**ECF5 Spec Feedback and Considerations – 14**

This document contains additional questions and commentary resulting from a review at the Electronic Court Filing Version 5.0 Working Draft 18. The issues in this document were originally raised in the prior document in this feedback series (i.e. ‘ECF5 Spec Considerations-13.docx’). Issues form the prior installment that have been fully addressed have been removed in this edition. Responses to the prior document by James Cabral are included in red text. Additional responses to these responses are included beneath in brown text.

1. **Normative Attorney to Party Association Method**

At the June 13, 2017 ECF Conference call it was agreed that although there are several ways that attorney/party representation relationships can be expressed or implied, a normative approach should be defined. It was also agreed that the normative approach was to use ecf:CaseRepresentedParty in an Attorney/CaseOfficial element. This approach was similar to that illustrated in ecf-v5.0-wd16.docx, section 6.2.11.1. It was also noted during the call that this example should be corrected to place ecf:CaseRepresentedParty into j:CaseInitiatingAttorney, as it is not necessary or even desirable to use both j:CaseInitiatingParty and j:CaseOfficial.

Whereas the example in section 6.2.11.1 has been revised and is structurally correct and consistent with the agreed changes from the 6-13-2017 conference call, this example does not define the illustrated approach for attorney/party representation relationships as ‘normative’. The specification does not state that the example is normative nor is it intended as an example of attorney/party representation relationship, as it is an example of Content References and Element References (although it can serve both illustration functions).

I continue to feel strongly that using ‘Reference Elements’ within ecf:CaseRepresentedParty instead of ‘Content References’ is a superior approach (see ECF5 Spec Considerations – 7.docx, item 1 ‘Attorney to party Association’).

To allow flexibility, the normative approach should permit implementers to choose to employ either ‘content references’ or reference elements’, when referring to the ecf:CaseRepresentedParty.

Again, I am providing suggested text for the specification document. This suggestion is from ECF5 Spec Considerations – 7.docx, with some updated revisions:

**Attorney Party Relationships**

The relationship of a case party or parties to the attorney or attorneys representing the party or parties in a law suit MUST be done using a j:CaseOfficialType element (e.g. j:CaseDefenseAttorney, j:CaseInitiatingAttorney, j:CaseProsecutionAttorney, j:CaseRespondentAttorney, j:CaseOtherEntityAttorney, and j:CaseOfficial ) within j:CaseAugmentation.

The attorney in the relationship MUST be identified using nc:RoleOfPerson. When identifying the attorney using nc:RoleOfPerson reference, then two methods are supported. One and only one of these two methods MUST be used. The two methods are:

1. The attorney is identified using the structures:ref attribute on nc:RoleOfPerson. This is the preferred and recommended method.
2. The attorney details are elaborated within nc:RoleOfPerson.

When using the first method, the value of structures:ref MUST correspond to a person type entity with a matching value for structures:id. The nc:RoleOfPerson element MUST not have any sub-element content.

When using the second method, the structures:ref attribute in nc:RoleOfPerspon MUST not be used. The attorney person details MUST be elaborated with sub-elements of nc:RoleOfPerson.

The element j:JudicialOfficialBarMembership and j:JudicialOfficialRegistrationIdentification MAY only be used within j:CaseOfficial, when they have not been used within the attorney person elaboration.

(Not sure what to say about j:CaseOfficialCaseIdentification – seems like the rule above should apply here too).

The attorney SHOULD be provided with an identifier in ecf:CaseOfficialAugmentation/ecf:AttorneyID/nc:IdentificationID.

To identify the parties represented by the attorney, ecf:CaseOfficialAugmentation MUST be substituted for j:CaseOfficialAugmentationPoint. An ecf:CaseRepresentedParty element MUST be used within ecf:CaseOfficialAugmentation to identify each party represented by the attorney. If the attorney represents more than one party on the case, then multiple ecf:CaseRepresentedParty elements should appear. When an attorney represents more than one party on a case, multiple j:CaseOfficialType elements SHOULD NOT be used as an alternative to using multiple ecf:CaseRepresentedParty elements within a single j:CaseOfficialType element.

