



i2b2

Software Documentation

i2b2 Software Architecture

Data Repository (CRC) Cell

TABLE OF CONTENTS

- TABLE OF CONTENTS..... 2**
- DOCUMENT MANAGEMENT 4**
- ABSTRACT 5**
- 1 OVERVIEW..... 6**
 - 1.1 CRC DEFINITIONS, ACRONYMS AND ABBREVIATIONS 6
 - 1.1.1 *Patient Data Object (PDO)* 6
 - 1.1.2 *Observation Fact*..... 7
 - 1.2 USER ROLE 7
 - 1.3 SECURITY..... 8
 - 1.4 SCOPE OF THE SYSTEM..... 8
 - 1.5 ASSUMPTIONS / CONSTRAINTS 9
 - 1.6 TECHNICAL PLATFORM 9
 - 1.6.1 *Transaction* 9
 - 1.6.2 *Security* 10
 - 1.6.3 *Persistence* 10
 - 1.6.4 *Reliability / Availability* 10
 - 1.6.5 *Performance* 11
- 2 USE CASE 12**
 - 2.1 USE CASE: RUN A QUERY FROM PANEL DEFINITION 12
 - 2.1.1 *CRC Query execution using Queue Model*..... 13
 - 2.1.1.1 Sequence Diagram 13
 - 2.1.1.2 Context Diagram 13
 - 2.2 USE CASE – GET PDO FROM PATIENTSET..... 14
- 3 ARCHITECTURE DESCRIPTION 15**
 - 3.1 COMPONENTS AND CONNECTOR VIEW 15
 - 3.1.1 *Client-Server Style* 15
 - 3.1.1.1 Primary Presentation 15
 - 3.1.1.2 Element Catalog 16
 - 3.1.1.3 Relations and their Properties 17
 - 3.1.1.4 Design Rationale, Constraints 17
 - 3.2 MODULE VIEW TYPE 18
 - 3.2.1 *Decomposition Style*..... 18
 - 3.2.1.1 Primary Presentation 18
 - 3.2.1.2 Element Catalog 18
 - 3.2.1.3 Relations and their Properties 19
 - 3.2.1.4 Context Diagram 19
 - 3.2.2 *Uses Style* 19
 - 3.2.2.1 Primary Presentation 19
 - 3.2.2.2 Element Catalog 20
 - 3.2.2.3 Relation and their Properties 20
 - 3.2.2.4 Context Diagram 20
 - 3.3 MAPPINGS OF STYLES 21
- 4 DATA VIEW..... 22**
 - 4.1 QT_QUERY_MASTER 23
 - 4.2 QT_QUERY_INSTANCE 23

4.3	QT_QUERY_RESULT_INSTANCE	24
4.4	QT_QUERY_RESULT_TYPE	25
4.5	QT_QUERY_STATUS_TYPE	25
4.6	QT_PATIENT_SET_COLLECTION	26
4.7	QT_XML_RESULT	26
4.8	QT_PATIENT_ENC_COLLECTION	27
4.9	QT_PRIVILEGE	27
4.10	QT_PDO_QUERY_MASTER	28
4.11	VOLUMES	28
5	DEPLOYMENT VIEW.....	29
5.1	GLOBAL OVERVIEW	29
5.2	DETAILED DEPLOYMENT MODEL	29
	REFERENCES.....	30

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.7.1	09/12/12	Janice Donahoe	Created 1.7 version of document.
1.7.2	10/23/13	Mike Mendis	Updated Jboss and Axis2
1.7.00-003	07/27/2015	Janice Donahoe	Fixed <i>Document Version</i> information that appears on the cover page. This was never updated to 1.7.2. It has now been updated to 1.7.00-003. Also fixed some minor spelling issues.
1.7.08-004	04/27/2016	Janice Donahoe	Fixed minor issue with the numbering of headers.

ABSTRACT

This is a software architecture document for the **CRC (Clinical Research Chart) 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 CRC message format.

1 OVERVIEW

The Clinical Research Chart (CRC) repository cell is one of the core cells in the i2b2 Hive. The CRC cell is designed with several requirements. The main requirements are:

1. It must be able to hold healthcare information from many different venues and allow it to be queried rapidly even if there are hundreds of millions of rows.
2. It must be easily combined with other project repositories to form large unified repositories.
3. Finally, it must allow objects to be stored that are present in the genomic data.

Currently information in the CRC cell is related to clinical data and hence it's also called Clinical Research Chart. For the remainder of this document, the terms **CRC** and **Data Repository Cell** will be used interchangeably to refer to the same cell.

