**ECF 4.1 WD05 and Web SIP WD03 Feedback**

This document provides review feedback to the recently released ECF 4.1 WD05 specification and ECF Web Services SIP WD03 specification.

**ECF 4.1 WD05**

1. Typo on line 606. There is an extra leading space character before the word ‘or’ and a missing space character after the word ‘or’ so that it reads “… existing case, orafter the …”.

2. Feedback for WD04 reported that “the section numbering for ‘Filer and Party Identifiers’ (was 3.3.1.8) got whacked and needs restoration’.

Unfortunately, the restoration in WD05 is worse than the ‘whacked’ version.

It now appears as:



Here is what may need to be done:

a) Remove the section number 3.3.1.8, so that this paragraph begins as “ECF 4.1 messages that support asynchronous … that include these elements.” This text is in the wrong style (e.g., font size and text color).

b) add “3.3.1.8 Filer and Party Identifiers” using the appropriate section numbering and heading style (purple text, Arial 12 font, etc.). Currently the text “Filer and Party Identifiers” is included in the paragraph for ‘Asynchronous responses” and should be pulled out as a separate section heading.

3. Is there a missing ‘period’ character in the reference in the Revision History for WD05 ‘3.27’ such that is should actually say ‘3.2.7’? (e.g., referring to section 3.2.7 NotifyDocketingComplete).

4. Typo in 3.2.8 NotyifyFilingReviewComplete – A space character should be inserted between ‘MDE’ and ‘MUST’ on line 689.

5. A second typo in section 3.2.8 – line 692. There should be a space character between “documents,” and “but”.

6. Can we discontinue the struck-through ‘2’ digit in the FIPS 180-4 reference for ECF 4.1?

7. Re: 3.2.7 and 3.2.8 and other matters (bear with me on this one, there’s a lot to it).

At the Aug. TC meeting, we went through feedback that I had provided on ECF 4.1 WD04.

The collection of feedback included specific feedback on sections 3.2.8 and 3.2.9 which prompted a lot of wordsmithing discussion (not so much content discussion) chiefly lead by Jim McMillan. We did not settle on specific wording at the meeting, leaving it up to Jim Cabral to finalize in the wording in the next working draft (which is now available as WD05).

I believe that this needs further consideration and possibly further revision.

Here are the two sections from the ECF 4.1 WD04 specification that I provided feedback to:

### 3.2.7 NotifyDocketingComplete

The Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE as a callback message to the RecordFiling operation to indicate whether the filing was accepted or rejected by the court record system. If the Court Record MDE rejected the filing, an explanation MUST be provided. If the Court Record MDE accepts the filing, the docketing information (e.g. date and time the document was entered into the court record, judge assigned, document identifiers and next court event scheduled) MUST be provided. The Filing Review MDE responds synchronously with an acknowledgement of the callback message.

### 3.2.8 NotifyFilingReviewComplete

If the clerk rejects the filings or the Filing Review MDE receives the Notify Docketing Complete message, the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted and docketed by the clerk and court record system. The operation MAY return the filed documents or links to the documents,but MUST include the **[FIPS 180-~~2~~4]** SHA 256 document hash, a condensed representation of a document intended to protect document integrity.

If the filing included a payment, and the filing was accepted by the clerk and court record system, a receipt for the payment MUST be included in the operation. The Filing Assembly MDE responds synchronously with an acknowledgement of the callback message.

Below is the feedback that I provided regarding the WD04 sections presented above:

9. 3.2.8 NotifyFilingReviewComplete – if NotifyFilingReviewComplete is now optional (per Court Policy) then the word “MUST” (line 687) requires revision or needs to be qualified, such as:

If the clerk rejects the filings or the Filing Review MDE receives the Notify Docketing Complete message, the Filing Review MDE either MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted and docketed by the clerk and court record system, or if the Court Policy does not mandate the NotifyFilingReviewComplete operation, and the Review Filing Request provided SendingMDELocationID and SendingMDEPorfileCode, then the Filing Review MAY invoke NotifyFilingReviewComplete.

10. 3.2.7 NotifyDocketingComplete – If NotifyDocketingComplete is now optional, then this section may need similar treatment as above for NotifyFilingReviewComplete.

