JSON binding of IODEF

draft-ietf-mile-jsoniodef-01.txt

Takeshi Takahashi, NICT

Current status of the draft

1. Based on the discussion at the IETF 99 in Prague, a WG draft was established for this issue

2. 01 version was just published on Monday, which is the first version we ask for review

3. We are looking for co-editors

4. We have three issues we would like to discuss

Issue 1: ML string

The use of MLStringType requires some considerations in JSON.

Example of the Description element within Software using the MLStringType in JSON

Description : {"value": “MS Windows", }

Description : {
“value”: “MS ウィンドウズ",
“lang": "jp",
"ttranslation-id": "ip2en0023"}

MLStringType is efficient for XML, but not for JSON, especially when we write values in English (default)

Proposed approach

Description : "MS Windows"

Description_ML : {
“value”: “MS ウィンドウズ",
“lang": "jp",
"ttranslation-id": "ip2en0023"}

We propose to prepare two different elements, i.e., Description and Description_ML for this example. Likewise, all the elements defined as MLStringType will have the same types of new definitions.

Issue 2: Binary strings

The IODEF in XML allows to use base64 binary and hexadecimal binary.
For JSON, simplicity is preferred. Therefore, let us confirm here.

Can we choose one of them? Or should we allow to use both of them?
(In any case, the schema file will treat values of these types as string)

Issue 3: Omitting some of semantic classes

XML allows us to structure the semantics.
JSON can do the same, but simplicity is often preferred for JSON.
Therefore, can we omit these classes that exist for semantical purpose and furture extension (in my understanding)?

a. Flow class (Defined in Section 3.16 of RFC7970)
b. ApplicationHeader class (defined in Section 3.20.2 of RFC7970)
c. SignatureData class (Defined in Section 3.27 of RFC7970)
d. IndicatorData class (Defined in Section 3.28 of RFC7970)
e. ObservableReference class (Defined in Section 3.29.6 of RFC7970)

These classes have only one element or attribute.

Next steps

1. We will copy some of the content from RFC7970 so that this draft will become self-contained

2. Reflect today's discussion results to the draft

3. Summarize the notable differences between RFC7970 and this draft

and so on.