# BEA AquaLogic BPM Suite
# Version 6.0 MP4 (Build 95902)
# Security Target

Version 1.0

31 March 2009

**Prepared for:**
Oracle, Inc.

100 Oracle Parkway
Redwood Shores, CA

**Prepared By:**
Science Applications International Corporation

**Common Criteria Testing Laboratory**

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization.

The TOE is Version 6.0 Maintenance Pack (MP) 4 (MP4) (Build 95902) of the BEA AquaLogic BPM Suite. The TOE is a product suite for business process management (BPM), or creating, executing, and optimizing business processes. It enables collaboration, business, and information technology (IT) to automate and optimize business processes. The TOE provides security functions that control user access to business process definitions and active instances of those processes. Users may be granted or denied access based on their organization affiliation, assigned group, assigned role, assigned groups, or identity, which is verified by the TOE. In addition, the TOE provides functions to securely manage BPM objects and process participants.

The Security Target contains the following sections:

Section 1 **Security Target Introduction**
This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization.

Section 2 **Target of Evaluation (TOE) Description**
This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.

Section 3 **Security Environment**
This section details the expectations of the environment, the threats that are countered by TOE and IT environment, and the organizational policy that TOE must fulfill.

Section 4 **Security Objectives**
This section details the security objectives of the TOE and IT environment.

Section 5 **IT Security Requirements**
The section presents the Security Functional Requirements (SFR) for TOE and IT Environment that supports the TOE, and details the assurance requirements for EAL2.

Section 6 **TOE Summary Specification**
The section describes the TOE security functions represented that satisfy the security requirements.

Section 7 **Protection Profile Claims**
This section presents any protection profile claims.

Section 8 **Rationale**
This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness, and suitability.

## 1.1 Security Target, TOE and CC Identification

**ST Title –** BEA AquaLogic BPM Suite Version 6.0 MP4 Security Target

**ST Version** – Version 1.0

**ST Date** – 31 March 2009

**TOE Identification** – BEA AquaLogic BPM Suite Version 6.0 MP4 (Build 95902)

**CC Identification** – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

## 1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, CCMB-2005-08-002, August 2005.
    - Part 2 Conformant

- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, CCMB-2005-08-003, August 2005.
    - Part 3 Conformant
    - Assurance Level: EAL2 augmented with ALC_FLR.1
- This TOE is conformant to the following Protection Profile (PP):
    - No PP claim is made for this ST.

## 1.3 Conventions, Acronyms, and Terminology

This section specifies the formatting conventions used in the Security Target and provides a glossary of acronyms.

### 1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, a letter placed at the end of the component indicates iteration. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by bold brackets (e.g., **[assignment]**).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by bold brackets (e.g., **[*selection*]**). An assignment inside a selection is indicated using bold italics surrounded by bold italics brackets surrounded by bold brackets (e.g., **[*[selection]*]**)
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Explicitly stated Security Functional Requirements (i.e., those not found in Part 2 of the CC) are identified with "**(EXP)**".
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

### 1.3.2 Acronyms

| | |
|---|---|
| **ACM** | CM (SAR class) |
| **ADO** | Delivery and Operation (SAR class) |
| **ADV** | Development/Design (SAR class) |
| **AFL** | Authentication Failure family of FIA |
| **AGD** | Guidance documents (SAR class) |
| **ALBPM** | AquaLogic Business Process Management |
| **ALC** | Life-cycle Support (SAR class) |
| **ALUI** | AquaLogic User Interaction |
| **ANL** | Analyzer Analysis family of IDS |
| **ASE** | ST (SAR class) |
| **ATD** | User Attribute Definition family of FIA |
| **ATE** | Tests (SAR class) |
| **AVA** | Vulnerability Assessment (SAR class) |
| **BAM** | Business Activity Monitoring |
| **BPEL** | Business Process Execution Language |
| **BPM** | Business Process Management |

| | |
|---|---|
| **CC** | Common Criteria |
| **CCTL** | CC Testing Laboratory |
| **CEM** | Common Evaluation Methodology |
| **CCEVS** | CC Evaluation and Validation Scheme |
| **CI** | Configuration Item |
| **CM** | Configuration Management |
| **CMP** | CM Plan |
| **CVE** | Common Vulnerabilities and Exposures |
| **EAL** | Evaluation Assurance Level |
| **EAR** | Enterprise Application aRchive |
| **FAU** | Security Audit (SFR class) |
| **FIA** | Identification and Authentication (SFR class) |
| **FMT** | Security Management (SFR class) |
| **FPT** | Protection of the TOE Security Functions (SFR class) |
| **FSP** | Functional Specification |
| **GEN** | Security Audit Data Generation family of FAU |
| **GUI** | Graphical User Interface |
| **HLD** | High-level Design |
| **HTTP** | Hyper-text Transfer Protocol |
| **HTTPS** | Secure HTTP |
| **ID** | Identity/Identification |
| **IIOP** | Internet Inter-Orb Protocol |
| **IP** | Internet Protocol |
| **IT** | Information Technology |
| **ITT** | Internal TOE TSF Data Transfer family of FPT |
| **JDBC** | Java Database Connectivity |
| **JNDI** | Java Naming and Directory Interface |
| **JVM** | Java virtual machine |
| **MOF** | Management of Functions family of FMT |
| **MTD** | Management of TSF Data family of FMT |
| **NIAP** | National Information Assurance Partnership |
| **NIST** | National Institute of Standards and Technology |
| **OS** | Operating System |
| **PAPI** | Process Application Programming Interface |
| **PAPI-WS** | Process Application Programming Interface Web Services |
| **PKI** | Public Key Infrastructure |
| **PP** | Protection Profile |
| **SAIC** | Science Applications International Corporation |
| **SAR** | Security Assurance Requirement |
| **SFR** | Security Functional Requirement |
| **SSL** | Secure Socket Layer |
| **SMF** | Specification of Management Functions family of FMT |
| **SMR** | Security Management Roles family of FMT |
| **SOA** | Service oriented architecture |
| **SOF** | Strength of Function |
| **ST** | Security Target |
| **STG** | Security Audit Event Storage family of FAU |
| **TCP** | Transmission Control Protocol |
| **TOE** | Target of Evaluation |
| **TSC** | TOE scope of control |
| **TSF** | TOE Security Functions |
| **TSS** | TOE Summary Specification |
| **UAU** | User Authentication family of FIA |
| **UID** | User Identification family of FIA |
| **US** | United States |
| **XML** | Extensible Markup Language |