The ecf:CaseRepresentedParty element MUST contain a valid entity type element which has been substituted for nc:EntityRepresentation. The appropriate element will depend upon the nature of the case party; whether it is a person party, an organization party, or a property/item party.

When identifying the party represented by the attorney, there are two methods supported. One and only one of these two methods MUST be used. The two methods are:

1. The party is identified using the structures:ref attribute on the substituted entity type element. This is the preferred and recommended method.
2. The party entity details are elaborated within the substituted entity type element (e.g. EntityItem, EntityOrganization or EntityPerson).

When using the first method, the value of structures:ref MUST refer to an entity type element with a corresponding entity type and which has a structures:id attribute with a matching value. The entity type element substituted for nc:EntityRepresentation MUST not have any sub-element content. The element ecf:CaseParticipantRoleCode MUST not be used when using the first method.

When using the second method, the structures:ref attribute in the substituted entity type element MUST not be used. The entity details MUST be elaborated using sub-elements of the entity type element substituted for nc:EntityRepresentation. The ecf:CaseParticipantRoleCode within ecf:CaseOfficialAugmentation MAY only be used if the entity is an item type entity. For organization and person type entities, use the ecf:CaseParticipantRoleCode available within the entity augmentation.

It seems to me, that if a reader were looking in the specification for how to define attorney/party representation relationships, the section on ‘Identifier Rules’ may not be the section you would look to. Perhaps it should be a section on ‘Identifiers and Associations’.

I added the following to 6.2.11.2

The relationship of an attorney to the party being represented SHOULD be defined using a reference element in ecf:CaseOfficialAugmentation/ecf:CaseRepresentedParty.

The above revision still does not specify a normative method for identifying attorney to party representation; ‘SHOULD’ is not ‘MUST’. Even if the word ‘SHOULD’ was replaced with the word ‘MUST’, it still only goes part of the way in defining a normative approach. Doing only this would still leave too many implementer alternative choices. For example, consider the suggested text from above:

If the attorney represents more than one party on the case, then multiple ecf:CaseRepresentedParty elements should appear. When an attorney represents more than one party on a case, multiple j:CaseOfficialType elements SHOULD NOT be used as an alternative to using multiple ecf:CaseRepresentedParty elements within a single j:CaseOfficialType element.

The revision made to 6.2.11.2 does not address this possibility. If not specified, then one implementer may elect to use multiple attorney elements to record multiple parties, whereas other implementers may elect to just use multiple ecf:CaseRepresentedParty elements for the same purpose. This lack of consistency derides from interoperability. I think it would be easy to get all to agree that multiple ecf:CaseRepresentedParty elements should be used instead of multiple attorney elements. As such, even the word ‘should’ in the sample above would be replaced with “MUST”.

I have other concerns with the correction to 6.2.11.2 above:

* The statement added is not where it belongs in the specification. It’s tucked away under ‘Reference Elements’. No one will look for it in this location, and it’s off topic for its present location.
* It’s ambiguous. To whom does the requirement to use a ‘reference element’ apply? To the attorney or the represented parties? I’m guessing parties.
* Assuming it applies to the parties, then it contradicts statements in 6.2.8 Filer and Party Identifiers – i.e. “Attorney identifier elements, e.g. ecf:FilingAttorneyID and ecf:AttorneyID, MAY reference the parties they represent with party identifiers.“ To reference with party identifiers would require the use of content references, not reference elements.
* Again, defining a normative approach has been skirted (‘SHOULD’, not ‘MUST’).
* The simple statement does not fully define how attorney/party representation relationships should be recorded. It’s too simple, too presumptive. Different readers may still do it different ways, detracting from interoperability.

1. **Multiple CaseParticipantRoleCodes**

At present ECF 4.01 allows an entity to have multiple CaseParticipantRoleCodes.

