

OPEN XML COURT INTERFACE



Electronic Filing Manager Software Requirements

May 17, 2004



TABLE OF CONTENTS

	<u>Page</u>
I. OVERVIEW	2
A. COURT FILING OVERVIEW.....	2
B. DOCUMENT ORGANIZATION	4
II. FUNCTIONAL REQUIREMENTS	6
A. ATTORNEYS AND PRO SE DEFENDANTS AND PLAINTIFFS.....	6
B. JUDICIAL OFFICERS AND JUDICIAL SUPPORT STAFF.....	7
C. COURT CLERKS.....	8
D. CLERK STAFF	10
E. SYSTEM ADMINISTRATORS (SUPERUSERS).....	11
III. TECHNICAL REQUIREMENTS	14
A. INFORMATION REQUIREMENTS.....	14
B. INTEGRATION REQUIREMENTS.....	15
IV. BUSINESS AND POLICY REQUIREMENTS.....	19
A. POLICY REQUIREMENTS	19
B. BUSINESS MODEL REQUIREMENTS	21

APPENDIX A – GLOSSARY
APPENDIX B – REVISION HISTORY

I. OVERVIEW

I. OVERVIEW

The Open eXtensible Markup Language (XML) Court Interface (OXCI) consortium of state courts intends to produce a middleware implementation for electronic filing for use within all levels of state courts for the receipt, transmission, and validation of electronic filings, court orders, and associated data. The middleware will provide a uniform open source implementation of an Electronic Filing Manager (EFM), compliant with the specifications developed by the LegalXML Electronic Court Filing Technical Committee (TC) of the Organization for the Advancement of Structured Information Standards (OASIS). However, these interface specifications are still in development and are not a sufficient basis for a complete implementation. This document is intended to define the technical design requirements for developing a complete architecture for electronic filing. The goal of this document is to provide a detailed description of the software requirements of the OXCI EFM system. The EFM is one piece of software that acts with other software systems to facilitate the electronic filing of court documents.

A. COURT FILING OVERVIEW

The general components of court filing and the flow of filing information are illustrated in EXHIBIT I. Each component and the overall court filing flow of information are described below.

1. Electronic Filing Service Provider

The Electronic Filing Service Provider (EFSP) is a front-end application for attorneys, pro se litigants, and the members of the general public wishing to review, or file, court documents. This filing component must gather all information necessary to accommodate the users and functions part of the listed functional requirements. This software should communicate with the EFM via Simple Object Access Protocol (SOAP) and World Wide Web Consortium (W3C) schemas outlined below in the Information Requirements and Integration Requirements subsections in Section III. The EFSP communication layer with the EFM is expected to be bidirectional; in other words, queries regarding case information can travel to the EFM, and notices and orders may travel through the EFM to the EFSP.

2. EFM

The EFM provides a front-end user interface for court staff members. This allows the court staff to meet the requirements outlined in the users and functions part. The EFM also provides bidirectional query and response capability with any EFSP. The requirements for this communication layer are outlined in the Information Requirements and Integration Requirements subsections. The EFM also

provides a generic interface for bidirectional query and response capability with the Case Management System (CMS) and Document Management System (DMS). The requirements for these layers of communication are also outlined in the Information Requirements and Integration Requirements subsections. The communication layer for CMS and DMS query and response is implemented in a generic format. This format is intended to be easily customizable in order to meet the needs of many different courts through out the country.

3. CMS

The CMS is local to any given court. This project will not attempt to provide any of the courts with a complete CMS solution. Instead, this project will define a basic communication layer and interface to be used for bidirectional communication between EFM and CMS. The requirements outlined in the Information Requirements and Integration Requirements subsections will define both the Application Program Interface (API) for query and response information, along with the W3C schemas used to encapsulate the interface.

4. DMS

The DMS is local to any given court. This project will not attempt to provide any of the courts with a complete DMS solution. Instead, this project will define a basic communication layer and interface to be used for bidirectional communication between EFM and DMS. The requirements outlined in the Information Requirements and Integration Requirements subsections will define both the API for query and response information, along with the W3C schemas used to encapsulate the interface.

