**ECF5 Spec Feedback and Considerations – 10**

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 14.

1. **CancelFiling**

See section 3.2.1 ‘The Filing and Service Process’

Once the new/revised definition of party, filer and submitter have been written, then the following statement prior to the CancelFiling bullet will need to be revised to replace ‘party’ with ‘submitter’ or whatever is appropriate:

At any point during or after the ReviewFiling operation and before the RecordFiling operation a party MAY request cancellation of the filing through the following operation:

The CancelFiling operation is invoked from the FAMDE, and is provided by the FRMDE. When an FRMDE is split into an EFM and a separate Clerk Review, there are no suggestions on how CancelFiling should be implemented. It seems reasonable that Clerk Review must also support cancellation.

Is cancellation automatically granted if requested prior to the RecordFiling request?

No – it is up to the clerk or court to decide.

I presume that Court Policy should declare whether or not cancellation is supported. If not supported, and a cancellation request is received by the FRMDE anyway, what should be done? Return an error?

Yes.

It seems reasonable and mandatory that the Filing Identifier would need to be specified in the CanceFiling request message (cancel:CancelFilingMessage). Section 6.2.6 ‘Filing Identifiers’ specifies that this must be the ../nc:DocumentIdentification/nc:IdentificationID element.

There is no specification requirement that the cancellation requester be identified in the CancelFilingMessage, but the example (cancel.xml) shows that ecf:FilingPartyID being included. There is also no requirement that the cancelling entity is the same as the original submitter entity. How are spoof cancellation requests detected or prevented?

Preventing spoofed requests is beyond the scope of the specification.

The example shows a ServiceRecipientID included in the cancellation request message. If cancelled, should there be a service impact? Where is this described in the specification?

Added a new section 6.1.7 CancelFiling

If this operation is enabled by court policy, a Filing Assembly MDE MAY invoke this operation on Filing Review MDE to request cancellation of the filing but the decision to cancel the filing is the responsibility of the court or clerk. The authentication of requests and the impact of a cancellation on service is beyond the scope of this specification.

1. **Reviewed Lead and Connected Documents**

In the RecordDocketingMessage, there should be a ReviewedLeadDocument for each FilingLeadDocument reviewed. This ReviewedLeadDocument must identify the FilingLeadDocument that it references.

Additionally, for each FilingConnectedDocument that is associated with the FilingLeadDocument that has been reviewed, there should be a corresponding ReviewedConnectedDocument. The ReviewedConnectedDocument must identify the FilingConnectedDocument that it references. The exception is when a new connected document is added in clerk review for a ReviewedLeadDocument. In this circumstance, there will not be a FilingConnectedDocument for the newly added ReviewedConnectedDocument to reference.

In wd14, there is only one RecordDocketingMessage example (docket.xml). In this example the references from the review documents to the filing documents are done using reference elements as shown below:

 <ecf:ReviewedConnectedDocument structures:ref="Document2" xsi:nil="true"/>

 <ecf:ReviewedLeadDocument structures:ref="Document1" xsi:nil="true"/>

Although this is not technically incorrect and perhaps this is an example for an automated acceptance process, when review is a manual, the reviewer’s document disposition must be communicated in the RecordDocketingMessage. To do this, child elements for ReviewedLeadDocument and ReviewedConnectedDocument will be required, such as nc:DocumentSubmitter to identify the reviewer, and nc:DocumentStatus to record the outcome and the date and time of the review.

As such, the reference from the reviewed document to the filing document cannot be done using a structures:ref attribute. To do so would violate NIEM NDR rule 12-2 Element with structures:ref does not have content. The ECF5 specification does not define how this is to be done in this circumstance nor are there any examples suggesting an approach or best practice.

ECF4 at least provided some ‘hints’ as to how these references should be done, e.g. Figures 1 & 2 in section 2.3.3, which implies the use of the Document Identifier. However, even ECF4 does not provide an example illustrating these references. The example (ECF-4.0-RecordDockeingMessage.xml) does not show the use of Document Identifiers.

I believe this omission should be corrected in ECF5.

Here are three options to consider when sub-element content is required:

* nc:DocumentIdentification/nc:IdentificationID
* nc:DocumentIdentification as reference element using structures:ref
* nc:DocumentAssociation
1. **nc:DocumentIdentification/nc:IdentificationID**

This approach uses document identifiers as specified in section 6.2.4 Document Identifiers. Note that nc:DocumentIdentification is a mandatory element and must be included for all of the three options presented. This is a difference from ECF4 where nc:DocumentIdentification was optional.

