**Informatics for Integrating Biology and the Bedside**


i2b2 Software Architecture

# Project Management (PM) Cell

*Document Version:* *1.6.2*

*I2b2 Software Version:* *1.6*

## Table of Contents

# DOCUMENT MANAGEMENT

| Revision Number | Date | Author | Description of change |
|---|---|---|---|
| 1.6.1 | 07/22/10 | Janice Donahoe | Created 1.6 version of document. |
| 1.6.2 | 09/22/11 | Mike Mendis | Removed the ADMIN role |

## ABSTRACT

This is a software architecture document for Project Management (PM) 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 PM message format.

# 1. OVERVIEW

The Project Management cell (PM) is an i2b2 Hive core cell. This cell has two basic functions: to control user access to various services and to keep track of where these services are located.

User access is determined by a user's '*role*', which is a variable associated with a user that serves to define the actions that a user may perform. The role may determine how much data to return and whether or not there is access to a particular service.

***Roles are further defined in the next section which is called User Roles.***

In addition to roles, there is the concept of a 'target location' or 'domain' that further defines the environment and associated permissions. The target location is a variable that defines the PM server location to be accessed. When a person logs in to the i2b2 workbench, a login screen comes up that requires the username, password and target location to be entered. The target location is also called the domain and it is used to authenticate the user. The domain is actually shorthand for the domain name. The i2b2 cells have mappings of domain names to URLs, which tell where the service is that will authenticate the user. If the domain does not exist in the lookup table, the person is not authenticated. If the domain exists, the user is authenticated. In effect, the mapping of the domain name to the URL provides an extra layer of security to the authentication process.

The PM cell next performs authorization, the process of determining the user's roles and permissions and privileges, and returns what the user is allowed to see. The message used by the PM cell for this information is **get_user_configuration.**

## 1.1 User Role

The PM determines when and how data is presented to a user based on their project user roles. Each user will have at least two roles per user_id and product_id combination. These two roles can be further defined as a **Data Protection role** and a **Hive Management role**.

The data protection role/path establishes the detail of data the user can see while the hive management role/path defines their level of functionality the user has in a project.

| Data Protection Track | |
|---|---|
| **Role** | **Access Description** |
| DATA_OBFSC | OBFSC = Obfuscated<br>▪ The user can see aggregated results that are obfuscated (example: patient count).<br>▪ The user is limited on the number of times they can run the same query within a specified time period. If the user exceeds the maximum number of times then their account will be locked and only the Admin user can unlock it. |
| DATA_AGG | AGG = Aggregated<br>▪ The user can see aggregated results like the patient count.<br>▪ The results are not obfuscated and the user is not limited to the number of times they can run the same query. |
| DATA_LDS | LDS = Limited Data Set<br>▪ The user can see all fields except for those that are encrypted.<br>▪ An example of an encrypted field is the *blob fields* in the *fact* and *dimension tables*. |
| DATA_DEID | DEID = De-identified Data<br>▪ The user can see all fields including those that are encrypted.<br>▪ An example of an encrypted field is the *blob fields* in the *fact* and *dimension tables*. |
| DATA_PROT | PROT = Protected<br>▪ The user can see all data, including the identified data that resides in the Identity Management Cell. |

| Hive Management Track | |
|---|---|
| **Role** | **Access Description** |
| USER | Can create queries and access them if he/she is the owner of the query. |
| MANAGER | Can create queries and can access queries created by different users within the project. |

***Additional roles can be added to the* PM_PROJECT_USER_ROLES *table but there will not be any recognized hierarchy to those roles.***

## 1.2 Security: User Authentication

Users may access PM with a user_id and password combination, secure http (https) can be used to encrypt the username, password and all transmitted data to and from the PM cell.

## 1.3 Technical Platform

The technology used to build the product is as follows

- Java 2 Standard Edition 5.0 version 11
- Oracle Server 10g database (optional)
- JBoss Application server version 4.2.2 and higher
- Axis2 1.1 web service (SOAP/REST messaging)

### 1.3.1 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

### 1.3.2 Persistence

This application utilizes JDBC calls to retrieve persisted data.

