Software Architecture Document

File Repository Cell 1.6

Partners/i2b2.org

Abstract:

This is a software architecture document for File Repository (FRC) cell. It identifies and explains important architectural elements. This document will serve the needs of stake holders to understand system concepts and give a brief summary of the use of the FR message format.

Revision History

Revision Number	Date	Author	Description
1.0	4/29/08	Mike Mendis	Initial Document
1.1	11/18/08	Mike Mendis	
1.2	1/25/2010	Mike Mendis	Upgraded to 1.4
1.3	7/12/2010	Mike Mendis	Upgraded to 1.5
1.4	7/22/2010	Janice Donahoe	Upgraded to 1.6

Table of CFRents

1. Overview

The File Repository Cell (FRC) is an i2b2 Hive Core cell. This cell manages i2b2 files between cells and the within i2b2 applications such as the i2b2workbench.   Under most conditions users will use the REST/SOAP service calls to access files on the FRC. In the rare conditions where users will need to have more direct access, a administer of the server will need to create a account on the server with the same username and password as the one in the Project Management Cell. At this point the user can use with the File Viewer in the workbench to access the server either with FTP or SFTP or a third-party client.

1.1. FR Definitions, Acronyms and Abbreviations

1.1.1 Patient Data Object (PDO)

This Object mirrors the star schema database model of the data mart. It holds patient information such as clinical observations, demographics and provider data.

1.1.1 Scheme

Each distinct vocabulary and their associated codes is called a scheme. A distinction is made between codes from different sources by pre-pending a unique prefix to each code.

1.2 Roles

The primary roles/participants in the FR system are as follows:
- Admin/Manager – Upload and download of files provided they have an account on the Project Management server.

1.3 Security

Users may access FR with a user-id and password combination, which is authorized through the Project Management Cell. The implementation detail of Project Management Cell is considered out-of scope to this document.

1.4 Scope of the system

Some other participants, currently outside the scope of FR are:

- Project Management Cell
- CRC Cell

1.5 Assumptions/Constraints

- Large files over 1 GB should try to use the direct connection method

1.6 Technical Platform

The technology used to build the product is as follows

- Java 2 Standard Edition 6.0
- Oracle Server 10g database
- Xerces2 XML parser
- JBoss Application server version 4.2.2
- Axis2 v1.1 web service (SOAP/REST messaging)

1.6.1 Transaction

The FR system is transactional, leveraging the transaction management model of the J2EE platform.

1.6.2   Security

The application must implement basic security behaviors:

- Authentication: Authenticate using at least a user name and a password
- Authorization: User may only access categories that they are allowed to by role
- Confidentiality: Sensitive data must be encrypted
- Data integrity : Data sent across the network cannot be modified by a tier
- Auditing: In the later releases we may implement logging of sensitive actions

1.6.3   Persistence

This application utilizes JDBC calls to retrieve persisted data.

1.6.4   Reliability/Availability

The Reliability/Availability will be addressed through the J2EE platform

Targeted availability is 16/7: 16 hours a day, 7 days a week

The time left (8 hours) is reserved for any maintenance activities

1.6.5   Performance

The user authentication with project management cell must be under 1 second.

2. Use Case

The diagram below depicts common use cases a user may perform with the FR cell.

2.1 Use Case 1

Client:

Assumes:

Patient ID in database
Encounter ID in database

Action:

Batch Tab
Step 1
1. Allow end user to get sample csv file

Step 2
1. Allow end user to upload filled in csv file
2. Convert csv to PDO containing either Patients, Observations, MRN Mapping, Visit Mapping or Visits
3. Validate csv data contains numbers in number fields, dates in date fields, the not null columns have data and files exist on local client, this is done with a wizard
4. If the CSV file has a value in the FileName column, than set the valType to a 'F', Convert tval in ObservationType to a (/[Project ID]/[Core Cell]/[Destination]) CWhere Project ID is the ID of the current Project such as Demo or Asthma, Core Cell is the ID of the cell which the file is related to, such as CRC, Destination is the remote directory and filename to place the local file which is gotten from the CSV file.