The filing document must be provided an identifier value. The reviewed document identifies the filing document that it references by matching document identifier value.

1. **nc:DocumentIdentification as reference element using structures:ref**

This approach employs nc:DocumentIdentification as a reference element, as shown:

<ecf:ReviewedLeadDocument >

 <nc:DocumentIdentification structures:ref="Lead1ID" xsi:nil="true"/>

…

<filing:FilingLeadDocument structures:id="Document1" >

 <nc:DocumentIdentification structures:id="Lead1ID" xsi:nil="true"/>

…

1. **nc:DocumentAssociation**

This approach using the ecf:DocumentAugmentation/nc:DocumentAssociation element.

<ecf:ReviewedLeadDocument >

 <ecf:DocumentAugmentation>

 <nc:DocumentAssociation>

 <nc:PrimaryDocument structures:ref="LeadDocument1" xsi:nil="true"/>

 <ecf:DocumentAssociationAugmentation>

 <ecf:DocumentRelatedCode>reviewed</ecf:DocumentRelatedCode>

 </ecf:DocumentAssociationAugmentation>

 </nc:DocumentAssociation>

 </ecf:DocumentAugmentation>

Added the following to 6.2.4

ecf:ReviewedLeadDocument MUST reference filing:FilingLeadDocument and ecf:ReviewedConnectedDocument MUST reference filing:FilingConnectedDocument using nc:DocumentIdentification/nc:IdentificationID.

Changed docket.xml to use nc:DocumentIdentification/nc:IdentificationID rather than structures:ref to reference the lead and connected documents.

1. **Attorney Identifiers**

Attorney Identifiers are not explicitly discussed in section 6.2 Identifier Rules.

Yet the element ecf:FilingAttorneyID is mandatory in ecf:DocumentAugmentation.

There are no elements available on the attorney elements (e.g. j:CaseInitiatingAttorney, etc.) for populating an Attorney ID.

Also see ‘ECF5 Spec Considerations-6’, item 7.

ecf:FilingAttorneyID is described in 6.2.8

Added ecf:FilingAttorneyID to ecf:CaseOfficialAugmentation

1. **Corrected Filing**

Also see ‘CorrectedFiling-1.xml’ example.

In wd14, docket.xml example, J. Cabral illustrates the use of docket:CorrectedCase.

This example serves to illustrate Jim’s understanding of how corrections would be marked up. There is no clarification in the specification on how this should be done, so inference is derived strictly by observance from the example.

First of all, it appears that Jim would essentially provide a complete over-lay (e.g. full replacement) of nc:Case from filing:FilingMessage with docket:CorrectedCase.

An alternative approach would be to only include the revised information in CorrectedCase, then merge these revisions with nc:Case data from filing:FilingMessage.

This ‘complete over-lay’ approach rather than ‘merge’ approach is also what I recommended in ‘ECF-5 RDM Issue 53 Options-2.docx’.

The replacement docket:CorrectedCase in the example is not a fully elaborated replacement, in that case parties and attorneys are included by reference to their corresponding nc:Case party and attorney elements, rather than by full sub-element content replacement (a sort of ‘over-merge’). This makes sense since in the docket.xml example, party and attorney information is unchanged in clerk review. This also keeps it cleaner and lighter weight.

To follow this practice more broadly would require expanded use of structures:id. For example, j:CourtCase information is also not revised in clerk review, but was fully replicated in docket:CorrectedCase/j:CaseAugmentation as shown below:

 <j:CaseAugmentation>

 <j:CaseCourt>

 <nc:OrganizationIdentification>

 <nc:IdentificationID>1</nc:IdentificationID>

 </nc:OrganizationIdentification>

 <j:CourtName>King County Circuit Court</j:CourtName>

 </j:CaseCourt>

If a structures:id attribute had been used on j:CaseCourt in filing:FilingMessage/nc:Case/j:CaseAugmentaion, such as:

 <j:CaseCourt structures:id="Court1">

Then in docket:CorrectedCase, j:CaseCourt could have been ‘over-layed’ by reference, as:

 <j:CaseCourt structures:ref="Court1">

Just as had been done for the case parties and initiating attorney.

