



i2b2

Software Documentation

i2b2 Software Architecture

Workplace Framework (WORK) Cell

TABLE OF CONTENTS

DOCUMENT MANAGEMENT	4
ABSTRACT	5
1 OVERVIEW.....	6
1.1 WORK DEFINITIONS, ACRONYMS AND ABBREVIATIONS	6
1.1.1 <i>Work Data Object (WDO)</i>	6
1.2 ROLES.....	6
1.3 SECURITY.....	8
1.4 SCOPE OF THE SYSTEM.....	8
1.5 ASSUMPTIONS / CONSTRAINTS	8
1.6 TECHNICAL PLATFORM	8
1.6.1 <i>Transaction</i>	9
1.6.2 <i>Security</i>	9
1.6.3 <i>Persistence</i>	9
1.6.4 <i>Reliability / Availability</i>	9
1.6.5 <i>Performance</i>	10
2 USE CASE	11
2.1 OPERATIONS	11
3 ARCHITECTURE DESCRIPTION	13
3.1 COMPONENTS AND CONNECTOR VIEW.....	13
3.1.1 <i>Client-Server Style</i>	13
3.1.1.1 Primary Presentation	14
3.1.1.2 Element Catalog.....	15
3.1.1.3 Design Rationale, Constraints	15
3.2 MODULE VIEW TYPE.....	15
3.2.1 <i>Decomposition Style</i>	16
3.2.1.1 Primary Presentation	16
3.2.1.2 Element Catalog.....	16
3.2.1.3 Context Diagram	16
3.2.2 <i>Uses Style</i>	17
3.2.2.1 Primary Presentation	17
3.2.2.2 Element Catalog.....	17
3.2.2.3 Context Diagram	18
3.2.2.4 Sequence Diagram	19
3.3 MAPPINGS OF STYLES	19
4 DATA VIEW.....	21
4.1 SELECTING THE DATA SOURCE	21
4.2 SCHEMAS WITHIN THE METADATA DATA SOURCE.....	23
4.2.1 <i>TABLE_ACCESS Table</i>	23
4.2.2 <i>WORKPLACE Table</i>	24
5 DRAG AND DROP ACTIONS.....	26
5.1 CURRENT.....	26
6 DEPLOYMENT VIEW.....	27
6.1 GLOBAL OVERVIEW	27

6.2 DETAILED DEPLOYMENT MODEL28

REFERENCES..... 29

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.7.0	03/11/13	Janice Donahoe	Created 1.7 version of document.
1.7.1	10/23/13	Mike Mendis	Added jboss and Axis
1.7.00-002	08/14/2015	Janice Donahoe	Fixed <i>Document Version</i> information that appears on the cover page. This was never updated to 1.7.1. It has now been updated to 1.7.00-002. Also fixed some spelling and grammar errors.
1.7.00-003	04/27/2016	Janice Donahoe	Fixed title page and some minor formatting. Content did not change.
1.7.08-004	10/04/2016	Janice Donahoe	Fixed minor formatting issues. Content did not change.

ABSTRACT

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

1 OVERVIEW

The Workplace Framework cell (WORK) is an i2b2 Hive core cell. This cell manages i2b2 project specific XML data objects for user of a given project. The project specific XML data objects originate in other views or cells, such as Ontology or Previous Query and are stored in the WORK cell as a convenience.

The project specific XML data objects in the WORK cell are organized in hierarchical structures that represent the relationships between elements. The top levels in the hierarchy are called the “**parents**” or “**roots**” with the lower levels being their “**children**”. Elements occurring on the same level are known as “**siblings**”. A level in a hierarchy is sometimes referred to as a “**node**”.

The WORK cell does the following:

1. Accepts new XML data object for storage
2. Provides a listing of those items previously stored.
3. Allows the user to organize, label and annotate stored data objects however they choose.

1.1 WORK Definitions, Acronyms and Abbreviations

1.1.1 Work Data Object (WDO)

This object holds workplace folder definitions and information about the folder contents.

1.2 Roles

When and how data is presented to a user is based on their user roles, which are specified in the PM Cell. 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 establishes the detail of data the user can see while the hive management role defines the level of functionality the user has in a project. The following tables summarize the roles in a hierarchical order of least to most access.

summarize the roles in a hierarchical order of least to most access.