Step 3
1. Send a web service call to Data Repository Uploader, with PDO XML object as a SOAP attachment

```
<message_body>
     <request_header>
     <request_type>Data_Load</request_type>
     </request_header>
     <request>
          <input_option>
               <data_file  location_uri="irods://irods_url" type="PDO"
                         source_system_cd="MGHDIAGS"
                         upload_label="MGHDIAGS"/>
          </input_option>
          <load_option commit_flag="true" delete_staging="true">
               <load_observation_set append_flag="false"/>
               <load_event_set   />
               <load_patient_set />
               <load_observer_set />
               <load_concept_set />
```

```
            </load_option>
            <output_option summary="true">
                    <event_set onlykeys="" status="" blob="" techdata=""/>
                    <observation_set />
                    <patient_set  />
                    <observer_set />
                    <concept_set />
            </output_option>
        <request>
</message_body>
```

With Location_uri, Soure_system_cd, Upload_label unique, the rest of the message_body will be the same as above.

The response of an error will be reported to the user, a successful start will not be reported. The user can see status on the 'status' tab.

The server will take the file and sent it to the File Repository via SFTP. The destination will be /[Project ID]/CRC/[Destination Dir and File])

It will than buffer read the file and insert the records into the CRC database.
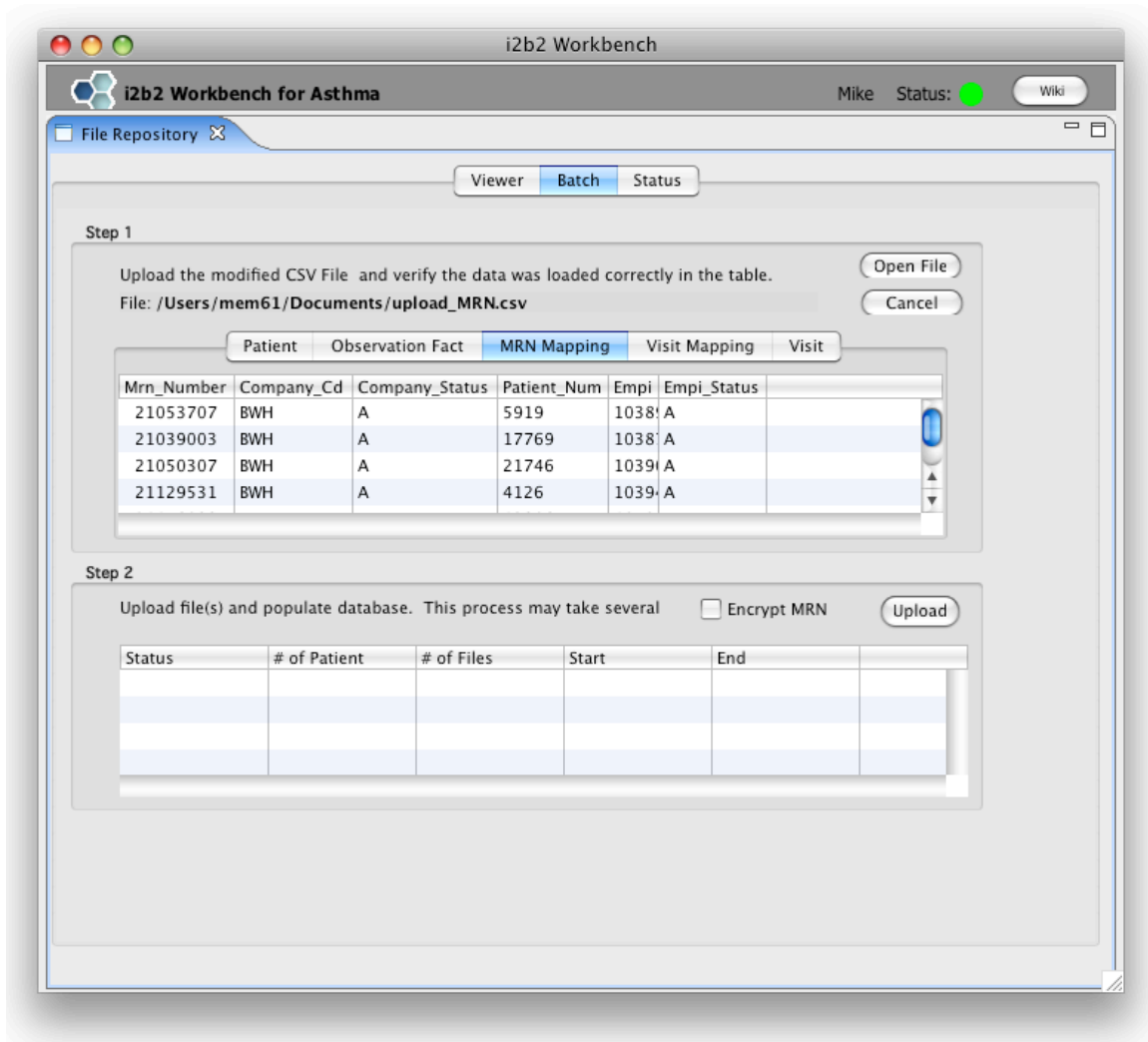
Status Tab