To facilitate this ‘over-lay by reference’ approach for corrected case information, more liberal use of the structures:id attribute in filing:FilingMessage would be recommended. Although the structures:id attribute could be added to filing:FilingMessage elements retrospectively as needed following clerk review, this is probably not a recommended practice.

At first blush, this ‘over-lay’ approach appears both easy and practical.

For simple corrections, this approach can work well. For instance, if the case number in nc:Case is added or revised in clerk review, then the nc:CaseTrackingID element in docket:CorrectedCase can just be added.

But what should be done if the correction is not so simple?

For example, what if the driver’s license number for the case initiating party is corrected in clerk review?

When there is no correction to the case initiating party, then the following is sufficient as in the docket.xml example:

<docket:CorrectedCase>

 <j:CaseAugmentation>

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:ref="Person1"/>

 </j:CaseInitiatingParty>

However, if the driver’s license number has been corrected, then the correction may require a ‘total replacement’ such as:

 <j:CaseInitiatingParty>

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

 <!--j:PersonAugmentation></j:PersonAugmentation-->

 <nc:PersonBirthDate>

 <nc:Date>1983-01-01</nc:Date>

 </nc:PersonBirthDate>

 <nc:PersonName>

 <nc:PersonGivenName>John</nc:PersonGivenName>

 <nc:PersonMiddleName>W.</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 </nc:PersonName>

 <nc:PersonOtherIdentification>

 <nc:IdentificationID>1234-56-789-Corrected</nc:IdentificationID>

 <ecf:PersonIdentificationCategoryCode>DriverLicense</ecf:PersonIdentificationCategoryCode>

 <nc:IdentificationSourceText>source</nc:IdentificationSourceText>

 </nc:PersonOtherIdentification>

 ...

There are a couple of challenges in using this approach:

First of all, the corrected case initiating party nc:EntityPerson element cannot have “Person1” as its structures:id value as the original case initiating party nc:EntityPerson element has already used this value and it cannot be duplicated in an XML document instance.

Secondly, the easy answer would be to use structures:ref instead of structures:id in the correction, such as:

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:ref="Person1">

 <!--j:PersonAugmentation></j:PersonAugmentation-->

 <nc:PersonBirthDate>

 <nc:Date>1983-01-01</nc:Date>

 </nc:PersonBirthDate>

 ...

However, the NIEM NDR rules prohibit this, specifically rule 12-2 Element with structures:ref does not have content.

In my proposal document (i.e. ‘ECF-5 RDM Issue 53 Options-2.docx’) I sort of side stepped this issue as seen below:

Case Initiating Party Name Correction

Example 6 from the Issue Document explores a name change correction. With this option, this is straight forward. The corrected last name would appear in the RCF within the RDM. As noted in the Issue Document, when copying the CFM data to the RCF, s:id attribute values will need treatment to avoid conflicts.

When creating the accompanying CorrectedFiling-1.xml example from the docket.xml example, the structures:id attribute value for the docket:CorrectedCase/j:CaseAugmentation/j:InitiatingParty/nc:EntityPerson element was modified by hand to a value that was not being used (i.e. “Person4”).

Perhaps an alternative approach could be recommended, such as creating a new ID by appending ‘docket:CorrectedCase:” to the copied ID. In the above example, this would have produced a new ID of “docket:CorrectedCase:Person1”. Of course other methods or procedures could be used. It should not matter to the message consumer how the IDs are reworked as long as it is done correctly.

The revised ID value may need to replace other occurrences of the uncorrected ID value. For example, in the CorrectedFiling-1.xml example, the new replacement ID value ‘Person4’ also needed to replace the prior ID value ‘Person1’ in nc:PersonAssociation/nc:Person in ecf:CaseAugmentation of docket:CorrectedCase.

The problem with structures:id values does not necessarily get any simpler if the structures:id attribute is placed at a more granular level in the XML. More specifically, would the problem be easier to resolve if in filing:FilingMessage, the case initiating party’s nc:PersonOtherIdentification for Driver’s License had a structures:id as shown:

 <nc:PersonOtherIdentification structures:id="DriverLicense1">

The answer is that it would not be necessarily be any easier to resolve since the nc:PersonIdentification element is inside of j:CaseInitiatingParty/nc:EntityPerson, and therefore all of the outer elements must be present to allow for the inner element (i.e. nc:PersonIdentification).

