FileNet Webservice

Version Number# 1.00
Creation Date: May 28, 2009

1 REQUIREMENTS 3

2 ASSUMPTIONS 3

3 PREREQUISITES 3

3.1 FILENET 3

4 USE CASES 3

4.1 RETRIEVE DOCUMENT 3

5 NETWORK ARCHITECTURE 4

5.1 TOPOLOGY 4

5.2 ISRA OVERVIEW 4

6 SOLUTION DESIGN 5

6.1 DESIGN CONSIDERATIONS 5

7 TECHNICAL DESIGN 6

7.1 WEBSERVICE IMPLEMENTATION 6

8 APPENDIX 7

8.1 APPENDIX A 7

1 Requirements

1) As a customer, I am able to view my current billing statement online.
2) As a customer, I am able to view my prior billing statements online.

2 Assumptions

- The statements will be presented to the customer via a PDF document.
- This module is not responsible for the authentication and/or authorization of the customer to view the data in the document.
- The proposed solution will be used until the P8 upgrade system is in place. At which time a new solution will be implemented that satisfies the same requirements, but uses P8.

3 Prerequisites

3.1 FileNet

The billing documents are stored in two areas. For P&C the documents are stored in a FileNet version 4.0 system. For Life the documents are stored in a FileNet 3.6 ESM system. Access to these documents via a programmatic API is a necessary prerequisite.

4 Use Cases

4.1 Retrieve Document

1) Customer clicks a hyperlink (or submits a form) to request the billing statement.

2) The request is processed.

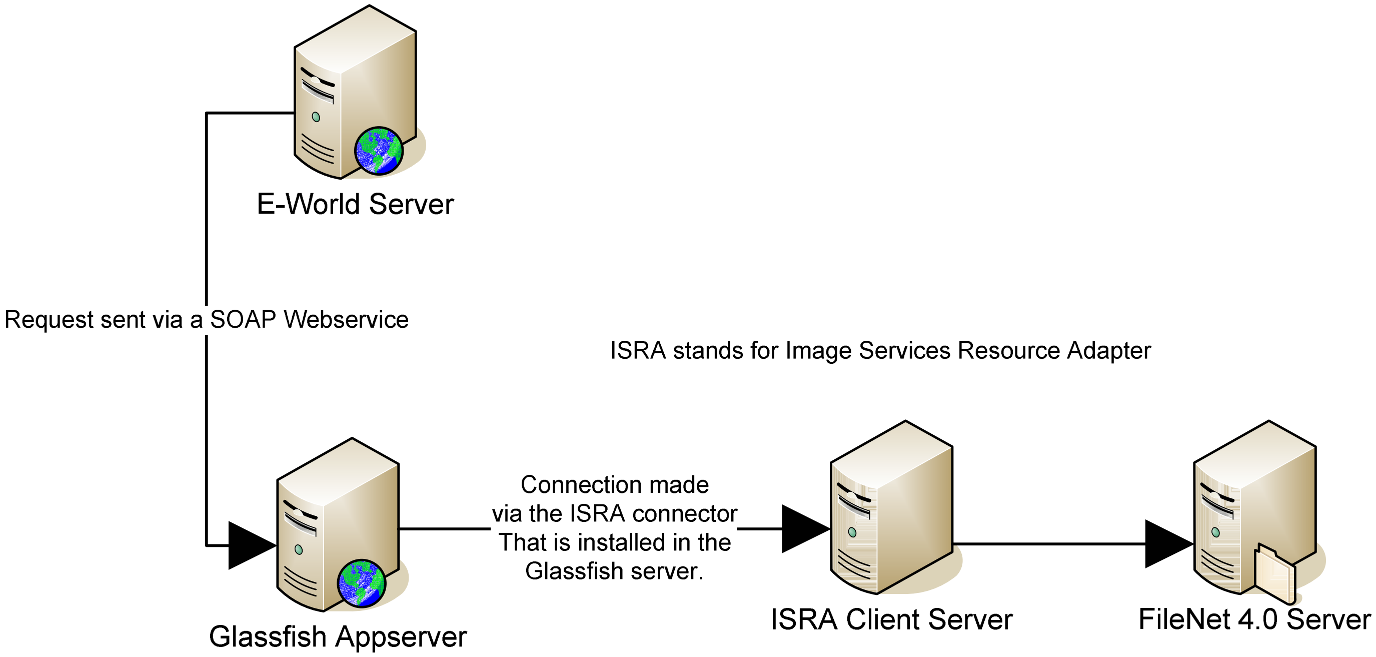
3) A PDF containing the customer's billing statement is returned to the customer.

Error Condition

If an error occurs during processing an appropriate message will be displayed to the customer.

5 Network Architecture

5.1 Topology


E-World Server
Request sent via a SOAP Webservice
ISRA stands for Image Services Resource Adapter
Connection made via the ISRA connector That is installed in the Glassfish server.
Glassfish Appserver
ISRA Client Server
FileNet 4.0 Server

The customer will be making the original request through E-world. This request will then contact a Glassfish application server via HTTP using SOAP. Glassfish will invoke a service to obtain the document via the ISRA connector. ISRA then contacts the FileNet Server to retrieve the binary document.