1. Display status of all current and past loads for the specific user
2. Send Web Service Call to Data Repository Cell to get loads

```
<message_body>
     <request_header>
     <request_type>Get_Load_Status</request_type>
     </request_header>
     <request>
         <user_id />
         <load_id />
     </request>
```

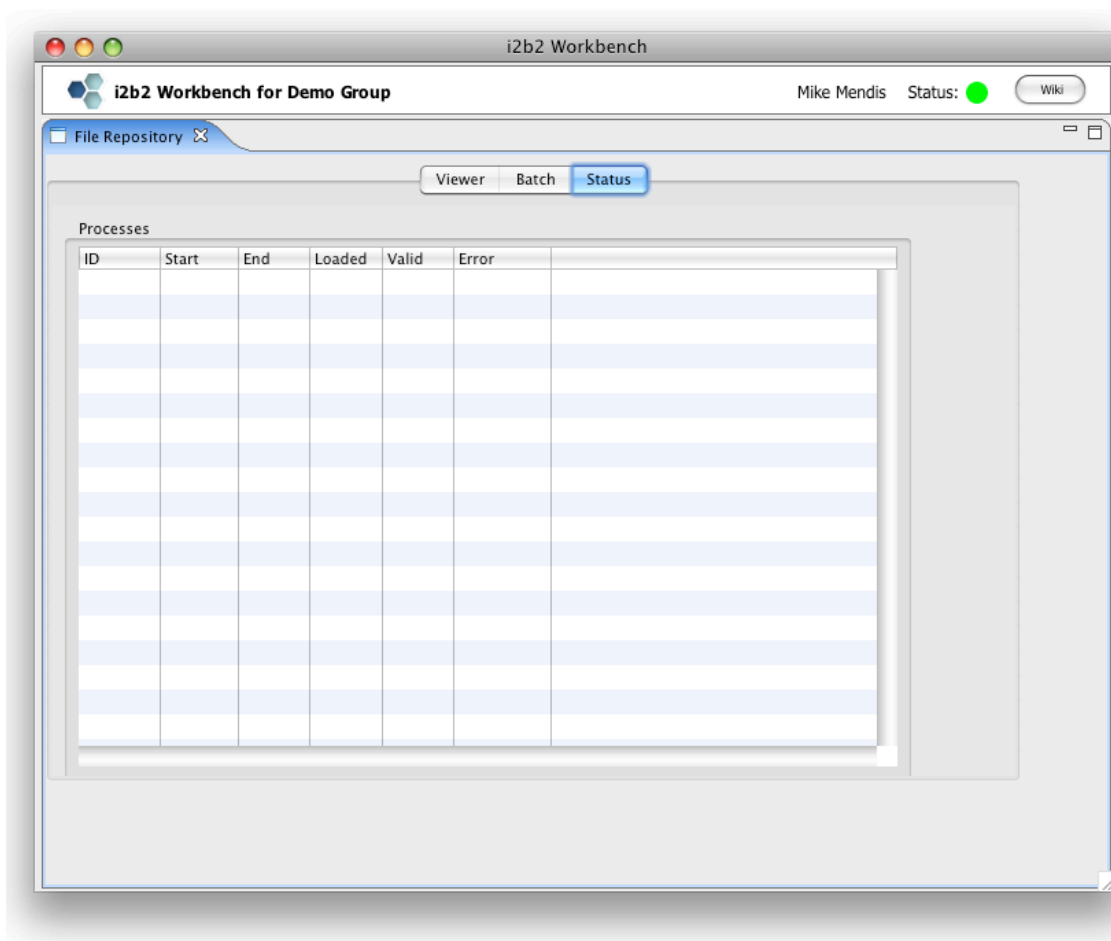The Userr_id will be the current user and the response will be:
```
     <response>
```

```
                <status load_id="100">
                        <upload_label/>
                        <user_id/>
                        <source_cd/>
                        <loaded_record/>
                        <load_date/>
                        <load_status/>
                        <message/>
                        <input_file_name/>
                        <log_file_name/>
                        <transform_name/>
                        <deleted_record/>
                </status>
        </response>
```

The client will parse the response and populate the table

Server:

Create 4 web services:, the first 2 are part of the CRC Loader, while the last 2 are part of the File Repository.