However, if all of the nc:EnityPerson inner elements contained a structurs:id attribute in filing:FilingMessage, then when just the dirver’s license is corrected, the corrected mark-up could be simplified such as shown below:

 <j:CaseInitiatingParty>

 <nc:EntityPerson>

 <nc:PersonBirthDate structures:ref="Person1BirthDate"/>

 <nc:PersonName structures:ref="Person1PersonName"/>

 <nc:PersonOtherIdentification>

 <nc:IdentificationID>1234-56-789-Corrected</nc:IdentificationID>

 <ecf:PersonIdentificationCategoryCode>DriverLicense</ecf:PersonIdentificationCategoryCode>

 <nc:IdentificationSourceText>source</nc:IdentificationSourceText>

 </nc:PersonOtherIdentification>

 <nc:PersonRaceText structures:ref="Person1Race"/>

 <j:PersonSexCode structures:ref="Person1Sex"/>

 <nc:PersonTaxIdentification structures:ref="Person1TaxID"/>

 ...

Implementations may want to carefully consider the case and document information that can be revised (i.e. changed, added to or removed) in clerk review, and require that the elements that convey this information to have a structures:id attribute in the filing:FilingMessage.

The same ‘full replacement’ approach used for CorrectedCase does not fit as well for reviewed documents. The reason for this is that some of the reviewed document sub-elements are needed to record the results of clerk review; e.g. DocumentStatus to record the review disposition and the review date and time, and DocumentSubmitter to record the reviewer. Other elements could also be used.

If DocumentStatus is used by the filing document, and this document status is revised or corrected in clerk review, then how can both needs be accommodated with a single DocumentStatus element?

One possible solution to this dilemma is to have clerk review specific elements for clerk review information (e.g. who, what, when, etc.). Perhaps this clerk review information could be provided by a DocumentAugmentation (e.g. DocumentReviewAugmentation that substitutes for nc:DocumentAugmentationPoint).

It would be disconcerting if the approach for ReviewedLeadDocument and ReviewedConnectedDocument is dissimilar to that used for CorrectedCase. To be more specific, if corrections to document information in clerk review is added using a ‘merge’ approach for documents, rather than a ‘full replacement’ approach as for corrected case information. But doing this does provide an alternative solution to the clerk review information needs.

To be more clear, let’s say that the clerk who performs the review of a document (e.g. ReviewedLeadDocument) is recorded in ecf:ReviewedleadDocument/nc:DocumentSubmitter, and the resulting disposition (e.g. ‘accepted’ or ‘rejected’) and the date and time of the review are recorded in ecf:ReviewedLeadDocument/nc:DocumentStatus.

If an implementation never used filing:FilingLeadDocument/nc:DocumentSubmitter and filing:FilingLeadDocument/nc:DocumentStatus in filing:FilingMessage, then ‘merging’ the clerk review document elements (e.g. DocumentSubmitter and DocumentStatus) with the FilingLeadDocument elements (e.g. DocumentCategoryText, CoumentTitleText, DocumentAugmentation, etc.) may seem a perfectly reasonable and practical thing to do. Of course if these elements were not used in filing:FilingMessage, then there would never be a conflict with clerk review usage, and a ‘replacement’ approach seems more warranted and certainly more consistent.

If however, separate and distinct clerk review elements are not provided as suggested above, then a ‘merge’ approach might be the only solution to allow an element such as DocumentStatus to be used both by the FAMDE in the lead document, and by clerk review in the reviewed document.

Furthermore, a ‘merge’ approach is fraught with other difficulties. Consider a simple case of DocumentTitleText. Let’s say that the filing document has been provided a DocumentTitleText element value of “Original Title”. If in clerk review the title is revised, then a DocumentTitleText element is included with a value such as “Revised Title”. When the document data is merged, then it seems reasonable that the value “Revised Title” replaces “Original Title”. But what if instead of revising the title in clerk review, the title was stricken? Well for text data this could be communicated by providing an empty string value for DocumentTitleText.

