

---

# **Software Architecture Document**

## **Pulmonary Function Test Processing Cell 1.0**

---

**Abstract:**

This is a software architecture document for Pulmonary Function Test Processing (PFT) 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 PFT message format.

**Revision History**

<b>Revision Number</b>	<b>Date</b>	<b>Author</b>	<b>Description</b>
1.0	9/03/07	Lori Phillips	Version 1.0

---

## Table of Contents

Revision History .....	2
1. Overview .....	4
1.1. PFT Definitions, Acronyms and Abbreviations .....	4
1.1.1 Patient Data Object (PDO): .....	4
1.2 Roles .....	4
1.3 Security .....	4
1.4 Scope of the system .....	4
1.5 Assumptions/Constraints .....	4
1.6 Technical Platform .....	5
1.6.1 Transaction .....	5
1.6.2 Security .....	5
1.6.3 Persistence .....	5
1.6.4 Reliability/Availability .....	5
1.6.5 Performance .....	5
2. Use Case .....	6
2.1 Operations .....	6
3. Architecture Description .....	7
3.1 Components and Connector View .....	7
3.1.1. Client-Server Style .....	8
3.2 Module View type .....	9
3.2.1 Decomposition Style .....	9
3.2.2 Uses Style .....	10
3.3 Mappings of Styles .....	11
4. Data View .....	12
4.1 Data Elements .....	12
4.2 Schemas .....	<b>Error! Bookmark not defined.</b>
5. Deployment View .....	13
5.1 Global Overview .....	13
5.2 Detailed deployment model .....	13
References .....	14

---

## **1. Overview**

The Pulmonary Function Test (PFT) Processing cell is an optional i2b2 Hive cell. The Pulmonary Function Test Processing Cell parses a pulmonary function test report and extracts embedded test values. The test values returned may be deposited into a data repository and used to create patient set queries or other types of analysis. This cell works as part of the i2b2 hive for this specific localized purpose.

### **1.1. PFT Definitions, Acronyms and Abbreviations**

#### **1.1.1 Patient Data Object (PDO):**

This object holds patient information such as clinical observations, demographics and provider data.

### **1.2 Roles**

The PFT system is not role-limited. Any participant (USER, MANAGER, ADMIN, etc) may submit a report to the PFT cell for processing.

### **1.3 Security**

There are no security requirements for this cell. As such, no user-id / password combination is required to submit a report to the PFT cell for processing.

### **1.4 Scope of the system**

The PFT cell is self-contained; no other cells are required to perform this function.

### **1.5 Assumptions/Constraints**

- The PFT report to be processed is in a pre-determined format.

---

## **1.6 Technical Platform**

The technology used to build the product is as follows

- Java 2 Standard Edition 5.0 version 11
- Xerces2 XML parser
- Axis2 v1.1 web service (SOAP/REST messaging)

### **1.6.1 Transaction**

There are no transactional elements within this cell.

### **1.6.2 Security**

There are no security requirements for this cell.

### **1.6.3 Persistence**

This cell does not retrieve or post 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

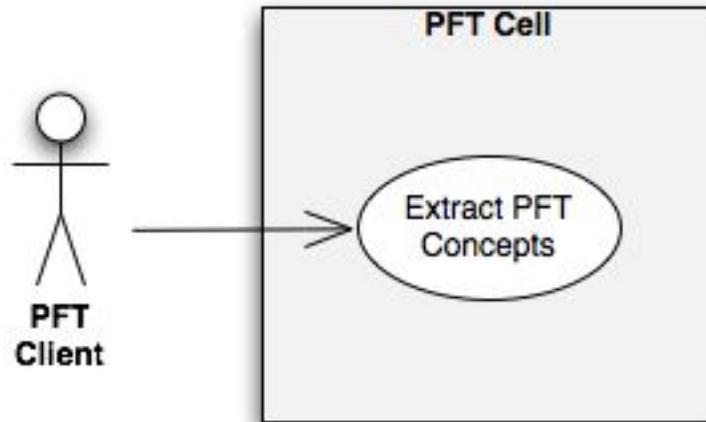
### **1.6.5 Performance**

There are no performance requirements for this cell.

---

## 2. Use Case

The diagram below depicts a common use case a user may perform with the PFT cell.



### 2.1 Operations

The PFT Cell provides services that support the extraction of pulmonary function data from pulmonary function test reports.

