**Metadata and Augmentations**

1. We are nottalking about metadata *properties,* such as the DoD Minimum Essential Metadata properties. So far as the NTAC is concerned, those are much like any other property in the NIEM model.
2. We are talking about the metadata *mechanism,* which is a way for a message designer to add properties to objects without changing the object type. The added properties are usually thought of as metadata – that is, data about data – but they do not have to be such.
3. For example, a message designer might wish to add privacy metadata to PersonName objects. He cannot do this by changing the definition of PersonNameType, but he can do it via the metadata mechansim, as shown below. (In all examples, some components and markup is omitted for clarity.)

<nc:Person>

<nc:PersonName>Clark Kent

<nc:PersonName s:metadata="#priv">Superman 🡨 *make this private*

1. Observe that a message designer must provide for metadata in the message schema. If the message schema does not contain any element with a type derived from s:MetadataType, then there can be no metadata in a conforming message.
2. The metadata mechanism is painful for the developers of software that consumes messages. They would be much happier with something like this:

<nc:Person>

<nc:PersonName>Clark Kent

<nc:PersonName>Superman

<my:PrivateIndicator>true

1. In the above example, the very *existance* of the name "Superman" is marked private. But that's not what we want. Everybody knows about Superman. We want the *relationship* between the person who is also named Clark Kent and the name "Superman" to be private – like this:

<nc:Person>

<nc:PersonName>Clark Kent

<nc:PersonName s:relationshipMetadata="#priv">Superman

1. We don't have a good way to handle relationship metadata without a relationship object. We can do it in NIEM XML, and we can do it in NIEM RDF using multiple graphs, but the NIEM JSON representation is irredeemably ugly. We really need something like this:

<nc:Person>

<nc:PersonName>Clark Kent

<nc:PersonNameRelationship>

<nc:PersonName s:relationshipMetadata="#priv">Superman

1. And, of course, the software developers would prefer this instead:

<nc:Person>

<nc:PersonName>Clark Kent

<nc:PersonNameRelationship>

<nc:PersonName>Superman

<my:PrivateIndicator>true

1. But of course, there is no PersonNameRelationship property in PersonNameType, so that won't work. If only we had a way to augment an existing type with additional properties…
2. Wait! We have augmentations! Everything shown above can be accomplished through augmentations, without using the metadata mechanism at all.
3. Augmentations have to be written into the message schema. But as we saw in #4, the metadata mechanism also requires some provision in the message schema. That really isn't very different.
4. So why don't we make our life ever so much easier, and completely remove the metadata mechanism from NIEM 6? Show people how to use augmentations instead, and poof! we're done.
5. What am I missing here?
6. An unredacted example, with all the markup and components:

<nc:Person>

<nc:PersonName>

<nc:PersonFullName>Clark Kent</nc:PersonFullName>

</nc:PersonName>

<nc:PersonName s:metadata="#priv">

<nc:PersonFullName>Superman</nc:PersonFullName>

</nc:PersonName>

</nc:Person>

<my:Metadata s:ref="#priv">

<my:PrivateIndicator>true</my:PrivateIndicator>

</my:Metadata>