5.2 ISRA Overview

The following is from the ISRA Deployment Guide, page 5

"*The J2EE Connector Architecture (JCA) provides a Java solution to the problem of connectivity between Application Servers and Enterprise Information Systems (EIS). JCA defines a standard architecture to connect the Java 2 Enterprise Edition (J2EE) platform to heterogeneous EIS's.*

ISRA is a JCA-compliant, system-level software driver, which is used by a Java application component or client to connect to FileNet Image Services (IS).

ISRA provides an alternative to Integrated Document Management (IDM) Web Services for IS customers. In addition, it provides a Web solution that does not require Microsoft technology or products."

The following is from the ISRA Deployment Guide, page 15
"*Software Requirements [for ISRA]: FileNet IS 3.6 SP2 and above*"

6 Solution Design

Here we show a high level diagram for the solution.

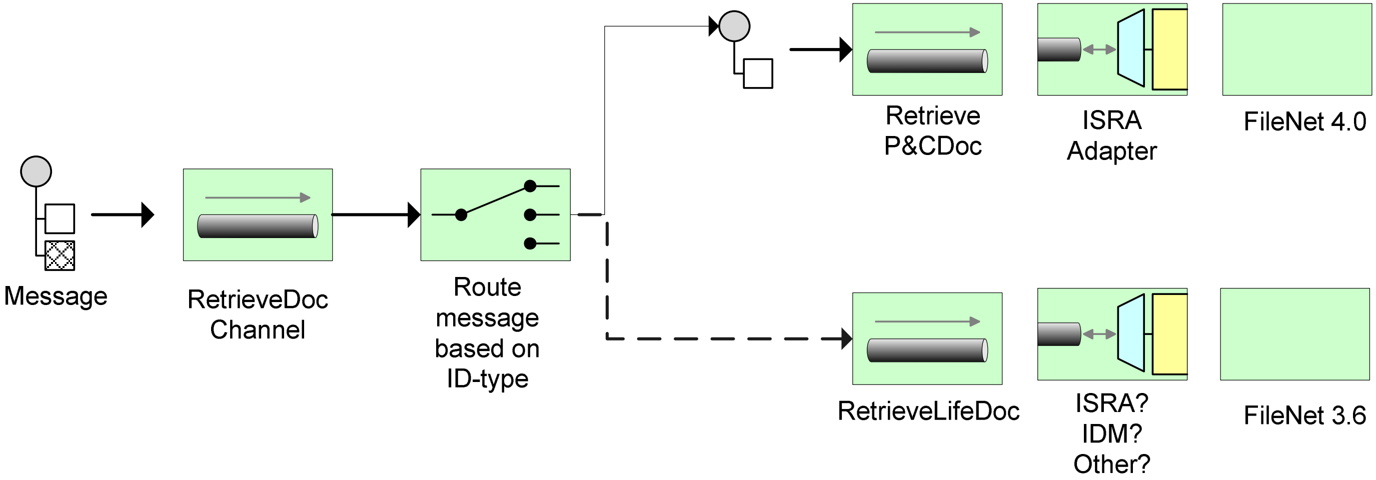


Figure 1

A message is sent to the RetrieveDoc Channel. In this message will be the Policy Identifier, the Policy Type, and a Time Period. Depending on the Policy Type, the message will either be routed to the P&C channel, or the Life Channel. From here it will have to translate the Policy Identifier into a document identifier (as shown in figure 2). Once the document ID is obtained, the message is sent through an adapter into the appropriate system.

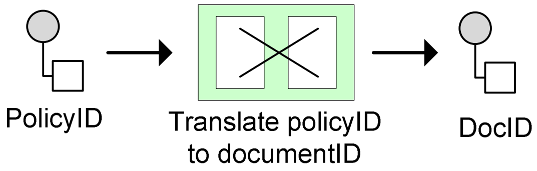


Figure 2

6.1 Design Considerations

The system who is making the initial request into the system should not have to know about how the document is stored, where it is stored, or how it gets retrieved. To the caller, they are simply requesting a document for a given policyID and type. Changes to the backend system should provide minimal impact to the calling system. That is, if the how and where of the document changes, the calling system should not know this. Instead, it should be decoupled from these types of changes.

7 Technical Design

7.1 Webservice Implementation

In order to provide a loose decoupling of the system, the message will be sent over HTTP. This message will be in SOAP format, and the response message will also be a SOAP formatted message.

Because the document that is being returned is a binary PDF document, Message Transmission Optimization Mechanism (MTOM) will be utilized to optimize the transmission of the binary document.

JAX-WS will be used to implement the webservice. Specifically, the Metro package within Glassfish will be utilized.

