**ECF5 Spec Feedback and Considerations – 5**

Jim Cabral’s responses in red

This document contains additional questions and commentary resulting from a review at the Electronic Court Filing Version 5.0 Working Draft 10.

1. **CourtEventActor and CourtEventOnBehalfOfActor**

The ecf:CourtEventActor element can be found in CourtEventAugmentation (e.g. wrapper:RecordDocketingRequest/docket:RecordDocketingMessage/filing:FilingMessage/nc:Case/j:CaseAugmentation/j:CaseCourtEvent/CourtEventAugmentation/ecf:CourtEventActor).

|  |  |
| --- | --- |
| In ECF4, CourtEventActor is permitted to either be a person, organization or property: | In ECF5, only person is provided: |
|  |  |

Why are organization and property removed?

NIEM nc:RoleOf allows for nc:RoleOfItem and nc:RoleOfOrganization in addition to nc:RoleOfPerson.

Element CourtEventOnBehalfOfActor has the same issue as CourtEventActor.

Correct. I added nc:RoleOfOrganization and nc:RoleOfItem.

1. **CaseDefenseAttorney, CaseProsectionAttorney, and CaseInitiatingParty**

Why has j:CaseDefenseAttorney been dropped from j:CaseAugmentation in ECF5?

|  |  |
| --- | --- |
| Snippet from ECF4: | Snippet from ECF5: |
|  |  |

Also, why has the maxOccurs for j:CaseInitiatingParty been reduced in ECF5 to 1 and not ‘unbounded’ as it was in ECF4?

Same question for j:CaseProsecutionAttorney.

Corred. I added j:CaseDefenseAttorney and changed the cardinality of j:CaseInitiatingParty and j:CaseProsecutionAttorney to 0,unbounded.

1. **Reviewed Filing Lead/Connected Documents**

Discussion on this item began with installment 3 (Item 8) and was continued in installment 4 (Item 1).

In a nutshell, this issue addresses the need for ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument in addition to ecf:FilingLeadDocument and ecf:FilingConnectedDocument in the RecordFiling operation call to the CRMDE.

The most recent response (installment 4) is:

I was mistaken. ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument are actually mapped to the callback messages, docketcallback:NotifyDocketingCompleteMessage and filingreviewcallback:NotifyFilingReviewCompleteMessage.

Whereas I do not disagree that ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument should be included in docketcallback:NotifyDocketingCompleteMessage (docketcallback.xsd), they are also needed in docket:RecordDocketingMessage (docket.xsd).

At the most recent TC conference call (4-11-2017) we did discuss 2 options with the TC. As such, I believed that it is understood that ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument need to be available in both docket:RecordDocketingMessage and docketcallback:NotifyDocketingCompleteMessage. .

However, Working Draft 10 has now been delivered and the elements ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument are still not present in docket:RecordDocketingMessage (docket.xsd) as needed.

 The following table provided a side by side comparison of ECF4 to ECF5:

|  |  |
| --- | --- |
| So, here is an excerpt from ECF-4.0-RecordDocketingMessage.xsd: | Here is a corresponding excerpt from RecordDocketingMessage in ECF5 WD10: |
|  |  |

What’s missing are ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument.

Additionally, it is not clear what has happened to CorrectedFiling.

When invoking the RecordFiling operation, the FRMDE needs to provide the original submission and original submission document renditions (i.e. filing:FilingMessage, containing filing:FilingLeadDocuments and filing:FilingConnectdDocuments) and clerk review results (i.e. RecordDocketingMessage, containing ReviewedLeadDocuments and ReviewedConnectedDocuments, and optionally, CorrectedFiling) along with any new or revised document renditions.

I added ecf:ReviewedLeadDocument and ecf:ReviewedConnectedDocument to docket:RecordDocketingMessage. The original filing is included in docket:RecordDocketingMessage/filing:FilingMessage. The rest of docket:RecordDocketingMessage (including nc:Case, nc:ReviewedLeadDocument and nc:ReviewedConnectedDocument) constitutes the corrected filing. Did I miss anything?

1. **CaseCourt Augmentation**

Within NIEM, j:CaseCourt can contain j:CourtAugmentationPoint.

Within ECF, j:CaseCourt provides nc:OrganizationAugmentationPoint, and not j:CourtAugmentationPoint.

Why?