1. Display status of current and past loads for a specific user
2. Load a set of observations from a PDO file on the File Repository into the Data Repository. The URI send will contain the location of PDO file, the format of the SRB URI is (srb:// [userName . domainHome @] host [: port][/ path) The file will be downloaded by using the URI, storage broker, and the password/token from the message header. The storage resource can be retrieved from the PM cell, under the parameters name of 'SRBDefaultStorageResource' All the pdo objects will be loaded into the database directly without encrypting or mapping the patient or encounter id. Use case 2 will deal with preprocessing the database.
3. Create a Web Service which will be able to Download a File
4. Create a Web Service which will be able to Upload A File

2.4 Use Case 2

Allow end users direct access to the FRC. A administer of the server will need to create a account on the server with the same username and password as the one in the Project Management Cell. At this point the user can use with the File Viewer in the workbench to access the server either with FTP or SFTP or a third-party client. Users will be able to remove files/directories, add files/directories, rename, and also upload/download files, using a friendly UI interface built into the workbench.
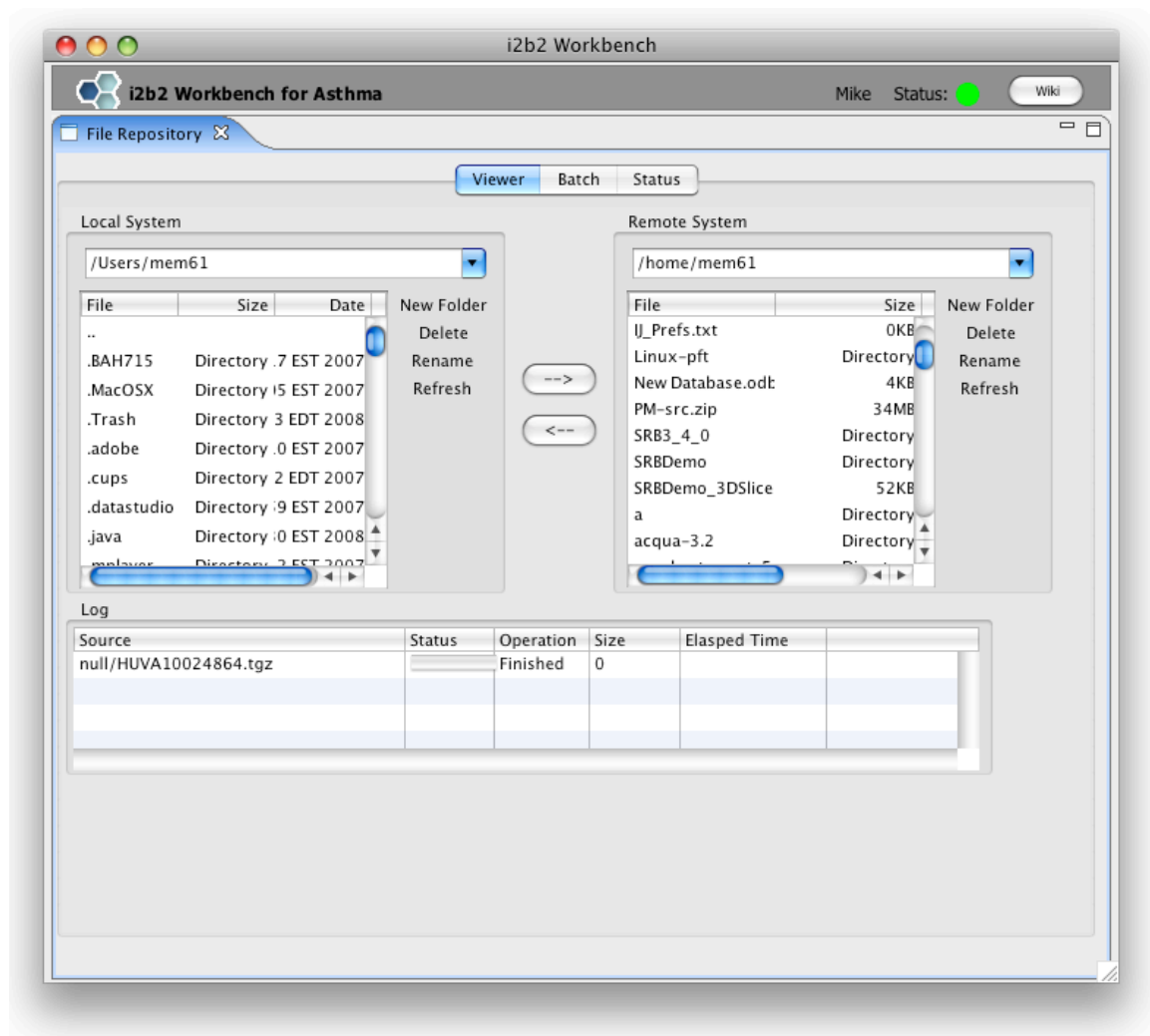
2.5 Operations

