draft-ietf-cbor-edn-literals

- EDN: Originally defined in RFC 7049 ➔ 8949 and an appendix to RFC 8610
- edn-literals started out to add **extension point**: application-specific literals
  - implemented (e.g., in cbor.me)
- Severe mission creep (in part after WGLC and IETF LC)
  - Add ABNF (no ABNF in 8949/8610)
  - Roll up base documents and this extension, editorial, …
- Now stuck in discussion about implementation preferences

draft-ietf-cbor-edn-e-ref stuck behind (in active use!)

EDN requirements

https://github.com/cbor-wg/edn-literal/wiki/Architecture-and-Requirements

➔ Humane presentation (and keyboarding) of CBOR data
(normal interchange uses CBOR)

EDN = Superset of JSON
(which is both interchange and presentation format)

- because CBOR ⊃ JSON
- because we want to improve usability

CBOR ecosystem: extensibility

CBOR ecosystem is friendly to third-party extension

- CBOR: Tags, simple values (type system)
- CDDL: Control operators (processing beyond grammar)
- EDN: Application-specific extensions
  (can go along with tag extensions)

Registry-based, many concurrent contributors

CBOR ecosystem: third party extensibility

— **Isolation**: New extension MUST NOT destabilize EDN base or other extensions

— **Pluggable Implementability**: Implementations can add extension without destabilizing themselves or other extensions.
➔ **bazaar** style of evolution

— **Permissionless Innovation**: enable use of extension even if not all libraries on the way know it yet

EDN: usability with editors and revision control

Specific to CBOR-related **languages** (not CBOR itself):

- can work with source code editors and revision control
  - Needs to work with Unicode
  - Needs to work with base repertoire
    (ASCII printable %x20-7E + newline LF/CRLF)
- can automatically convert to base repertoire (and back)
- keep **third party extensibility** requirements in mind!

CBOR ecosystem: implementation

- extensions:
  enable use of existing application specific libraries
  - do not require special implementations of URI, date/time, IP address, ...
- do not prescribe specific implementation technique
  - isolated ("two-level"), enabling non-ABNF extension
  - integrated in base ABNF ("one-level")