

# CBOR: Semantic Tagging

Main items of progress from JSON to CBOR:

1. embracing binary (byte strings, binary encoding, ...)
2. embracing evolution (extensibility): **semantic tagging**

# Tags as main extension point for CBOR

CBOR Basic generic data model: ~ JSON + binary

CBOR Extended generic data model:

- **Tag** data (from Basic or Extended generic data model)  
→ express some semantics, use existing representation

"Batteries included":

Tags are used for some basic data types of CBOR itself (time, big numbers, certain conversions, ...)

# Tag allocation: IANA registry

- Simple unsigned 64-bit number
- Tag namespace is **global**: IANA CBOR Tags Registry
  - Compare ASN.1's four tag spaces
    - UNIVERSAL — comparable to CBOR, but smaller
    - APPLICATION — typically local to module
    - PRIVATE — we didn't want that in CBOR
    - Context-specific — e.g., in CBOR-Packed

# Tag allocation: ranges

Vast space (but small ones **better** than large ones)  
Easy registration (most ranges accessible to FCFS)

- Different levels of curation
  - 0 to 23 ("1+0"): **Standards Action**
  - 24 to 255 ("1+1") and 256 to 32767 (lower half of "1+2"): **Specification Required** (via Designated Expert)
  - 32768 to 18446744073709551615 (upper half of "1+2", "1+4", "1+8"): **FCFS** (First Come First Served)

# Tag allocation: design for decades

These need to last for decades!

Designated Expert (DE) officially only for 0..32767

FCFS space was 256..∞ (7049), now 32768..∞ (8949)

range	used	%	free	total
0 1+0	13	54.17	11	24
1 1+1	70	30.17	162	232
2 1+2	434	0.66	64846	65280
3 1+4	65284	0.00	4294836476	4294901760
4 1+8	2	0.00	18446744069414584318	18446744069414584320

# Tag allocation: policies

- FCFS (32768.. $\infty$ ) is free for all  
(in practice: some advice is given)
- 0..32767 is **curated**  
saving the good ranges for good uses
  - 1+0 is really hard to get now
  - 1+1 needs good justification
  - lower half of 1+2: apply basic checks

# Case study: JSON-LD

(RDF in JSON, W3C)

Limited data transparency:

- Uses @type map key as extension point
- Can only use maps as root of typed data
- Uses URIs as namespace
- Has typical problems of dereferenceable identifiers

# Case study: RFC 9290

(concise problem details, CoRE WG)

Existing XML/JSON spec: RFC 7807

- map with predefined and application-specific keys
- type key: use URI reference for type-tagging

draft-ietf-7807bis tries to nail this down further

RFC 9290: **concise** problem details

Replace URIs with registration (integer; URI as fallback)



# Case study: draft-rundgren-cotx-03

("CBOR Object Type Extension (COTX)", Anders Rundgren)

Use text strings for semantic tagging

- + Easy to come up with text strings
- No curation, no collision avoidance
- Could use URIs, could use anything

Contrast to [CBOR-native](#) tag system:

- + Comes with CBOR, well-implemented
- + Easy registration, central collision avoidance
- Can't just make up text strings

# COTX vs. CBOR WG

This looks like a valid registration request  
Registry can handle this; no WG action required

Likely Designated Expert response:

- Specification required: I-D is sufficient  
(nice if that becomes an RFC, e.g., independent stream)
- Tag range: URIs are already big, so 1+2 is appropriate

Presentation Score ("B-Note"):

- Should not present itself as a replacement for native tags