RDAP Extension for TTL Values

draft-ietf-regext-rdap-ttl-extension

Gavin Brown <gavin.brown@icann.org>

IETF 124, Montreal

Document Status

- Document adopted by the Working Group (thank you!)
- Discussion on the mailing list about best way to represent TTL values, but no consensus.
- I will present two options and outline their pros and cons.

Option #1: Array-Oriented

Example:

```
"ttl": [
  {
    "types"   : [ "DS", "NS" ],
    "value"   : 300,
    "events"  : [ /* events */ ],
    "remarks" : [ /* remarks */ ]
  },
  /* other types */
]
```

Pros:

- Compresses the response (value, events and remarks are not duplicated).

Cons:

- Clients have to grep through arrays to find the value they are searching for.

Option #2: Object-Oriented

Example:

```
"ttl": {
  "NS": {
    "value"   : 300,
    "events"  : [ /* events */],
    "remarks" : [ /* remarks */]
  },
  "DS": {
    "value"   : 300,
    "events"  : [ /* events */],
    "remarks" : [ /* remarks */]
  }
  /* other types */
}
```

Pros:

- Clients that want to directly access the data about a specific RR type can do so easily.

Cons:

- Clients that use mapping frameworks will have difficulty handling unknown RR types.
- The payload is larger on the wire due to duplication of events, remarks, etc.

My View

- I continue to lean towards Option #1 (Array-Oriented) because:
    1. A smaller payload is better for clients and servers.
    2. The array-oriented approach is widely used elsewhere (events, entities, remarks, etc).
    3. The extra complexity for clients is minimal.

Question

Is the array-oriented approach a deal-breaker?

That is, do you dislike it enough that you would object during Working Group Last Call?

Thanks!