Data Protection Track	
Role	Access Description
DATA_OBFSC	<p>OBFSC = Obfuscated</p> <ul style="list-style-type: none"> The user can see aggregated results that are obfuscated (example: patient count). 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	<p>AGG = Aggregated</p> <ul style="list-style-type: none"> The user can see aggregated results like the patient count. The results are <u>not</u> obfuscated and the user is <u>not</u> limited to the number of times they can run the same query.
DATA_LDS	<p>LDS = Limited Data Set</p> <ul style="list-style-type: none"> The user can see all fields except for those that are encrypted. An example of an encrypted field is the <i>blob fields</i> in the <i>fact</i> and <i>dimension tables</i>.
DATA_DEID	<p>DEID = De-identified Data</p> <ul style="list-style-type: none"> The user can see all fields including those that are encrypted. An example of an encrypted field is the <i>blob fields</i> in the <i>fact</i> and <i>dimension tables</i>.
DATA_PROT	<p>PROT = Protected</p> <ul style="list-style-type: none"> 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 as well as access queries created by different users within the project



Additional Resources

Further details regarding roles can be found in the *Project_Management_Design* document.

1.3 Security

Users can access the WORK with user-id and password combination, which is authenticated through the Project Management Cell. The implementation detail of the Project Management Cell is considered out-of-scope to this system context.

✔ Additional Resources

Further details about the implementation of the Project Management Cell can be found in the following documents:

- Project_Management_Architecture
- Project_Management_Design
- Project_Management_Messaging
- i2b2 Installation Guide

1.4 Scope of the system

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

- Project Management Cell

1.5 Assumptions / Constraints

The Workplace database shall not contain protected health information.

1.6 Technical Platform

The product is designed to run on the following platform:

- Java 2 Standard Edition 7.0
- Oracle Server 10g/11g database
- SQL Server 2005/2008

- Xerces2 XML parser
- Jboss Application server version 7.1.1
- Spring Web Framework 2.0
- Axis2 1.6.2 web service (SOAP / REST)

1.6.1 Transaction

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

1.6.2 Security

The application must implement basic security behaviors:

Category	Behavior
Authentication	Authenticate using at least user name and password.
Authorization	Based on the user role, the user may only access those categories allowed to by the 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

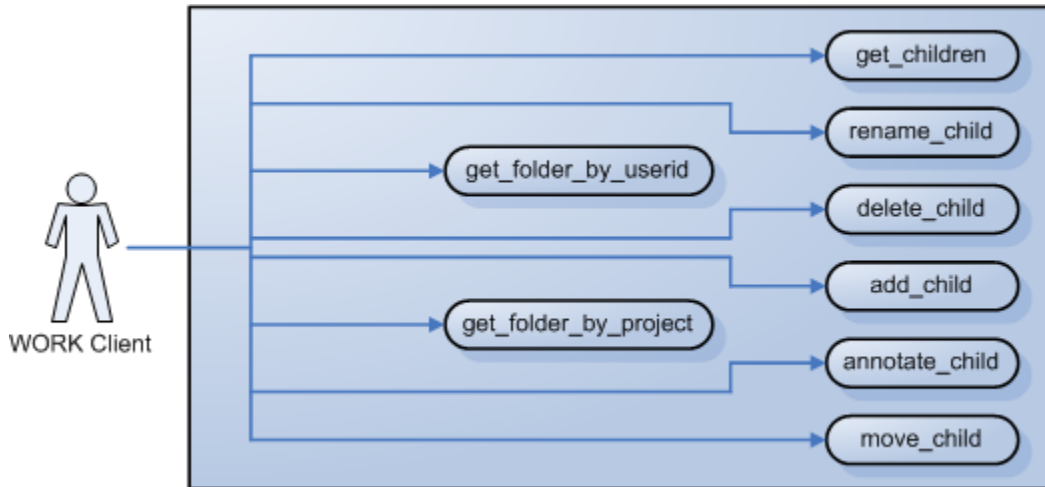
- The remaining time (8 hours) is reserved for any maintenance activities

1.6.5 Performance

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

2 USE CASE

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



2.1 Operations

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

Service Operation	Description
get_folder_by_userid	Returns a list of root folders available for a given user. These folders are displayed in a tree format. The top level of the tree consists of all the folders a particular user has permission to see as determined by his / her role.
get_folder_by_project	Returns a list of root folders available for a given project across all users in the project. These folders are displayed in a tree format. A user must have manager permission and be configured as such to see all folders in a project.
get_children	Expands any level of a folder, providing information about its contents, for a given user.
rename_child	Renames a work item (leaf or folder) in the workplace tree.
delete_child	Deletes a work item (leaf or folder) in the workplace tree.
add_child	Adds a work item (leaf or folder) in the workplace tree.
annotate_child	Annotates a work item (leaf or folder) in the workplace tree by modifying or adding a

Service Operation	Description
	tooltip.
move_child	Moves a work item (leaf or folder) in the workplace tree and reassigns all its children.
export_child	Export a work item (leaf or folder) as XML to a local file.