The CRC is a data warehouse of patient's phenotype and genotype information. It is supported by a powerful metadata management module (the Ontology Cell). Currently the CRC handles concepts such as diagnoses, procedures, medications, and lab tests, but the structure of the table gives enough flexibility to expand this to include virtually any kind of observation. The presence of both genotype and phenotype information makes this cell a powerful tool for researchers.

All patient data present in the CRC is de-identified; the only exception is the patient notes from hospitals. These notes are stored in encrypted form, so only users enabled with an encryption key can view them.

1.1 CRC 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.

Setfinder Query

Setfinder queries are used to create a set of patients that satisfy a criteria presented in the query. The setfinder query is composed of query constraints, a list of panels and its items.

1.1.2 Observation Fact

Any observation made on a patient can be stored as fact information in the CRC data mart. The user can fetch the fact information via the PDO queries.


1.2 User Role

The CRC determines when and how data is presented to a user based on their user roles, which are specified in the Project Management 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.

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 CRC with domain-id, project-id, 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 CRC are:

- Project Management Cell

- Ontology Cell
- edu.harvard.i2b2.common

1.5 Assumptions / Constraints

- The data in the CRC data mart database will not have identified data. The exception to this is the patient notes are stored inside the OBSERVATION_BLOB and it will be encrypted.
- The client will make “Patient Data Object Query / Request” in multiple requests if the input list (Patient Set or Observation Set) is big.

1.6 Technical Platform

The technology used to build the product is as follows:

- 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 CRC system is transactional, leveraging the technical platform capabilities. The transaction management model of the J2EE platform will be reused intensively.

Note

In the current implementation, to support long running setfinder queries, transaction management will be manually turned off until the completion of the query.

1.6.2 Security

The application must implement basic security behaviors:

Category	Behavior
Authentication	Authenticate using the combination of domain id, project id, user name and a password.
Authorization	Based on the user role, the user may access setfinder queries created by other users, view patient notes, etc.
Confidentiality	Sensitive data must be encrypted (Patient Notes).
Data Integrity	Data sent across the network cannot be modified by a tier.
Auditing	All queries and retrieval of patient data is stored for auditing purposes.
User Lockout	Users with the role of DATA_OBFSC will be limited to the number of times they can run the same query in a project. Once they reach that limit their account will be locked out and they will not be able to run queries again until an administrator unlocks the account.

1.6.3 Persistence

This application utilizes JDBC calls to retrieve persisted data.

1.6.4 Reliability / Availability

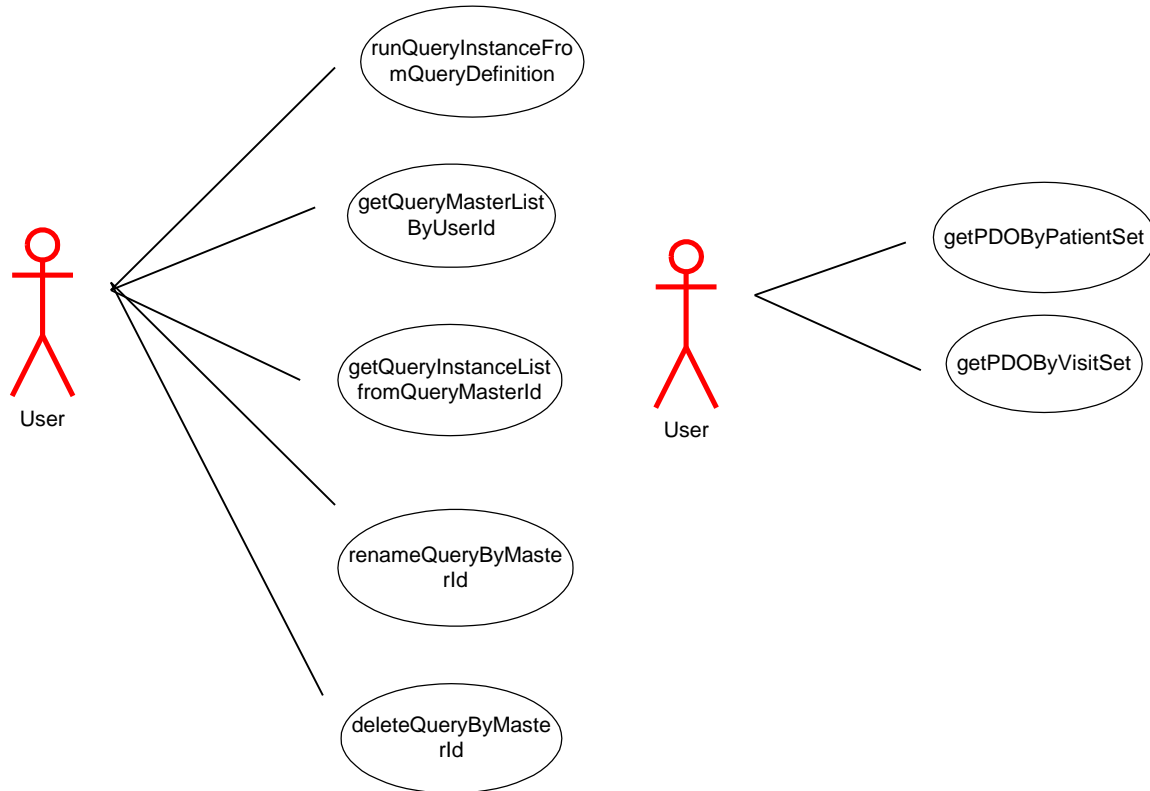
- The reliability / availability will be addressed through the J2EE platform
- Targeted availability is 16 / 7: 16 hours a day, 7 days a week
- The remaining time (8 hours) is reserved for any maintenance activities