**getPulmonaryData:** returns a list of pulmonary function concept data contained within a pulmonary function report

---

## 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

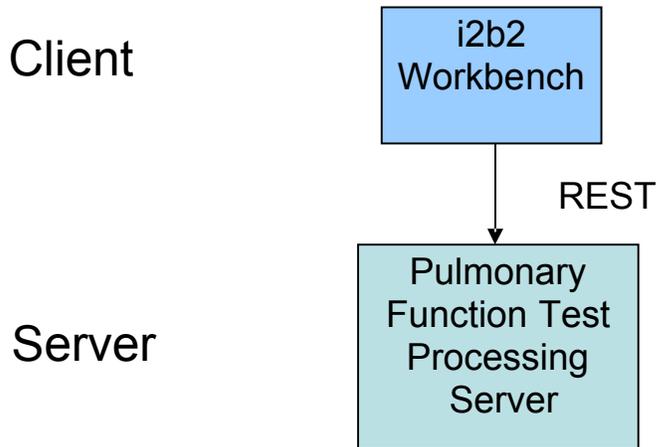
A Components and Connector view represents the runtime instances and the protocols of connection between the instances. The connectors represent the properties such as concurrency, protocols and information flows. Following diagram represents the Components and Connector view for the multi-user installation.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

---

### 3.1.1. Client-Server Style

The PFT system is represented using the C&C 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 PFT Server components and renders response XML.
Pulmonary Function Test Processing Server	Server Component	Provides Web Service Interface for the PFT system. It supports the REST protocol.
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.

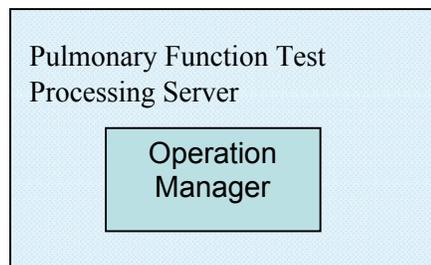
#### 3.2.1.1 Primary Presentation

System	Segment
Pulmonary Function Test Processing Server	Operation Manager

#### 3.2.1.2 Element Catalog

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

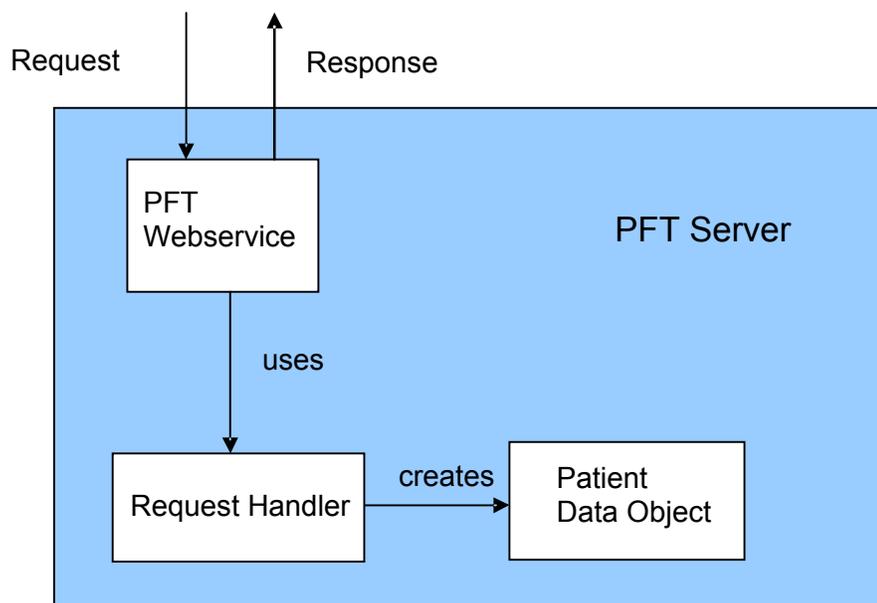
#### 3.2.2.1 Primary Presentation

System	Segment
Pulmonary Function Test Processing Server	PFT Module
Operation Manager Subsystem	PFT Webservice
	Request Handler
	Patient Data Object