3 ARCHITECTURE DESCRIPTION

This document provides the 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

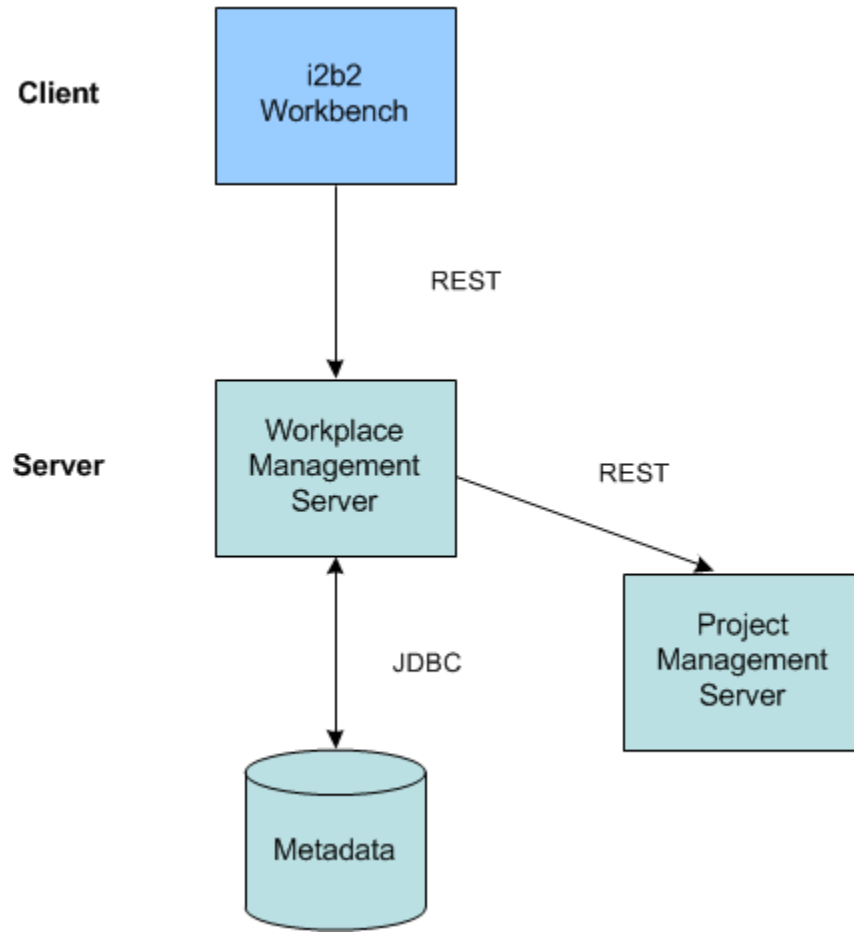
3.1 Components and Connector View

A **Component and Connector view** (C&C) 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 diagram shown in the *Primary Presentation* section represents the Component and Connector view for the multi-user installation. As seen in the diagram, component instances are shown in more detail with specific connectors drawn in different notations.

3.1.1 Client-Server Style

The WORK system is represented using the components and connector 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 request to the WORK Server components and renders a response XML.
Workplace Framework Server	Server Component	Provides a Web Service Interface for the WORK system. It supports the REST protocol. It uses the Project Management server to handle user authentication.
Project Management Server	Server Component	WORK cell uses the Project Management cell to authenticate the user. WORK cell constructs the PM request message and makes a web service call to the Project Management cell.
Workplace	Data Repository Component	This repository is a database for the i2b2 workplace data.
JDBC	Query Connector	SQL query used as a connector between the WORK System and the Metadata database.
Web Service	Request Connector	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 the 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.

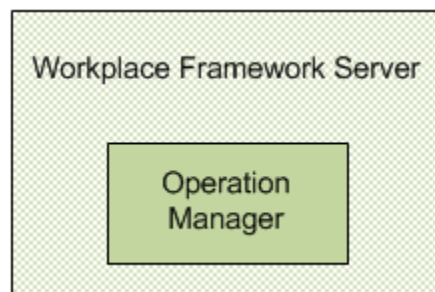
3.2.1.1 Primary Presentation

System	Segment
Workplace Framework Server	Operation Manager

3.2.1.2 Element Catalog

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

3.2.1.3 Context Diagram



3.2.2 Uses Style

The Uses style shows the relationship between modules and sub-modules. This view is very helpful for implementing, integrating and testing the system.

