DNS in JSON

Stéphane Bortzmeyer
AFNIC
bortzmeyer@nic.fr

14 March 2013
Reminder

- RFC 1034, 1035 and many others
- RFC 4627
- draft-bortzmeyer-dns-json-01
Big questions

1. Is it a good idea to define new formats for DNS data?
Big questions

1. Is it a good idea to define new formats for DNS data?
2. **Bikeshed warning** If yes, JSON, XML, other?
1. How to define the fields of the JSON object for each RR type? Manually? It doesn’t scale. Using levine-dnsexextlang when it will be implemented? Or an IANA registry with a (non-mandatory) JSON mapping entry (written in JSON?) for each RR type?
Smaller questions

1. How to define the fields of the JSON object for each RRtype? Manually? It doesn’t scale. Using levine-dnsextlang when it will be implemented? Or an IANA registry with a (non-mandatory) JSON mapping entry (written in JSON?) for each RR type?

2. Explicit RRsets (one more level) or not? (Already a clear consensus: yes, RRsets)
Smaller questions

1. How to define the fields of the JSON object for each RRtype? Manually? It doesn’t scale. Using levine-dnsextlang when it will be implemented? Or an IANA registry with a (non-mandatory) JSON mapping entry (written in JSON?) for each RR type?

2. Explicit RRsets (one more level) or not? (Already a clear consensus: yes, RRsets)

3. Use zyp-json-schema or natural language description of the format?
Smaller questions

1. How to define the fields of the JSON object for each RRtype? Manually? It doesn’t scale. Using levine-dnsextrlang when it will be implemented? Or an IANA registry with a (non-mandatory) JSON mapping entry (written in JSON?) for each RR type?

2. Explicit RRsets (one more level) or not? (Already a clear consensus: yes, RRsets)

3. Use zyp-json-schema or natural language description of the format?

4. Representation of unknown RR types?
Smaller questions

1. How to define the fields of the JSON object for each RRtype? Manually? It doesn’t scale. Using levine-dnsexxtlang when it will be implemented? Or an IANA registry with a (non-mandatory) JSON mapping entry (written in JSON?) for each RR type?

2. Explicit RRsets (one more level) or not? (Already a clear consensus: yes, RRsets)

3. Use zyp-json-schema or natural language description of the format?

4. Representation of unknown RR types?

5. Representation of binary data? Base64 in a string?
RRsets or not?

"AnswerSection": [  
  {"Name": "facebook.com", "TTL": 6666, "Type": "AAAA",  
    "Address": "2a03:2880:10:8f01:face:b00c::25"},  
  {"Name": "facebook.com", "TTL": 6666, "Type": "AAAA",  
    "Address": "2a03:2880:2110:3f01:face:b00c::"},  
...

Or:

"AnswerSection": [  
  {"Name": "facebook.com", "TTL": 6666, "Type": "AAAA",  
    [  
      {"Address": "2a03:2880:10:8f01:face:b00c::25"},  
      {"Address": "2a03:2880:2110:3f01:face:b00c::"},  
    ...

Warning: JSON arrays have an order (not the DNS)
"AnswerSection": [
    {
        "Name": "a.example", "Type": "TYPE731",
        "Data": "XXXXXX"
    }
    ...

No need for an explicit length in JSON
No better way in JSON than:

```json
{"Name": "signed.example", "Type": "DNSKEY", "Key": "AwEAAYz/bZVFyefKTiBBF..." }
```

Explicitely mark fields with Base64? No tagging in JSON, so tag in the name of the field?