Also, per the NIEM 3.0 NDR, Rule 10-29 ‘Augmentation point reference must be last particle’, the augmentation point, either nc:OrganizationAugmentationPoint, or j:CourtAugmentatinPoint must be the last element in the list of elements.

There are no extensions currently in ECF that would cause the need for j:CourtAugmentationPoint. However, implementers may need to add extensions to the NIEM court element so I added j:CourtAugmentationPoint.

1. **Case Type – Traffic or Citation?**

Section 2 (Informative) Scope calls out the case type as “Violations (including traffic, ordinances and parking).

Section 4.2. Case Type Augmentations refers to the [Statistical Reporting Guide] which lists it as ‘Traffic, Parking and Local Ordinance’. The table beneath characterizes it (through namespace mnemonic) as ‘citation’

The CaseTpeCode.gc file lists it as ‘traffic’.

Should we be consistent?

Note: the hyperlink to **[Statistical Reporting Guide]** (section 1.3 Non-Normative References) (i.e. <http://www.courtstatistics.org/~/media/Microsites/Files/CSP/State%20Court%20Guide%20to%20Statistical%20Reporting%20v%20211.ashx>) is not working.

Try:

<http://courtstatistics.org/~/media/Microsites/Files/CSP/State%20Court%20Guide%20to%20Statistical%20Reporting%20v%202point1point2.ashx>

 Good suggestion. I changed the CaseTypeCode.gc list to use “citation” to be consistent with our file naming and updated the link to the NCSC guide.

1. **CourtCase**

Why are we using nc:Case and not cyfs:CourtCase?

NIEM’s nc:Case is a general purpose element that can be used for any kind of case (e.g. doctor’s case, court case, etc.) It is defined as “an aggregation of information about a set of related activities and events.” It is because of this very generic idea of a case that we have been forced to use nc:CaseTrackingID instead of ‘case number’.

Then along comes cyfs and they understand the specialization of case as court case (i.e. cyfs:CourtCase).

cyfs:CourtCase is defined as “an element that contains the details of a court case.”

There definition is more correct for our purpose than the definition of nc:Case.

Both elements can have the very same content:

|  |  |
| --- | --- |
| **nc:Case** can contain: | **cyfs:CourtCase** can contain: |
| J:CaseAugmentation | J:CaseAugmentation |
| nc:CaseCategoryText | nc:CaseCategoryText |
| nc:CaseDocketID | nc:CaseDocketID |
| nc:CaseFiling | nc:CaseFiling |
| nc:CaseGeneralcategoryText | nc:CaseGeneralcategoryText |
| nc:CaseResolutionText | nc:CaseResolutionText |
| nc:CaseSubCategoryText | nc:CaseSubCategoryText |
| nc:CaseTitleText | nc:CaseTitleText |
| nc:CaseTrackingID | nc:CaseTrackingID |
| nc:CaseYearDate | nc:CaseYearDate |

Why don’t we add j:CaseNumberText to our ‘case’ element (e.g. either nc:Case or cyfs:CourtCase)? This would presumably need to be done by adding it to ecf:CaseAugmentation for now (perhaps we can get NIEM to add it natively to cyfs:CourtCase which appears to be hs:CourtCase in NIEM4). Even, better, perhaps we can get NIEM to move CourtCase into jxdm where appears it truly belongs.

As you said, nc:Case and cyfs:CourtCase are both based nc:CaseType so it really doesn’t make much difference. I’ll submit your suggestion to rename hs:CourtCase to j:CourtCase for NIEM 4.

1. **Document Association** (continued)

After revisions to Working Draft 9, characterizing the association of documents is done using ecf:RelatedDocumentCode rather than nc:AssociationDescriptionText. This is shown in the following ECF5 XML excerpt from the civil.xml example:

 <nc:DocumentAssociation>

 <nc:PrimaryDocument structures:ref="Document1" xsi:nil="true"/>

 <ecf:DocumentAssociationAugmentation>

 <ecf:DocumentRelatedCode>parent</ecf:DocumentRelatedCode>

 </ecf:DocumentAssociationAugmentation>

 </nc:DocumentAssociation>

In the interests of interoperability, it seems that a standardized set of allowable values should be specified for ecf:RelatedDocumentCode. Is “parent” the only allowable value? Should there be a code list of allowable values?