5. The Court Filing Flow

The EFSP is used by attorneys, pro se litigants, court staff, and the general public as an interface to view public court information, such as previous filings that have been made, and to allow for the filing of documents with new or previously filed cases. The EFSP passes these filings and requests for information to the EFM. The EFM can receive requests for information and query the CMS or DMS to send a response back to the EFSP with the information. The EFM is also usable by the court staff to review case filings as they are received from the EFSP. The court staff can accept or reject a filing. Accepted filings can be passed on to the court CMS and DMS. Rejected filings never make it past the EFM, therefore not cluttering up a CMS or DMS. Filings already processed, rejected, or approved can be removed from the EFM to save storage space. The EFM can also send acceptance or rejection notifications back to an EFSP or to a filer's e-mail address.

B. DOCUMENT ORGANIZATION

Section II describes compliance of the OXCI EFM with the functional requirements listed in the OXCI EFM Architecture. It groups potential end users of the system and lists which features will be required by which group. Each individual requirement listed will contain a detailed description and an indication of whether or not the OpenEFM, on which the OXCI EFM is based, currently meets this requirement.

Section III describes compliance of the OXCI EFM with the technical requirements listed in the OXCI EFM Architecture, including the information and integration standards.

Section IV describes compliance of the OXCI EFM with the business and policy requirements listed in the OXCI EFM Architecture.

APPENDIX A will include a list of definitions and acronyms used throughout the document.

APPENDIX B will contain a revision history of this document.

II. FUNCTIONAL REQUIREMENTS

II. FUNCTIONAL REQUIREMENTS

The OXCI EFM Architecture lists the following users to be supported by the OXCI EFM:

- Attorneys and Pro Se Defendants and Plaintiffs
- Judicial Officers and Judicial Support Staff
- Court Clerks
- Clerk Staff
- System Administrators (Superusers)

This section describes the functions to be supported by the OXCI EFM for each of these users.

A. ATTORNEYS AND PRO SE DEFENDANTS AND PLAINTIFFS

1. File Pleadings

Documents can be filed to the EFM in the form of a court filing envelope. Any EFSP capable of adhering to the court filing requirement outlined in subsection III.A., Information Requirements, can access the functionality. These filings can be subsequent, associated with current cases active in the court system, or initial filings that are not associated with a current case.

Status: OpenEFM has met this requirement. Previous configurations of the EFM have accepted filings in the form of the LegalXML Court Filing 1.1 specification. These filings have been accepted via HyperText Transfer Protocol Secure (HTTPS) and SOAP.

2. Send and Receive Notifications

OpenEFM will be expected to act as a conduit for query and response messages. The messages may include the passing of court notifications. OpenEFM will be expected to receive notifications from a CMS or DMS and send the notifications on to an EFSP. Likewise, messages may come from an EFSP and require being passed on to a CMS or DMS. No user interface will be required for this. The sending and receiving of notifications will be available through the query and response API.

Status: OpenEFM has not met this requirement. In the past, query and response specifications have not been part of any of the functional integration tests the OpenEFM has successfully been used in. However, a framework is currently in place to support these features. This framework will rely on the query and response specification schemas currently being produced.

3. Review Pleadings, Orders, and Notices of Individual Cases

The EFM will provide access to pleadings, orders, and notices of individual cases that have been entered into the CMS and/or DMS. They will be available through the query and response schemas outlined in the Information Requirements subsection. No user interface is provided for this; instead an API (query/response) will allow access to these documents.

Status: OpenEFM has not met this requirement. In the past, query and response specifications have not been part of any of the functional integration tests the OpenEFM has successfully been used in. However, a framework is currently in place to support these features.

4. Open Criminal and Civil Cases

Filers will be able to file documents that are not currently associated with an active case at the court. Court personnel can then create new court cases from these initial filings. These filings must be submitted through the filing interface provided by the EFM. The interface is an API, not a user interface.

Status: OpenEFM has met this requirement. Once the EFM has processed a filing, it is then placed in a queue for court staff review. After court personnel have accepted the filing, it is passed to the CMS connector so that it can be passed directly to a court's CMS and/or DMS.