### 1.3.3 Reliability/Availability

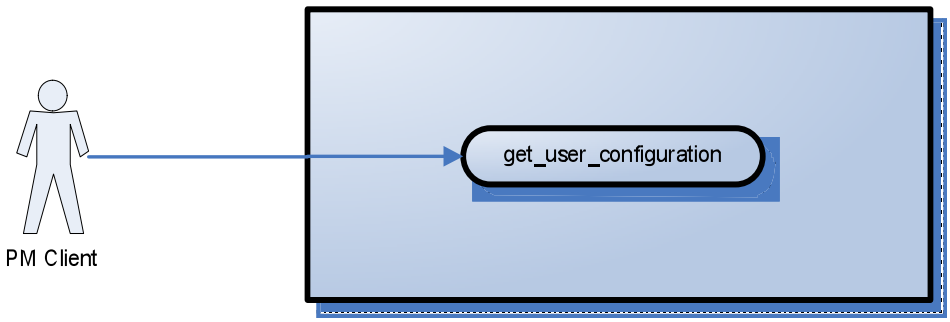
- The Reliability/Availability will be addressed through the J2EE platform
- Targeted availability is 16/7: 16 hours a day, 7 days a week
- The time left (8 hours) is reserved for any maintenance activities

### 1.3.4 Performance

The user authentication and authentication must be under 10 seconds.

# 2. USE CASE

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




***The PM Messaging document contains a complete list of detailed use cases.***

## 2.1 Operations

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

| | |
|---|---|
| **get_user_configuration** | Returns a list of project and roles available for a given user. Also all the services cell information for the hive is provided. |

***The PM Messaging document contains a complete list of operations.***

# 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

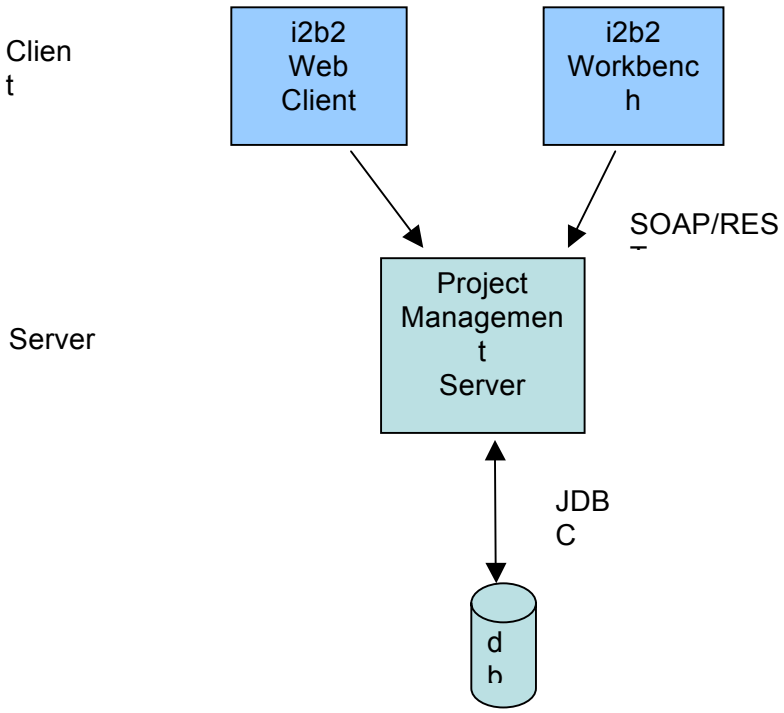
## 3.1 Components and Connector View

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. The following diagram in the Primary Presentation section 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 PM system is represented using the client-server view.

3.1.1.1 PRIMARY PRESENTATION




### 3.1.1.2 ELEMENT CATALOG

| Element Name | Type | Description |
|---|---|---|
| i2b2 Workbench | Client Component | Webservice client submits the requests to the PM server components and renders a response XML. |
| Project Management Server | Server Component | The ONT cell uses the PM cell to authenticate the user.<br>The ONT cell constructs the PM request message and makes a web service call to the PM cell. |
| db | Data Repository Component | This repository is a database for the cell, group, role and user information. |
| JDBC | Query Connector | SQL query used as a connector between the PM system and the metadata database. |
| Web Service | Request Connector | SOAP or REST protocol used to communicate with the external system. |

#### 3.1.1.3 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 thus providing for a high degree of portability.

## 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.

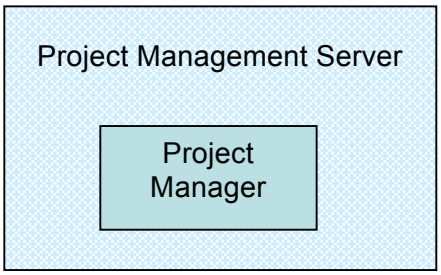
#### 3.2.1.1 PRIMARY PRESENTATION

| System | Segment |
|---|---|
| Project Management Server | Project Manager |

#### 3.2.1.2 ELEMENT CATALOG

| Element Name | Type | Description |
|---|---|---|
| Project Manager | Subsystem | This subsystem manages queries for the user and cell operations. |