The definition for ecf:DocumentRelatedCode (i.e. ‘A type of related document that was previously filed in this case. For instance, the document to which this document is a response.’), does not seem fit well with the intended usage as illustrated in the civil example. In the civil example, nc:DocumentAssociation appears in filing:ConnectedDocument (i.e. /filing:FilingMessage/filing:FilingConnectedDocument/ecf:DocumentAugmentation/nc:DocumentAssociation). Its purpose is to identify the filing:FilingLeadDocument that it is connected to (i.e. is a child of). As such, its parent has not yet been filed in the case as the definition for ecf:DocumentRelatedCode states.

The civil example appears to suggest that the proper usage for nc:PrimaryDocument is to just use structures:ref to reference the parent FilingLeadDocument and xsi:nil=”true” to identify that the element does not have content. The nc:ParentDocument element is of type nc:DocumentType and can therefore can contain any and all document type elements (e.g. nc:DocumentIdentification, nc:DocumentTitleText, etc.) The same is true for nc:SecondaryDocument.

In the context of using nc:DocumentAssociation within filing:FilingConnectedDocument to identify its parent FilingLeadDocument, are there any anticipated usages whereby nc:PrimaryDocument would contain sub-element content? Would nc:SecondaryDocument ever be used?

Should nc:DocumentAssociation never be used in filing:FilingLeadDocument?

For a filingConnectedDocument, should nc:DocumentAssociation be required?

When using nc:DocumentAssociation within FiilingConnectedDocument, should nc:PrimaryDocument be required? Should nc:SecondaryDocument be prohibited?

Using nc:DocumentAssociation was an attempt to provide a general and extensible framework for relating any two documents. DocumentRelatedCode.gc currently defines two possible values “parent” and “prior-related”. Are there others we should include?

nc:DocumentAassociation can be included within ecf:DocumentAugmentation which is part of both filing:FilingLeadDocument and filing:FilingConnectedDocument. Since we can infer one document from the context, we only need to reference a single related document (nc:PrimaryDocument). We cannot make nc:DocumentAugmentation required only in filing:FilingConnectedDocument.

An alternative approach would be to abandon nc:DocumentAugmentation and DocumentRelatedCode.gc completely and define separate elements for references to parent and prior-related documents, ala

filing:FilingLeadDocument/filing:PriorRelatedDocument

filing:FilingConnectedDocument/filing:ParentDocument

filing:FilingConnectedDocument/filing:PriorRelatedDocument

This approach is not extensible but would give us fine control of the cardinality of each document relationship.

 Let’s discuss which approach works best this with the TC.

1. **CaseAugmentation**

One significant change with ECF5 is how the various case types (e.g. criminal, civil, citation, etc.) are used. In ECF4 these were used by substitution to nc:Case. In ECF5, these are used through augmentation.

In addition to j:CaseAugmentation and ecf:CaseAugmentation that we are already accustomed to in ECF4, there are now (in ECF5), there are CaseAugmentation elements for each case type (i.e. criminal:CaseAugmentation, civil:CaseAugmentation, citation:CaseAugmentation, bankruptcy:CaseAugmentation, appellate:CaseAugmentation and domestic:CaseAugmentation).

So in ECF5, instead of substituting a case type element for nc:Case, a case-type-specific CaseAugmentation is substituted for nc:CaseAugmentationPoint.

Unlike ECF4, the order in which the CaseAugmentation elements appear is not defined. In ECF4 j:CaseAugmentation must always appear before ecf:CaseAugmentation. If an implementation has done an extension (e.g. aoc:CaseAugmentation), then this element must appear after ecf:CaseAugmentation.

This CaseAugmentation element sequencing is not enforced in ECF5. With ECF5, ecf:CaseAugmentation is allowed to appear before j:CaseAugmentation.

This is easily demonstrated using a tool such as XMLSpy. If you use the civil example (civil.xml), and cut the j:CaseAugmentation element, then paste it back in immediately following ecf:CaseAugmentation, the resulting XML is still well formed and valid.

In addition to the lack of sequence restrictions or enforcement for elements substituted for nc:CaseAugmentationPoint, there is also no restriction on how many times an element can be substituted. I confirmed that it is valid to use j:CaseAugmentation a second time, even if placed following ecf:CaseAugmentation.

Some XML tools such as XMLSpy provide assistance when writing XML documents. When adding a new element, a pop-up list of elements allowed at that location is provided. This is what XMLSpy provides:



This was done using the civil case example (civil.xml). Note that the criminal namespace is defined in the nc:Case element rather than the civil namespace. I believe if it had been the civil namespace instead, then civil;CaseAugmentation would have been provided in the allowable element listing instead of criminal:CaseAugmentation.