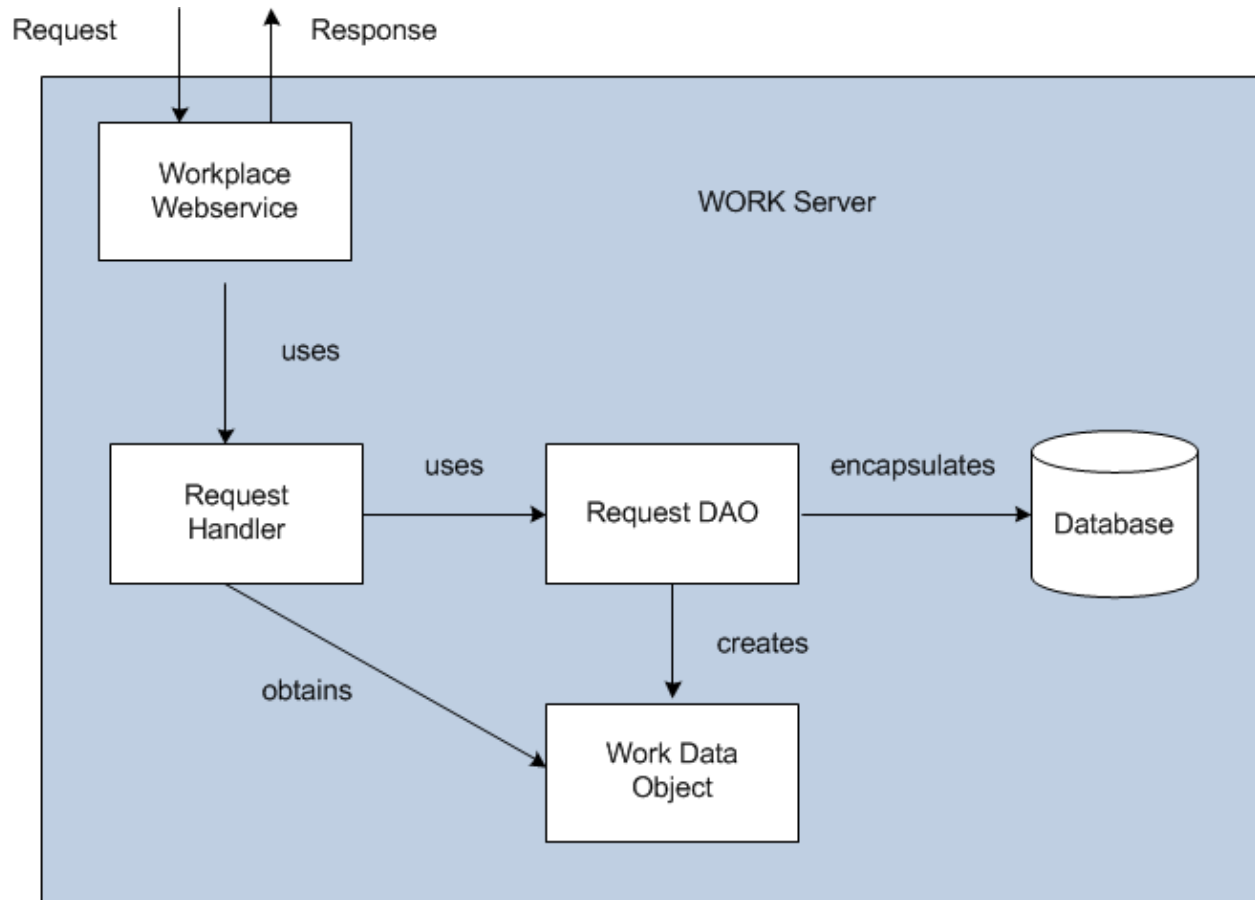
3.2.2.1 Primary Presentation

System	Segment
Workplace Framework Server	WORK Module
Operation Manager Subsystem	Workplace Webservice Request Handler Request DAO Work Data Object