B. JUDICIAL OFFICERS AND JUDICIAL SUPPORT STAFF

1. File Orders

OpenEFM will be expected to act as a conduit for query and response messages. The messages may include the filing of court orders. OpenEFM will be expected to receive orders from a CMS or DMS and send the order on to an EFSP. Likewise, messages may come from an EFSP and require being passed on to a CMS or DMS. No user interface will be available directly to the EFM. The orders will be required to be passed to the EFM through the query and response API.

Status: OpenEFM has not met this requirement. In previous functional integration tests that the OpenEFM has been used in, no filings were initiated by court staff via the EFM. Also, the filing of documents from one EFM to another EFM was not done. While the EFM should contain all query and response interfaces to facilitate this, it was decided that providing EFSP interfaces inside of an EFM would needlessly complicate the success of the previous projects.

2. Send and Receive Notifications

OpenEFM will be expected to act as a conduit for query and response messages. The messages may include the passing of court notifications. OpenEFM will be expected to receive notifications from a CMS or DMS and send the notifications on to an EFSP. Likewise, messages may come from an EFSP and require being passed on to a CMS or DMS. No user interface will be required for this. The sending and receiving of notifications will be available through the query and response API.

Status: OpenEFM has not met this requirement. In the past, query and response specifications have not been part of any of the functional integration tests the OpenEFM has successfully been used in. However, a framework is currently in place to support these features. This framework will rely on the query and response specification schemas currently being produced.

3. Review Pleadings, Orders, and Notices of Individual Cases

The EFM provides access to reviewed queued filings that have yet to be accepted into the CMS or DMS. These filings are accessible through a Web front end. Old filings may be stored at the EFM; however, we recommend they be removed frequently to save storage space. Also, the CMS and DMS should already provide access to all current pleadings, orders, and notices of individual cases.

Status: OpenEFM has met the requirement by providing the access to review filings prior to acceptance or rejection into a DMS or CMS. OpenEFM does not provide a user interface for review of filings already in a DMS or CMS. In prior functional integration projects, which include OpenEFM, it was decided that this functionality was already provided to court personnel through a CMS or DMS.

C. COURT CLERKS

1. File Orders and Notices Within Court

OpenEFM will be expected to act as a conduit for query and response messages. The messages may include the passing of court notifications and orders. OpenEFM will be expected to receive notifications and orders from a CMS or DMS, then send the notifications on to an EFSP. Likewise, messages may come from an EFSP and require being passed on to a CMS or DMS. No user interface will be required for this. The sending and receiving of notifications and orders will be available through the query and response API.

Status: OpenEFM has not met this requirement. In previous functional integration tests that the OpenEFM has been used in, no filings were initiated by court staff via the EFM. Also, the filing of

documents from one EFM to another EFM was not done. While the EFM should contain all query and response interfaces to facilitate this, it was decided that providing EFSP interfaces inside of an EFM would needlessly complicate the success of the previous projects.

2. Send and Receive Notifications

OpenEFM will be expected to act as a conduit for query and response messages. The messages may include the passing of court notifications. OpenEFM will be expected to receive notifications from a CMS or DMS and send the notifications on to an EFSP. Likewise, messages may come from an EFSP and require being passed on to a CMS or DMS. No user interface will be required for this. The sending and receiving of notifications will be available through the query and response API.

Status: OpenEFM has not met this requirement. In the past, query and response specifications have not been part of any of the functional integration tests the OpenEFM has successfully been used in. However, a framework is currently in place to support these features. This framework will rely on the query and response specification schemas currently being produced.

3. Review Pleadings, Orders, and Notices of Individual Cases

The EFM provides access to reviewed queued filings that have yet to be accepted into the CMS or DMS. These filings are accessible through a Web front end. Old filings may be stored at the EFM; however, we recommend they be removed frequently to save storage space. Also, the CMS and DMS should already provide access to all current pleadings, orders, and notices of individual cases.

Status: OpenEFM has met the requirement by providing the access to review filings prior to acceptance or rejection into a DMS or CMS. OpenEFM does not provide a user interface for review of filings already in a DMS or CMS. In prior functional integration projects, which include OpenEFM, it was decided that this functionality was already provided to court personnel through a CMS or DMS.