The FR service is designed as a collection of operations, or use cases:

upload_file: insert a new record into the FR cell.

download_file: download a new record from the FR cell.

3. Architecture Description

This section provides a description of the architecture as multiple views. Each view conveys the different attributes of the architecture.

> 1) Components and Connector View
>> a) Client-Server Style
> 2) Module View
>> a) Decomposition Style
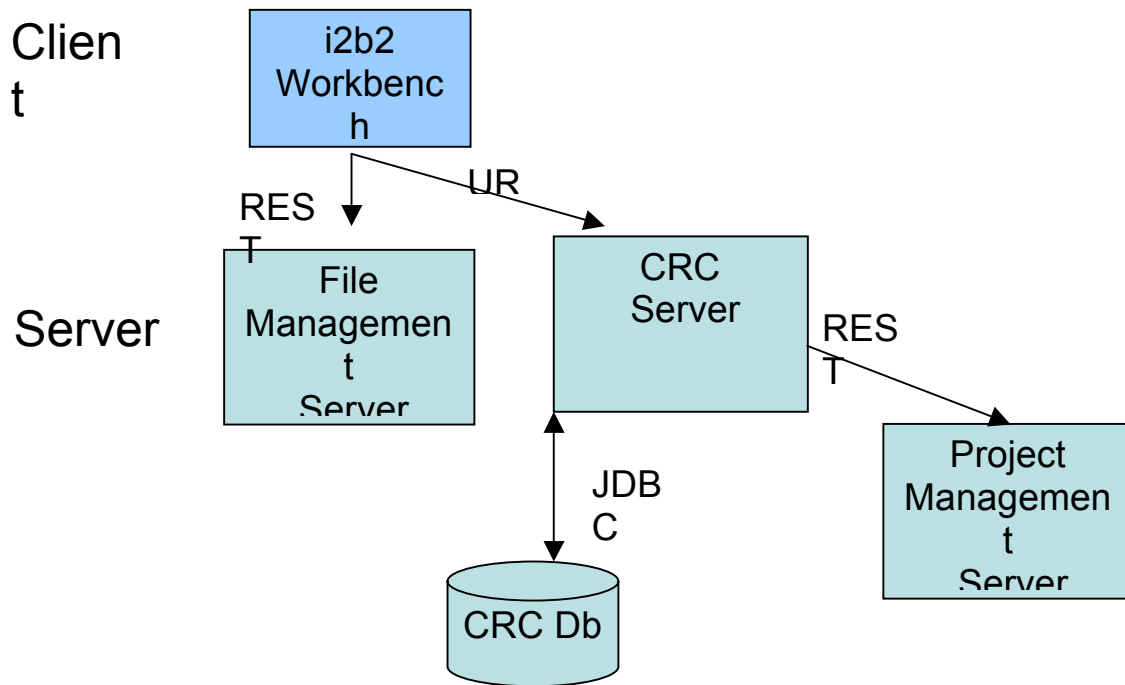>> b) Uses Style
> 3) Data View
> 4) Deployment View

3.1 Components and Connector View

A Components and Connector view represents the runtime instances and the protocols of connection between the instances. The connectors represent the properties such as concurrency, protocols and information flows. Following diagram represents the Components and Connector view for the multi-user installation. As seen below, component instances are shown in more detail with specific connectors drawn in different notations.

3.1.1. Client-Server Style

The FR system is represented using the C&C Client-Server view.