### 1.3.3 Terminology

| Term | Definition |
|---|---|
| **Activity** | Defines a manual or automated task that performs one step within a process design. |
| **Activity Instance** | The representation of an activity within a (single) enactment of a process, i.e. within a process instance. |
| **Application server** | A software engine that delivers applications to client computers, which typically handles most, if not all, of the business logic and data access of the application. |
| **Business process** | An abstract process model that gets realized in a runtime environment by being published and deployed. It represents a specific set of business tasks and activities that must be executed to reach a well-defined outcome. When the goal is reached, the process is complete. Examples of simple processes are: hiring an employee, processing a sales order, or reimbursing a business expense. |
| **PBL** | A high-level scripting language used to define the business rules and the logic of activity types and certain transitions within a process; a script written in PBL and associated with an activity. |
| **Event** | Represent each time the instance entered or exited an activity (simple activity, group, procedure). The Process Execution Engine generates one event per action. |
| **Activity group** | A compound activity. It is composed of a set of activities that may include other Groups. |
| **Interactive activity** | An activity that brings end users into the process (i.e. requires participant intervention). |
| **J2EE** | This term stands for the Java Platform, Enterprise Edition specification. |
| **Java virtual machine** | A virtual machine that executes Java bytecode. |
| **Organizational Group** | Represent a profile. Groups have members associated to them and can be assigned roles and other groups. |
| **Organizational Role** | Represent job functions performed within the organization. |
| **Organizational Unit** | Represent department or division within the organization. |
| **Participant** | A member of an organization that participate in any task within a business process. |
| **Portlet** | A Java-technology-based web component, managed by a portlet container that processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems. |
| **Process instance** | A specific item proceeding through a business process. For example, in a business process that handles purchases, each individual purchase order is a business process instance. Any number of instances can traverse a business process at any given time. |
| **Publish and Deploy** | Activates the process design into a real-time situation where the activities and roles can be fulfilled automatically or by human users. |
| **Servlet container** | Comprises essentially the component of an application server that hosts and interacts with Java servlets. A servlet container controls the servlets that are deployed within the Web Server and is responsible for forwarding the requests and responses for them. It has the functionality of mapping a URL to a particular servlet and of ensuring that the process requesting the URL has the correct access rights. |
| **Supervisory application** | A business process that has been published and desployed. A supervisory application is a business process that has been transformed into a (runtime) executable set of Java classes. Each time the deployed process is invoked, a process instance is associated with the supervisory application to maintain the state of the process for that invocation. A supervisory application can be thought of as the template from which process instances are created and executed. |
| **Swing (or Swing application)** | Swing is a GUI toolkit for Java and is one part of the Java Foundation Classes (JFC). Swing includes graphical user interface (GUI) widgets such as text boxes, buttons, split-panes, and tables. |
| **Task** | Consists of one or more actions that need to be executed in order to achieve an activity's goal. |
| **Work Item** | The representation of the work to be processed (by a workflow participant) in the context of an activity within a process instance. (See “Task.”) |

| Term | Definition |
|---|---|
| **Worklist** | A list of work items associated with a role or participant. |
| **Worklist item portlet** | A portlet that displays work items for a particular user/role. Columns can be custom/business-specific. Users can use this portlet to perform actions on one or more of the work items in the list and can also use the portlet to search for particular Work Items. |

# 2. TOE Description

## 2.1 Product Overview

BEA AquaLogic BPM Suite is a complete product suite for creating, executing, and optimizing business processes. It includes two distinct sets of products that are separately installed and licensed:

- BEA AquaLogic BPM Studio (Studio)
- BEA AquaLogic Enterprise Server (Enterprise Server)

The two product sets correspond to two distinct phases of the business process management cycle:

- Design time
- Runtime

**Design Time:**

During **design time**, business analysts, and business architects use Studio to design and run simulations of a complete process without involving IT. When the process fulfills the business specifications, they hand it over to developers to implement the necessary connectivity to existing IT systems. User interfaces for human interaction with the process are generated automatically and provided as standards-based portlets or web applications.

The suite thus enables collaboration between business and IT to automate and optimize business processes. The projects jointly created by the business architects and analysts on the one side and developers on the other serve as the contracts between them regarding the requirements of the process.

When a project is complete, the developer exports it. This involves compressing XML files containing all the project information into a project file and saving it in a specified location.