4. Keep the Court Files, Including Sealed, Confidential Records

The EFM provides access to reviewed queued filings that have yet to be accepted into the CMS or DMS. These filings are accessible through a Web front end. Old filings may be stored at the EFM; however, we recommend they be removed frequently to save storage space. Also, the CMS and DMS should already provide access to all current pleadings, orders, and notices of individual cases.

Status: OpenEFM has met the requirement by providing the access to review filings prior to acceptance or rejection into a DMS or CMS. OpenEFM does not provide a user interface for review of filings already in a DMS or CMS. In prior functional integration projects, which include

OpenEFM, it was decided that this functionality was already provided to court personnel through a CMS or DMS.

5. Provide Access to Court Files

EFM provides access to all files as they are queued in the system prior to acceptance into CMS/DMS. These filings are accessible through a Web front end. Files are not immediately removed from the EFM at the time of acceptance into a CMS or DMS; however, it is recommended to remove these filings to save space. Once these filings are in a court CMS or DMS, that system should be used to access the files.

Status: OpenEFM has met the requirement by providing the access to review filings prior to acceptance or rejection into a DMS or CMS. OpenEFM does not provide a user interface for review of filings already in a DMS or CMS. In prior functional integration projects, which include OpenEFM, it was decided that this functionality was already provided to court personnel through a CMS or DMS.

D. CLERK STAFF

1. Receive, Index, and File Pleadings, Orders, and Notices for Litigants, Attorneys, Judges, and Clerk of Court
2. Review Queued Entries Prior to Docketing
3. Review Pleadings, Orders, and Notices of Individual Cases

EFM provides access to all files as they are queued in the system prior to acceptance into CMS/DMS. These filings are accessible through a Web front end. Files are not immediately removed from the EFM at the time of acceptance into a CMS or DMS; however, it is recommended to remove these filings to save space. Once these filings are in a court CMS or DMS, that system should be used to access the files.

Status: OpenEFM has met the requirement by providing the access to review filings prior to acceptance or rejection into a DMS or CMS. OpenEFM does not provide a user interface for review of filings already in a DMS or CMS. In prior functional integration projects, which include OpenEFM, it was decided that this functionality was already provided to court personnel through a CMS or DMS.

E. SYSTEM ADMINISTRATORS (SUPERUSERS)

1. Maintain (Add, Delete, Modify) User Lists

EFM user account lists can be edited as need by a simple user interface.

Status: OpenEFM has met this requirement. A clean installation of OpenEFM has only an administrator account setup. The administrator is the only person who can add, delete, and modify other user accounts. The administrator can also create other administrator accounts. All of this is available through a simple Web interface.

2. Maintain Court Policy

The administrator has the ability to update the location from which the EFM retrieves the court policy. This allows for a court to post an updated policy and for the EFM administrator to make the EFM aware of the new policy's existence. The EFM is also required to provide hosting capabilities for the Court Policy XML file. The Court Policy XML should be made available via HTTP access by the EFM.

Status: OpenEFM has not met this requirement. In prior integration projects, court policy standards were developed but were not tested. OpenEFM currently provides a framework for accessing the court policy; however, no interface exists for editing the location of the posted policy file. Also, no database elements support storing the location of the policy file.

3. Control Query, Response, and Filing Access to the EFM

The administrator will be able to assign usernames and passwords to organizations that wish to access the query, response, and filing capabilities of the EFM. Organizations that might wish to access these features would be EFSPs or other court EFM systems. The administrator can add, edit, and remove username and password pairs in order to control access to the filing interfaces.

Status: OpenEFM has met this requirement. OpenEFM currently provides a simple Web interface to allow the addition, removal, or editing of user/passwords pairs in order to facilitate access control. All incoming requests for information or filings are subject to access validation checks.

4. Remove Previously Accepted or Rejected Filings From the System

The administrator will be able to search cases currently in the system that have already been accepted or rejected. These cases will be searchable based on how long they have been in the system. The administrator can then choose to remove individual filings from the system.

