# CBOR
# Concise Binary Object Representation Maintenance and Extensions

https://datatracker.ietf.org/wg/cbor/

https://github.com/cbor-wg

Chairs:

Francesca Palombini
Jim Schaad

Mailing List:

cbor@ietf.org

Jabber:

cbor@jabber.ietf.org

Etherpad:

https://etherpad.tools.ietf.org/p/notes-ietf-106-cbor

# Note Well

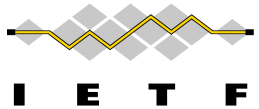
This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- https://www.ietf.org/privacy-policy/ (Privacy Policy)



# Reminder

- Minutes are taken
- This meeting is recorded
- Presence is logged

# Agenda Bashing

- Introduction [5'] : Chairs
  Agenda bashing and WG status update
- CBOR specification status [25'] : Carsten
  https://tools.ietf.org/html/draft-ietf-cbor-7049bis
- CDDL cont. - Ways forward [15'] : Chairs / Carsten
- Flextime [5']
- Wrap-up [5'] : Chairs

# Status Update

- 6 Interims since IETF105: https://datatracker.ietf.org/wg/cbor/meetings/
- Rechartering finalized: https://datatracker.ietf.org/wg/cbor/about/
- CBOR Array Tags and CBOR Sequences in RFC Editor queue (EDIT)
- 7049bis v-09 :
  - In WGLC until December 12

# Next Interims

- Continuing on the same time slot (Wednesday 16:00-17:00 UTC)

- December 18
- January 15
- January 29

- When needed

# CBOR Bis

# CDDL 2.0

# Goal for today

We identified interesting features for CDDL users (started…)

- Discussion today:
  - Do not focus on technical solution, but rather
  - Define scope of the features and
  - Identify possible pitfalls

# New features

**Extend domain into semantic interoperability**

-- semantic augmentation

-- post-validation output (~ PSVI in XML), e.g. for defaults

**Expressiveness for structural interoperability**

-- variants: both CBOR and JSON variants in one spec

-- co-occurrence constraints

-- more cuts

-- computed literals

**More Control Operators**

-- Alternative regexp forms (.pcre)

-- Operating with bitwise data: .bits variants, .bitfield

**Syntactic sugar**

-- regexp literals

**Language features**

-- CDDL/ABNF integration

-- module superstructure, import/export

**Tool Interoperation**

-- representation of CDDL in JSON

-- Provide an annotation feature to retain additional information in a CDDL spec

# New features

**Extend domain into semantic interoperability**

-- semantic augmentation 3 1

-- post-validation output (~ PSVI in XML), e.g. for defaults 2 1

**Expressiveness for structural interoperability**

-- variants: both CBOR and JSON variants in one spec 4 1

-- co-occurrence constraints 3

-- more cuts 1

-- computed literals 4 1

**More Control Operators**

-- Alternative regexp forms (.pcre) 1

-- Operating with bitwise data: .bits variants, .bitfield 1 2

**Syntactic sugar**

-- regexp literals

**Language features**

-- CDDL/ABNF integration 3 1

-- module superstructure, import/export 5 1 1

**Tool Interoperation**

-- representation of CDDL in JSON 2 2

-- Provide an annotation feature to retain additional information in a CDDL spec 2 1

YES

Maybe

This worries me

# Top 4 Features

1. Module superstructure, import/export
2. Computed literals
3. Variants: both CBOR and JSON variants in one spec
4. Co-occurrence constraints

# 1. Module superstructure, import/export

*CDDL definitions to be defined in modules and be referenced from other modules. Additionally, this would require some of the following features to be included in CDDL:*

- **namespacing**: to allow for internal and external referencing, with some possible control of namespace pollution.
- **import / export**: to make use of modules from other modules. Allow modules to specify which types are being either imported or exported from a module.
- **module naming**: define a global naming system so that modules can be uniquely identified and downloaded for a web site.
- **versioning**: to allow for unambiguous referencing of modules

# 2. Computed literals

*CDDL is not defined to be able to compute. We are considering to add the functionality to:*

*1.define integers as components of a computed operation.*

Example:
base = 100
a = base + 5

*2. define string literals of a computed operation, such as concatenation and substitution. (Analogously, some these operations could be defined to work on CBOR byte strings as well.)*

Example:
base = "CORE"
real = base /O/o

*3. represent CBOR tags as string literals tailored to their semantics rather than serialized CBOR*

Example: date tag represented as specified by ISO 8601 notation: dt'2019-07-21T19:53Z'

# 3. Variants: both CBOR and JSON variants in one spec

*Define a way to express the same structure for different encoding variants.*

For example, a user may want to define a structure in both CBOR and, with some differences:

- CBOR uses byte string where JSON uses b64;
- CBOR encodes a date as a UNIX time offset where JSON uses ISO encoded string).

*This feature would allow to write one specification that specifies both variants.*

# 4. Co-occurrence constraints

- *CDDL currently allows a constraint to be placed on a field. Allow for placing a constraint on a field which depends on the value of a second field. Such a constraint could be expanded to use multiple fields for the purpose of selection a constraint pattern.*

Example: in the US the postal codes match the pattern "\d\d\d\d\d (-\d\d\d\d)? while in Canada the postal code looks like "\w\d\w \d\w\d". This would be a simple single level constraint. The constraint selection could be expressed using two input values. Thus, the selection could be based on both the country and state. For the US in the state of Oregon, the postal code must start with the digits “97”.

- *Another selector allows for the use of fields at a different level than where the constraint is being applied. Two different ways of doing this would be either to specify a path by listing a set of operations (i.e. “../field1/field2”) or by doing a pattern match on the tree using either field names or type names to locate the input fields.*

```
Document ::= {
   key1 => [
      value3 .fromTable ValueTable ..\key1
      value4 .fromTable  ValueTable2 [..\key1, ..\key2]
   ]
}
```

In that example, ‘value3’ comes from a table that is keyed by the value associated with key1. The value ‘value4’ comes from a table that is keyed by the values associated with key1 and key2.

# Ways Forward

- Start with the features discussed today
- Send out survey to mailing lists and collect more input and feedback
- Report on general comments about CDDL as well as features