1.6.5 Performance

- The user authentication with the project management cell must be under 10 seconds.
- The concept code lookup to the ontology cell must be under 10 seconds.

2 USE CASE

The diagram below depicts the common use cases a user can perform with the CRC cell.



2.1 Use Case: Run a query from Panel Definition

- Validate the user by calling the Project Management cell.
- Select a data mart based on the combination of domain_id, project_id, and user_id.
- Call the Ontology cell with the item key and determine the dimension table to join with the fact table.
- Save the query panel definition and the generated SQL statements.
- Generate the list of output like the patient count, patient gender count, patient set, etc.
- To scale the application and to support long running SQL, the execution of SQL is handled inside a set of queues. At first the query SQL statements will be executed inside

a small job queue. If it doesn't complete within a certain time period then the jobs will be transferred to a mid-size job queue and then to a large size job queue.

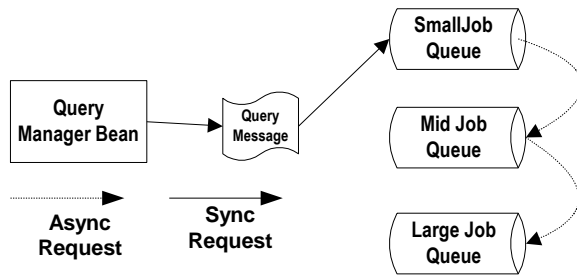
- If the SQL execution completes before the "result_waittime_ms" which is specified in the request, then the query results are passed in the response message. Otherwise the status of the query is passed in the response message.

2.1.1 CRC Query execution using Queue Model

2.1.1.1 Sequence Diagram



2.1.1.2 Context Diagram



2.2 Use Case – Get PDO from PatientSet

- Validate the user via the Project Management Cell
- Select the data mart based on the domain_id, project_id and user_id.
- Call Ontology Cell with the item key and determine the dimension table to join with the fact table.
- Using the given patient set or observation set, apply the panel filters and return PDO.

3 ARCHITECTURE DESCRIPTION

As noted in “Documenting Software Architectures”⁰, software architecture is a complex entity that cannot be described in a simple one-dimensional fashion. 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

