2017-01-09: CBOR WG

- Concise Binary Object Representation Maintenance and Extensions

1. Formal process: Take RFC 7049 to IETF STD level
2. Standardize CDDL as a data definition language
3. (Maybe define a few more CBOR tags, as needed.)
CDDL
Henk Birkholz, Christoph Vigano,
draft-ietf-cbor-cddl
ABNF

• BNF (Backus-Naur form): grammars for strings

• RFC40 (1970): first RFC with BNF

• “Internet” BNF: Augmented BNF (ABNF)

• RFC 733 (1977): “Ken L. Harrenstien, of SRI International, was responsible for re-coding the BNF into an augmented BNF which compacts the specification and allows increased comprehensibility.”
ABNF in the IETF

- 752 RFCs and I-Ds reference RFC 5234 (the most recent version of ABNF) [cf. YANG: 160]
- Tool support (e.g., BAP, abnf-gen; antlr support)
- Pretty much standard for text-based protocols that aren’t based on XML or JSON
ABNF is composed of productions

\[
\begin{align*}
\text{addr-spec} &= \text{local-part} \ "@" \ \text{domain} \\
\text{local-part} &= \text{dot-atom} / \ \text{quoted-string} / \ \text{obs-local-part} \\
\text{domain} &= \text{dot-atom} / \ \text{domain-literal} / \ \text{obs-domain} \\
\text{domain-literal} &= \text{[CFWS]} \ "[" \ *([FWS] \ \text{dtext}) \ [FWS] \ "]" \ [CFWS] \\
\text{dtext} &= \%d33-90 / \quad ; \text{Printable US-ASCII} \\
&\quad \%d94-126 / \quad ; \text{characters not including} \\
&\quad \text{obs-dtext} \quad ; \ "[", "]", \text{or } "\"
\end{align*}
\]

- **Names** for sublanguages
- **Compose** using
  - Concatenation
  - Choice: /
- **Literals** terminate nesting
From ABNF to CDDL

- Build **trees** of data items, not **strings** of characters
- Add literals for primitive types
- Add constructors for containers (arrays, maps)
  - Inspiration: Relax-NG (ISO/IEC 19757-2)
Rule names are **types**

`bool = false / true`
`label = text / int`
`int = uint / nint`

- Types are **sets** of potential values
- Even literals are (very small) types

`participants = 1 / 2 / 3`
`participants = 1..3`
`msgtype = "PUT"`
`msgtype = 1`
Groups: building containers

• Containers contain sequences (array) or sets (maps) of entries

• Entries are types (array) or key/value type pairs (maps)

• Unify this into **group**:

  • sequenced (ignored within maps)

  • labeled (ignored within arrays)
How RFC 7071 would have looked like in CDDL

reputation-object = {
  application: text
  reputons: [* reputon]
}

reputon = {
  rater: text
  assertion: text
  rated: text
  rating: float16
  ? confidence: float16
  ? normal-rating: float16
  ? sample-size: uint
  ? generated: uint
  ? expires: uint
  * text => any
}

; This is a map (JSON object)
; text string (vs. binary)
; Array of 0-∞ reputons

; Another map (JSON object)

; OK, float16 is a CBORism
; optional…

; unsigned integer

; 0-∞, express extensibility
Named groups

```python
header_map = {
    "Generic_Headers",
    * label => values
}

Generic_Headers = (
    ? 1 => int / tstr, ; algorithm identifier
    ? 2 => [+label],    ; criticality
    ? 3 => tstr / int,  ; content type
    ? 4 => bstr,        ; key identifier
    ? 5 => bstr,        ; IV
    ? 6 => bstr,        ; Partial IV
    ? 7 => COSE_Signature / [+COSE_Signature]
)
```

- Named groups allow re-use of parts of a map/array
- Inclusion instead of inheritance
• Generic Autonomic Signaling Protocol (GRASP)
• For once, try not to invent another TLV format: just use CBOR
• Messages are arrays, with type, id, option:
  
  ```
  message /= [MESSAGE_TYPE, session-id, *option]
  MESSAGE_TYPE = 123 ; a defined constant
  session-id = 0..16777215
  ; option is one of the options defined below
  ```

• Options are arrays, again:
  