And here is the revision in the most recent working draft (WD05) (Note: I added the yellow highlighting so that the changes are more readily apparent):

### 3.2.7 NotifyDocketingComplete

If the <RequireAsynchronousResponsesIndicator> in the CourtPolicyResponseMessage is “true”, the Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE as a callback message to the RecordFiling operation to indicate whether the filing was accepted or rejected by the court record system. If the Court Record MDE rejected the filing, an explanation MUST be provided. If the Court Record MDE accepts the filing, the docketing information (e.g. date and time the document was entered into the court record, judge assigned, document identifiers and next court event scheduled) MUST be provided. The Filing Review MDE responds synchronously with an acknowledgement of the callback message.

### 3.2.8 NotifyFilingReviewComplete

If the clerk rejects the filings or the Filing Review MDE receives the NotifyDocketingComplete message and the <RequireAsynchronousResponsesIndicator> in the CourtPolicyResponseMessage is “true”, the Filing Review MDEMUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted and docketed by the clerk and court record system. The operation MAY return the filed documents or links to the documents,but MUST include the **[FIPS 180-~~2~~4]** SHA 256 document hash, a condensed representation of a document intended to protect document integrity.

If the filing included a payment, and the filing was accepted by the clerk and court record system, a receipt for the payment MUST be included in the operation. The Filing Assembly MDE responds synchronously with an acknowledgement of the callback message.

Okay, now for the point of this discussion/consideration – Are the most recent modifications in WD05 accurate, and correct?

In both 3.2.7 and 3.2.8, Cabral makes the CourtPolicyResponseMessage an integral part of the condition for whether the MDE MUST invoke the specified operation.

I think this is not right, since CourtPolicy is optional, and therefore there may not be any CourtPolicyResponseMessage. When there is no CourtPolicyResponseMessage, then <RequireAsynchronousResponseIndicator> cannot be true. As such, the operation invocation would not be required.

Consider:

1. New in ECF 4.1 is that asynchronous responses are no longer mandatory.

This has been indicated in the ECF spec by:

* Relaxing the cardinality for ecf:SendingMDELocationID and ecf:SendingMDEProfileCode in ecf:CaseFilingType which are now both optional elements.
* Proving an explanation in the written specification in section 3.3.1.7 ‘Asynchronous responses’ as follows: ECF 4.1 messages that support asynchronous responses include <SendingMDELocationID> and <SendingMDEProfileCode> to support the return of the asynchronous response to the sending MDE. If the <RequireAsynchronousResponsesIndicator> in the CourtPolicyResponseMessage is “true”, then both <SendingMDELocationID> and <SendingMDEProfileCode> MUST be included in all ECF 4.1 messages that include these elements.
* Removing the bold-type font for NotifyDocketingComplete and NotifyFilingReviewComplete operations in section 3.1. Bold type was used to indicate that an operation was required in an ECF compliant implementation. Removing the bold-type font indicates that the operation is optional.
* Support for this new asynchronous response optionality has been extended into CourtPolicy through the inclusion of the RequireAsynchronousResponseIndicator in the CourtPolicyResponseMessage as described in section 3.3.1.7 (provided above).

For asynchronous responses to be optional, then the either or both <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> must be absent or empty in a request message that allows an asynchronous response. It does not require that <RequireAsynchronousResponseIndicator> in the CourtPolicyResponseMessage must be false.

2. Since CourtPolicy is optional, then when not used by an implementation, there will not be any CourtPolicyResponseMessage.

3. So, when there is no CourtPolicyResponseMessage, then are asynchronous responses required?

A partial answer appears to be provided in 3.3.1.7 Asynchronous responses. This section provides that “if the <RequireAsynchronousResponseIndicator> in the CourtPolicyResponseMessage is “true”, then ...” (in so many words) asynchronous responses are required.

When something is ‘not true’, then it is ‘false’ (or perhaps unknown/null if ‘nillable’).

4. What ECF does provide (as described by section 3.3.1.7) is a means for an implementation to require asynchronous responses through Court Policy. What ECF does not say is that asynchronous responses are only valid or can only be provided when required by Court Policy (through <RequireAsynchronousResponseIndicator>).

When <RequireAsynchronousResponseIndicator> is ‘true’, then this has the impact of saying the <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> are mandatory in exchanges that provide these elements.

5. So, when there is no CourtPolicyResponseMessage, and hence no <RequireAsynchronousResponseIndicator>, are asynchronous responses required?

When ECF CourtPolicy is not used then whether an asynchronous response is required is determined from the exchange instance (e.g., the ‘Request’). When the ‘request’ provides <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> (with valid element content) then the asynchronous response is required, otherwise it is an error (invalid <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> element content, or not both <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> provided) or the asynchronous response is not required (and presumably should not be provided).