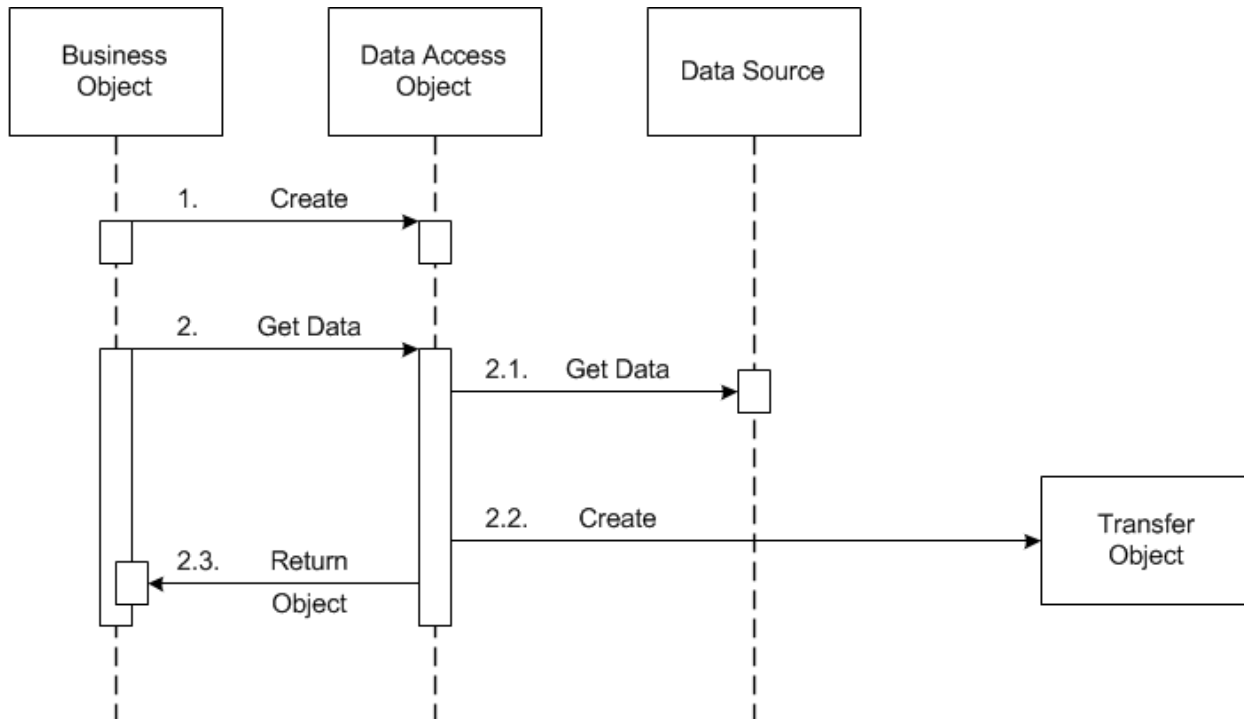
3.2.2.2 Element Catalog

Element Name	Type	Description
WORK Module	Module	Authenticates the user through the PM Server System.
Workplace Webservice	Communication Module	Provides web service interface to workplace operations.
Request Handler	Business Object	Delegates' workplace requests to Data Access Object layer to perform database operations.
Request DAO	Data Access Object	Supports database query operations.
Work 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 *Component & Connector Client-Server* view shown in section 3.1.1, and the *Modules Uses* view and *Decomposition* view 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 and "X" in the corresponding cell.

	WORK Server	Project Management Server	Workplace Database
WORK Service	X	X	
Workplace Webservice	X		

	WORK Server	Project Management Server	Workplace Database
Request Handler	X		
Request DAO	X		X
Work Data Object	X		

4 DATA VIEW

4.1 Selecting the Data Source

Stored Workplace data is distributed to projects through the existence of independent databases (in SQL Server) or schemas (in Oracle). These will be referred to in the rest of the document as the “**persistent storage location**” or **PSL**. These PSL’s are organized so that the data from two metadata representations can be merged to a “Super” data set. While a person is working on a specific project, they will be directed to data in a PSL associated with that project.

In order to support the i2b2 project distribution strategy, the user is enrolled in numerous projects recorded within the i2b2 project management cell. The projects available to the user are returned in the web service call to the Project Management cell. The logic of selecting the correct PSL for the project is embodied in the following table.

DB_LOOKUP		
PK	C_DOMAIN_ID	VARCHAR(255)
PK	C_PROJECT_PATH	VARCHAR(255)
PK	C_OWNER_ID	VARCHAR(255)
	C_DB_FULLSCHEMA	VARCHAR(255)
	C_DB_DATASOURCE	VARCHAR(255)
	C_DB_SERVERTYPE	VARCHAR(255)
	C_DB_NICENAME	VARCHAR(255)
	C_DB_TOOLTIP	VARCHAR(255)
	C_COMMENT	CLOB
	C_ENTRY_DATE	DATE
	C_CHANGE_DATE	DATE
	C_STATUS_CD	CHAR(1)

The logic for selecting the PSL is as follows:

1. There are two methods to select the correct PSL, an implicit one, and an explicit one. Both rely only on information available within the i2b2 header.

- a. The implicit one relies upon the data within the <domain> tag, the <username> tag, and the <project_id> tag.
- b. The explicit one relies upon the data only within the <project_id> tag. It has the format represented as the following string:

[“DOMAIN” | “PROJECT” \ “sub-project” \ “sub-sub-project”\ | “USER_ID”]

Note

These may not actually match the domain and username that is actually being used (since it is being built by the client), and must be checked when the PM cell is accessed.

2. The table is meant to provide a series of default locations if ones are not specifically listed. If a project is listed in the *C_PROJECT_PATH* column, then that PSL may be used, otherwise a domain source will be used.
3. If a username is listed in the *C_OWNER_ID* column, and the project also matches the *PROJECT_ID*, the PSL in that row may be used otherwise a project PSL will be used. If the project PSL does not exist, the domain PSL will be used.

For example, only if the *domain \ project \ user_id* is an EXACT match to the entries in the database will that PSL be used.