  ```
  option /= waiting-time-option
  waiting-time-option = [O_WAITING, waiting-time]
  O_WAITING = 456 ; a defined constant
  waiting-time = 0..4294967295 ; in milliseconds
  ```
<table>
<thead>
<tr>
<th>Document</th>
<th>Title</th>
<th>Status</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-bormann-chord-time-tag</td>
<td>Concise Binary Object Representation (CBOR) Tags for Time, Duration, and Period</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-bormann-lwpan-chord-template</td>
<td>Concise Binary Object Representation (CBOR) Tag for CBOR Templates</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-bormann-t2tg-sworn</td>
<td>SWORN: Secure Wake On Radio Nudging</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-carpenter-anima-anis-objectives</td>
<td>Technical Objective Formats for the Autonomic Network Infrastructure</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-froath-chord-tags</td>
<td>Concise Binary Object Representation (CBOR) Tags for Typed Arrays</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-man/yam-tokbind-attest</td>
<td>Attested TLS Token Binding</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-hirsholz-tuda</td>
<td>Time Based Uni-Directional Attestation</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-hartke-t2tg-cbor-forms</td>
<td>CBOR-encoded Form Data</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-hartke-t2tg-coral</td>
<td>The Constrained RESTful Application Language (CoRAL)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-ace-chord-web-token</td>
<td>CBOR Web Token (CWT)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-anima-grasp</td>
<td>A Generic Autonomic Signaling Protocol (GRASP)</td>
<td>?proposed Standard</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-anima-prefix-management</td>
<td>Autonomic IPv6 Edge Prefix Management in Large-scale Networks</td>
<td></td>
<td>informational</td>
</tr>
<tr>
<td>draft-ietf-care-links-json</td>
<td>Representing CoRE Formats in JSON and CBOR</td>
<td>?proposed Standard</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-care-object-security</td>
<td>Object Security of CoAP (OSCoAP)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-care-senml</td>
<td>Media Types for Sensor Measurement Lists (SenML)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-case-cose</td>
<td>CBOR Object Signing and Encryption (COSE)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-dsop-dns-capture-format</td>
<td>C-DNS: A DNS Packet Capture Format</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-ietf-asm-cosevid</td>
<td>Concise Software Identifiers</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-liu-opentrustprotocol-cbor</td>
<td>Open Trust Protocol CBOR Encoding</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-schaad-cose-x509</td>
<td>CBOR Encoded Message Syntax (Cose): Headers for carrying and referencing X.509 certificates</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>draft-selan-diere-cose-ecdh</td>
<td>Ephemeral Diffie-Hellman Over COSE (EDHCC)</td>
<td>Re's Ref'd by</td>
<td>normatively references</td>
</tr>
<tr>
<td>rfc8007</td>
<td>Content Delivery Network Interconnection (CDN) Control Interface / Triggers</td>
<td></td>
<td>?proposed Standard</td>
</tr>
</tbody>
</table>

This is an experimental product. These dependencies are extracted using heuristics looking for strings with particular prefixes. Notably, this means that references to I-Ds by title only are not reflected here. If it's really important, please inspect the documents' reference sections.
SDOs outside of IETF

- CDDL is being used for specifying both CBOR and JSON in W3C, ___, and _____________ ___

- Data in flight in a variety of protocols, e.g.
  - Access to specific features in wireless radios
  - Aggregation of metadata, enabling visualization of network topologies
From draft to RFC

• **Do not**: break it

• Editorial improvements required

• Any additional language features needed?
  • Should stay in the “tree grammar” envelope
  • Should be mostly done with that, anyway.

• What can we take out?
  Not much without breaking specs.
Avoid the kitchen sink

• This is not a Christmas wish list

• Each feature has a cost
  • specification complexity
  • learning effort
  • implementation effort
Improvements of definition

- Editors’ draft, “matching” branch: new appendix B, matching rules
  - Concisely summarizes CDDL semantics
- Is this
  - Useful
  - Correct
  - Complete?
“Map validation” issue

• CDDL semantics are generative (production system)

• All elements of a group in a map are equal

• Wildcard match (for extensibility) can enable what was not intended to be enabled  
  { ? 4=>text,  
    * uint=>any }  

• How to create priority for “more specific”?
cuts (better error messages)

a = ant / cat / elk
ant = ["ant", ^ uint]
cat = ["cat", ^ text]
ant = ["elk", ^ float]

["ant", 47.11]

- Tool will not tell you "can't match a", but "can't match rest of ant"

- Worth adding?
Proposal: use cuts here, too

• A cut after recognizing a map key cuts off any alternative matches

\[
{ ? 4 ^ =>text, \\
* uint=>any }
\]

• Proposal: Make existing “:” a shortcut for “^ =>”

\[
{ ? 4: text, \\
* uint=>any }
\]

• TO DO: fully define

• TO DO: check for breakage

• TO DO: implement
Take CBOR to STD

• **Do not**: futz around

• **Do**:
  • Document interoperability
  • Make needed improvements in specification quality
    • At least fix the errata :-)  
  • Check: Are all tags implemented interoperably?
Take CBOR to STD

Process as defined by RFC 6410:

• independent interoperable implementations ✔

• no errata (oops) ✔ in draft

• no unused features [__]

• (if patented: licensing process) [N/A]
• –00 had already fixed errata
• –01: 2017-10-14
• Amplification of chosen Simple encoding (1-byte only for false/true/null etc.)
• Add a changes section
  • Maybe sort this into fixes and new information?
• New: Section 2.5 CBOR Data Models
CBOR data models

- Biggest failing of JSON: Data model now entirely implicit
- Observant reader could infer CBOR data model from RFC 7049
- Now more explicit: “generic data model” (as opposed to any specific data model realized in CBOR)
  - Unextended (basic) data model
  - Extension points: Simple, Tags
    - Pre-extension by false/true/null/undefined, 18 pre-defined tags
    - Further extension by Simple/Tag definitions (IANA)
Why is a generic data model important?