So when an operation, such as RecordDocketing, receives a message containg nc:Case (such as RecordDocketingRequest), how does the operation determine the type of case?

Is an implementation required to derive the case type from the namespace of the case-type-specific CaseAugmentation?

Consider that some submissions may require elements from multiple case types. This can be observed in the citation FilingMessage example (ciatation.xml). The citation.xml example uses elements from both the criminal namespace (i.e. criminal:ChargeAugmentation) as well as the citation namespace. The criminal namespace elements (e.g. criminal:ChargeAugmentation, criminal:ChargeAmendedIndicator) are within the j:CaseAugmentation element. Only one case-type-specific CaseAugmentation appears (i.e. citation:CaseAugmentation).

So, it is valid to have ecf:CaseAugmentation appear before j:CaseAugmentation, it is valid to have multiple ecf:CaseAugmentations or multiple j:CaseAugmentations, it is even valid to have multiple case-type-specific CaseAugmenations (whether these are for the same case-type of different case-types). To demonstrate this, I copied the ‘citation:CaseAugmentation’ from citation.xml and pasted it into civil.xml, and the xml is still valid!

The specification provides support for multiple case type augmentations in a single submission (i.e. see section 4.1 ‘Messages’, bullet 3 ‘Optionally, one or more case type augmentations …’). So if an implementation receives a message like this which contains multiple case-type-specific CaseAugmentations for multiple case-types, what kind of a case is it? Is it a civil case or is it a citation case?

Section 4.2 ‘Case Type Augmentations’ does not make it clear how the case type of the case would be known, if or when there are multiple different case type CaseAugmentations. If fact, in this section it states “case type augmentations MAY ONLY substitute for nc:CaseAugmentationPoint.” This appears to say that you cannot create your own implementation specific CaseAugmentation (such as aoc:CaseAugmentation above) that would be available for all case types. But certainly this must be an incorrect interpretation, since in section 5.2.1 ‘Case-Specific Augmentations’ is says “a court may add elements required for a particular case type (e.g. civil) by defining an extension that includes an augmentation element (e.g. court:CivilCaseAugmentation) that substitutes for an ECF augmentation point (e.g. nc:CaseAugmentationPoint).” Unless, by way of the phrase “a particular case type”, it is disallowing CaseAugmentations that are case general purpose augmentations (e.g. can be applied to all case types much like j:CaseAugmentation and ecf:CaseAugmentation). If this is the restriction, then any extension elements that a court would want to be added to all case types would need to be added to each case type specific CaseAugmentation (e.g. court:CivilCaseAugmentation).

Another option would be to impose a rule that an nc:Case element must have one and only one case-type-specific CaseAugmentation from the table in section E.2 Case-type Augmentations.

Even with this rule, to know the case type of a case, the namespaces of all CaseAugmentations would need to be examined. For example, consider the citation.xml example. It contains 3 CaseAugmentations: 1) j:CaseAugmentation, 2) ecf:CaseAugmentation, and 3) citation:CaseAugmentation. Since the namespace mnemonics are not normative, the case type determination would need to be made on the actual namespace URI, i.e. if a CaseAugmentation is from the "https://docs.oasis-open.org/legalxml-courtfiling/ns/v5.0/citation" namespace, then this is a ‘citation’ case.

Use of the ecf:CaseTypeCode element might be considered, however it is not available in an ReveiwFilingRequest or a RecordDocketingRequest. Should we add it so that the case type can be specified?

Inspecting the value of ecf:CaseTypeCode would be a lot easier that inspecting the namespace URI.

You cannot determine the case type easily by examining the augmentations – you must inspect j:CaseCourt/ecf:CaseAugmentation/ecf:CaseCategoryCode which is a court-specific code list which we assume is more detailed than ecf:CaseTypeCode. Should we require courts to provide a mapping from ecf:CaseTypeCode to ecf:CaseCategoryCode or should we include ecf:CaseTypeCode in ecf:CaseAugmentation? Let’s discuss with the TC.

1. **Contact Information Duplicity?**

The nc:PersonAugmentation element contains nc:ContactInformation. It also contains nc:ContactInformationAssociation which in turn, contains nc:ContactInformation.