4. The project id may have associated sub-projects that will be represented as *project \ sub-project \ sub-sub-project* string. If a sub-project is identified but only the project exists in the table then the project PSL would be used.
5. The project may not have an entry in the table and in that case any project (and sub-projects) would be designated the PSL of the domain.
6. If a general domain PSL is not available in the table and only a specific project is associated with the domain in the table, then any incoming messages not associated with that project will return an error.
7. In the table, the “@” character is used to represent the absence of an entry (rather than a blank or a null).
8. In the explicit string and in the <project_id> a “@” can be used to optionally represent a blank column.

Other columns are specified as follows:

1. The column `C_DB_FULLSCHEMA` is used to contain the path to a table when the data source is used. Software is written so that the absence of the delimiter (usually a ".") does not need to be explicitly stated.
2. The column `C_DB_DATASOURCE` is used to contain a short string that represents a data source configured in some other location.
3. The column `C_DB_SERVERTYPE` can be "ORACLE" or "SQLSERVER".
4. The column `C_DB_NICENAME` is a string that can be used in the client software to describe a data source.
5. The column `C_DB_TOOLTIP` contains a longer (hierarchical) representation of the nice name.

To restate, many cells need to access some kind of persistent storage, and these cells will organize their persistent storage so that it is self-contained and can be apportioned in a way consistent with the project-based requirements of i2b2 that are described above. To that end, a table exists in many cells to make the decision of what persistent storage location to which a specific user will be directed, depending on the project and domain to which they are associated.

4.2 Schemas within the Metadata Data Source

The following schemas provide data used by the WORK system.

4.2.1 TABLE_ACCESS Table

The **WORKPLACE_ACCESS** schema expands upon the WDO, containing security information for the root level work nodes. It is used to obtain a list of "Workplace" tables within the **Persistent Storage Location (PSL)**. Each workplace table within the PSL is represented by a single row in this table. The primary identifier of each workplace table in the **WORKPLACE_ACCESS** table is in the `C_TABLE_CD` column. All messages that need to point to a specific workplace table will use this identifier. For example, in the <key> element of many messages this identifier is represented as `\\C_TABLE_NAME\` (see messaging specification). Within the `C_TABLE_NAME` column is the actual name in the PSL of the workplace table. The `C_PROTECTED_ACCESS` column designates whether one must have the protected access

role (DATA_PROT) to obtain data from this table. The other columns are defined in a similar manner to the columns of the Metadata table, with the following special notes:

The *C_HLEVEL* is almost always 0 in this table.

WORKPLACE_ACCESS		
PK	C_INDEX	VARCHAR(255)
	C_TABLE_CD	VARCHAR(255)
	C_TABLE_NAME	VARCHAR(255)
	C_PROTECTED_ACCESS	CHAR(1)
	C_HLEVEL	INT
	C_NAME	VARCHAR(255)
	C_USER_ID	VARCHAR(255)
	C_GROUP_ID	VARCHAR(255)
	C_SHARE_ID	VARCHAR(255)
	C_VISUALATTRIBUTES	CHAR(3)
	C_PARENT_INDEX	VARCHAR(255)
	C_TOOLTIP	VARCHAR(255)
	C_ENTRY_DATE	DATE
	C_CHANGE_DATE	DATE
	C_STATUS_CD	CHAR(1)