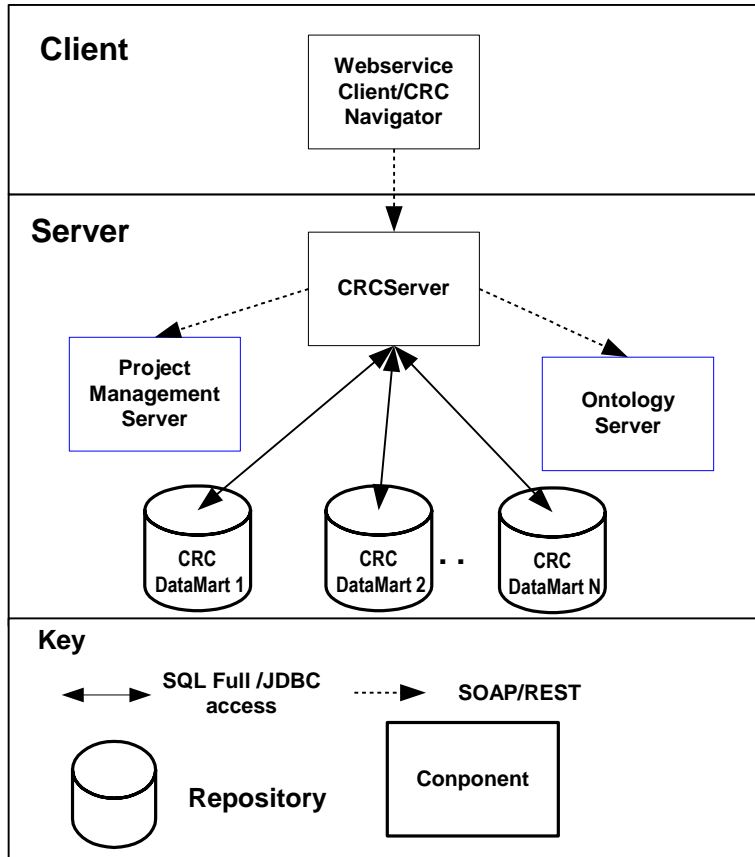
3.1 Components and Connector View

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 CRC system is represented using the components and connector client-server view.

3.1.1.1 Primary Presentation



3.1.1.2 Element Catalog

Elements and their Properties

The properties of the CRC cell elements are:

- *Element Name:* listed in the table shown below.
- *Type:* whether the element is a data repository, a data accessor, a communication method, a query, a client or a server component.
- *A description* of the element

Element Name	Type	Description
--------------	------	-------------

Webservice Client	Client	Webservice client (i2b2 Workbench / Navigator) submits the requests to CRC Server components and renders response XML.
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. Select the CRC data mart based on domain_id, project_id and user_id 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	CRC cell uses the Project Management cell to authenticate the user. The CRC cell constructs PM Cell request message and makes a web service call to Project Management Cell.
Ontology Server	Server	CRC sends web service requests to the Ontology cell to get metadata information about an Observation fact's concepts.
CRC Data mart DB	Data Repository	This repository is mainly a data mart for patient's clinical observation information represented in star schema. The server supports multiple data marts; the data marts are selected based on the domain_id, project_id and user_id combination. This database also holds CRC user queries (setfinder query) information and its results like patient sets, etc.
Full SQL	Query Connector	SQL query used as a connector between the CRC System and the CRC Datamart DB.
Web Service	Request Connector	SOAP or REST request used to communicate with the external system.

3.1.1.3 Relations and their Properties

The relation of this C&C view is *attachment*, dictating how components and connectors are attached to each other. The relations are as shown in the primary presentation section; there are no additional ones.

3.1.1.4 Design Rationale, Constraints

N-tier Architecture

The client-server style depicts the n-tier architecture that separates presentation layer from business logic and data access layer; thus providing for a high degree of portability through the application of the principle of Separation of Concerns.

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 Style.

3.2.1 Decomposition Style

The Decomposition style presents the functionality in terms of manageable work pieces. They can be further decomposed to present higher level of details. The decomposition view identifies modules and breaks them down into sub-modules and so on, until a desired level of granularity is achieved. The “Uses” style shows the relationships between modules and sub-modules. This view is very helpful for implementation, integration and testing the system.

3.2.1.1 Primary Presentation

System	Segment
CRC	Setfinder Manager
	PDO Manager

3.2.1.2 Element Catalog

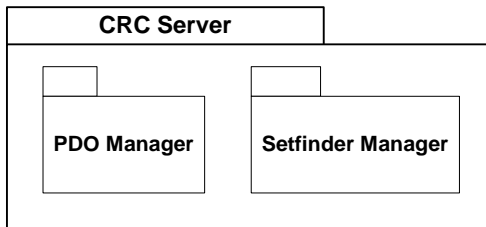
Elements and their properties

Element Name	Type	Description
Setfinder Manager	Subsystem	This subsystem manages user’s Setfinder queries. Keeps track of query information like the query definition, its SQL, owner of the query, etc. Also the results of the query like the patient set, visit set, etc. are stored.
PDO Manager	Subsystem	This manages both plain and table Patient Data object queries.

3.2.1.3 Relations and their Properties

The subsystem elements form the *is-part* of the relation with the overall CRC system.

3.2.1.4 Context Diagram



3.2.2 Uses Style

3.2.2.1 Primary Presentation

System	Segment
CRC	CRC Module
Setfinder Manager Subsystem	Setfinder Web Service
	Setfinder EJB
	Setfinder DAO
	edu.harvard.i2b2.common
PDO Manager Subsystem	PDO Web Service
	PDO EJB
	PDO DAO
	edu.harvard.i2b2.common