#### 3.2.1.3 CONTEXT DIAGRAM




### 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

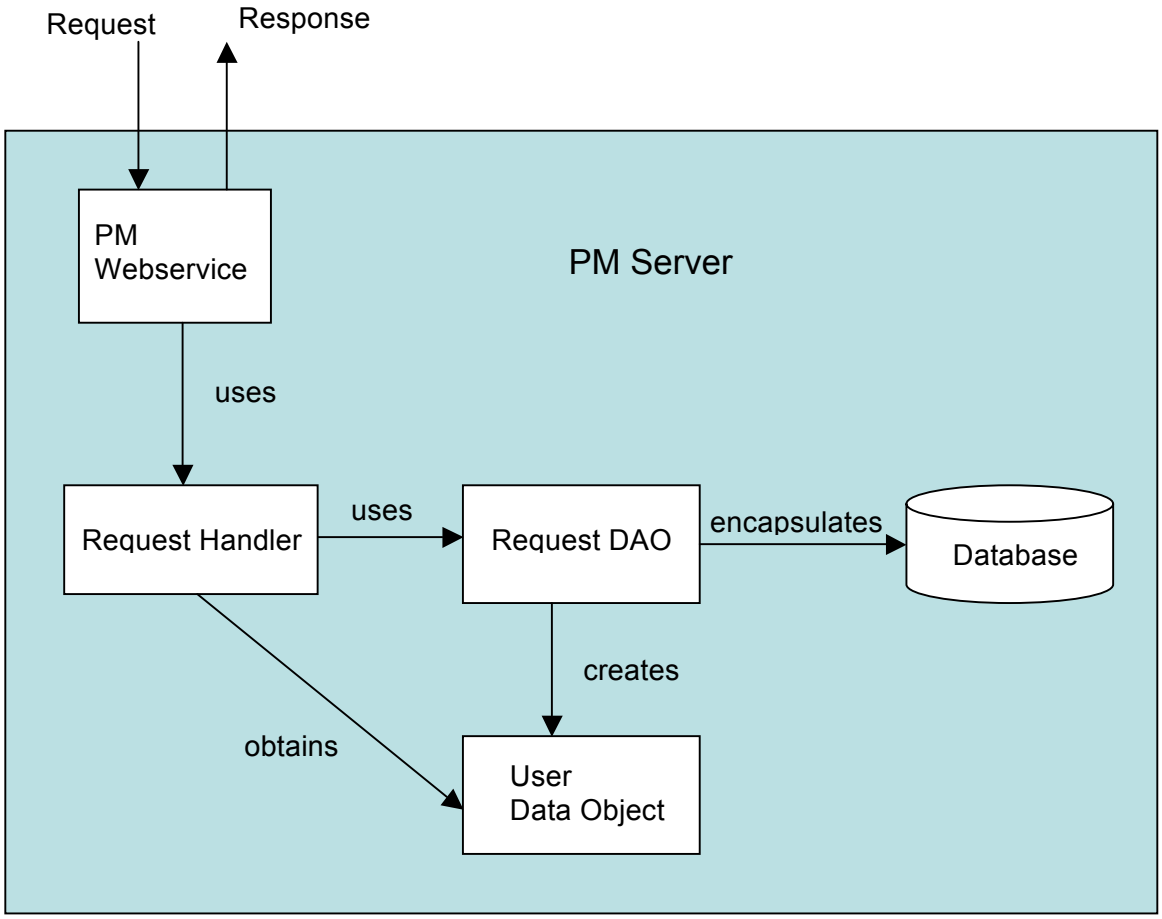
| System | Segment |
|---|---|
| Project Management Server | PM Module |
| Project Manager Subsystem | PM Webservice<br>Request Handler<br>Request DAO<br>User Data Object |

