— CBOR: packed
— Packed/Array (RFC 8746): functional tag validity
— Progress on Alternative/Union (Choice) Tags: § 9.1 of draft-bormann-cbor-notable-tags-06
Packed

Already prepared at previous interim:

- merge prefix and suffix tables ➔ argument tables
- default function: concatenation
- specific function: function tag

This round (draft-ietf-cbor-packed-06):

- type 0/1 ➔ straight/inverted
- set up lhs (left-hand-side), rhs (right-hand-side)
- function tag, if any, is lhs
Pre-define one function

Function tags are extension points
"Use it or lose it": Join function

" | ".join(['foo', 'bar'])
(Python)

→ "foo | bar"

["foo", "bar"].join(" | ")
(Ruby)

One array of items, one joiner to intersperse

Not commutative (!)
Example: merged argument table

["https://packed.example/foo.html",
 "https://packed.example/bar.html",
 "https://packed.example/ant.cbor"]

113([ [],
    [106("packed.example")],
    [6(["https://", "/foo.html"]),
     6(["coap://", "/bar.cbor"]),
     6(["mailto:support@", "]")]]

6 = straight reference 0

113([ [],
    ["packed.example"],
    [216(105(["https://", "/foo.html"]),
     216(105(["coap://", "/bar.cbor"]),
     216("mailto:support@"))]

216 = inverted reference 0

Carsten Bormann • 2022-cbor-12 2022-07-13
Non-commutative function: May need two function tags

105: joiner + array, 106: array + joiner both directions are needed:

```
113([ [],
    ["packed.example"],
    [216(105(["https://", "/foo.html"]),
      216(105(["coap://", "/bar.cbor"]),
      216("mailto:support@"))
]
```

```
113([ [],
    [105(["coaps://[2001::db8::1]/s/", ".senml"]),
      6("temp-freezer"),
      6("temp-fridge"),
      6("temp-ambient"))
])
```
What else to do?

- Sequences (RFC8742): unwrap-splicing
- Further function tags? CURIE?
- Type pairings beyond:
  - string + string (*rump* determines text vs. byte)
  - array + array, * map + map
  - tag + any
Tag Validity

RFC 8949: Structural Tag Validity, shape of tag content
E.g., RFC 8746 Multi-dimensional Array (Tags 40/1040):

Data Item: Array (major type 4) of two arrays:
• one array (major type 4) of [...] unsigned integers distinct from zero; and
• one array (any one of a CBOR array of major type 4, a Typed Array, or a Homogeneous Array) of elements.

(Typed Array = Tags 64 to 87)
Incompatible with CBOR-Packed!

CBOR packed substitutes reference tags for actual data

Reference Tags stand in for their referents

Should we turn this into a concept?
Equivalence Principle

— not new: Tags can define shape of their valid tag content

— NEW: Tags can define what shape they stand in for (equivalence of tag to a shape)

— ➔ a data compression tag can stand in for a byte string

— ➔ a new typed array tag can stand in for a CBOR array

— like the existing typed arrays already do for all intents and purposes
How to write this up?

Use cases: lz4 compression tags, bfloat16 (1+8+7) floats

Could simply say this in those tag definitions
A bit unilateral 😐

Does equivalence principle update RFC 8949?
Can it be opt-in?
Status choice/alternative/union tag:

To be registered with IANA based on text in notable-tags
Interim Cadence

Proposal from CoRE WG chairs (Wed 14–15Z):

**CBOR WG:**
- 2022-08-24
- 2022-09-07
- 2022-09-21
- 2022-10-05
- 2022-10-19

**alternate weeks for CoRE:**
- 2022-08-31
- 2022-09-14
- 2022-09-28
- 2022-10-12
- 2022-10-26