3.2.2.2 Element Catalog

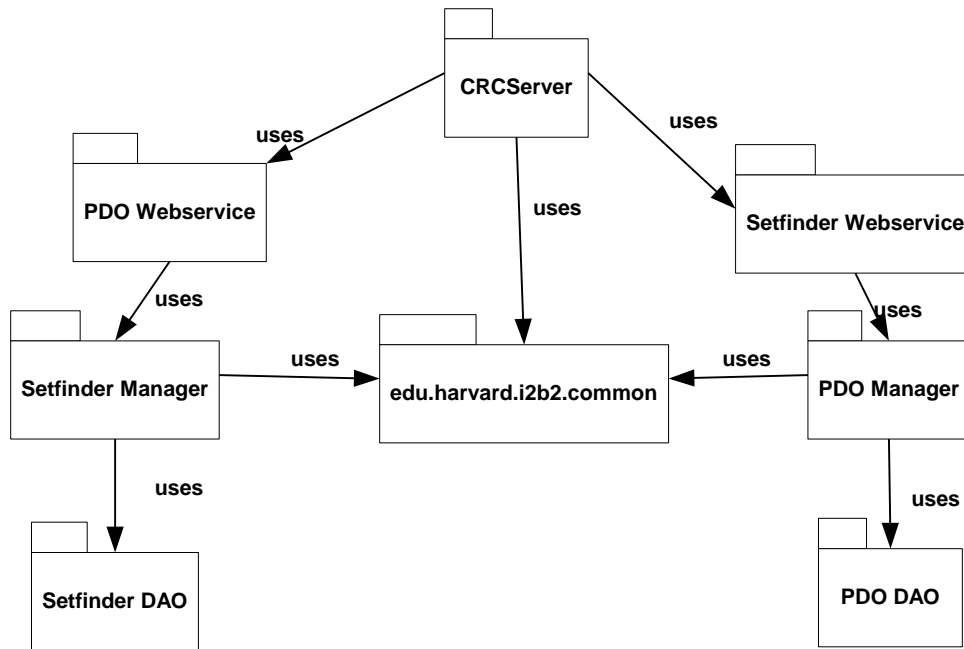
Elements and their Properties

Element Name	Type	Description
CRC Module	Module	User Login Module authenticates through PIN Server System with user id and PIN.
Setfinder Webservice	Module	Provides web service interface to Setfinder operations.
Setfinder EJB	Module	Delegates Setfinder requests to DAO layer to perform database operations.
Setfinder DAO	Module	Supports operations like create query master, delete query, saving query definition and its results.
PDO Webservice	Module	Provides web service interface for PDO requests.
PDO EJB	Module	Module to delegate PDO requests to corresponding PDO and to build PDO response message.
PDO DAO	Module	Module to query database based on PDO requests.
edu.harvard.i2b2.common	Module	This module provides utility classes to handle JAXB, JNDI, etc.
Persistence Service	Module	Provides SQL interface to database.

3.2.2.3 Relation and their Properties

The modules in this style follow a ***depends-on*** relation.