#### 3.2.2.2 ELEMENT CATALOG

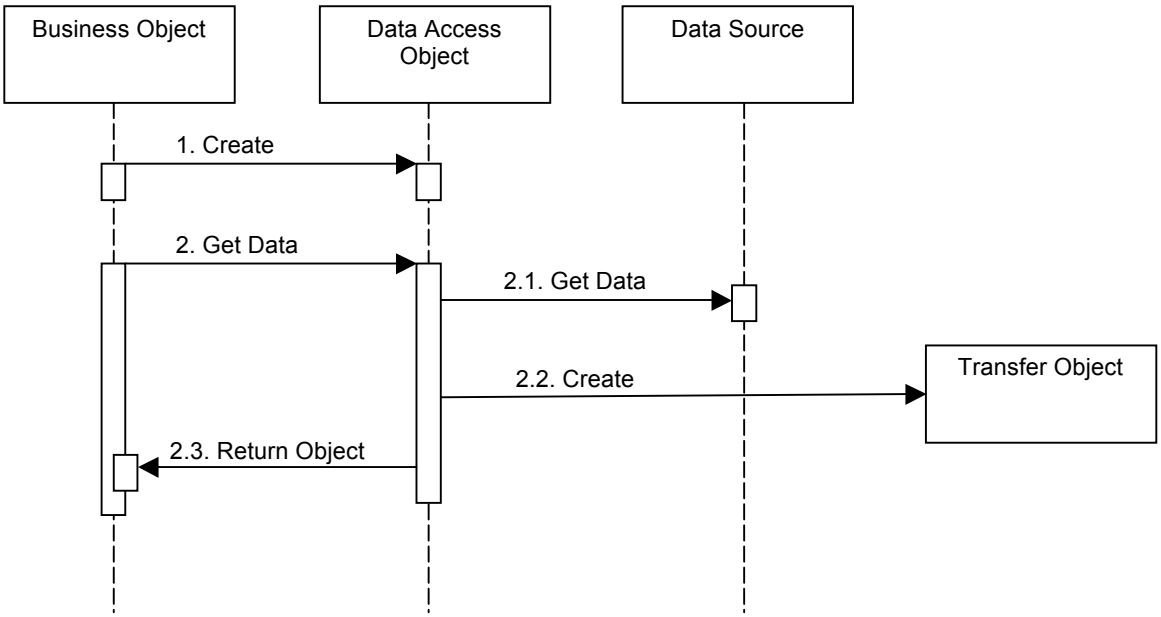
| Element Name | Type | Description |
|---|---|---|
| PM Module | Module | Authenticates the user through local user name or active directory. |

| | | |
|---|---|---|
| PM Webservice | Communication Module | Provides a web service interface to project manager operations. |
| Request Handler | Business Object | Delegates requests to the data access object layer to perform database operations. |
| Request DAO | Data Access Object | Supports database query operations. |
| User Data Object | Transfer Object | Object representation of persisted data |

### 3.2.2.3 CONTEXT DIAGRAM




### 3.2.2.4 SEQUENCE DIAGRAM




## 3.3 Mappings of Styles

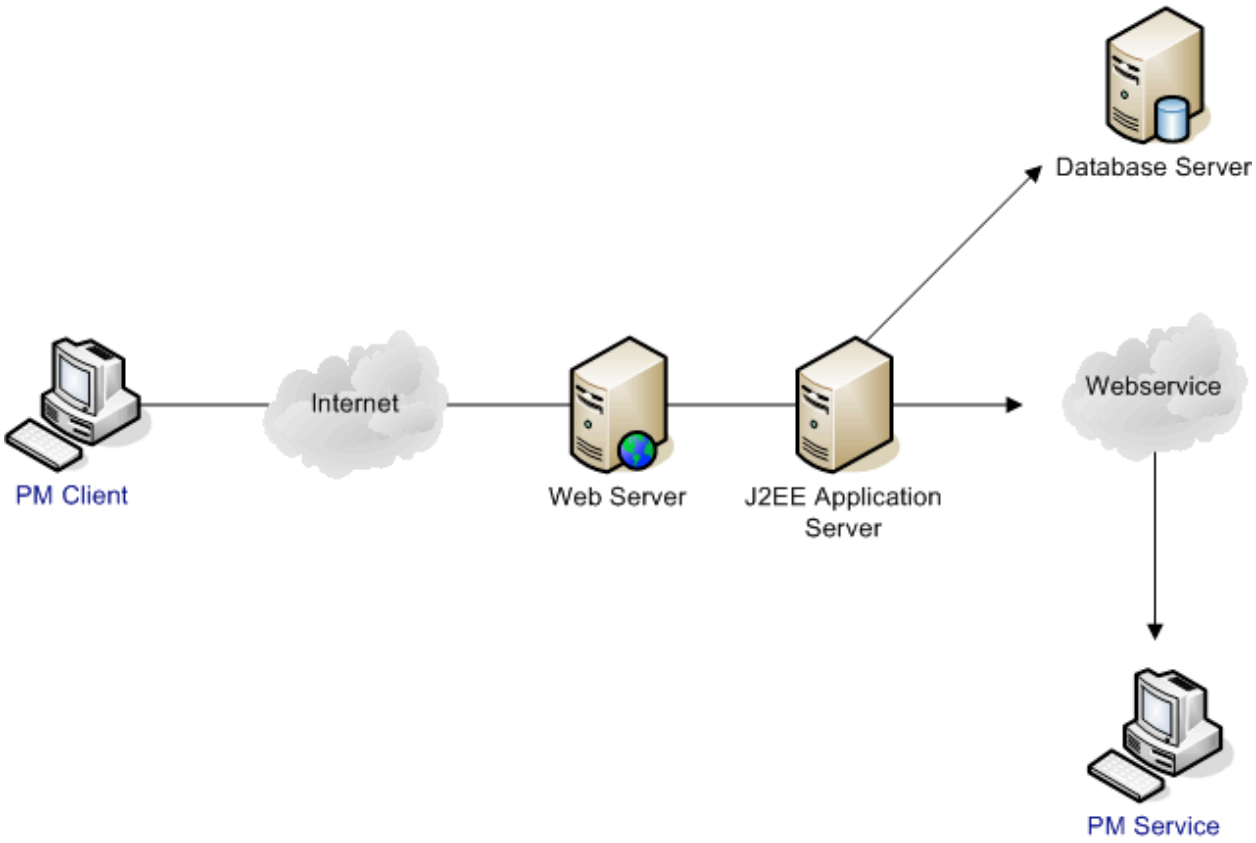
The following table is a mapping between the elements in the *Components and Connector Client-Server view* shown in section 3.1.1, and the *Modules Decomposition and Uses views* shown in sections 3.2.1 and 3.2.2.