This was straight forward in ECF 4.0 when cardinalities were generally more liberal. You just listed as many codes as needed as shown in the following example:

<ecf:CaseParticipant>

<ecf:CaseParticipant>

<ecf:EntityPerson s:id="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

</ecf:EntityPerson>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

</ecf:CaseParticipant>

</ecf:CaseParticipant>

In ECF 4.01, the cardinality was limited to only one. However, it was stated that multiple CaseParticipantRoleCodes could still be had by using multiple ecf:CaseParticipant elements. The above ECF 4.0 example must be expressed as:

<ecf:CaseParticipant>

<ecf:CaseParticipant>

<ecf:EntityPerson s:id="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

</ecf:EntityPerson>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

</ecf:CaseParticipant>

</ecf:CaseParticipant>

<ecf:CaseParticipant>

<ecf:CaseParticipant>

<ecf:EntityPersonReference s:ref="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

</ecf:CaseParticipant>

</ecf:CaseParticipant>

In retrospect, the method we were instructed to use in ECF 4.01 (i.e. applying multiple CaseParticipantRoleCodes to an entity through the use of multiple CaseParticipant elements) is not valid in ECF 5. We now know that NIEM 3.2 Rule 12-2 (e.g. element with structures:ref does not have content) would forbid this approach.

In ECF 5, applying multiple ecf:CaseParticipantRoleCodes to an entity is even more obtuse than in ECF 4.01. One approach that can be used when the entity is a case party with attorney representation is illustrated in the example for section 6.2.11.1 ‘Element Content References’.

Other approaches are possible. The Samuel Adams example from ECF 4 above could be done as shown below:

<j:CaseOtherEntity>

<ecf:EntityPerson structures:id="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

<j:CaseOtherEntity>

<ecf:EntityPerson>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

Being forced to use Content References rather than Reference Elements makes this decidedly more difficult to process.

Here’s another ECF 5 approach to accomplish the Samuel Adams example:

<j:CaseOtherEntity>

<ecf:EntityPerson structures:id="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

The multiple cardinality approach in ECF 4.0 is still the most elegant and practical way to achieve multiple CaseParticipantRoleCodes for an entity.

However, the example above is not the only way you can do this in ECF 5!

It could also be done like this:

<j:CaseOtherEntity>

<ecf:EntityPerson structures:id="Person1">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

<j:CaseOtherEntity>

<ecf:EntityPerson structures:id="Person2">

<nc:PersonName>

<nc:PersonGivenName>Elizabeth</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Elizabeth Adams</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Wife</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>17</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

…

<nc:PersonAssociation>

<nc:Person>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</nc:Person>

<nc:Person structures:ref="Person2" xsi:nil="true"/>

<ecf:PersonAssociationAugmentation>

<ecf:EntityAssociationTypeCode>spouse</ecf:EntityAssociationTypeCode>

</ecf:PersonAssociationAugmentation>

</nc:PersonAssociation>

Having so many different ways to do one thing makes it difficult for implementers and does not promote interoperability.

So for those of you who like puzzles, how many and what CaseParticipantRoleCodes does Samuel Adams have in the ECF 5 XML snippet below?

<j:CaseOtherEntity>

<ecf:EntityPerson structures:id="ia7bcafff-e6f4-400c-9d36-fc46473c199a">

<nc:PersonName>

<nc:PersonGivenName>Samuel</nc:PersonGivenName>

<nc:PersonSurName>Adams</nc:PersonSurName>

<nc:PersonFullName>Samuel Adams</nc:PersonFullName>

</nc:PersonName>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Brewer</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

<j:CaseOtherEntity>

<ecf:EntityPerson>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Patriot</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Delegate</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</ecf:EntityPerson>

</j:CaseOtherEntity>

…

<j:CaseInitiatingAttorney>

<nc:RoleOfPerson structures:id="Person3">

<nc:PersonName>

<nc:PersonFullName>Jane Doe, Esq.</nc:PersonFullName>

</nc:PersonName>

</nc:RoleOfPerson>

<ecf:CaseOfficialAugmentation>

<ecf:AttorneyID>

<nc:IdentificationID>10</nc:IdentificationID>

</ecf:AttorneyID>

<ecf:CaseParticipantRoleCode>Attorney</ecf:CaseParticipantRoleCode>

<ecf:CaseRepresentedParty>

<nc:EntityPerson>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>TaxCollector</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Father</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</nc:EntityPerson>

