Discussion of Tyler ECF 5 Proposals
December 8, 2015

Background

The ECF TC discussed Tyler's ECF5 proposals at the face-to-face meeting in September. This document provides Tyler's feedback in regards to those discussions based upon the meeting minutes.

Proposal #1: CancelFiling API

Notes from the TC meeting

CancelFiling – optional, court policy, error responses (consider separate return for correction and resubmission message perhaps via capture of original filing ID)

Tyler Responses

Optional: Indeed, we do agree that this would be an optional operation.

Court Policy: Yes, GetPolicy's SupportedOperationName would be used to reflect whether the system/court supports this operation.

Error Responses: The mention of "return for correction" suggests the need for clarification. The proposed CancelFiling API is intended as a filer-initiated action – not a clerk rejecting or returning for correction, in which case return for correction and subsequent resubmission does not apply. The filer would simply resubmit as they see fit with no correlation to the original filing.

NOTE: This operation is proposed as a query operation – to facilitate a synchronous response that indicates whether the cancellation was successful or not, thereby allowing a court/system to set business conditions defining when a cancellation is allowed (e.g. cancel cannot be performed after the clerk has already begin reviewing the filing).

Discussion of Tyler ECF 5 Proposals
December 8, 2015

Proposal #2: CreateCase/NotifyCaseAssignmentComplete

<u>Notes from the TC meeting</u>

Case Assignment (CreateCase, NotifyCaseAssignmentComplete – may consider renaming these and/or allow for modified process where case record may not actually need to create the case – may want to adjust sequence so the RecordFiling is still used to get what's needed for stamp - something has to exist to support stamping)

<u>Tyler Responses</u>

We are not quite clear on the points above, but since they make mention of renaming the operations and a modified process, let us reiterate and/or clarify the context as to our proposal in preparation for additional discussion with the TC:

1. This proposal only applies to architectures whereby clerk review/acceptance takes place within the EFM. Thus, clerk review occurs prior to making calls (CreateCase and RecordFiling) to the CMS.
2. Once accepted, funds are captured for the filing. This also occurs prior to making calls to the CMS, thereby ensuring funds are captured prior to sending any information to the official system of record for the court.
3. CreateCase is named as such to indicate what is needed from the CMS – to create the case record in order to provide case assignment information in the callback to the EFM.
4. Likewise, NotifyCaseAssignmentComplete is named to indicate the purpose of passing the newly assigned case information to the EFM.

Proposal #3: Payment Information (to the CMS)

<u>Notes from the TC meeting</u>

Payment information –
o Provision for fee breakdown?
o Include UBL element for fee categories
• Include payment message as option on RecordDocketing
• The TC generally agrees with intent of suggested operations
• Phillip Baughman should recommend an approach to meet requirements of suggested operations

<u>Tyler Responses</u>

Note that our original proposal included two possible approaches. While we (Tyler) initially recommended the first approach (adding the existing ECF PaymentMessage to the RecordFiling operation), subsequent feedback suggests that the second approach would be preferred in order to avoid impact at the WSDL level. This, the responses below pertain to the second approach in the original proposal:

Our proposal's ProviderCharge structure is essentially a PaymentMessage embedded within the CoreFilingMessage. Alternate approaches would be:

1. Embedding it within the RecordDocketingMessage
2. Modeling it after a PaymentReceiptMessage instead of a PaymentMessage

However, on both points we believe the current proposal is more appropriate in order to maintain neutrality as to whether the payment processing takes place within the EFM or the CMS.

The child element AllowanceCharge would be repeated as needed in order to reflect a breakdown of each relevant fee category.