DNS data representation for use in RPP

IETF124 Montreal, November 2025

Christian Simmen, DENIC eG

Recap: DNS resource records as a data model

DNS	`NAME TTL CLASS TYPE RDATA` `de. 21600 IN NS a.nic.de.`
RPP	see below

```
NAME    TTL      CLASS    TYPE    RDATA
de.     21600    IN       NS      a.nic.de.
```

RPP

```
{
  "@type": "Domain",
  "name": "de",
  "dns": {
    "records": [
      { "name": "@", "type": "ns", "rdata": { "nsdname": "a.nic.de." } }
    ],
    "controls": {
      "ttl": { "ns": "21600" }
    }
  }
}
```

Recap: Host attribute model

Host attribute model

```
{
  "@type": "Domain",
  "name": "example.com",
  "dns": {
    "records": [
      { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.de." } },
      { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.com." } },
      { "name": "ns.example.com.", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "ns.example.com.", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": {
      "ttl": {
        "ns":  "21600",
        "a": "21600",
        "aaaa": "21600"
      }
    }
  }
}
```

Recap: Host object model

Host object model

```json
{
  "@type": "Domain",
  "name": "example.com",
  "_object_references": {
    "nameserver": [
      { "name": "ns.example.com.", "href": "nameservers/ns.example.com" }
    ]
  },
  "dns": {
    "records": [],
    "controls": {}
  }
}
```

```json
{
  "@type": "Host",
  "name": "ns.example.com",
  "dns": {
    "records": [
      { "name": "@", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "@", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": {
      "ttl": {
        "a": "21600",
        "aaaa": "21600"
      }
    }
  }
}
```

Recap: DNSSEC

DNSSEC

```
{
  "@type": "Domain",
  "name": "example.com",
  "dns": {
    "records": [
      { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.de." } }
      {
        "name": "@",
        "type": "ds",
        "rdata": {
          "keyTag": "12345",
          "algorithm": "13",
          "digestType": "2",
          "digest": "BE74359954660069D5C632B56F120EE9F3A86764247C"
        }
      }
    ],
    "controls": {
      "ttl": {
        "ns": "21600,
        "ds": "86400"
      },
      "maximumSignatureLifetime": {
        "ds": "86400"
      }
    }
  }
}
```

Updates –00 to -01

- Included operational controls into the dns structure ($.dns.controls)
- Switched to **camel case** for consistency with other drafts
- Moved examples to appendix (also added a few RRTypes)
- Explicitly referencing the DNS RR **presentation format**
- Set datatypes to all "string"

Challenges - Host object model

- Host object may be in control of a different sponsoring client ("name server operator") than the domain object ("domain owner")
- With NS-style delegation the host name is enough information to "synthesize" resource records just with the relation between domain and host
- With DELEG other information can be provided (including future extensions):
  - Server IPs
  - Additional inclusion of DELEGI resource records
  - Nameless delegation
  - Available transports
  - DS records for multi-signer setups

Who is in control of the delegation?

Example: Domain object in control of delegation

Domain Object

```
{
  "@type": "Domain",
  "name": "example.com",
  "_object_references": {
    "nameserver": [
      { "name": "ns.example.com.", "href": "nameservers/ns.example.com" }
    ]
  },
  "dns": {
    "records": [
      { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.com." } },
      { "name": "@", "type": "deleg", "rdata": { "deleginfos": {
                  "server-name": "ns.example.com.", "server-ipv4": "192.0.2.1", "server-ipv6": "2001:DB8:::1" } }
    ],
    "controls": { "ttl": { "ns":  "21600" } }
  }
}
```

Host Object

```
{
  "@type": "Host",
  "name": "ns.example.com",
  "dns": {
    "records": [
      { "name": "@", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "@", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": { "ttl": { "a": "21600" }, { "aaaa": "21600" } }
  }
}
```

Domain object in control of delegation

Yes...