So, the implication for these considerations is that the WD05 modifications in sections 3.2.7 and 3.2.8 for when the MDE MUST invoke the specified operation are incomplete and inaccurate.

Here is a Decision Tree:

a. Does the Request allow an asynchronous response?

 If no, then ‘Done’.

b. Does the Request include <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode>?

 B1. If yes, then:

B1.1 Do both <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> provide valid values?

 B1.1a If yes, then ‘Done’ – asynchronous response is expected (e.g., required).

B1.ib If no, then ‘Error’ – synchronous responses are expected, but necessary <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> do not provide valid values.

 B2. If no, then

B2.1. Did the MDE receive a CourtPolicyResponseMessage with <RequireAsynchronousResponseIndicator> with a value ‘true’?

 B2.1a If yes, then ‘Done’ – asynchronous response is required.

B2.1b If no, then ‘Error’ – synchronous responses are required, but necessary <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> do not provide valid values.

Footnote to B2.1 above – Court Policy is described in the ECF specification as an overall implementation-wide policy, seemingly applicable to all MDEs (does this also make sense for Service MDE?). As such, if in an implementation, a ‘Court’ wants asynchronous responses to ReviewFiling requests to be optional, but required for RecordFiling requests asynchronous responses, then this cannot be done with a single Court Policy (e.g., since the single implementation-wide Court Policy applies to both the FilingReview MDE and the CourtRecord MDE).

So finally, below is my attempt to redraft sections 3.2.7 and 3.2.8 to include the above considerations (revised text is in purple):

### 3.2.7 NotifyDocketingComplete

The Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE as a callback message to the RecordFiling operation to indicate whether the filing was accepted or rejected by the court record system, when either:

1) the RecordFiling message provides valid <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> elements, or

2) the Filing Review MDE received a CourtPolicyResponseMessage with <RequireAsynchronousResponsesIndicator> with a value ‘true’.

If the Court Record MDE rejected the filing, an explanation MUST be provided. If the Court Record MDE accepts the filing, the docketing information (e.g., date and time the document was entered into the court record, judge assigned, document identifiers and next court event scheduled) MUST be provided. The Filing Review MDE responds synchronously with an acknowledgement of the callback message.

### 3.2.8 NotifyFilingReviewComplete

If the clerk rejects the filings or the Filing Review MDE receives the NotifyDocketingComplete message, the Filing Review MDE MUST invoke the NotifyFilingReviewComplete operation on the Filing Assembly MDE as a callback message to the ReviewFiling operation, to indicate whether the filing was accepted and docketed by the clerk and court record system, when either:

1) the NotifyFilingReviewComplete message provides valid <ecf:SendingMDELocationID> and <ecf:SendingMDEProfileCode> elements, or

2) the Filing Review MDE received a CourtPolicyResponseMessage with <RequireAsynchronousResponsesIndicator> with a value ‘true’.

When the NotifyFilingReview operation is invoked, the operation request MAY return the filed documents or links to the documents but MUST include the **[FIPS 180-4]** SHA 256 document hash, a condensed representation of a document intended to protect document integrity.

If the filing included a payment, and the filing was accepted by the clerk and court record system, a receipt for the payment MUST be included in the operation request. The Filing Assembly MDE responds synchronously with an acknowledgement of the callback message.

Note that I have highlighted through the words “the clerk rejects the filings or” in 3.2.8 above. I did this because these words may require additional consideration.

It is still true that the RecordFiling operation is a mandatory ECF 4.1 operation, even when a submission is fully or partially rejected by a clerk (in FilingReview MDE). As such, clerk determination of the acceptability of the submission is irrelevant to the need to call RecordFiling.

However, now with ECF 4.1, even when RecordFiling has been called (as required), it does not follow that there will/must be any NotifyDocketingComplete request.

So, is it the ECF TC’s intention to make invocation of NotifyDocketingComplete mandatory when a clerk has rejected a submission (in full? Or in-part?) without regard for Court Policy and/or inclusion of ecf:SendingMDELocationID/ecf:SendingProfileCode in the RecordFilingRequest?

Congratulations, you have made it through Issue #7.