The relationship shown is *is-implemented-by*, i.e. the elements from the components and connector 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.

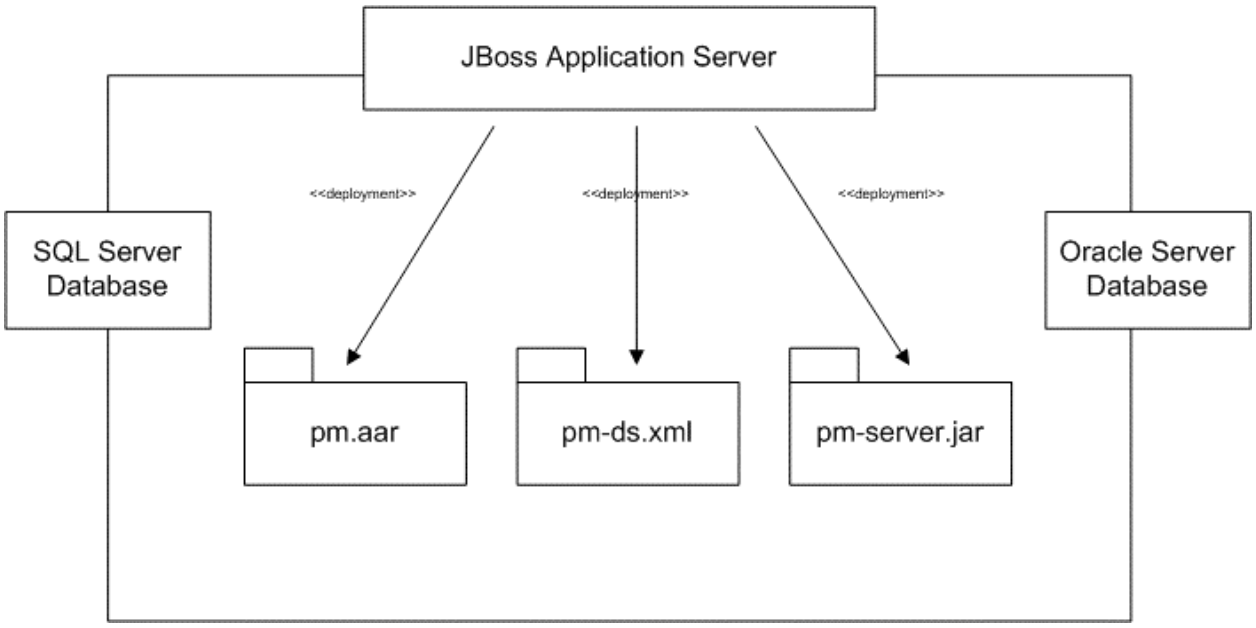
| | **PM Server** | **Metadata Database** |
|---|---|---|
| PM Service | X | |
| PM Webservice | X | |
| Request Handler | X | |
| Request DAO | X | X |
| User Data Object | X | |

# 4. DEPLOYMENT VIEW

## 4.1 Global Overview



## 4.2 Detailed deployment model



# REFERENCES

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