- Domain owner has full control of the delegation
- Domain label resource records in domain object
- Host label resource records (GLUE) in host object
- DELEG resides on domain level

But …

- Each domain has to be updated while renaming a host object or updating server capabilities
- DELEG capabilities configured per domain (unless "include-delegi" is used) limiting reuse

Example: Host object in control of delegation

Domain Object

```
{
  "@type": "Domain",
  "name": "example.com",
  "_object_references": {
    "nameserver": [
      { "name": "ns.example.com.", "href": "nameservers/ns.example.com" }
    ]
  },
  "dns": {
    "records": [],
    "controls": {}
  }
}
```

Host Object

```
{
  "@type": "Host",
  "name": "ns.example.com",
  "domain": {
    "dns": {
      "records": [
        { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.com." } }
        { "name": "@", "type": "deleg", "rdata": { "deleginfos": {
                     "server-name": "ns.example.com.", "server-ipv4": "192.0.2.1", "server-ipv6": "2001:DB8:::1" } }
      ],
      "controls": { "ttl": { "ns": "21600", "deleg": "21600" } }
    }
  }
  "dns": {
    "records": [
      { "name": "@", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "@", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": { "ttl": { "a": "21600", "aaaa": "21600" } }
  }
}
```

Host object in control of delegation

Yes ...

- Assumes the domain owner and the name server operator having trust
- Name server operator is in control of name server capabilities through DELEG

But …

- DELEG might be switched on/off without consent of the domain owner, shifting traffic from DELEG-aware name servers
- Domain label resource records defined in host label
- The meaning of "@" is different in "domain" and "dns" section which might be confusing
- DELEG has no concept of GLUE, therefore "Host object" might not be the right place

Example: Shared control of delegation

Domain Object

```
{
  "@type": "Domain",
  "name": "example.com",
  "_object_references": {
    "nameserver": [
      { "name": "ns.example.com.", "href": "nameservers/ns.example.com","type": ["ns", "deleg"] }
    ]
  },
  "dns": {
    "records": [],
    "controls": {}
  }
}
```

Host Object

```
{
  "@type": "Host",
  "name": "ns.example.com",
  "domain": {
    "dns": {
      "records": [
        { "name": "@", "type": "ns", "rdata": { "nsdname": "ns.example.com." } },
        { "name": "@", "type": "deleg", "rdata": { "deleginfos": {
                  "server-name": "ns.example.com.", "server-ipv4": "192.0.2.1", "server-ipv6": "2001:DB8:::1" } }
      ],
      "controls": { "ttl": { "ns": "21600" } }
    }
  }
  "dns": {
    "records": [
      { "name": "@", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "@", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": { "ttl": { "a": "21600", "aaaa": "21600" } }
  }
}
```

Shared control of delegation

Yes ...

- Separation of controlled aspects
- Domain owner in control of delegation type
- Name server operator in control of name server capabilities through DELEG

But …

- DELEG might be switched off without consent of the domain owner, shifting traffic from DELEG-aware name servers
- Domain label resource records defined in host label
- The meaning of "@" is different in "domain" and "dns" section which might be confusing
- DELEG has no concept of GLUE, therefore "Host object" might not be the right place

Templating?

Domain Object

```
{
  "@type": "Domain",
  "name": "example.com",
  "_object_references": {
    "nameserver": [
      { "name": "ns.example.com.", "href": "nameservers/ns.example.com", "type": ["ns", "deleg"] }
    ]
  },
  "dns": {
    "records": [],
    "controls": {}
  }
}
```

Host Object

```
{
  "@type": "Host",
  "name": "ns.example.com",
  "domain": {
    "dns": {
      "records": [
        { "name": "%domain%", "type": "ns", "rdata": { "nsdname": "ns.example.com." } },
        { "name": "%domain%", "type": "deleg", "rdata": { "deleginfos": {
                  "include-delegi": "%domain%.ns.example.net." } }
      ],
      "controls": { "ttl": { "ns": "21600" } }
    }
  }
  "dns": {
    "records": [
      { "name": "@", "type": "a", "rdata": { "address": "192.0.2.1" } },
      { "name": "@", "type": "aaaa", "rdata": { "address": "2001:DB8::1" } }
    ],
    "controls": { "ttl": { "a": "21600", "aaaa": "21600" } }
  }
}
```

Templating?

Yes ...

- Identifiers make the context explicit
- May also be useful for other objects (e.g. abuse contact URL)

But ...

- Should be defined in core protocol
- Added complexity

What is your opinion?

Thank you

Christian Simmen:
simmen@denic.de
https://www.linkedin.com/in/christian-simmen-47007b242/

Pawel Kowalik:
pawel.kowalik@denic.de
https://www.linkedin.com/in/pawelk/