Is this a redundancy, or do these serve different purposes?

For example, is PersonAugmentation/nc:ContactInformation used to specify contact information for the individual, whereas PersonAugmentation/nc:ContactInformationAssociation is used to itemize the individual’s contacts (i.e. persons or organizations on the individual’s contact list)?

The contact association details provided for ECF5 are not the same as for ECF4:

|  |  |
| --- | --- |
| ECF4 | ECF5 |
|  ECF4 | ECF5 |
|  | Same as above for organization contact association in ECF5.The following ECF4 elements do not have a ECF5 correlary:* nc:ContactInformationIsPrimaryIndicator
* nc:ContactInformationIsEmergencyIndicator
* nc:ContactInformationIsDayIndicator
* nc:ContactInformationIsEveningIndicator
* nc:ContactInformationIsNightIndicator

The last three on the above list are perhaps replaced by ContactInformationAvailibilityCode. |

To preserve the greatest level of ‘backward compatibility’ or to at least minimize the version migration challenges and costs, then at a minimum, all ECF4 supported information should also be supported in ECF5. There may be ECF4 implementations that are unknown to the TC that utilize elements which have been removed in ECF5. If there are compelling and/or necessary reasons for replacing nc:PersonContactInformationAssociationType and nc:OrganizationContactInformationAssociationType with just nc:ContactInformationAssociationType, then the person contact specific elements (i.e. nc:ContactInformationIsHomeIndicator, and nc:ContactInformationIsWorkIndicator, could be accessed through a ContactInformationAssociationAugmentationPoint, such as PersonContactInformationAssociationAugmentation.

Both nc:ContactInformation and nc:ContactInformationAssociation are required as the actual contact information is provided in nc:ContactInformation while the indicators (home, work, etc.) are provided in nc:ContactInformationAssociation.

1. **Other Augmentation**

Many elements, such as EntityPerson, contain two augmentation points (e.g. structures:ObjectAugmentationPoint and nc:PersonAugmentationPoint).

When and how is structures:ObjectAugmentationPoint used?

structures:ObjectAugmentationPoint is included in all NIEM complete types. There is no removing it.

1. **DocumentIdentification cardinality**

ECF 4.01 permits unlimited nc:DocumentIdentification elements in CoreFilingMessage. In ECF5 working draft 10, the limit in FilingMessage is at most one. Also, in ECF4, nc:DocumentIdentification is optional in CoreFilingMessage. In ECF5 it is mandatory in FilingMessage.

Why has this changed?

As used in FilingMessage, DocumentIdentification is a ‘Filing Identifier’ (see section 6.2.6 Filing Identifiers).

As stated in section 6.2.6, a ‘Filing Identifier “will be generated by the court in response to a ReviewFiling operation.” As suggested in section 6.2.6, this court generated ‘Filing Identifier’’ is returned to the FAMDE in a cbrn:MessageStatus message (cbrn.xsd). The example in section 6.2.6 shows this being done in ecf:MessageStatusAugmentation.

However, I do not see any allowed substitutions for cbrn:MessageStatusAugmentationPoint in MessageStatus (cbrn.xsd). The MessageStatus example (messagestatus.xml) contains ecf:MessageStatusAugmentation substituted for cbrn:MessageStatusAugmentationPoint and the XML is valid. Which schema establishes that ecf:MessageStatusAugmentation is permitted to substitute for cbrn:MessageStatusAugmentationPoint?

The ecf:MessageStatusAugmentation element also makes nc:DocumentIdentification mandatory and singular (i.e. minOccurs=1, maxOccurs=1). In this message, and when used in response to ReviewFiling operation request, this makes sense.

However, restricting nc:DocumentIdentification to one and only one in the FilingMessage is overly constraining. Currently in Arizona, using ECF4, we use up to three instances of nc:DocumentIdentiifcation in the CoreFilingMesage in both the ReviewFilingRequest and the RecordDocketingRequest.

Below is the text for nc:DocumentIdentification from the Arizona specification:

8.2 nc:DocumentIdentification

Standard Definition:

An identification that references a document.