**Runtime:**

During **runtime**, an administrator uses the components of the Enterprise Server to import the project file, and then to publish, deploy, and administer it in an enterprise. During that process, an administrator can choose whether or not to include in the runtime version the information about participants, organizational units, roles, and auditing as specified by the developer in the imported project.

The Enterprise Server is available both as a standalone and as part of an application server configuration using either BEA WebLogic Server or IBM WebSphere.

## 2.2 Components and Interfaces of BEA AquaLogic BPM Suite

This section describes the components and external interfaces of BEA AquaLogic BPM Suite. It first discusses the design-time component, and then the runtime components.

### 2.2.1 Design-Time Component

*BEA AquaLogic BPM Studio:* An environment for both designing and developing a process. It has an external interface that can be used by business analysts, business architects, and developers. The interface changes depending on which role the user chooses.

- Business analysts and architects can design and simulate a process inside Studio without writing any code or having to work with IT. When the process is considered complete from a business perspective, they hand it over to IT for completion and deployment.
- Once the developer receives a process completed by the business analysts and architects, he or she writes business logic, connects to existing applications, and assembles user interfaces for human

interaction. No web design or coding is necessary. Studio automatically generates the necessary web components based on the interaction and message formats specified in the process.

Studio is a Java application, based on the Eclipse open platform.

### 2.2.2 Runtime Components

*BEA AquaLogic BPM Process Execution Engine (Process Execution Engine):* Orchestrates all processes and their resources—people, organizations, applications, and systems—managing proper sequence, enforcing business rules, and auditing each step to ensure process execution, escalation, and exception management. The Process Execution Engine executes processes designed in Studio as well as processes written in BPEL 2.0, the Business Process Execution Language. The Process Execution Engine runs either as a standalone Java application or alternatively as a J2EE Web Application in a variety of application containers. Optional application containers include a pure Java virtual machine (JVM) as well as major application servers and open source containers such as WebLogic Server and IBM WebSphere.

*BEA AquaLogic BPM WorkSpace (WorkSpace):* The component with an external interface through which a participant interacts with the Process Execution Engine. Process activities that require human interaction are automatically surfaced into a web interface without any need for manual web page design or coding. Participants can access and manipulate tasks according to their assigned roles, permissions, and organizational units.

WorkSpace can be integrated with other (non-TOE) products such as the AquaLogic User Interaction (ALUI) suite. The resulting product suite would combine portal, collaboration, and BPM technologies into a single system. The integrated product suite supports the capture of ad-hoc, collaborative activities associated with business processes in an integrated environment.

The external interface can be exposed as portlets on ALUI and on any JSF-compliant container like BEA WebLogic Portal. Integration with ALUI enables:

- Using AquaLogic Interaction as the user interface layer for ALBPM and for handling user authentication
- Using AquaLogic Interaction Collaboration as the document repository for attachments

WorkSpace is implemented as a J2EE Web Application.

*Process Application Programming Interface for Web Services (PAPI-WS)*: A web services wrapper that is functional equivalent of PAPI. It enables developers to build custom applications that participants can then use to interact with process activities and workflows and read audit records. It adheres to the WS-Security specification using the UserNameToken Profile implementation as well as HTTP Basic Authentication.

*BEA AquaLogic BPM Process Administrator (Process Administrator):* The console with an external interface that enables administrative management of:

- Organizational information: participants, roles, groups, permissions, categories)
- Engines: add, remove, start, stop
- General configuration information: connectivity information to external databases and systems
- Publish, Deployment, and undeployment of projects

The Process Administrator is a J2EE Web Application.

*BEA AquaLogic BPM Archive Viewer (Archive Viewer):* A web application with an external interface that provides IT administrators and business users with historical activity data for business processes that had previously run in the Process Execution Engine. The Process Execution Engine periodically purges the Process Execution Engine database and moves and aggregates process instance audit information into an archiving database. This database can be queried by the Archive Viewer. The Archive Viewer is implemented as a J2EE web application.

*BEA AquaLogic BPM Log Viewer (Log Viewer):* A Swing application with an external interface that enables administrators to read information logged by the Process Execution Engine. The Process Execution Engine creates a

set of log files following a rotating implementation. Log Viewer reads the files and displays them to help administrators monitor and trace engine execution.

*BEA AquaLogic BPM Admin Center (Admin Center):* a Swing application with an external interface designed to help administer a BEA AquaLogic BPM implementation. The primary focus of Admin Center is to setup and configure the BEA AquaLogic BPM installations.

## 2.3 TOE Overview

The Target of Evaluation (TOE) comprises:

1. BEA AquaLogic BPM Studio (Studio)
2. BEA AquaLogic BPM Process Execution Engine (Engine)
3. BEA AquaLogic BPM WorkSpace (WorkSpace)
4. BEA AquaLogic BPM Process Administrator (Process Administrator)
5. BEA AquaLogic BPM Log Viewer (Log Viewer)
6. BEA AquaLogic BPM Archive Viewer (Archive Viewer)
7. BEA AquaLogic BPM Admin Center (Admin Center)
8. BEA AquaLogic Process Application Programming Interface for Web Services (PAPI-WS)

There are two configurations of the Process Execution Engine TOE component in the evaluated configuration – a stand-alone configuration and an application-server configuration. Figure 1 below shows a high-level view of the TOE components as they are distributed in the physical environment. Note that this view shows the TOE components in only the "standalone" configuration in its other configuration (the "application server" configuration) the Process Execution Engine component is implemented as a Java Enterprise application and is hosted by the same J2EE application server that presents the Workspace, WorkSpace Administrator, Process Administrator, and Archive Viewer components. In the standalone configuration, the Process Execution Engine is implemented as a standalone Java application. In either configuration, the Directory Service can be hosted remotely, in which case communications should be protected by SSL[1].