However, this ‘merge’ approach does not work as well for other data. Consider ecf:DocumentRendition. If in clerk review, the filer provided rendition is modified, say by adding a file stamp, or perhaps through redaction, or both, then a new instance of DocumentRendiiton would need to be added. This new DocumentRendition instance would appear in the reviewed document element (e.g. ReviewedLeadDocument). With a ‘merge’ approach, is this understood as supplementing the FilingLeadDocument set of document renditions, or replacing one of the FilingLeadDocument’s set of renditions? If multiple document renditions were provided for a FilingLeadDocument, and in clerk review one of these renditions is to be pruned out, how would this be communicated?

It therefore appears that even for reviewed documents, the ‘merge’ approach is not practical.

In the docket.xml example, the reviewed document information in the RecordDocketingMessage is done as shown below:

 <ecf:ReviewedConnectedDocument structures:ref="Document2" xsi:nil="true"/>

 <ecf:ReviewedLeadDocument structures:ref="Document1" xsi:nil="true"/>

This may make sense in an implementation that did auto-accept (no manual clerk review) since document acceptance is implied in this scenario.

However, manual clerk review is more common, and elements to capture the clerk review results need to be provided. As pointed out above, then this precludes the use of the structures:ref attribute for ReviewedLeadDocument and ReviewedConnectedDocument.

Additionally, if eschewing the ‘merge’ approach in favor of the ‘replacement’ approach, then even when there are no data changes in clerk review for a document, then all filing document information would need to be copied into the reviewed document element.

Can we redesign to avoid this?

What if there were an outer element, such as <docket:DocumentReview> that contained <ecf:ReviewedLeadDocument> or <ecf:ReviewedConnectedDocument>?

Also, what if there was a complex element, such as <docket:ReviewResults> that provided for the clerk review results, such as acceptance, reviewer and date/time.

For example, if there were no changes in clerk review:

<docket:DocketReview>

 <docket:ReviewResults>

 <nc:DocumentSubmitter>

 <nc:EntityPerson structures:id=”Clerk1”>

 <nc:PersonName>

 <nc:PersonFullName> Mary Smith</nc:PersonFullName>

 </nc:PersonName>

 </nc:EntityPerson>

 <nc:DocumentStatus>

 <nc:StatusDate>

 <nc:DateTime>2017—5-26T13:47:42.0Z</nc:DateTime>

 </nc:StatusDate>

 <nc:StatusText>accepted</nc:StatusText>

 <nc:DocumentStatus>

 </docket:ReviewResults>

 <nc:DocumentAssociation>

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

 <ecf:DocumentAssociationAugmentation>

 <ecf:DocumentRelatedCode>reviewed</ecf:DocumentRelatedCode>

 </ecf:DocumentAssociationAugmentation>

 </nc:DocumentAssociation>

 <ecf:ReviewedLeadDocument structures:ref=”Document1” xsi:nil=”true” />

</docket:DocketReview>

If there were simple corrections in clerk review, such as correction to a document category text:

<docket:DocketReview>

 <docket:ReviewResults>

 <nc:DocumentSubmitter>

 <nc:EntityPerson structures:id=”Clerk1”>

 <nc:PersonName>

 <nc:PersonFullName> Mary Smith</nc:PersonFullName>

 </nc:PersonName>

 </nc:EntityPerson>

 <nc:DocumentStatus>

 <nc:StatusDate>

 <nc:DateTime>2017—5-26T13:47:42.0Z</nc:DateTime>

 </nc:StatusDate>

 <nc:StatusText>accepted</nc:StatusText>

 <nc:DocumentStatus>

 </docket:ReviewResults>

 <nc:DocumentAssociation>

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

 <ecf:DocumentAssociationAugmentation>

 <ecf:DocumentRelatedCode>reviewed</ecf:DocumentRelatedCode>

 </ecf:DocumentAssociationAugmentation>

 </nc:DocumentAssociation>

 <ecf:ReviewedLeadDocument>

 <nc:DocumentCategoryText>Apperance-Corrected</nc:DocumentCategoryText>

 <nc:DocumentSoftwareName>Microsoft Word</nc:DocumentSoftwareName>

 <nc:DocumentDescriptionText>Appearance</nc:DocumentDescriptionText>

 <nc:DocumentEffectiveDate>

 <nc:Date>2008-07-07</nc:Date>

 </nc:DocumentEffectiveDate>

 <nc:DocumentFileControlID>2</nc:DocumentFileControlID>

 <nc:DocumentIdentification>

 <nc:IdentificationID>2</nc:IdentificationID>

 </nc:DocumentIdentification>

 <nc:DocumentSequenceID>2</nc:DocumentSequenceID>

 <nc:DocumentSubmitter>

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

 </nc:DocumentSubmitter>

 <ecf:DocumentAugmentation structures:ref="Document1Augmentation" xsi:nil="true"/>