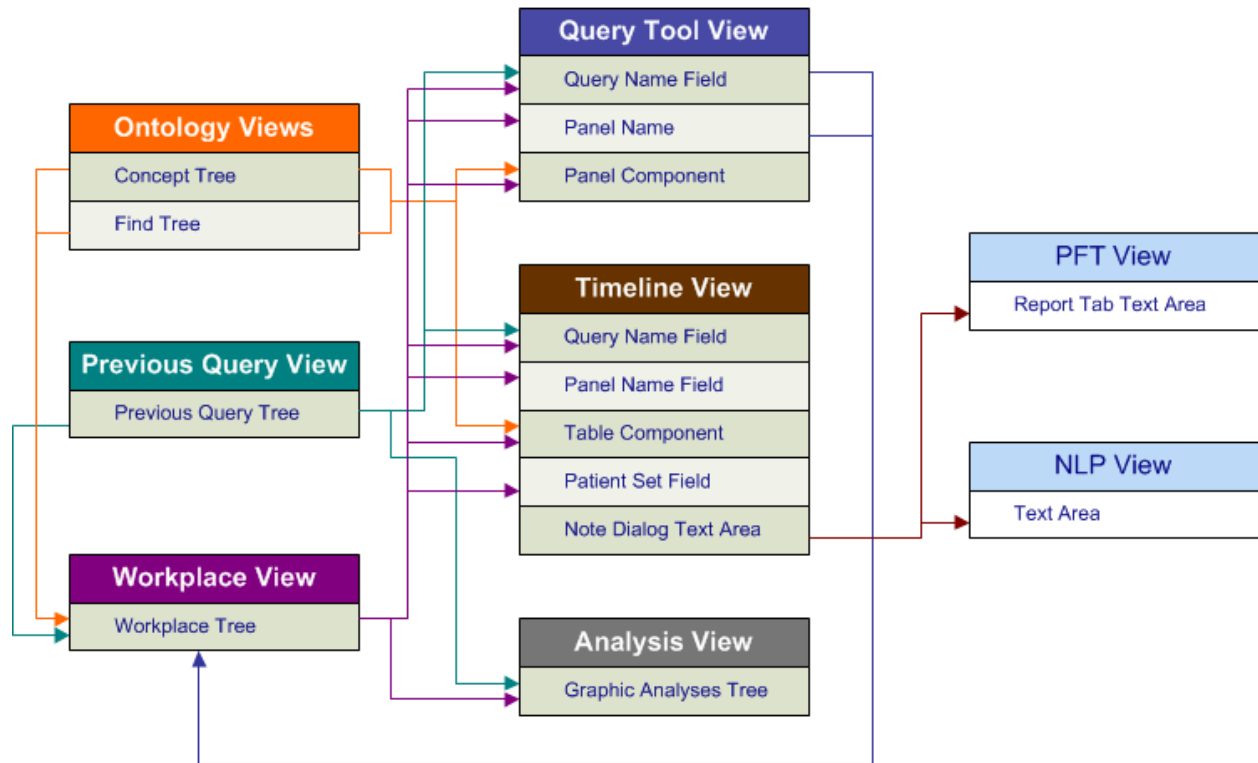
4.2.2 WORKPLACE Table

The **WORKPLACE** table encapsulates workplace storage items used in the data repository. A folder is a work data object encapsulation of a row from the Workplace table. It is the primary object used to pass workplace items to the requesting client. A PSL may contain numerous Workplace tables, each with a name that indicates the domain that the workplace storage items contained within represent.

WORKPLACE		
PK	C_INDEX	VARCHAR(255)
	C_NAME	VARCHAR(255)
	C_USER_ID	VARCHAR(255)
	C_GROUP_ID	VARCHAR(255)
	C_SHARE_ID	VARCHAR(255)
	C_VISUALATTRIBUTES	CHAR(3)
	C_PARENT_INDEX	VARCHAR(255)
	C_TOOLTIP	VARCHAR(255)
	C_PROTECTED_ACCESS	CHAR(1)
	C_WORK_XML	CLOB
	C_WORK_XML_SCHEMA	CLOB
	C_WORK_XML_I2B2_TYPE	VARCHAR(255)
	C_ENTRY_DATE	DATE
	C_CHANGE_DATE	DATE
	C_STATUS_CD	CHAR(1)