</ecf:CaseRepresentedParty>

</ecf:CaseOfficialAugmentation>

</j:CaseInitiatingAttorney>

…

<nc:PersonAssociation>

<nc:Person>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Revolutionary</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Colonist</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</nc:Person>

<nc:Person structures:ref="Person2" xsi:nil="true"/>

<ecf:PersonAssociationAugmentation>

<ecf:EntityAssociationTypeCode>spouse</ecf:EntityAssociationTypeCode>

</ecf:PersonAssociationAugmentation>

</nc:PersonAssociation>

…

<payment:Payer>

<ecf:PersonAugmentation>

<ecf:CaseParticipantRoleCode>Bostonian</ecf:CaseParticipantRoleCode>

<ecf:FilingPartyID>

<nc:IdentificationID>16</nc:IdentificationID>

</ecf:FilingPartyID>

</ecf:PersonAugmentation>

</payment:Payer>

The answer is 8 – Brewer, Patriot, Delegate, TaxCollector, Father, Revolutionary, Colonist, and Bostonian.

Hopefully, this little puzzle illustrates how out of control this appears to be! It seems to be neither elegant nor practical to have to search multiple nooks and crannies in the XML to assemble the full participation picture. I think we need to discuss this with the TC.

We can discuss this at the next TC meeting but I just think this is an example of the flexibility of the specification.

Just because you can spread CaseParticipantRoleCodes for a single participant far and wide across the message, does not mean that you should and it certainly does not mean that it is good to do so. I believe most implementers will want some reasonable expectation as to where the information is expected to be found. I think we can agree on where and how this should be done in a consistent and standardized way which promotes interoperability and consistency.

1. **CorrectedCase Integrity Rules**

Some integrity rules were first suggested in ‘ECF5 Spec Considerations 6’, item 2, such as:

CorrectedCase must be the same case type and use the same augmentation point type as FilingMessage/nc:Case.

The above example may not be such a good integrity rule, since it seems to say that the case type cannot be changed in clerk review.

However, I do think that how to use and how to understand the usage of CorrectedCase should be more specifically described in the specification. Presently it only gets the following mention in Section 6.3.3:

If the clerk made any modifications to the original filing case information, then the modified case information SHOULD be included in the docket:CorrectedCase element.

The specification does not go on to say, or even suggest, how this should be done. It may be more challenging than simply stated.

Whereas the docket.xml example may provide some additional (non-normative) guidance, this still leaves a lot of room for differing interpretations and approaches. This derides from interoperability.

Writing up this additional specification guidance will not be easy. So if the TC agrees that this should be done, then it will need to be assigned to someone (perhaps me) to draft the write-up for review. Many elements of this guidance may already be included in ‘ECF5 Spec Considerations – 10’, item 4 Corrected Filing.

I’m not sure this detail is warranted. I think providing the non-normative example (docket.xml) should be sufficient.

In Tyler’s recent response (‘Tyler Technologies – ECF 5 Specification WD18 Feedback.docx’) in the section titled ‘Payment in NFRC’, they use the phrase “avoid potential implementer specific scenarios”.

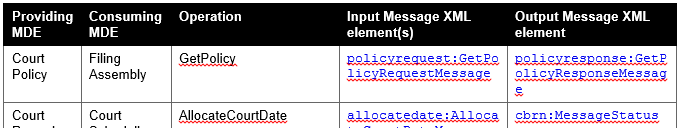
This is exactly the reason to provide additional specification clarity – to promote interoperability.

When the specification does not provide for a specific normative approach, simply leaving things up to the reader’s interpretation of schema, element definitions and weak XML examples, then results are not hard to predict - “implementer specific scenarios”.

1. **Policy Response**

Section 6.1.1 GetPolicy, states “an MDE (typically, a Filing Assembly MDE) MAY obtain a court’s machine-readable court policy, …”.

The table in section 4.1 Message lists the Filing Assembly MDE exclusively as a GetPolicy operation consumer.