3.2.2.4 Context Diagram



3.3 Mappings of Styles

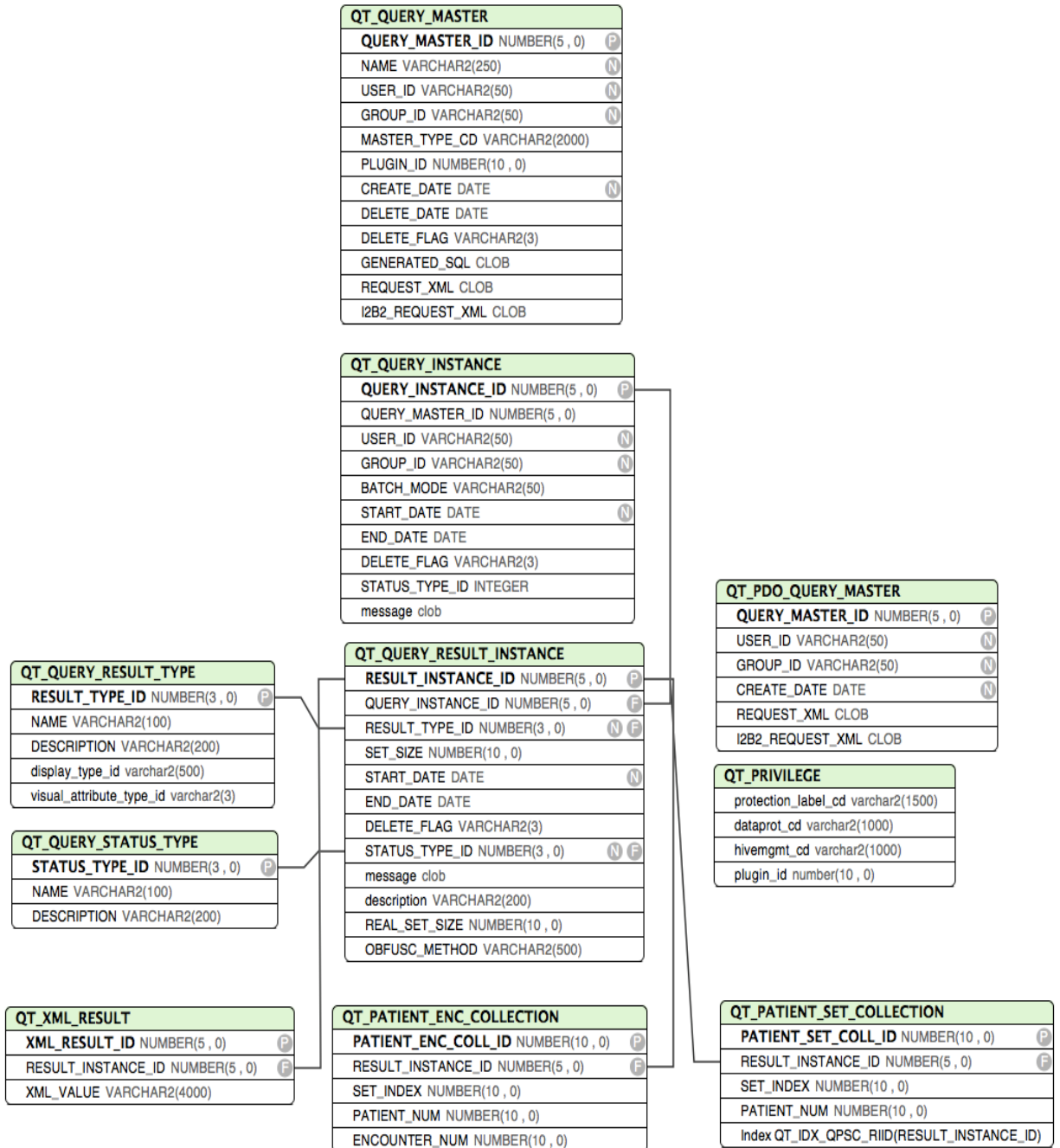
The following table is a mapping between the elements in the *Component & Connector Client-Server* view shown in section 4, and the *Modules Uses* view and *Decomposition* view show in sections 5 and 6.

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.

	CRC Server	PM Server	Ontology Server	CRC Data Mart DB
CRC Service	X	X		
Setfinder Webservice	X			
PDO Webservice	X			
SetFinderEJB	X			
PDOEJB	X		X	
SetFinderDAO	X			X
PDODAO	X			
Persistence Service				X

4 DATA VIEW

The data mart tables are defined in the CRC design doc; below are the query tables in the CRC.



4.1 QT_QUERY_MASTER

The setfinder query definition and the analysis plug-in definition information is stored in the QT_QUERY_MASTER table.

QT_QUERY_MASTER			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	QUERY_MASTER_ID	Unique id for the query	NO
	NAME	Name of the query	NO
	USER_ID	The user's login id	NO
	GROUP_ID	The project id	NO
	MASTER_TYPE_ID	If it is a setfinder query this field will be empty. If it is an Analysis plug-in, then this field will have the name of the plug-in.	YES
	PLUGIN_ID	Not used in the setfinder query	YES
	GENERATED_SQL	The database SQL transformation of the query definition	YES
	REQUEST_XML	The query definition part of the request xml	YES
	I2B2_REQUEST_XML	The full request xml including the i2b2 header	NO

4.2 QT_QUERY_INSTANCE

The information related to the status of the query is stored in the QT_QUERY_INSTANCE table.

QT_QUERY_INSTANCE			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	QUERY_INSTANCE_ID	Unique id for the query	NO
FK	QUERY_MASTER_ID	Many to one reference to the query master table	NO
	USER_ID	The user's login id	NO

	GROUP_ID	The project id	NO
	BATCH_MODE	The name of the queue in which the query is running or last ran	YES
	START_DATE	The date the query was started (run)	YES
	END_DATE	The date the query finished running	YES
	DELETE_FLAG	A flag that denotes whether or not the query was deleted. (Y / N)	NO
FK	STATUS_TYPE_ID	Reference to the QT_STATUS_TYPE table	NO
	MESSAGE	To store the query error message, warning, etc.	YES

4.3 QT_QUERY_RESULT_INSTANCE

The result status for each setfinder query is stored in the QT_QUERY_RESULT_INSTANCE table