AOC Usage Notes:

Per ECF 4.01 section 3.3.1.4, “Document identifiers are assigned by the court record system and MUST be unique within a court.” This element appears up to three times in each RFR and RvFR. In both messages, the first instance is used to carry the unique IdentifierID assigned to the submission by the EFM upon receipt of the ReviewFilingRequest (or RvFR) message. The population of this element is mandatory and the EFM is responsible for ensuring that IdentificationID values are unique across all participating FAMDEs. The assigned submission IdentifierID is returned to the submitting FAMDE via the MessageReceiptMessage (MRM). It is also added to the RvFR by the EFM before forwarding the RvFR to Clerk Review. This element must be present in the CFM in the RFR.

Examples

Example 1: This example describes the first usage of nc:DocumentIdentification as a submission identifier provided by the EFM. This value is not provided by the FAMDE, so this would only be present in a subsequent submission; it is not present in initial submissions.

<nc:DocumentIdentification>

<nc:IdentificationID>64756644313174747</nc:IdentificationID> <nc:IdentificationCategoryText>SubmissionID</nc:IdentificationCategoryText>

</nc:DocumentIdentification>

The second nc:DocumentIdentification element is supported for use by an FAMDE that wishes to include a local e-Filing or submission identifier.

Although the use of this element is optional, its use is also recommended. If used, this value is to be unique to the specific FAMDE in question.

The FAMDE provider should not expect the CRMDE to persist or otherwise report the value of this element.

Example 2: This example describes the second usage of nc:DocumentIdentification as a submission identifier provided by the FAMDE

<nc:DocumentIdentification>

<nc:IdentificationID>200704</nc:IdentificationID>

<nc:IdentificationCategoryText>eFilingID</nc:IdentificationCategoryText>

</nc:DocumentIdentification>

Another optional nc:DocumentIdentification element may be used include a reference number or other identifier meaningful to the FAMDE filer.

Example 3: See section 6.1.6

<nc:DocumentIdentification>

<nc:IdentificationID>CV0070476</nc:IdentificationID>

<nc:IdentificationCategoryText>ClientMatterID</nc:IdentificationCategoryText>

</nc:DocumentIdentification>

Also, the definition of Document Identifier in section 6.2.4 may need some additional clarification or context. In 6.2.6 ‘Filing Identifiers’ the additional context of the relevant messages is provided (i.e. cbrn:MessageStatus, filing:FilingSttausRequest, and filing:FilingSttausResponse). In 6.2.4, the context would be when used in/for a Document (e.g. FilingLeadDocument) and not in a message header or even in a DocumentRendition. Perhaps something such as when used in an nc:DocumentType, and not in a filing:FilingMessageType and not in an ecf:DocumentRenditionType.

ecf.xsd defines ecf:MessageStatusAugmentation which substitutes for cbrn:MessageStatusAugmentationPoint. I changed the maxOccurs cardinality of nc:DocumentIdentification to unbounded (except in cbrn:MessageStatusAugmentation) to support the use cases you described. However, minOccurs needs to be 1 because there are a number of messages (e.g. CancelFilingMessage, GetFilingStatus) that are meaningless without it. I also added the context “within a nc:Document element” to Section 6.2.4

1. **MessageStatusAugmentation ServiceRecipientID**

The ecf:MessageStatusAugmentationType is defined as shown below:



We know from section 6.2.6 that ecf:MessageStatusAugmentation is used by cbrn:MessageStatus, filing:FilingStatusRequest, and filing:FilingStatusResponse. It may also be used in other messages, such as a ServeFiling Response (see 6.1.5 ServeFiling), as a RecordDocketing response (6.1.7), in response to NotifyDocketingComplete (6.1.8), ReserveCourtDate (6.1.16), AllocateCourtDate, etc.

When used in cbrn:MessageStatus, as a response to a FilingReview operation request, why would either ecf:ServiceRecipient or ecf:ServiceStatusCode be relevant? Especially, why would ecf:ServiceReceipientID be required?

Of the 12 operations to which cbrn:MessageStatus is a response, which really require ecf:ServiceReceipientID?

Which really require nc:DocumentIdentification?

Is there a way for substitutions to cbrn:MessageStatusAugmentationPoint to be specific to the operation to which they are a response? In other words, can the MessageStatusAugmentation used in a response to ReviewFiling be different that the augmentation content when used as a response for NotifyDocumentStampInformation?

If this is possible, and appropriate to do, then how is the relevant substitution allowed element established? It seems that the answer to this question may be the same or similar to the question asked in the prior issue (i.e. “Which schema establishes that ecf:MessageStatusAugmentation is permitted to substitute for cbrn:MessageStatusAugmentationPoint?“).