- Generic data model enables the implementation of generic encoders and decoders
- An ecosystem of generic encoders and decoders
  - makes interoperability so much more likely
  - guides definition of specific data models
“Expectations”

• “Batteries included”: not always appropriate
• But some of the pre-extensions are really basic
  • Which ones?
• Section 2.5 states \texttt{false/true/null} are \textit{expected} to be provided in a generic encoder/decoder
• Anything else (Simple: \texttt{undefined}, 18 tags) is “truly optional and a matter of implementation quality”.

26
Implementations

- Parsing/generating CBOR easier than interfacing with application
- Minimal implementation: 822 bytes of ARM code
- Different integration models, different languages
- > 40 implementations

http://cbor.io
Houston, we have an interoperability problem

- Tags 21, 22, 33, 34: base64url, base64 classic
- Those can be used with or without padding. Which one is it?
- Defined for tag 21: base64url **without** padding.
- But what about tag 22, 34? Reference to RFC4648 not helpful.
- Tag 33: is this also limited to base64url without padding? (And what about tag 34?)
- (Is white space allowed? I don’t think so. Weird line length limitations? Of course not.)
Being permissive is not solving this

• Tag 21, 22 are intended to be acted upon by a CBOR-to-JSON converter — need to know how

• Tag 33, 34 could be interpreted in a more permissive way?

  • Depending on specific data model, might require re-encoding on conversion to JSON (!)
How are base64, base64url being used in practice?

• Easy: Base64url is almost always without padding
  • Interoperability benefits from nailing this down

• Base64 more variable
  • Usually used with padding, but exceptions
  • Bikeshed
Solutions?

• Be more explicit about tag 33: base64url is used without padding in this case, too

• Could define tag 22/34 as with or without padding
  • Tag 22 defines JSON side, tag 34, CBOR side

• Could define additional tags for padding/none (probably only for base64 classic)

• Also, tag 23 (base16): lower or upper case?
Proposal

• Padding designed to help with indeterminate length

• We do know the length, so *no padding* is “right”

• RFC 7049 was unclear about this

• ➔ for base64 classic, go for no padding, too

  • add an implementation note explaining the clarification and asking to be particularly liberal about what you accept
Continuing work on implementation matrix


- Need to fill in more columns
  - Certainly not for all 45 implementations :-)

- Who?
CBOR tag definitions

Carsten Bormann, 2017-11-16
Batteries included

• RFC 7049 predefines 18 Tags
  • Time, big numbers (bigint, float, decimal), various converter helpers, URI, MIME message

• Easy to register your own CBOR Tags

• > 20 more tags: 6 for COSE; UUIDs, Sets, binary MIME, Perl support, language tagged string, compression
CWT: CBOR Web Token

- JWT: JSON Web Token (RFC 7519)
  - Package **Claim Set** into JSON
  - Apply JOSE for Signing and Encryption

- CWT: Use CBOR and COSE instead of JSON and JOSE

- CWT can replace unstructured misuse of certificates for Claim Sets

- Tag 61 assigned; WGLC completed in IETF ACE WG (draft-ietf-ace-cbor-web-token)
Status of Tags drafts

- **OID**: On charter, kitchen sink, expired. Needs work.
- **Array**: On charter, ready for adoption
- **Time**: Off charter; solved for now by FCFS registration (3-byte tag 1001); move spec to RFC how?
- **Template**: Off charter (will likely be done with SCHC anyway)
- **“Useful tags”**: Maybe document some of the more useful registered tags in an RFC on its own (could include Time)?
• Provide tags for homogeneous arrays represented in byte strings

  • Inspired by JavaScript

  • 12×2: Both LSB and MSB first

  • Reserves 24 contiguous tags in 2-byte space

• Provides a tag for other homogeneous arrays

• Provides a tag for multidimensional arrays
Array tags: 2-byte space?

- 2-byte Tags: Tags 24 to 255
- 2017: ~20 taken of 232; be careful with the space
- This is taking out 24 more — would this be a waste of 2-byte space?
  - Yes; arrays can be large; fine with 3-byte tags
  - No; arrays can also be small (e.g., RGB)
- Could partition 2 vs. 3 by size of basic type; ugly
- Would like to move this ahead (technical decision should not be an obstacle for draft adoption anyway)
Time tag

- Document 1001 as is
  - Could do this on independent stream, WG allowing
- Develop 1001 into a more general time tag
http://cbor.io
http://cbor.me
http://cddl.space