QT_QUERY_RESULT_INSTANCE			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	RESULT_INSTANCE_ID	Unique id for the query	NO
FK	QUERY_INSTANCE_ID	Foreign key reference to the QT_QUERY_INSTANCE table	NO
FK	RESULT_TYPE_ID	Foreign key reference to the QT_QUERY_RESULT_TYPE table	NO
	SET_SIZE	The size of the results; the size of the patient set or encounter set. The value is based on the user role; an obfuscated value may be stored in this column.	YES
	START_DATE	The timestamp for the start of each query result generated.	NO
	END_DATE	The timestamp for the end of each query result generated.	YES
	DELETE_FLAG	A flag that denotes whether or not the query was deleted. (Y / N)	NO
FK	STATUS_TYPE_ID	Reference to the QT_STATUS_TYPE table	NO

	MESSAGE	To store the query error message, warning, etc.	YES
	DESCRIPTION	The description displayed to the user in the client (i2b2 Web client and Workbench)	YES
	REAL_SET_SIZE	The real set size. Unlike the SET_SIZE this column does NOT store obfuscated results.	YES

4.4 QT_QUERY_RESULT_TYPE

The metadata for the query result types are stored in the QT_QUERY_RESULT_TYPE table.

QT_QUERY_RESULT_TYPE			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	RESULT_TYPE_ID	Unique id for the result type	NO
	NAME	The name of the result type	NO
PK	DESCRIPTION	The description for the result type. This description is displayed to the user in the client (i2b2 Web client and Workbench)	NO
PK	DISPLAY_TYPE_ID	Information to UI as how to handle the result data. The values are: LIST CATNUM	NO
PK	VISUAL_ATTRIBUTE_TYPE_ID	Defines whether or not the result type is visible to the user. The values are: LA = Visible LH = Hidden	NO

4.5 QT_QUERY_STATUS_TYPE

The metadata for the query statuses are stored in the QT_QUERY_STATUS table.

QT_QUERY_STATUS_TYPE			
----------------------	--	--	--

Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	STATUS_TYPE_ID	Unique id for the status Type	NO
	NAME	The name of the status type	NO
PK	DESCRIPTION	A description of the query status type	YES

4.6 QT_PATIENT_SET_COLLECTION

This is one of the result tables for the query and it captures the patient set.

QT_PATIENT_SET_COLLECTION			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	PATIENT_SET_COLL_ID	Unique id for the patient set	NO
FK	RESULT_INSTANCE_ID	Foreign key reference to the QT_QUERY_RESULT_INSTANCE table	NO
	SET_INDEX	The set index for each patient number	NO
	PATIENT_NUM	The patient number	NO

4.7 QT_XML_RESULT

If the query result is in free form then the user has the choice to store the result in an xml format.

QT_XML_RESULT			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	XML_RESULT_ID	Unique id for the xml result	NO
	RESULT_INSTANCE_ID	Foreign key reference to QT_QUERY_RESULT_INSTANCE table	NO

PK	XML_VALUE	The result in an XML format	NO
----	-----------	-----------------------------	----

4.8 QT_PATIENT_ENC_COLLECTION

This is one of the result tables and it holds the patient's encounter set information.

QT_PATIENT_ENC_COLLECTION			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	PATIENT_ENC_COLL_ID	Unique id for the patient encounter set	NO
FK	RESULT_INSTANCE_ID	Foreign key reference to the QT_QUERY_RESULT_INSTANCE table	NO
	SET_INDEX	The set index for each patient number	NO
	PATIENT_NUM	The patient number	NO
	ENCOUNTER_NUM	The encounter number for the PATIENT_NUM	NO

4.9 QT_PRIVILEGE

This table holds the minimum user role required in CRC operation like whether to obfuscate the result or whether the user has access to blob field in the fact table, etc.

QT_PRIVILEGE			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	PROTECTION_LABEL_CD	Unique label used in the CRC to check for data access based on user role	NO
	DATAPROT_CD	The Data Track's minimum role	NO
	HIVEMGMT_CD	The Management track's minimum role	NO
	PLUGIN_ID	The Analysis plug-in id value if the user privilege is used in the analysis plugin	YES

4.10 QT_PDO_QUERY_MASTER

The QT_PDO_QUERY_MASTER table is used as an audit table for the PDO query. The PDO request information will be stored in this table.