3.1.1.1 Primary Presentation

Client

i2b2 Workbench

REST

UR

File Management Server

CRC Server

REST

Server

JDBC

Project Management Server

CRC Db

3.1.1.2 Element Catalog

Element Name	Type	Description
i2b2 Workbench	Client Component	Webservice client submits the requests to FR Server components and renders response XML.
File Repository Server	Server Component	Provides socket Interface for the FR system. It supports the REST protocol. It uses Project Management server to handle user authentication.
CRC Server	Server	Provides Web Service Interface for the CRC system. It supports both SOAP and REST protocols. It uses Project Management server to handle user authentication. It uses Ontology server to lookup the concepts metadata. It stores Setfinder query definition, query run instance and the corresponding query results. The user can then request Patient Data Object using the Setfinder results.
Project Management Server	Server Component	FR cell uses Project Management cell to authenticate user. FR cell constructs PM request message and makes a web service call to Project Management Cell.
CRC Datamart DB	Data repository	This repository is mainly a data mart for patient's clinical observation information represented in star schema. This database also holds CRC user's queries (Setfinder query) information and its results like

		patient sets, visit sets, etc.
JDBC	Query Connector	SQL query used as a connector between the FR System and the Metadata database.
Web Service	Request Connector	REST protocol used to communicate with the external system.

3.1.1.2 Design Rationale, Constraints
N-tier Architecture

      The client-server style depicts an n-tier architecture that separates the presentation layer from business logic and data access layer.

3.2 Module View type

The module view shows how the system is decomposed into implementation units and how the functionality is allocated to these units. The layers show how modules are encapsulated and structured. The layers represent the "allowed-to-use" relation.

The following sections describe the module view using Decomposition and Uses Styles.

3.2.1 Decomposition Style

The "Decomposition" style presents system functionality in terms of manageable work pieces. It identifies modules and breaks them down into sub-modules and so on, until a desired level of granularity is achieved.
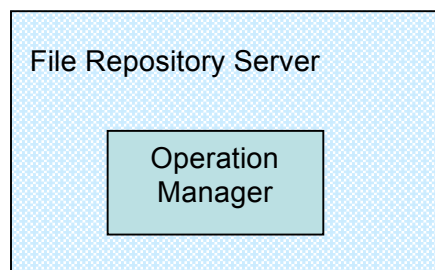
3.2.1.1 Primary Presentation

System	Segment
File Repository Server	Operation Manager

3.2.1.2 Element Catalog

Element Name	Type	Description
Operation Manager	Subsystem	This subsystem manages queries for FR operations.

3.2.1.3 Conext Diagram

File Repository Server

Operation Manager

3.2.2 Uses Style

The "Uses" style shows the relationships between modules and sub-modules. This view is very helpful for implementing, integrating and testing the system.
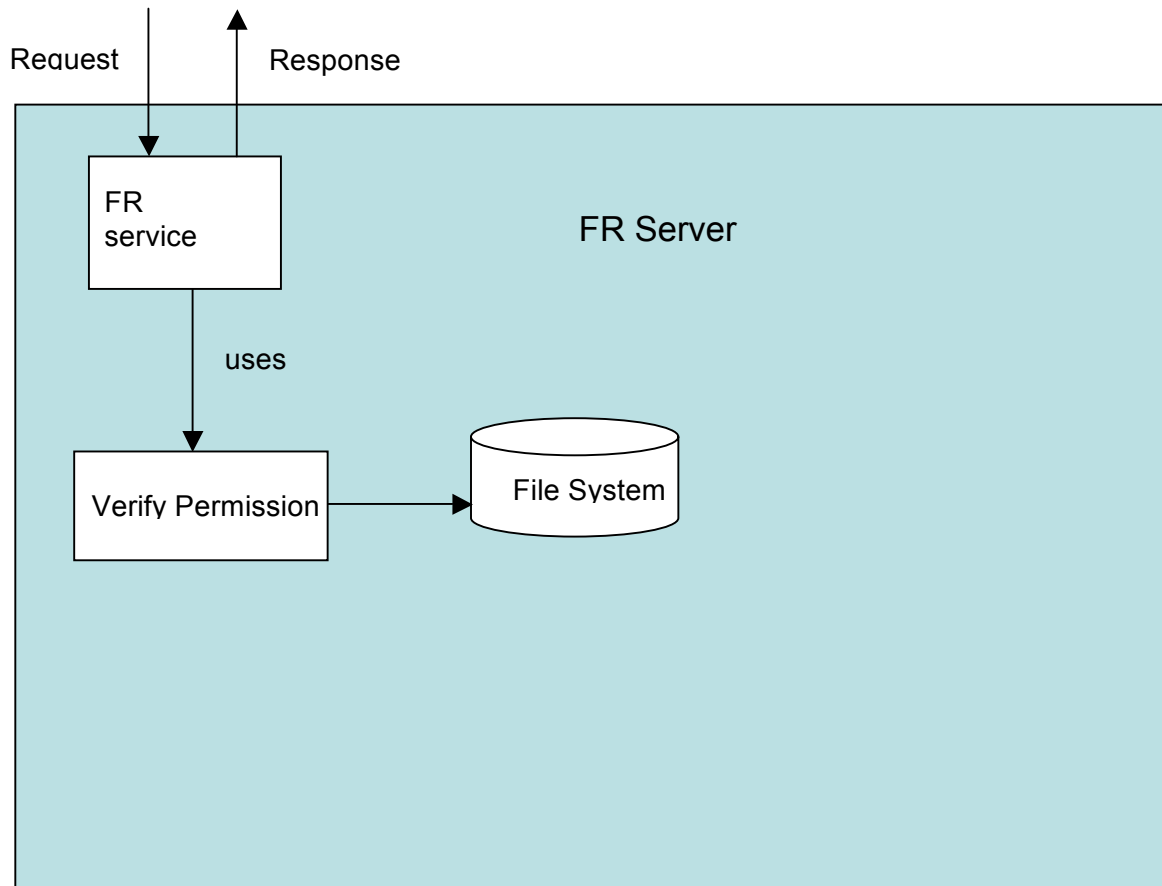
3.2.2.1 Primary Presentation