How should the table in section 4.1 be interpreted? Does this table only list allowed MDE Provider/Operation/Consumers triads or only ‘typical’ triads?

The policy response example (policyresponse.xml) contains:

<policyresponse:MajorDesignElementTypeCode>CourtRecord</policyresponse:MajorDesignElementTypeCode>

<policyresponse:SupportedOperations>

<policyresponse:OperationNameCode>AllocateCourtDate</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>DocumentStampInformation</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>GetCase</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>GetCaseList</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>GetDocument</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>GetServiceInformation</policyresponse:OperationNameCode>

<policyresponse:OperationNameCode>RecordDocketing</policyresponse:OperationNameCode>

</policyresponse:SupportedOperations>

If the FAMDE is the only MDE recipient of policyresponse:GetPolicyResponseMessage (see table in Section 4.1 Messages), is there any reason to list operations that are not consumed by the FAMDE, such as AllocateCourtDate (consumed by CSMDE), DocumentStampInformation (consumed by FRMDE), and RecordDocketing (consumed by FRMDE)?

Or is it now true in ECF5 that MDEs other than the FAMDE will utilize GetPolicy to see if operations are supported, such as the FRMDE using GetPolicy to see if DocumentStampInformation supported? If so, would this be ‘typical’?

If the new normal is that GetPolicy is typically used by MDEs other than FA, then the table in section 4.1 may need some additional GetPolicy triad entries.

Updated the preface before the table in 4.1 to say:

The following table lists each ECF 5.0 operation, the MDEs that MUST provide and MUST consume the operation, and the input and output XML messages that define the data content exchanged. Other MDEs MAY also consume the operation.

It is not clear how this added statement clarifies the purpose of the table. In fact, I would say it makes things murkier. It seems to be saying that the table lists mandatory operations and mandatory operation consumers. But this cannot be correct. Consider the first entry: GetPolicy, provided by CPMDE and consumed by FAMDE – the GetPolicy operation is optional not mandatory (see 3.2.1).

1. **Policy Requester Filer ID**

The GetPolicyRequest message contains a mandatory ecf:FilingPartyID (“A unique identifier for a person or organization that requests information.”).

Of course the policy requesting entity may not be a case party; it could be an attorney, or some other participant. So perhaps this element should be ecf:RequesterID instead.

By whatever element name, would this element be considered a ‘Filer or Party’ Identifier as referenced in section 6.2.8 Filer and Party Identifiers?

If so, does the requirement that the identifier “MUST be unique within a case and will be generated by the court in response to a ReviewFiling operation” make any sense or even apply?

If ‘generated by the court in response to a ReviewFiling operation’, then would it not be unreasonable to expect that the very first time an entity accessed an ECF compliant efiling implementation, with the intent to submit an efiling or otherwise, and began the process by requesting the Court Policy for this efiling implementation, that the requester’s identifier element is empty?

I suspect that portions of your (J. Cabral) response in ‘ECF5 Spec Consideration – 6’, item 7 is relevant here too. Particularly number 4. “Filing party and attorney IDs are unique within the e-filing system. In practice, these will need to be assigned by an identity management system (not the EFSPs) that is outside the scope of the ECF specification. If there is a single EFM, the EFM will likely be the source of identity for filers and service recipients.”

Revised 6.2.8 to

Identifiers for filers and parties to a case, including person, organizations and property, labeled as ecf:FilingPartyID/nc:IdentificationID, ecf:AttorneyID/nc:IdentificationID, or ecf:FilingAttorneyID/nc:IdentificationID, MUST be unique within a case.

Since the policy requester identifier element was not changed (it’s still ecf:FilingPartyID) I have to presume you do not agree . Is this because you consider the policy requester to be a ‘party’ (as in participant rather than litigant)? If so, then at least consider changing the element name to ecf:FilingParticipantID. Or is it because you consider that a policy request message is a form of a ‘filing’? I do not think most people would consider it a filing, but all would agree that it is a message; so perhaps: ecf:MessageParticipantID?

However, I still think ecf:RequesterID is better.