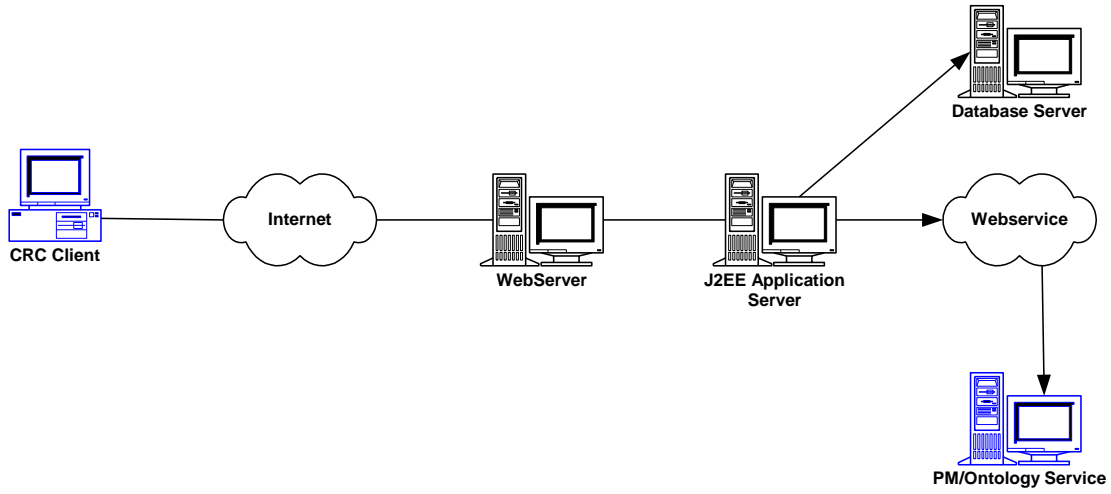
QT_PDO_QUERY_MASTER			
Key	Column Name	Column Definition	Allow Nulls? (Default = YES)
PK	QUERY_MASTER_ID	Unique id for the patient encounter set	NO
	USER_ID	The user's login id	NO
	GROUP_ID	The project id of the user	NO
	CREATE_DATE	The timestamp of when the PDO query was created	NO
	REQUEST_XML	The query definition part of the PDO request	YES
	I2B2_REQUEST_XML	The full PDO request including the i2b2 header	NO

4.11 Volumes

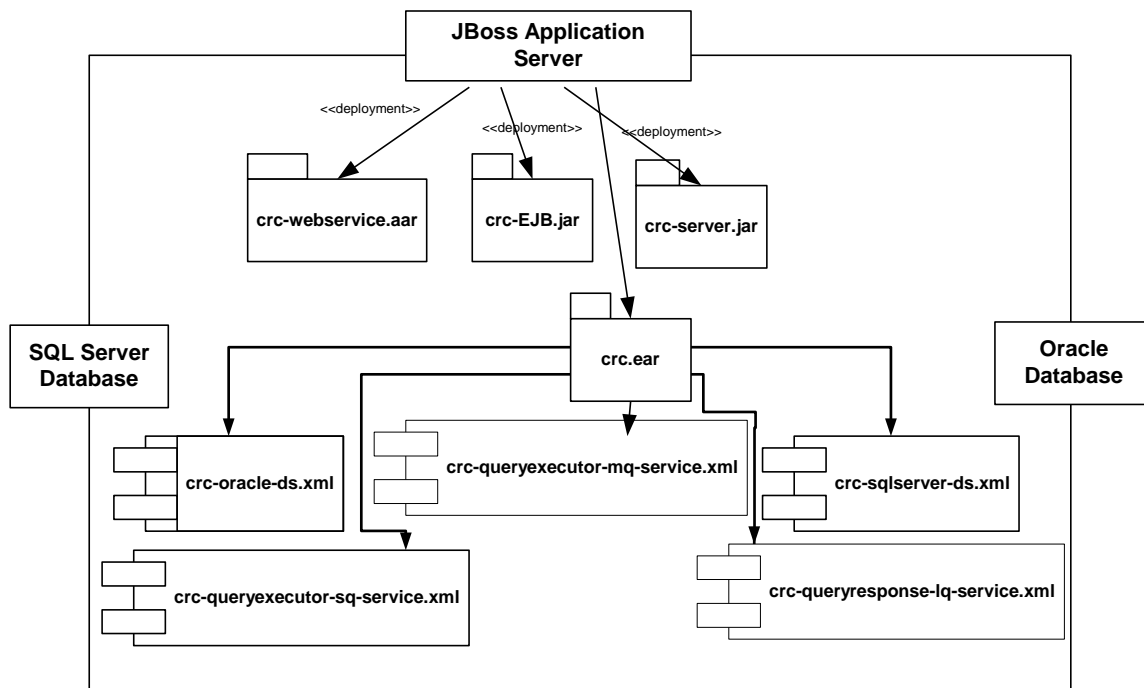
- Estimated new setfinder query: 100 a day, with peaks in the morning
- Average Patient set size 100,000
- CRC registered individual user: about 150

5 DEPLOYMENT VIEW

5.1 Global Overview



5.2 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>