[1] The IT environment is relied upon to provide the certificates for secure SSL communications

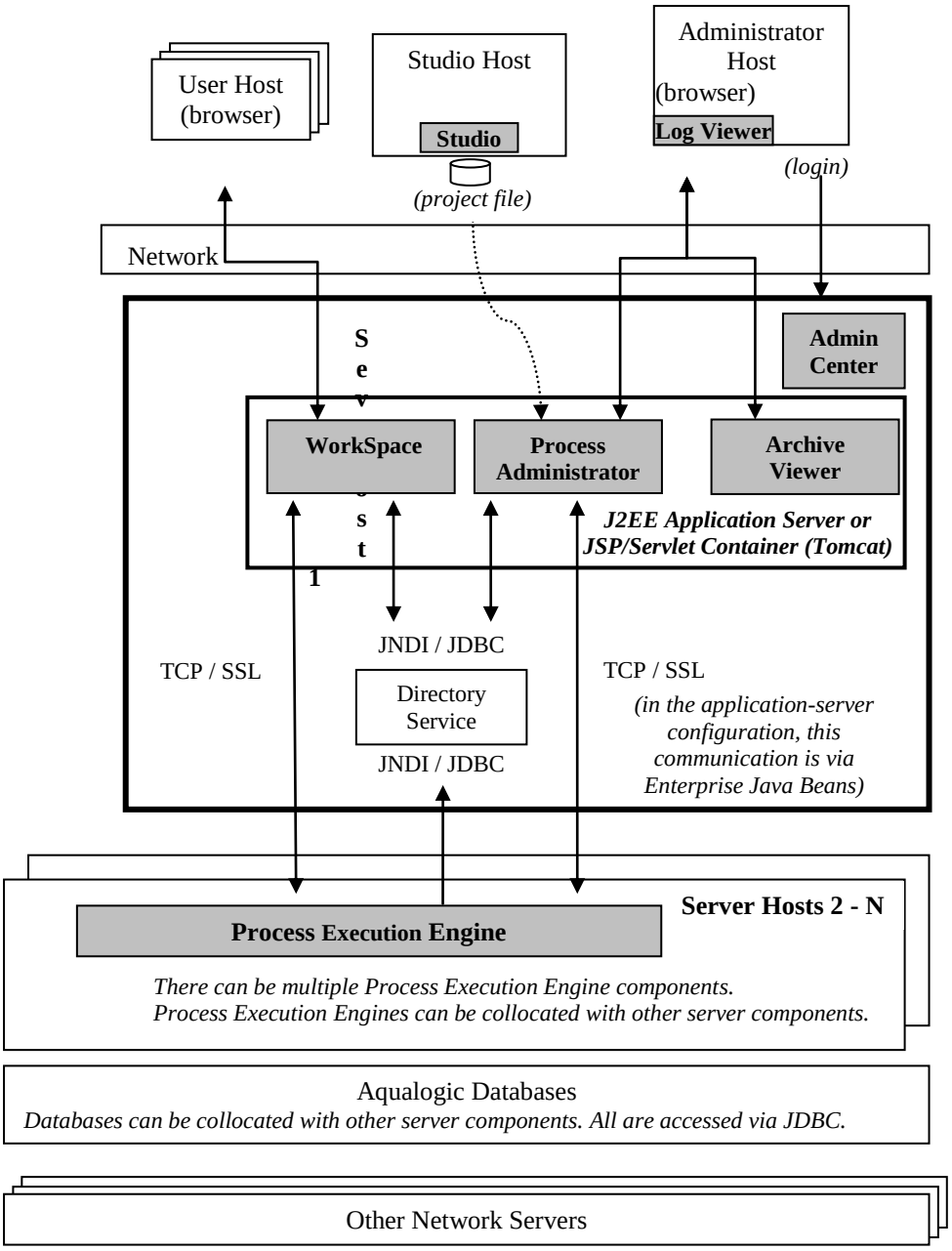



**Figure 1 TOE Components High-level View**

In the stand-alone configuration, the Process Execution Engine component can reside on a separate server host from the auxiliary application server,[2] or they can reside on the same server host. In either case, the Process Execution Engine is a separate Java application that is directly accessible as a service. The application server in the stand-alone configuration provides the service interfaces for access to other TOE components such as the Process Administrator and WorkSpace. In the application-server configuration, an application server provides the service interface for all TOE components that are part of the runtime infrastructure (i.e., including the Process Execution Engine but excluding Studio, Admin Center, and Log Viewer, which are Swing applications).

[2] The "auxiliary" application server is the application server that hosts other runtime components of the TOE, such as the Process Administrator component. It is auxiliary in the sense that it is part of the IT environment and it supports the TOE, but it does not exclusively support the TOE as a dedicated component.