Status: OpenEFM has met this requirement. OpenEFM currently provides a simple Web interface to allow for the search and removal of aged filings.

III. TECHNICAL REQUIREMENTS

III. TECHNICAL REQUIREMENTS

The section describes compliance of the OXCI EFM with the technical requirements listed in the OXCI EFM Architecture, including relevant information and integration standards.

A. INFORMATION REQUIREMENTS

All information that is passed through the EFM will need to be stored in a common format. The advantage of this is that many different systems will be capable of communicating with the EFM. For example, multiple EFSPs may wish to offer filing capabilities to a single local court. Also, other EFM, or eventually other governmental agencies, may need to access the public information.

1. W3C XML Schema

All data passing through the system will be required to be encoded in XML format. The definitions of the different data objects will all be defined using the W3C standard of XML schemas. The schemas will be publicly available, allowing potential developers of systems that may wish to integrate with the EFM to verify and validate the structure of data objects that will be passed between systems.

Status: OpenEFM has not met this requirement. Integration projects that the OpenEFM has been utilized in before have used Document Type Definitions (DTDs) to define the XML formation of information passed through the system. Converting these features to use a schema instead should not be a difficult task.

2. LegalXML Court Filing

The LegalXML Court Filing standard defines the structure of a filing object. This includes all information necessary for a court to identify the involved parties, identify involved cases, contain the document(s) being filed, and relay any other pertinent information to the CMS and/or DMS. The Court Filing standard will be defined in a W3C XML schema.

Status: OpenEFM has not met this requirement. The court has only been defined in a very preliminary version. While the EFM has used court filing standards before, it has not yet been modified to work with this specific version.

3. Query and Response

Query and response will be a set of schemas defining a communication API. This includes queries (requests for information or actions) and responses (the requested information or outcome of the requested action), which will enable the transmission of common data between all the components of the electronic filing process. This includes the EFSP, EFM, CMS, and DMS.

Status: OpenEFM has not met this requirement. A framework exists for integrating query and response functionality into OpenEFM. However, this has not yet been a previous interoperability requirement. Also, the current query and response schema set planned to be utilized has not yet been produced.

4. Court Policy

Every court has rules and policies that vary. These rules might include such information as hours of operation, fees and fines associated with filing, or information regarding court holidays. The OXCI EFM project aims to encapsulate this information in a way that allows for the automation of electronic filing functionality, which complies with these policies. A court policy schema will be defined. This schema will contain all information deemed necessary for containing the entire court policy. A court administrator will then be able to construct an XML document, which complies with the court policy schema. This document will be made available to the OpenEFM. The EFM will then be able to make logical decisions based on the policy of the court.

Status: OpenEFM has not met this requirement. A framework exists for integrating the court policy functionality into OpenEFM. However, this has not yet been a previous interoperability requirement. Also, the current court policy schema planned to be utilized has not yet been produced.

B. INTEGRATION REQUIREMENTS

Many of the requirements of the EFM involve communication between the different components of an electronic filing system. This requires defining a standard means of communication. OXCI EFM will accomplish this through the use of SOAP, Web Services Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI).

1. SOAP

SOAP provides a basic messaging service relying on XML and HyperText Transfer Protocol (HTTP). This allows for simple XML communication to occur over a standard communication layer: the World Wide Web. A direct benefit of using SOAP is platform independence. Since

SOAP allows applications to communicate via a defined protocol, XML, over a standard communication layer, the Web, the underlying technology used to implement the queries and requests is irrelevant. This should allow for any organization wishing to integrate with the EFSP, EFM, CMS, or DMS to do so without having to use one particular development environment. SOAP is currently a W3C Technical Note. Version 1.1 of the specification has been widely implemented, and many libraries exist for almost every programming language available for development.

Status: OpenEFM has met this requirement. SOAP has been used as a filing interface in prior implementations of OpenEFM. This functionality will be further utilized when the query and response functionality is implemented. It will likely also be utilized when the CMS and DMS adapters are implemented.