System	Segment
File Repository Server	FR Module
Operation Manager Subsystem	FR Fileservice

3.2.2.2 Element Catalog

Element Name	Type	Description
FR Module	Module	Authenticates user through PM Server System
FR Fileservice	Communication Module	Providesfileservice interface to FR operations.

3.2.2.3 Conext Diagram

3.2.2.4 Sequence Diagram
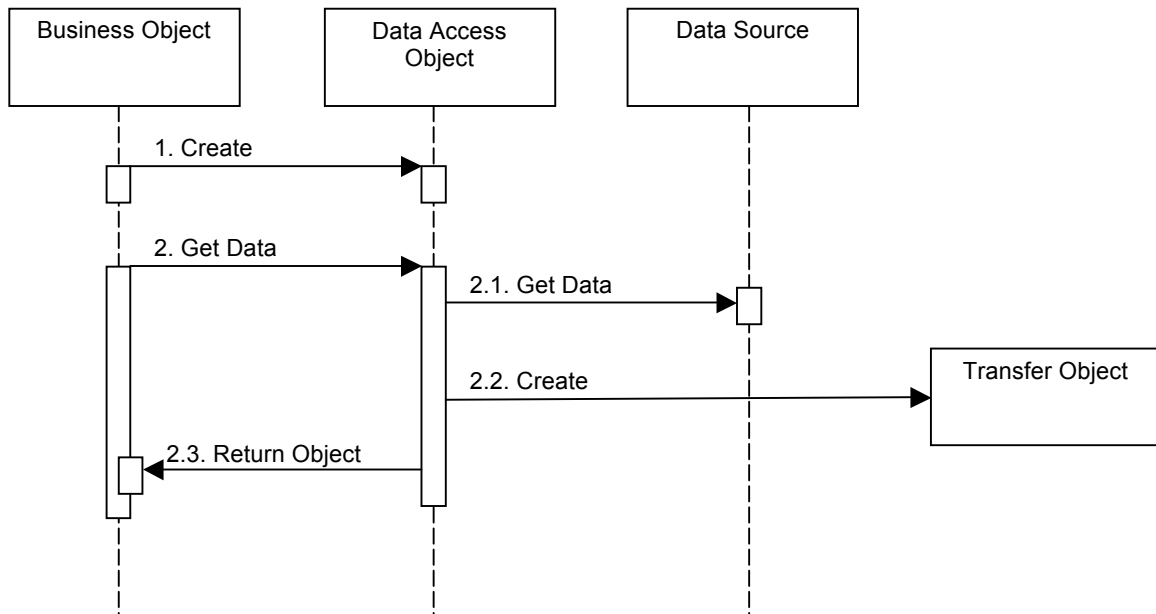
3.3 Mappings of Styles

The following table is a mapping between the elements in the Component & Connector Client-Server view shown in section 3.1.1, and the Modules Decomposition and Uses views shown in sections 3.2.1 and 3.2.2.

