** for subsequent requests.

draft-brown-rdap-referrals-01

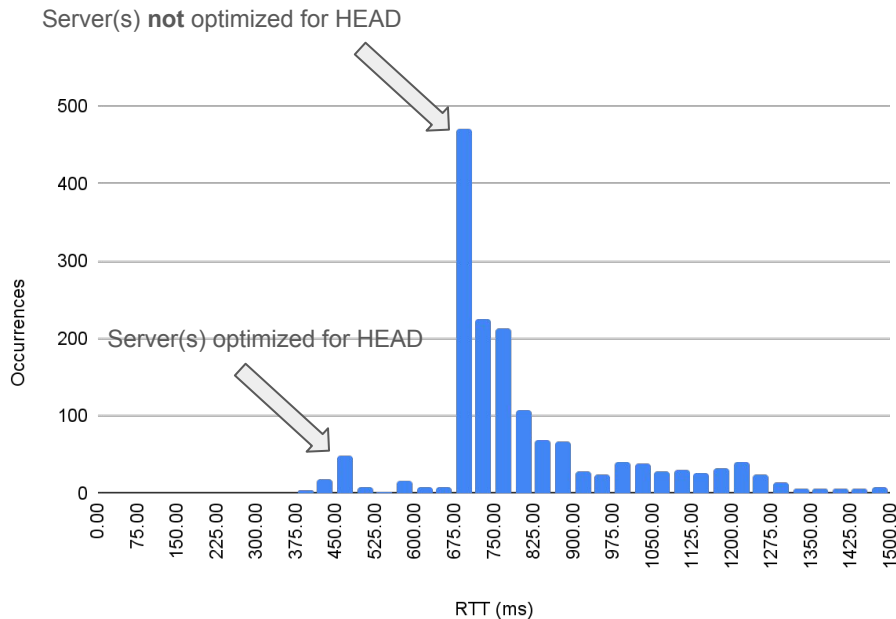
- Describes how RDAP link objects can be represented in **Link:** header fields, e.g.

```
Link: <https://rdap.example.com/domain/example.com>;  
      title="URL of Sponsoring Registrar's RDAP Record";  
      rel="related";  
      type="application/rdap+json"
```

- Conformant servers include one or more **Link:** header fields in responses to **HEAD** requests, and the **link_headers** value in the **rdapConformance** array.

Performance benefits

- Most gTLD RDAP servers don't seem to have been optimized for **HEAD** requests: the average RTT for a **HEAD** request is only marginally shorter than for **GET**.
- The histogram of the RTT of **HEAD** queries suggests that if the server is optimised for **HEAD**, it could be twice as fast as **GET**.



Questions for the WG

- The authors think this is a real problem.
- Does the WG agree this is a problem worth solving?
- Would an Experimental RFC be a better choice?