 </filing:FilingLeadDocument>

 </ecf:ReviewedLeadDocument >

</docket:DocketReview>

Or if a new redacted rendition is added:

<docket:DocketReview>

 <docket:ReviewResults>

 <nc:DocumentSubmitter>

 <nc:EntityPerson structures:id=”Clerk1”>

 <nc:PersonName>

 <nc:PersonFullName> Mary Smith</nc:PersonFullName>

 </nc:PersonName>

 </nc:EntityPerson>

 <nc:DocumentStatus>

 <nc:StatusDate>

 <nc:DateTime>2017—5-26T13:47:42.0Z</nc:DateTime>

 </nc:StatusDate>

 <nc:StatusText>accepted</nc:StatusText>

 <nc:DocumentStatus>

 </docket:ReviewResults>

 <nc:DocumentAssociation>

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

 <ecf:DocumentAssociationAugmentation>

 <ecf:DocumentRelatedCode>reviewed</ecf:DocumentRelatedCode>

 </ecf:DocumentAssociationAugmentation>

 </nc:DocumentAssociation>

 <ecf:ReviewedLeadDocument>

 <nc:DocumentCategoryText>Apperance-Corrected</nc:DocumentCategoryText>

 <nc:DocumentSoftwareName>Microsoft Word</nc:DocumentSoftwareName>

 <nc:DocumentDescriptionText>Appearance</nc:DocumentDescriptionText>

 <nc:DocumentEffectiveDate>

 <nc:Date>2008-07-07</nc:Date>

 </nc:DocumentEffectiveDate>

 <nc:DocumentFileControlID>2</nc:DocumentFileControlID>

 <nc:DocumentIdentification>

 <nc:IdentificationID>2</nc:IdentificationID>

 </nc:DocumentIdentification>

 <nc:DocumentSequenceID>2</nc:DocumentSequenceID>

 <nc:DocumentSubmitter>

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

 </nc:DocumentSubmitter>

 <ecf:DocumentAugmentation>

 <ecf:DocumentRendition structures:ref="Document1Rendition1" xsi:nil="true"/>

 <ecf:DocumentRendition structures:id="Document1Rendition2" >

 <nc:DocumentIdentification/>

 <nc:Attachment structures:id="Attachment3">

 <nc:BinaryDescriptionText>Appearance</nc:BinaryDescriptionText>

 <nc:BinaryFormatText>application/pdf</nc:BinaryFormatText>

 <nc:BinaryURI>cid://Payload3</nc:BinaryURI>

 <nc:BinarySizeValue>35000</nc:BinarySizeValue>

 </nc:Attachment>

 </ecf:DocumentRendition>

 <ecf:FilingAttorneyID>

 <nc:IdentificationID>100</nc:IdentificationID>

 </ecf:FilingAttorneyID>

 <ecf:RedactionRequiredIndicator>false</ecf:RedactionRequiredIndicator>

 <ecf:RegisterActionDescriptionCode>Appearance</ecf:RegisterActionDescriptionCode>

 <nc:Metadata structures:id="Document1Metadata">

 <nc:SensitivityText>public</nc:SensitivityText>

 <nc:LanguageCode>eng</nc:LanguageCode>

 </nc:Metadata>

 </ecf:DocumentAugmentation>

 </filing:FilingLeadDocument>

 </ecf:ReviewedLeadDocument >

</docket:DocketReview>

Although these redesign suggestions provide considerable benefit to the ‘clerk review – no document modification’ scenario, and reasonable benefit in the ‘simple document data modification’ case, it is still a heavy-weight copy when there are any revisions to ecf:DocumentAugmentation. I’m not sure there is anything that can be done about this aside from flattening the structure.

Note: In the CorrectedFiling-1.xml example, reviewed document information is provided as it would be using a ‘merge’ approach even though this approach is not recommended. This was necessary for this example since proposed redesign elements are not available.

 Replaced docket.xml with the example. Let’s discuss the potential addition of docket:DocketReview with the TC.

1. **xxx**