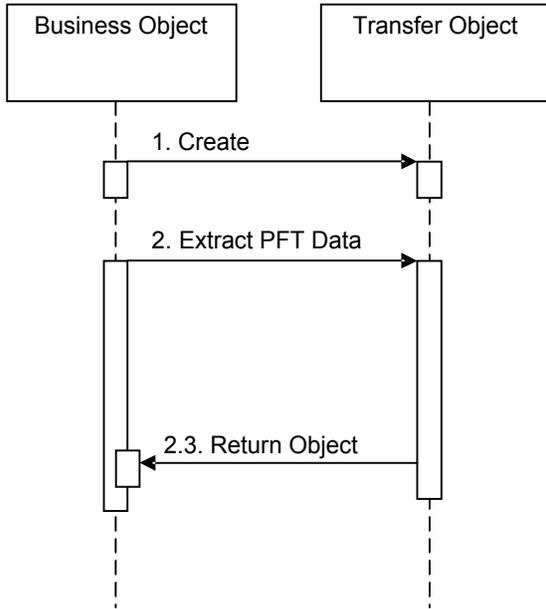
#### 3.2.2.2 Element Catalog

Element Name	Type	Description
PFT Module	Module	Manages PFT processing thread
PFT Webservice	Communication Module	Provides web service interface to PFT operations.
Request Handler	Business Object	Performs PFT request operation
Patient Data Object	Transfer Object	Object representation of PFT 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 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.

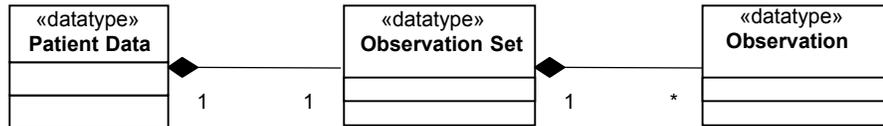
	PFT Server
PFT Service	X
PFT Webservice	X
Request Handler	X
Patient Data Object	X

---

## 4. Data View

### 4.1 Data Elements

The key data elements related to the PFT system are:

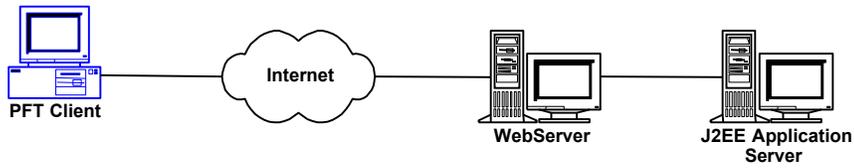


A Patient Data Object (PDO) is the primary means of communication between a client requesting PFT services and the PFT server. For requests, the PDO contains the report to be processed within the observation passed in the PDO. Responses contain a collection of observations each containing a concept extracted from the document.

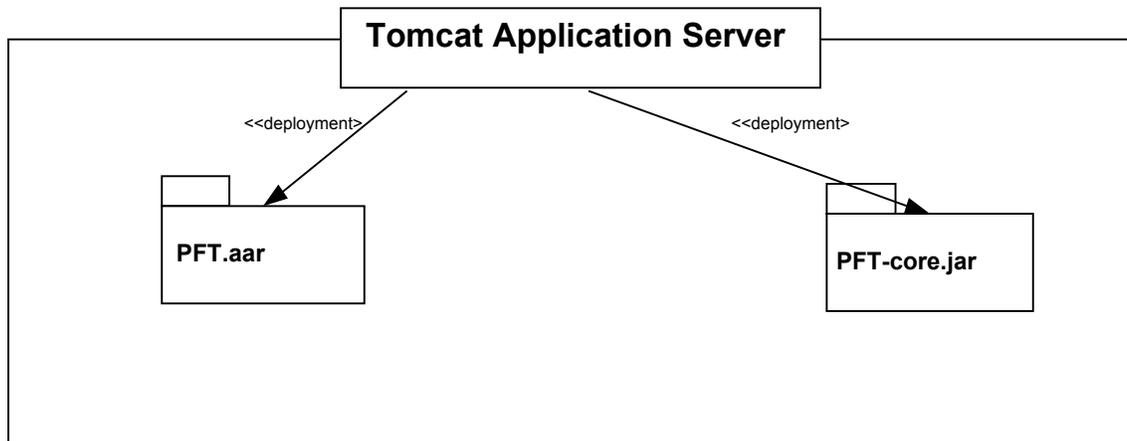
---

## 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., (2003). Documenting Software architectures – Views and Beyond. Addison Wesley, Boston, MA.

The “4+1” view model of software architecture, Philippe Kruchten, November 1995,  
<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

Object Management Group UML 2.0 Specification -  
<http://www.omg.org/technology/documents/formal/uml.htm>

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

---