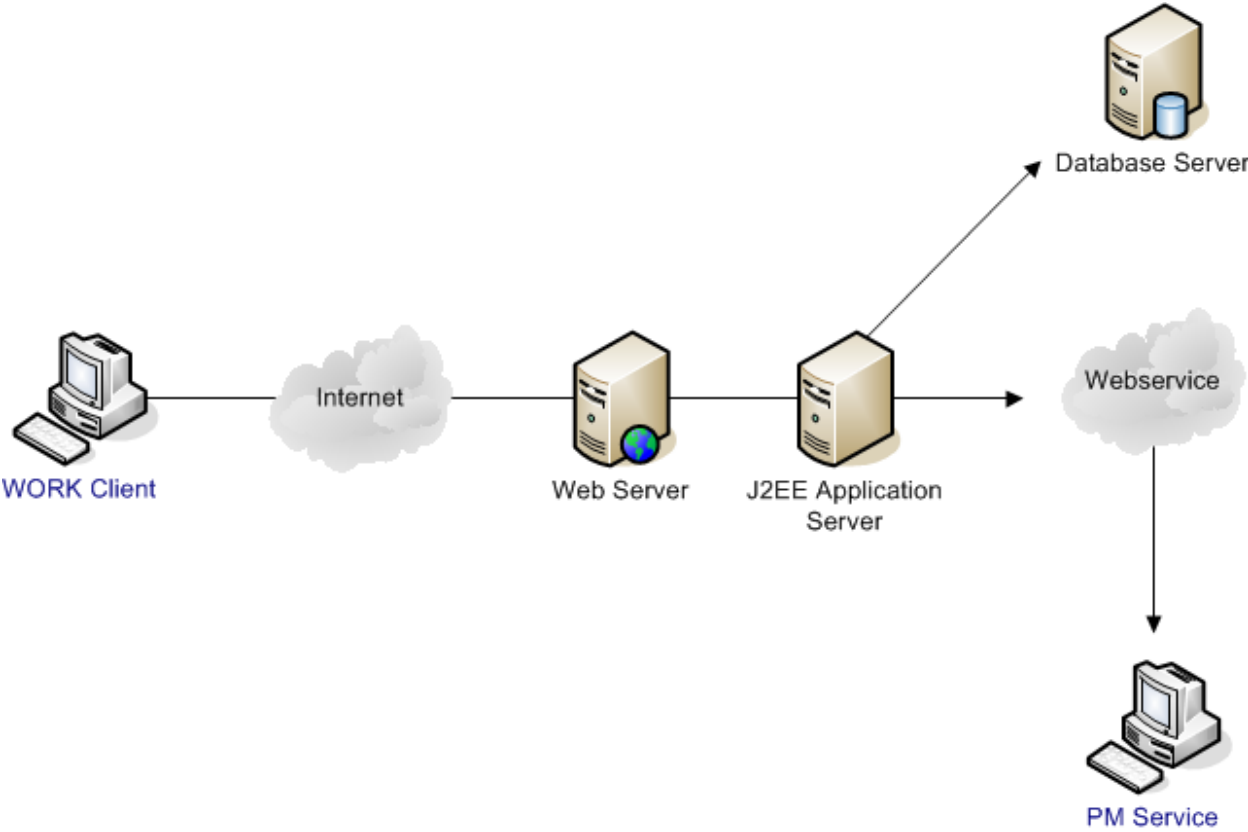
5 DRAG AND DROP ACTIONS

5.1 Current

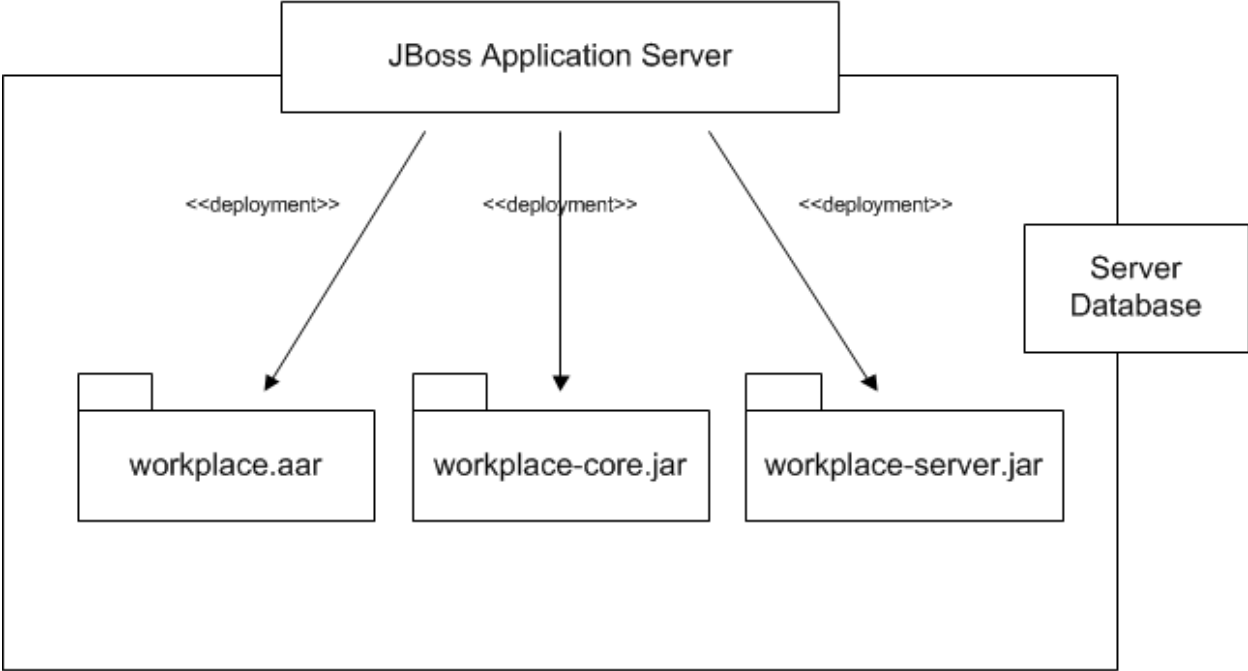


6 DEPLOYMENT VIEW

6.1 Global Overview



6.2 Detailed Deployment Model



REFERENCES

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R. and Stafford, J., *Documenting Software Architectures: Views and Beyond*. (Boston, MA: Addison-Wesley, 2003)

Philippe Kruchten, “Architectural Blueprints – The “4+1” View Model of Software Architecture, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf> (*IEEE Software* 12 (6), November 1996)

“Object Management Group UML 2.0 Specification”, <http://www.omg.org/technology/documents/formal/uml.htm> (Object Management Group)

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