I changed the cardinaliry of SrviceRecipientID to 0,1. As you said, cbrn:MessageStatus provides the response to a number of ECF request messages. The response to most request messages require nc:DocumentIdentification to provide the filingID. Only the response to ServeFiling requires the ecf:ServiceRecipientID and ecf:ServiceStatusCode. There is not a good way to limit the augmentations in the response to the specific request messages.

1. **ServiceRecipientID**

ServiceReceipientID is not listed in section 6.2. ‘Identifier Rules’.

I see section 6.2 as listing identifiers for things that are common enough and important enough that all implementations must have the same understanding; e.g. these are the things (and the identifiers for those things) that interoperable implementations must commonly recognize.

ServiceRecipientID would seem to fit this.

 I added the following to Section 6.2

### Service Recipient Identifiers

Identifiers for filers and parties to a case, including person, organizations and property, labeled as ecf:ServiceRecipientID/nc:IdentificationID, MUST correspond to filer and party identifiers. The following is a non-normative example of an identifier for filer number 100:

<ecf:ServiceRecipientID>

 <nc:IdentificationID>100<nc:IdentificationID>

</ecf:ServiceRecipientID>

1. **Filing Number**

The definition for ‘GetFilingStatusRequestMessage’ is “This is query to get a filing status by Filing Number.”

Is ‘Filing Number’ synonymous with “Filing Identifier”?

If so, should we not use consistent terminology? If not, then please clarify.

 I fixed the definition to use Filing Identifier.

1. **Incorrect Identification Type**

Certainly PersonIdentificationCategoryCode is misplaced within nc:IdentificationType when used for nc:DocumentIdentification (see filing.xsd):

ecf:PersonIdentificationCategoryCode substitutes for nc:IdentificationCategory in any identifier. Unfortunately, we cannot constrain it to person pdentifiers only.



1. **Typo**



1. **DocumentPostDate**

Okay, I initially thought that nc:DocumentPostDate had been removed in ECF5, however I now see that it has been relocated.

But isn’t it in the wrong place now?

In ECF5, it now does not appear until after both j:CaseCourt and nc:DocumentInformationCutOffDate (which has also been relocated).

However, in NIEM, nc:DocumentPostDate should appear before nc:DocumentReceivedDate as it did under ECF4.

DocumentInformationCutOffDate should appear immediately following nc:DocumentIdentiifcation.

1. **DocumentStatus absent**

ECF4 provides nc:DocumentStatus. This element is missing in ECF5.

This element is currently used in Arizona (in NotifyDocketingComplete – RecordDocketingCallbackMessage):

6.5 nc:DocumentStatus

Standard Definition: DocumentStatus is a status of a document.

AOC Usage Notes: This is the directive set in clerk review that describes the overall submission review completeness. This is used to identify when review of all documents in a CFM has been completed. This element may be used in a multi-episode clerk review scenario to identify RecordDocketingCallbackMessages (RDCM) resulting from the final review operation.

This element MUST be provided for all RDCBMs. The value provided in nc:StatusText MUST be the same for all RDCBMs within the NDC.

Example:

<nc:DocumentStatus>

<nc:StatusText>SubmissionFullyReviewed</nc:StatusText>

<nc:StatusDate>

<nc:DateTime>2011-08-24T02:34:22-07:00</nc:DateTime>

</nc:StatusDate>

<nc:StatusDescriptionText>SubmissionFullyDocketed</nc:StatusDescriptionText>

</nc:DocumentStatus>

1. **FilingMessageTypeAugmentationPoint?**

Is the augmentation point in filing:FilingMessageType correctly named nc:DocumentAugmentationPoint or should it be named like FilingMessageAugmentationPoint?

Does the name of an augmentation point affect the allowed substitutions? If renamed as suggested above, could DocumentAugmentation still be substituted for the renamed augmentation point?

Regardless of the name of the augmentation point in filing:FilingMessageType, is the content of DocumentAugmentation appropriate for FilingMessageType?

Aside from a couple 9e.g. ecf:SpecialHandlingInstructionsText, nc:Metadata), these elements in DocumentAugmentation do not appear appropriate for FilingMessageType, but would be appropriate for nc:DocumentType.

1. **RedactionRequiredIndicator**

There seems to be a typo in the definition for RedactionRequiredIndicator – “Indicator by the filer that the document must be redacted nu the court.”

1. **xxx**