8. In Appendix F, please add ‘Civil’ to Arizona Administrative Office of the Courts, as in:

* Arizona Administrative Office of the Courts
	+ Appellate, Civil

**ECF 4.1 WSSIP WD03**

1. Typo on line 285 – there needs to be a space character inserted between “operation” and “being”.

2. Section 1.2.4 W3C Web Services Description Language (WSDL) 1.1 includes the following:

An MDE implementation MUST consist of a [SOAP 1.1] web service that implements the SOAP HTTP binding for that MDE’s portType from the corresponding MDE WSDL document provided with this specification (e.g. CourtRecordMDE.wsdl). Further, the implementation MUST be accompanied by an implementation-specific WSDL document that imports the namespace defined in the MDE WDSL, and defines a <wsdl:service> element containing a <soap:address> element with a location attribute whose value provides an HTTP URL at which the MDE implementation can be invoked.

(Note that in the previous paragraph, a namespace prefix of “wsdl” is assumed to map to the <http://schemas.xmlsoap.org/wsdl/> namespace, while the namespace prefix of “soap” is assumed to map to the <http://schemas.xmlsoap.org/wsdl/soap/> namespace.)

An example (non-normative) implementation-specific WSDL document for each MDE (e.g. examples/CourtRecordMDE-ImplementationExample.xsdl) is provided with this specification.

The language provided above for WD03 is similar to the language used in section 1.2.4 for ECF 4.0 WSSIP v2.01, having only been adjusted for changed WSDL file names (e.g., replacing “ECF-4.0-WebServicesProfile-Definitions.wsdl” with “e.g., CourtRecordMDE.wsdl”).

Although the updated ECF 4.1 WSSIP language in section 1.2.4 is not fundamentally changed from prior WSSIP versions, clear understanding of these statements is the point of this feedback item.

It appears that both the current WD03 language and the language of prior WSSIP versions require an “implementation-specific WSDL document that imports the namespace defined” (emphasis added) in the WebServices SIP provided WSDL.

For prior WS SIP versions, this WSDL is ECF-4.0-WebServicesProfile-Definitions.wsdl.

For ECF 4.1 WS SIP, this WSDL is either CourtRecordMDE.wsdl, FilingAssemblyMDE.wsdl, FilingReviewMDE.wsdl, or ServceMDE.wsdl.

The word ‘imports’ used in section 1.2.4 is understood to be the WSDL import element (e.g., statement) as shown on line 401 in Appendix C for the WSSIP 4.1 WD03 spec, and repeated below:

<import namespace="urn:oasis:names:tc:legalxml-courtfiling:schema:wsdl:CourtRecordMDE-4.1" location="CourtRecordMDE.wsdl"/>

Both the current ECF 4.1 WS SIP WD03 and prior WS SIP versions include non-normative example WSDL implementation that includes a WSDL import element similar to the WSDL import shown above (the namespace and location attribute values differ) (see Appendix C).

Also, all four ECF 4.1 WS SIP provided WSDL and the prior version’s WS SIP provided WSDL include the following commentary line at the bottom of the WSDL:

 <!-- services not defined here...defined in an implementation-specific WSDL that imports this one -->

So, it seems clear that:

* The WS SIP specification requires the use of the specification provided WSDL.
	+ Its less clear whether the exact/full specification provided WSDL must be used, or
	+ Whether only the binding element section must be used.
	+ It is also not clear whether implementations must use the specification provided WSDL filenames or whether they can provide any valid filename of their choosing.
* The specification requires implementation-specific WSDL
* Requires Implementation-specific WSDL to import the specification provided WSDL
* Does not prohibit implementation-specific edits or modifications to specification provided WSDL

Over the years, I have seen, and have heard about, various WS SIP WSDL implementations, and I have discussed these matters with various ECF implementers, both in and out of the TC. I would say that there is not a common understanding or agreement on the provisions of section 1.2.4 (frankly, nobody is especially cognizant of this section, perhaps focusing only on the SOAP 1.1 aspect).

Perhaps clarification is in order.

Also consider that svcutil.exe (ServiceModel Metadata Utility Tool), provided by Microsoft is a commonly used tool for code generation. However, svcutil has been reported as failing when importing one WSDL into another (such as importing the WS SIP provided WSDL into implementation-specific WSDL). As such, it is not uncommon for a single WSDL, that combines the WS SIP provided WSDL with the implementation-specific WSDL, to be used instead of multiple WSDL (as seemingly required).

Any possible resolution/revision here may be applicable to LEGALXMLCO-3 ‘Consider how to improve reusability to ECF implementations’ and/or LEGALXMLCO-2 ‘Provide .NET implementation guidance’.

3. xxx