The relationship shown is *is-implemented-by*, i.e. the elements from the C&C view shown at the top of the table are implemented by any selected elements from the Modules views, denoted by an "X" in the corresponding cell.

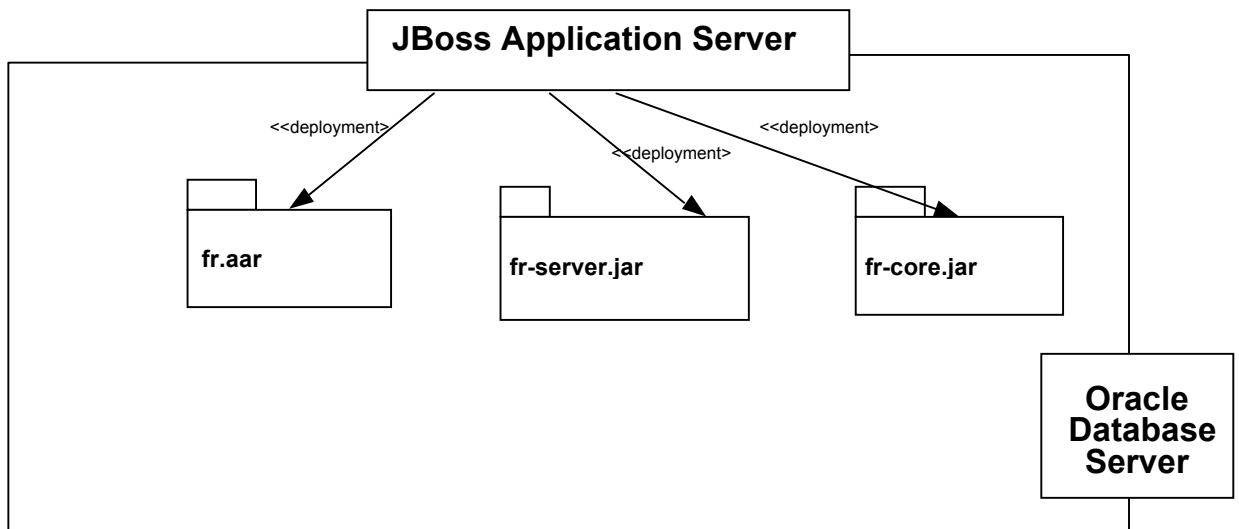|	FR Server	Project Management Server	Metadata Database
FR Service	X	X	
Request Handler	X		
Request DAO	X		X
Vocabulary Data Object	X		

5. Deployment View

5.1 Global Overview
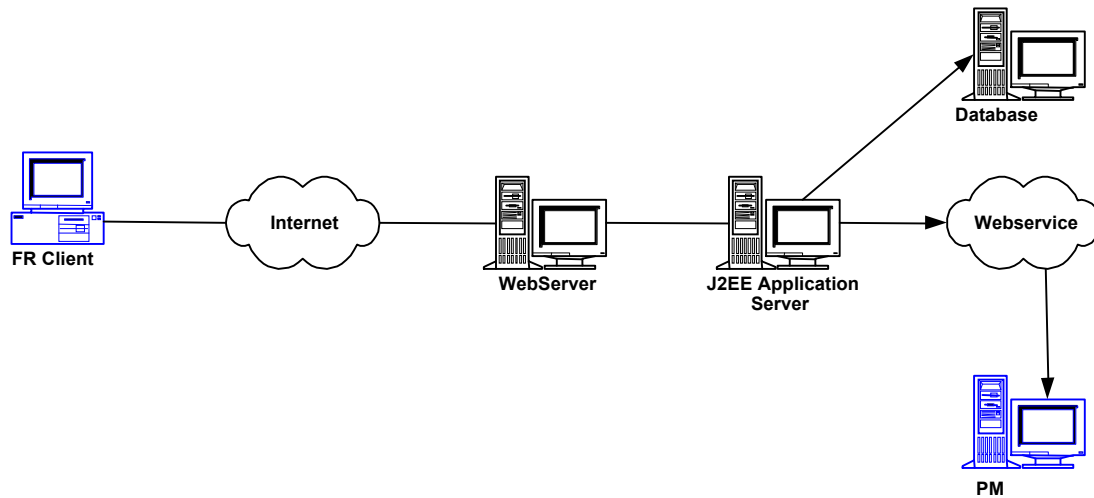
5.2 Detailed deployment model

References

i2b2 (Informatics for Integrating Biology and the Bedside)
https://www.i2b2.org/resrcs/hive.html