8 Appendix

8.1 Appendix A

WSDL for the webservice will be provided here.

```
<!--
 Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-751-SNAPSHOT.
-->

<definitions targetNamespace="http://www.fbfs.com/RetrieveDocument" name="RetrieveDocument">

<ns1:Policy wsu:Id="RetrieveDocumentPortBinding_MTOM_Policy">

<ns1:ExactlyOne>

<ns1:All>
<ns2:OptimizedMimeSerialization ns1:Optional="true"/>
</ns1:All>
</ns1:ExactlyOne>
</ns1:Policy>

<types>

<xsd:schema>
<xsd:import namespace="http://www.fbfs.com/RetrieveDocument"
schemaLocation="https://filenetws:443/FileNetWebService/RetrieveDocument?xsd=1"/>
</xsd:schema>
</types>

<message name="search">
<part name="parameters" element="tns:search"/>
</message>

<message name="searchResponse">
<part name="parameters" element="tns:searchResponse"/>
</message>

<message name="InvalidDataException">
<part name="fault" element="tns:InvalidDataFault"/>
</message>

<message name="SystemUnavailableException">
<part name="fault" element="tns:SystemUnavailableFault"/>
</message>

<message name="retrieveDocument">
<part name="parameters" element="tns:retrieveDocument"/>
</message>

<message name="retrieveDocumentResponse">
```

```
<part name="parameters" element="tns:retrieveDocumentResponse"/>
</message>

<portType name="RetrieveDocumentPortTypeImpl">

<operation name="search">
<input message="tns:search"/>
<output message="tns:searchResponse"/>
<fault message="tns:InvalidDataException" name="InvalidDataException"/>
<fault message="tns:SystemUnavailableException" name="SystemUnavailableException"/>
</operation>

<operation name="retrieveDocument">
<input message="tns:retrieveDocument"/>
<output message="tns:retrieveDocumentResponse"/>
<fault message="tns:InvalidDataException" name="InvalidDataException"/>
<fault message="tns:SystemUnavailableException" name="SystemUnavailableException"/>
</operation>
</portType>

<binding name="RetrieveDocumentPortBinding" type="tns:RetrieveDocumentPortTypeImpl">
<ns3:PolicyReference URI="#RetrieveDocumentPortBinding_MTOM_Policy"/>
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>

<operation name="search">
<soap:operation soapAction=""/>

<input>
<soap:body use="literal"/>
</input>

<output>
<soap:body use="literal"/>
</output>

<fault name="InvalidDataException">
<soap:fault name="InvalidDataException" use="literal"/>
</fault>

<fault name="SystemUnavailableException">
<soap:fault name="SystemUnavailableException" use="literal"/>
</fault>
</operation>

<operation name="retrieveDocument">
<soap:operation soapAction=""/>

<input>
<soap:body use="literal"/>
</input>

<output>
<soap:body use="literal"/>
</output>
```

```
<fault name="InvalidDataException">
<soap:fault name="InvalidDataException" use="literal"/>
</fault>

<fault name="SystemUnavailableException">
<soap:fault name="SystemUnavailableException" use="literal"/>
</fault>
</operation>
</binding>

<service name="RetrieveDocument">

<port name="RetrieveDocumentPort" binding="tns:RetrieveDocumentPortBinding">
<soap:address location="https://filenetws:443/FileNetWebService/RetrieveDocument"/>
</port>
</service>
</definitions>

================================================================================
================================= Schema =======================================
================================================================================
<!--
 Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3.1-hudson-751-SNAPSHOT.
-->

<xs:schema version="1.0" targetNamespace="http://www.fbfs.com/RetrieveDocument">
<xs:element name="InvalidDataFault" type="tns:InvalidDataException"/>
<xs:element name="SystemUnavailableFault" type="tns:SystemUnavailableException"/>
<xs:element name="retrieveDocument" type="tns:retrieveDocument"/>
<xs:element name="retrieveDocumentResponse" type="tns:retrieveDocumentResponse"/>
<xs:element name="search" type="tns:search"/>
<xs:element name="searchResponse" type="tns:searchResponse"/>

<xs:complexType name="retrieveDocument">

<xs:sequence>
<xs:element name="documentID" type="xs:long"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="retrieveDocumentResponse">

<xs:sequence>
<xs:element name="return" type="xs:base64Binary" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="InvalidDataException">

<xs:sequence>
<xs:element name="message" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="SystemUnavailableException">

<xs:sequence>
<xs:element name="message" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="search">

<xs:sequence>
<xs:element name="policyNumber" type="xs:long"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="searchResponse">

<xs:sequence>
<xs:element name="documents" type="tns:documents" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="documents">

<xs:sequence>
<xs:element name="docInfo" type="tns:documentInfo" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="documentInfo">

<xs:sequence>
<xs:element name="closed" type="xs:boolean"/>
<xs:element name="docID" type="xs:long" minOccurs="0"/>
<xs:element name="documentType" type="xs:string" minOccurs="0"/>
<xs:element name="entryDate" type="xs:dateTime" minOccurs="0"/>
<xs:element name="formNumber" type="tns:allowedForms" minOccurs="0"/>
<xs:element name="policyID" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:simpleType name="allowedForms">

<xs:restriction base="xs:string">
<xs:enumeration value="NB001"/>
<xs:enumeration value="NB014"/>
<xs:enumeration value="NB031"/>
<xs:enumeration value="NB010"/>
<xs:enumeration value="NB020"/>
<xs:enumeration value="NB021"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```