The logical functionality provided by one configuration is identical to that provided by the other configuration. The application server provided in the stand-alone configuration is the Tomcat private distribution, specifically integrated to service the supported TOE components. In this configuration, communications between the TOE components and the Process Execution Engine are external to the applications server, and are protected using TCP/SSL. In the application-server configuration, the application server is one of two supported products: BEA WebLogic or IBM WebSphere. These application servers are capable of hosting other applications simultaneously with the TOE. In this configuration, communications between the TOE components and the Process Execution Engine are internal to the applications server via internal "container" protocols (Enterprise JavaBeans communications). In this configuration, TOE communications are protected from other applications via J2EE file security descriptors, and rely on the J2EE Security Framework to restrict access to TOE-related applications. In addition, the TOE always performs an authorization check for each request it receives, to protect against unauthorized requests.

The IT environment and the TOE both work together to provide transaction protection. A transaction begins when the Process Execution Engine begins the processing of a request. In the application server configuration, the J2EE Container Transaction Manager provides rollback protection in the case where a request does not terminate correctly. In the standalone configuration, the Process Execution Engine has an internal, integrated transaction manager to provide the same protection.

A business *process* is an abstract process model that gets realized in a runtime environment by being published and deployed. It represents a specific set of business tasks and activities that must be executed to reach a well-defined outcome. When the goal is reached, the process is complete. Examples of simple processes are: hiring an employee, processing a sales order, or reimbursing a business expense.

A specific item proceeding through a business process is called a business *process instance*. For example, in a business process that handles purchases, each individual purchase order is a business process instance. Any number of instances can traverse a business process at any given time.

The activities of a process can be automated or interactive. Automated activities occur in the background from the perspective of external (human) subjects. For the purpose of security functionality, the only activities of interest are interactive activities. Unless otherwise noted, the term "activity" in this document refers to an interactive activity.

A business process is designed in Studio. When a process design is complete, the business process is saved in a "project" file. An administrator can retrieve the project file to publish and deploy the process within the Process Execution Engine.

An activity is composed of work items or "tasks." A task can be thought of as a discrete step taken in the processing of an activity. Tasks are typically items of work visible to end users (participants) that are implemented internally within a process instance implementation or externally via a call to an external application. A participant's user interface will present the active tasks that are applicable to that participant.

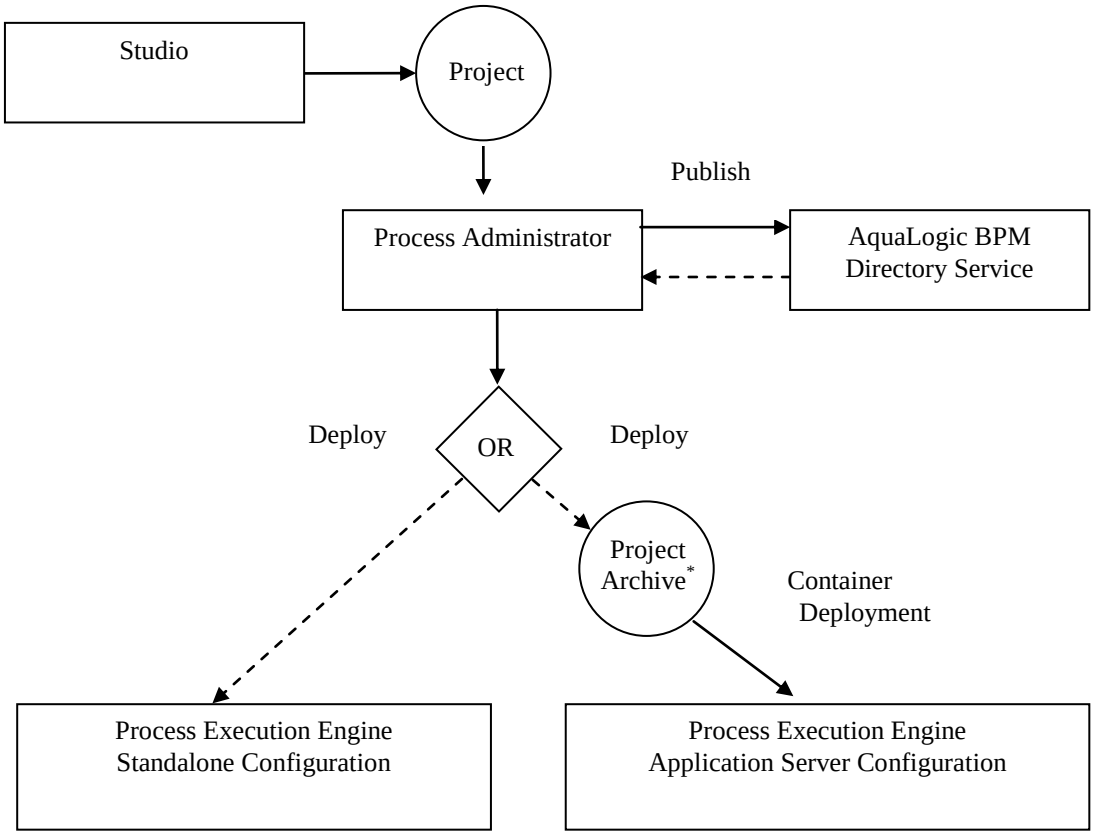
A business-process developer associates abstract roles with the activities of the process during modeling of a business process. Abstract roles are later realized as concrete roles (i.e., relevant to the operational environment) through the publishing and deployment of the process, which are activities the TOE restricts to administrators. The roles associated with the activities of a process carry permissions that define how the activity can be manipulated within the scope of the process.