2. Electronic Business eXtensible Markup Language

Electronic Business eXtensible Markup Language (ebXML) provides greater message service handling functionality. ebXML can be used in conjunction with SOAP in order to provide a more structured and reliable message service. Tasks such as authentication, encryption, and delivery verification can be handled through ebXML. Currently, Version 2.0 has been developed and released through OASIS.

Status: OpenEFM has not met this requirement. The EFM has not incorporated ebXML into the SOAP envelopes that it has utilized.

3. WSDL and UDDI

WSDL provides an XML protocol for describing Web services. WSDL includes information necessary for locating and utilizing SOAP Web services. The OXCI EFM will eventually use WSDL to describe the SOAP interface options that are available through the system. WSDL is a W3C Working Draft. Currently, Version 1.2 of the specification is available and recommended for use. The current round of development of the OXCI EFM will not include complete WSDL compliance. However, by identifying WSDL as an eventual requirement, development should not do anything that may complicate eventual WSDL compliance.

UDDI facilitates publishing of WSDL definitions and querying a central registry for available Web services. This can greatly aid in the development of systems to be integrated with the EFSP, EFM, CMS, or DMS. UDDI is being developed as a specification through OASIS. UDDI is planned as an eventual requirement for the OXCI EFM; however, it will not be an included requirement for this round of development. By identifying UDDI as an eventual requirement, development should not do anything that may complicate eventual UDDI compliance.

Status: OpenEFM has met this requirement. Currently there is nothing in the code base that will prohibit the eventual adoption of these features.

IV. BUSINESS AND POLICY REQUIREMENTS

IV. BUSINESS AND POLICY REQUIREMENTS

The section describes compliance of the OXCI EFM with business and policy requirements listed in the OXCI EFM Architecture.

A. POLICY REQUIREMENTS

The OXCI EFM has the stated goal of providing an easy-to-use, open source solution for courts across the country. In order to achieve this goal many requirements have been set regarding the policy used in making development decisions. They are listed below.

1. Use Freely Licensed and Open Source Technologies

Many courts are already strapped for cash. Adding a new, complex computer system to the budget of a courthouse is often a daunting task for court administrators. OXCI aims to provide a low-cost solution. No software is free, and there will always be costs associated with maintenance and support for any computer system put into place. However, OXCI is committed to using only freely available software for the design of the basic EFM system. This will allow court personnel to start working on their electronic filing solution without having to worry about complicated licensing or software acquisition issues.

Maintaining the OXCI EFM as an open source software project also allows for greater information sharing between courts. Open source software allows anyone to view and modify the source code. As courts customize the EFM, making such changes as feature enhancements or bug fixes, these changes can be incorporated back into the main project for all to share. This can allow for more robust software in a faster amount of time.

Status: OpenEFM has met this requirement.

2. Simplify the Architecture to Minimize the Cost and Complexity of Implementation

Recognizing the cost of implementing a new electronic filing system, OXCI requires that the process be as simple and time-saving as possible. Many small courts across the nation have a very limited information technology department. The OXCI EFM will strive to provide cost-effective, simple-to-use technologies. An example of this can be seen in the generic CMS and DMS software adapters that will be distributed with the EFM. These adapters will provide a skeleton-type framework with which to start the integration process.

Status: OpenEFM has met this requirement.

3. Scale to Support Both Large and Small Courts

The OXCI EFM should be a viable solution for all courts. This requires meeting the needs of both small and large courts. The EFM is designed to allow for scaling with larger courts. It is also designed in a modular manner. For example, if a larger court wishes to use an industrial, commercial database, the database module of the EFM is the only aspect of the system that needs to be modified in order to accommodate.

Status: OpenEFM has met this requirement.

4. Support All the Electronic Filings for a Single Court

The OXCI EFM is intended to be a complete electronic filing solution. Managing multiple EFMs would only make court personnel work harder to keep everyone happy. A single EFM should be used to process all of a court's filings. The OXCI EFM is capable of accepting any document type as a filing and can integrate with CMS and DMS, allowing all types of filings to flow into the court.

Status: OpenEFM has met this requirement.

