

Implementation Plan - caAERS

Table of Contents	
• Document History	
• Introduction	
• Technical Environment	
• Overview	
• Development Platform	
• Libraries and Frameworks	
• Databases	
• Web Browsers	
• Deployment	
• Systems Integration	

Document History

Version	Contributor	Description
Version 0.1	Denis Krylov	Initial Draft
Version 0.1.1	Paul Baumgartner	Conversion to wiki

Introduction

The purpose of this document is to capture and convey the technical and architectural decisions that were made to satisfy the functional and non-function requirements of the caAERS system. This document is complementary to documents created by the Architecture Stream. Only technology decisions about components that have been implemented (at least to some extent) will be described here.

Technical Environment

Overview

caAERS consists of a Java Web application based on the Java EE Servlets/JSP infrastructure, WSRF-compliant Web services built on top of caGrid, JBI components hosted by Apache ServiceMix container, and numerous third-party open-source frameworks and libraries. The following sections describe the components in details.

Development Platform

Java Standard Edition

caAERS uses Oracle's Java SE Development Kit version 1.5.0_22 for development and has been tested only with the standard Sun's JVM implementation included in the Kit. caAERS is expected, but not guaranteed to work properly with JVM implementations from other vendors.

Java Enterprise Edition

caAERS uses Oracle's Java Enterprise Edition version 1.4. The usage is limited to Web application development technologies, primarily Servlets 2.4 and JSP 2.0. caAERS does not utilize EJB.

Apache Ant

Apache Ant version 1.7.1 is used as a primary build tool for the project.

Apache Ivy

Apache Ivy is a dependency manager used in tandem with Apache Ant to automatically resolve and download dependencies required to build the application.

Version Control

Subversion 1.5 is used as a version control system.

Libraries and Frameworks

The following sections list main third-party frameworks and libraries used in the project.

Client-Side UI

Component	Version	Purpose
jQuery	1.3.2	General purpose JavaScript programming framework to speed up the development and ensure portability across different browsers. Also enables UI animation, drag-n-drop and other effects.
Prototype	1.6.0	General purpose JavaScript programming framework to speed up the development and ensure portability across different browsers. Need to clarify why two interchangeable frameworks (jQuery & Prototype) are used together. This might come up as a question during the review.
Scriptaculous	1.8.1	JavaScript animation framework.
DWR	2.0.3	Easy integration between server-side Java code and client-side JavaScript.
Extreme Components	1.0.1	JavaScript library to render dropdown lists.
YUI	2.7.0	UI Widget JavaScript library used to render and display data tables.

Server-Side UI

Component	Version	Purpose
Spring MVC	2.0.6	Java-based Web framework (part of Spring) for rendering HTML, handling user requests, providing input data validation.
JSTL	1.1.2	A collection of Java-based custom tags used to render HTML.
SiteMesh	2.2.1	Web page layout and decoration framework used to render HTML.

Security

Component	Version	Purpose
Acegi Security	1.0.3	Java-based security framework used to secure the application on both URL and method levels.

Middle Tier

There is a bunch of libraries used from caGrid, CTMS, CSM and other modules. I am not listing those here since, according to my understand, they are not really 3rd party libraries. Please confirm, thx.

Component	Version	Purpose
Spring Framework	2.0.6	Layered lightweight Java/J2EE application platform used to centralize and automate configuration, provide transaction management, AOP functionality, and integrate with ORM solutions.
Spring Modules	0.9	Integration with JackRabbit and jBPM
Jackrabbit	1.3	Implementation of the Content Repository for Java Technology API used to enable business rule processing in caAERS.
jBPM	3.2.3	Workflow management system used to execute caAERS business processes described in process definition language (jPDL).
Quartz	1.6.0	Job scheduling service used to schedule adverse event report notifications.
Drools	4.0.6	Java-based rule engine to enable business rule processing in caAERS.
xFire	1.2.6	Web Services engine used to expose some of caAERS business functions, such as Subject Import, as Web services to external applications.

Additionally, there multiple libraries used from caGrid, CTMS, CSM and other modules. These libraries are not listed here as they are not considered party libraries.

Back-End

Component	Version	Purpose
Hibernate	3.2.1	Object-relational mapping framework used to provide access to the database.

Miscellaneous

Component	Version	Purpose
Apache Commons	N/A	A collection of general-purpose reusable Java components to speed up and simplify the development: collections, connection pool, I/O, and other.
EhCache	1.2.3	Java framework for caching.
Log4J	1.2.13	Java framework for logging.
Apache POI	3.0	Java library used to produce Excel worksheets.
Apache FOP	0.93	Java library used to produce PDF copies of reports.

Testing

Component	Version	Purpose
Easy Mock	2.2	Java-based framework to support unit testing of Java classes.
JUnit	3.8.2	Automated unit testing framework.
Selenium Server	1.0.b1	JavaScript-based automated UI testing framework.
PMD	4.2.5	Static code analysis tool for Java.
Cobertura	1.8	Test coverage measurement tool for Java.

Databases

caAERS has been built with database-independence in mind, but tested only with PostgreSQL (version 8 or above) and Oracle (version 9i or above). The development team mainly uses PostgreSQL in day-to-day development, and additional QA is performed with Oracle as well.

Web Browsers

caAERS Web application supports the following Web browsers:

- Internet Explorer 6.0 or above.
- Mozilla FireFox 2.0 or above.

Deployment

Operating System

caAERS will run on any operating system that provides full Java 5 support. The list includes Windows 7, Windows Vista, Windows XP, Linux 32/64 bit, and MacOS. For complete details, please refer to <http://java.sun.com/j2se/1.5.0/system-configurations.html>.

Hardware

caAERS will run on any hardware that supports any of operating systems listed above. For complete details, please refer to <http://java.sun.com/j2se/1.5.0/system-configurations.html>.

Middleware

caAERS runs in Apache Tomcat version 5.5.17 or higher. Compatibility with other application servers is expected, but has not been explicitly verified.

Systems Integration

caAERS integrates with other systems through an Enterprise Service Bus. We are using the Apache ServiceMix ESB, which implements the Java

Business Integration (JBI) standard. The main components of a JBI-compliant ESB include binding components (BCs), service engines (SEs), and the normalized message router (NMR). Using ESB, caAERS exchanges information with other applications of the Suite and submits adverse event reports to AdEERS.