Publishing a project prepares the process for the run-time environment, where the business process modeling language constructs are compiled to Java classes. Abstract roles are associated with actual roles in the operational system. During deployment, the process instance is associated with a certain Organizational Unit and bound to a specific Process Execution Engine. The Engine is notified that a new process is available to users associated to that Organizational Unit, so that they can begin working with it.

A business process that has been published and deployed is referred to as a "supervisory application." A supervisory application is a business process that has been transformed into a (runtime) executable set of Java classes. Each time the deployed process is invoked, a process instance is associated with the supervisory application to maintain the state of the process for that invocation. A supervisory application can be thought of as the template from which process instances are created and executed.

In addition to the binding of concrete roles, process instances are associated with organizational units at publishing and deployment time. Any given abstract business model (e.g., travel request processing) could have applicability with different organizational units as defined by a company using the TOE. When a process instance is realized in the operational environment, each process instance has to be associated with a specific organizational unit so that the Process Execution Engine can control logically separate process instances separately. Organizational units also bound the scope for role and group identifiers. Groups are simply profiles that allows for a set of roles to be associated with a set of users, easing participant and role management.

Figure 2 shows how TOE components interact in either configuration for designing, developing, and deploying a project to the Process Execution Engine. This diagram demonstrates graphically how the business process definitions are linked to relevant TOE components and IT infrastructure elements through the lifecycle of the process. A project created in Studio can be exported into a project export file. The Process Administrator tool can deploy a project from that project export file.




** The project archive is the Enterprise Application aRchive (EAR)*

**Figure 2. Design, Development, and Deployment**

## 2.4 TOE Architecture

This section describes the TOE physical and logical boundaries.

### 2.4.1 TOE Physical Boundaries

This section describes the platforms supported for each of the TOE components and identifies the evaluated configuration of the TOE for testing purposes.

The configurations described in this section can run on a variety of supporting software and hardware. The full listing of TOE components and compatible support software and hardware is provided in the Compatibility Matrix at http://support.plumtree.com. The tables below identify the compatible support elements for TOE components in its evaluated configuration. Only the BEA AquaLogic BPM components identified in the table below are TOE components.

**Table 1: TOE Component Physical Requirements**

| Configuration | Requirement |
|---|---|
| Studio | Recommended:<br>• 2 GB RAM or more<br>• 4 GB or greater free disk space<br>• 1.8 GHz or faster Pentium Core Duo CPU or similar<br><br>Minimum:<br>• 1 GB RAM<br>• 2.5 GB or greater free disk space<br>• 1.5 GHz Pentium M CPU or similar |
| Enterprise Standalone | Recommended:<br><br>For high-volume deployment (more than 500 concurrent users), consult with ALBPM Professional Services.<br><br>For testing and low volume deployment (up to 500 concurrent users):<br>• 4 GB RAM or more<br>• 5 GB or greater free disk space<br>• 2.0 GHz or faster Pentium Dual-Core Xeon CPU or similar<br><br>Minimum (testing only):<br>• 2 GB RAM<br>• 3 GB or greater free disk space<br>• 1.5 GHz Pentium M CPU or similar |

The BEA WebLogic and IBM WebSphere application servers are not part of the TOE. Rather for the purposes of this evaluation, they are considered as supporting IT infrastructure outside the scope of the evaluation. Supporting database and directory service components are considered part of the IT environment, too. A complete list of acceptable supporting components can be found in the BEA AquaLogic BPM installation guide. However, the definitive list for support components that are applicable to the TOE definition is provided below, in Table 2.

| TOE Component | IT Support Components | Platform(s) |
|---|---|---|
| ALBPM Process Execution Engine Standalone (v6.0) | Operating System(s) | Windows Server 2003 SP1 (x86-32), Linux SUSE 10.0 (x86-32, x86-64, Itanium-64), Linux RHEL 4.x (x86-32, x86-64, Itanium-64), AIX 5.3, Solaris 9 and 10 (SPARC), HP-UX 11.23 (Itanium-64) |
| | Application Server | WorkSpace:<br>Tomcat Servlet 5.5.15, BEA WebLogic Server 9.2.1 and 9.2.2, WebLogic Server 10.0, IBM WebSphere 6.1.0.5;<br><br>Process Administrator:<br>Tomcat Servlet 5.5.15;<br>WorkSpace Administrator:<br>Tomcat Servlet 5.5.15, BEA WebLogic Server 9.2.1 and 9.2.2, WebLogic Server 10.0, IBM WebSphere 6.1.0.5;<br><br>Archive Viewer:<br>Tomcat Servlet 5.5.15, BEA WebLogic Server 9.2.1 and 9.2.2, WebLogic Server 10.0, IBM WebSphere 6.1.0.5 (other patch levels introduced problems);<br><br>PAPI-WS:<br>Tomcat Servlet 5.5.15, BEA WebLogic Server 9.2.1 and 9.2.2, BEA WebLogic Server 10.0, IBM WebSphere 6.1.0.5;<br><br>RSS Feeds: Tomcat Servlet 5.5.15, BEA WebLogic Server 9.2.1 and 9.2.2, BEA WebLogic Server 10.0, IBM WebSphere 6.1.0.5; |
| | Directory Service Database | Oracle 9i, 10g (including RAC), MS SQLServer 2005, IBM DB2 UDB 8.1 and 9.1 using DataDirect Embedded JDBC Drivers (DataDirect 3.6) |
| | JMS Provider | N/A |
| | LDAP Directory Service | Sun ONE System Directory Server 5.2<br><br>MS Active Directory 2003 |
| | Identity Service | AquaLogic Interaction Identity Service (for use with the AquaLogic User Interaction Portal) |
| | Browsers | On Microsoft Windows:<br>• Microsoft Internet Explorer 6.0 and 7.0<br>• Mozilla Firefox 2.0<br><br>On Linux: Mozilla Firefox 2.0<br><br>On Apple Mac OS X:<br>• Apple Safari 3.0<br>• Mozilla Firefox 2.0 |