5. Support Filing Fees and Payments Managed Through an External Payment System

Credit card payment and electronic fund transfer services are currently available on line by a number of different organizations. These groups provide libraries for programmers wishing to incorporate these payment methods into software. The OXCI EFM should be able to interface with these systems and process payments in a manner allowing the court to keep track of who has paid fees and fines associated with filings. The billing information itself should not reside in the Court Filing envelope. Instead the billing information (credit card number, etc.) should be contained in an ebXML element contained in the SOAP layer.

Status: OpenEFM has not met this requirement. In the past OpenEFM has utilized commercial, third-party electronic payment systems. However, the billing information which was used (i.e., credit card number) was passed to OpenEFM in the LegalEnvelope element of LegalXML. This will need to change to accommodate the ebXML elements of the SOAP layer.

6. Support Two-Way Operation

The OXCI EFM will be able to both send and receive messages from both EFSPs and CMS and DMS. This is necessary to allow a proper query and response information flow. The EFM should

act as a public information clearinghouse in regard to queries made by EFSP. Since the information requested by EFSPs will require two-way communication with the CMS and DMS, all channels of communication will need to be extensible.

Status: OpenEFM has not met this requirement. This requirement will be met once all functional requirements regarding query and response have been satisfied.

7. Support Anti-Virus Checking

The OXCI EFM will support the programmatic checking of any documents passing through the system for viruses. This includes documents being filed into the CMS or DMS system. Documents passing through the system for query and response will also be able to be checked. The implementation details of this functionality will be defined in the Design Document.

Status: OpenEFM has not met this requirement.

B. BUSINESS MODEL REQUIREMENTS

The OXCI EFM will be released as open source software. By definition this software will be available to any interested party. Due to this, private companies, as well as public courts, will have access to this system. This will allow for any of the four mentioned business models to be used in setting up an electronic filing process with OXCI EFM.

- *Court Control Model* – The court manages both the EFSP and the EFM.
- *Vendor Control Model* – A third-party vendor provides both EFSP and EFM.
- *Split Control Model* – A vendor provides the EFSP, and the court controls the EFM.
- *Single Source Control Model* – Courts and legal firms each provide their own EFSP and EFM.

It is the recommendation of Counterclaim that the split control model be utilized; however, the OXCI EFM could be used in any model.

Status: OpenEFM has met this requirement.

APPENDIX A
GLOSSARY

GLOSSARY

API	Application Program Interface
CMS	Case Management System
DTD	Document Type Definition
DMS	Document Management System
ebXML	Electronic Business eXtensible Markup Language
EFM	Electronic Filing Manager
EFSP	Electronic Filing Service Provider
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
OASIS	Organization for the Advancement of Structured Information Standards
OXCI	Open XML Court Interface
SOAP	Simple Object Access Protocol
TC	Technical Committee
UDDI	Universal Description, Discovery and Integration
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	eXtensible Markup Language

APPENDIX B
REVISION HISTORY

REVISION HISTORY

Version	Date	Revised By	Description
0.1	1/14/04	Mr. Jim Beard counterclaim.com, inc.	The first version of this document contained most of Section II only. It was the first attempt to flesh out requirements based on the proposed OXCI EFM Architecture document. An outline for this document was created, and the identified requirements were defined.
0.2	1/15/04	Mr. Beard counterclaim.com, inc.	The second version of this document introduced Sections I and III, along with APPENDICES A and B.
0.3	1/16/04	Mr. Beard counterclaim.com, inc.	This version corrected various grammatical issues and reworded potentially confusing sentences.
0.4	1/18/04	Mr. Beard counterclaim.com, inc.	This version included OpenEFM status elements for each requirement, added ebXML and administrative removal of aged documents as functional requirements, and worked on layout issues.
0.5	1/21/04	Mr. James E. Cabral Jr. MTG Management Consultants, L.L.C.	This was a reformatted and reorganized version.
0.6	1/26/04	Mr. Beard counterclaim.com, inc.	Updated a number of requirements based on design meeting held on January 23, 2004.
0.7	2/02/04	Mr. Beard counterclaim.com, inc.	Updated a number of requirements based on design meeting held on January 30, 2004.
0.8	5/17/04	Mr. Beard counterclaim.com, inc.	Added a requirement for anti-virus checking.