1. **Lest We Forget**

As referred to above, a lengthy response was provided to an even lengthier item 7 in ‘ECF5 Spec Consideration – 6’. The suggestions in the response are still pending. The full response is repeated below:

There is a lot to unpack here. It seems like we need to start with some assumptions that should probably be in the specification. Here are my suggestions:

1. We need common definitions for “filer”, “party” (or, if you prefer “participant”), “attorney”, “service recipient” and possibly “submitter” that we use consistently throughout the specification.
2. Every filing is associated with a filing party and, unless they are filing pro se, a filing attorney. There are no “litigant identifiers”.
3. The filing party and attorney are both legally responsible for submitting the filing regardless if someone else submits the filing on their behalf.
4. Filing party and attorney IDs are unique within the e-filing system. In practice, these will need to be assigned by an identity management system (not the EFSPs) that is outside the scope of the ECF specification. If there is a single EFM, the EFM will likely be the source of identity for filers and service recipients.

We should discuss and agree on assumptions with the TC. Based on your feedback, I expect some dissent. Once we agree to the assumptions, we should consider the following changes to the specification.

1. Rewrite 6.2.8 completely to provide the definitions and clarify the assumptions.
2. In the schema, provide a reference from the filing party and attorney IDs to the entities (usually person objects), rather than reference from one ID to another ID as you suggested.
3. Improve the non-normative examples to more clearly demonstrate how the specification is intended to be used.

This issue was discussed with the task team on May 23 and the changes listed above were already made to the specification.

I agree that some aspects of this was discussed during the May 23, 2017 TC conference call. But not all of it was discussed. I included your entire response from item 7 in ‘ECF5 Spec Consideration – 6’ because I did not want it to be out of context. I have highlighted the items above in yellow background that I do not remember discussing or concluding.

I think there is still considerable work to do in this area. Some of the issues I raised in item 7 in ‘ECF5 Spec Consideration – 6’ were not even addressed in the original response, except to note that there is a lot to unpack, e.g. the impracticality of generating Filer and Party Identifiers “in response to a ReviewFiling operation” (note: this provision was included in section 6.2.8 in WD11, and all the way through to and including WD17, but is absent in WD18). There was not any notation regarding this removal in Appendix H. Revision History, but the revision is documented in the response to the feedback document which immediately preceded WD18 (i.e. ‘ECF5 Spec Consideration – 13 – jec resposnes.docx’, item 15). Perhaps this was done to permit the possibility of an “identity management system”, but it does not address the issue presented in ‘ECF5 Spec Considerations – 13 .docx’, item 15 which is that the GetPolicyRequest element ecf:FilingPartyID is not appropriate by element name and definition.

1. **Judge ID**

A ‘Filer’ is defined in section 1.1 ‘Terminology’ as:

An attorney, judicial official or a *pro se* (self-represented) litigant acting as an individual who assembles and submits one or more filings (combinations of data and documents).

The term ‘judicial officer’ is understood to include judges, such as the case judge.

Section 6.2.8 Filer and Party Identifiers specifies “Identifiers for filers and parties to a case“ and specifically prescribes ecf:FilingPartyID and ecf:FilingAttorneyID. The case judge is not a party on the case, and not all judges are licensed to practice law (e.g. are not all attorneys). So when a case judges files on the case (such as an order), what ‘Filer’ ID should be used?

Perhaps instead of coining a new ecf:JudgeID or ecf:FilingJudgeID element, we can eliminate ecf:FilingPartyID, ecf:FilingAttorneyID, and ecf:AttorneyID with just a single element such as ecf:FilingEntityID or even just ecf:EntityID.

The judge is obviously not represented by an attorney and should just use ecf:FilingPartyID. We could discuss your idea of consolidating these IDs on a future TC call.

The problem with using ecf:FilingPartyID for a filing judge, is that a judge is not a party (party does not equal participant). Additionally, the definition for ecf:FilingPartyID is “a unique identifier for a person or organization that requests information” which is not even close. If schema is supposed to substitute for specification, then element names and definitions need to be correct.

Yes, let’s discuss consolidating entity participant identifiers with the TC. Also, any and all TC members can contribute to this conversation using the TC distribution list.