| TOE Component | IT Support Components | Platform(s) |
|---|---|---|
| | JVM Version for each operating system | Sun 1.5.0_12^ for Windows Server 2003 SP1 (x86-32), SUSE Linux 10.0 (x86-32), Red Hat Enterprise Linux 4 (x86-32), Solaris 9 and 10 (SPARC), JRockit 5 for Red Hat Enterprise Linux 4 (Itanium-64 and x86-64) and Suse 10.0 (Itanium-64 and x86-64), IBM 1.5.0 64 bits for AIX 5.3 running 32 bit JVM, HP-UX 11.23 Itanium-64 using HP-UX JVM 1.5.0_05 |
| Process Execution Engine for WebLogic (v6.0) | Operating System(s) | Microsoft Windows Server 2003 SP1 or R2 (x86-32)<br>Novell SUSE Linux 10.0 (x86-32, x86-64, IA-64)<br>Red Hat Enterprise Linux 4.x (x86-32, x86-64, IA-64)<br>IBM AIX 5.3<br>Sun Solaris 9 and 10 (SPARC)<br>HP-UX 11.23 (IA-64) |
| | Application Server | The following versions are supported for all Enterprise applications except Process Administrator, which runs only in the built-in Tomcat Servlet/JSP Container:<br>BEA WebLogic Server 9.2 (MP1 or MP2)<br>BEA WebLogic Server 10.0. |
| | Engine Database | Oracle 9i, 10g (including RAC), MS SQLServer 2005, IBM DB2 UDB 8.2 and 9.1 using BEA WebLogic Server Embedded DataDirect JDBC Drivers. |
| | JMS Provider | TIBCO EMS 4.1, WebLogic 8.1 Embedded Messaging and WebLogic 9.2 Embedded. (XA Compliant Resources) |
| | Directory Service Database | Single Source JDBC Plugins: Oracle 9i and 10g, MS SQLServer 2005, IBM DB2 UDB 8.2 and 9.1 using DataDirect JDBC Drivers 3.6; Hybrid Plugins: Sun ONE System Directory Server 5.2 and Oracle 9i, Sun ONE System Directory Server 5.2 and Oracle 10g, MS Active Directory 2003 and Oracle 9i, MS Active Directory 2003 and Oracle 10g, MS Active Directory 2003 and MS SQL Server 2005, Sun ONE System Directory Server 5.2 and IBM DB2 8.2 or 9.1 |
| | Browsers | On Microsoft Windows:<br>• Microsoft Internet Explorer 6.0 and 7.0<br>• Mozilla Firefox 2.0<br>On Linux: Mozilla Firefox 2.0<br>On Apple Mac OS X:<br>• Apple Safari 3.0<br>• Mozilla Firefox 2.0 |

| TOE Component | IT Support Components | Platform(s) |
|---|---|---|
| | JVM Version | JRockit 5 for Windows Server 2003 SP1, Red Hat Enterprise Linux 4 (x86-32, x86-64 and Itanium-64) and Suse 10.0 (x86-32, x86-64 and Itanium-64), Sun 1.5.0_12^ Solaris 9 and 10 (SPARC), IBM 1.5.0 64 bits for AIX 5.3 running 32 bits JVM, HP-UX 11.23 Itanium-64 using HP-UX JVM 1.5.0_05, |
| ALBPM Enterprise 6.0 for WebSphere | Operating System(s) | Windows Server 2003 SP1 (x86-32), Linux SUSE 10.0 (x86-32, x86-64, Itanium-64), Linux RHEL 4.x (x86-32, x86-64, Itanium-64), AIX 5.3, Solaris 9 and 10 (SPARC). |
| | Application Server | WorkSpace: Tomcat 5.5.15, IBM WebSphere 6.1.0.5; Process Administrator: Tomcat 5.5.15; WorkSpace Administrator: Tomcat 5.5.15, IBM WebSphere 6.1.0.5; Archive Viewer: Tomcat 5.5.15, IBM WebSphere 6.1.0.5; BPM Deployer: IBM WebSphere 6.1.0.5; PAPI-WS: Tomcat 5.5.15, IBM WebSphere 6.1.0.5; RSS Feeds: Tomcat 5.5.15, IBM WebSphere 6.1.0.5; ALSB Custom Transport EAR: N/A |
| | Engine Database | Oracle 9i, 10g (including RAC), MS SQLServer 2005, IBM DB2 UDB 8.2 and 9.1 using DataDirect Embedded JDBC Drivers (DataDirect 3.6). |
| | JMS Provider | TIBCO EMS 4.1, WebSphere 6.1.0.5 (other versions presented problems) Embedded Messaging, IBM MQ Series 5.3 (XA Compliant Resources) |
| | Directory Service Database | Single Source JDBC Plugins: Oracle 9i and 10g, MS SQLServer 2005, IBM DB2 UDB 8.2 and 9.1 using DataDirect JDBC Drivers 3.6; Hybrid Plugins: Sun ONE System Directory Server 5.2 and Oracle 9i, Sun ONE System Directory Server 5.2 and Oracle 10g, MS Active Directory 2003 and Oracle 9i, MS Active Directory 2003 and Oracle 10g, MS Active Directory 2003 and MS SQL Server 2005, Sun ONE System Directory Server 5.2 and IBM DB2 8.2 or 9.1 |
| | Browsers | On Microsoft Windows:<br>• Microsoft Internet Explorer 6.0 and 7.0<br>• Mozilla Firefox 2.0<br><br>On Linux: Mozilla Firefox 2.0<br><br>On Apple Mac OS X:<br>• Apple Safari 3.0<br>• Mozilla Firefox 2.0 |
| | JVM Version | Sun 1.5.0_12^ for Solaris 9 and 10 (SPARC); IBM 1.5.0 for Windows Server 2003 SP1 (x86-32), SUSE Linux 10.0 (x86-32, x86-64), Red Hat Enterprise Linux 4 (x86-32, x86-64), IBM 1.5.0 64 bits for AIX 5.3 running in 32 bits JVM |

| TOE Component | IT Support Components | Platform(s) |
|---|---|---|
| Studio (v6.0) | Operating System(s) | Microsoft Windows XP SP2 (x86-32)<br>Microsoft Windows 2003 Server SP1 or R2 (x86-32)<br>Novell SUSE Linux 10.0 (x86-32)<br>Red Hat Enterprise Linux 4 (x86-32) |
| | Application Server | Tomcat Servlet/JSP Container |
| | Database | Embedded Derby DB |
| | Browsers | On Microsoft Windows:<br>• Microsoft Internet Explorer 6.0 and 7.0<br>• Mozilla Firefox 2.0<br>On Linux: Mozilla Firefox 2.0<br>On Apple Mac OS X:<br>• Apple Safari 3.0<br>• Mozilla Firefox 2.0 |

**Table 2 IT Environment Support-Component Requirements**

#### 2.4.1.1 Runtime Components and the IT Environment

Figure 3 and Figure 4 show the relationships between TOE components and the supporting IT components for the standalone configuration and the application server configuration, respectively. In both figures, the TOE components with external interfaces are indicated. Communications between TOE components and components in the IT environment are shown with arrows.

Except as explained below, the arrows in the figures completely identify how TOE components interact or communicate with other TOE components or the IT environment. In the runtime environment, the TOE is primarily coordinating the execution of deployed process instances, although this aspect is not emphasized in the figures. The execution of events created by human interaction is characterized by communication between the Process Execution Engine and the WorkSpace components. This communication represents, for instance, tasks being pushed into a participant's workspace, and data and status changes returning to the Process Execution Engine as the participant completes tasks. Participants access their respective tasks via HTTP-enabled (default) or HTTPS enabled[3]. Events that do not involve human interaction are executed automatically by the engine.

The Process Execution Engine sequences a given process instance, pushing work items and receiving subsequent data and events as the state of the process instance changes. The Process Execution Engine keeps track of the state of the process instance and records audit and (auxiliary) log data. Other inter-TOE communication links shown in the figures are used for administrative functions. Typically, the latter types of communication are less frequent than the processing of business processes.

The primary communication links between the TOE and the IT environment are to the directory service and to two external databases that support the runtime environment. The external databases are the Process Execution Engine database and the Archive database.

Table 3 summarizes the functions of these IT components.

[3] The TOE offers the ability to secure communications between the browsers and TOE components via HTTPS. If configured to use HTTPS, the IT environment is relied upon to provide the certificates for secure SSL communications.

| IT Environment Component | Function |
|---|---|
| Directory Service | Stores:<br>• User identification and authentication data<br>• Data associating process instances, activities, and tasks with organizational units |
| Process Execution Engine database | Stores:<br>• Process instance data as the instance moves to completion<br>• Audit data until the specified time for offloading it to the archive database |
| Archive database | Stores per-process and per-instance audit data when it is offloaded from the Process Execution Engine database |

**Table 3: Primary Communication Links Between TOE and IT Environment**

Each component retrieves authentication and authorization information from the directory service by using the FDI abstraction framework. This API of the TOE consolidates the resolution of permissions for all TOE components regardless of whether they are deployed in a standalone or J2EE configuration of ALBPM.

The Admin Center is a standalone Swing application that is installed and launched from the machine whether the TOE runtime components have been installed. The Admin Center component is used primarily during initialization and setup, and is not typically used in the runtime environment. Administrators have to login to the runtime components' server to use this tool. For both the standalone configuration and application server configuration, the Admin Center is used for a variety of administrative functions, including the following:

- Initial setup and configuration of the TOE environment
- Starting and stopping the applications server
- Configuring the directory service
- Cconfiguring individual web applications
- Configuring the Process Administrator component
- Configuring WorkSpace settings
- Viewing various web application logs
- Applying maintenance packs

The Log Viewer is used to view Process Execution Engine logs where information about the instance processing can be recorded. When used remotely, the Log Viewer component retrieves events via the Process Administrator, which in turn retrieves log information directly from the files system of the runtime envirionment server host. When used locally, the Log Viewer retrieves log information directly from the file system without using Process Administrator.

All runtime components connect to the directory server either to store or retrieve information (especially for authentication), except for the Log Viewer and Archive Viewer. All of these connections use either JDBC or JNDI, depending on whether the directory service is an LDAP server or a database.

**Figure 3: Standalone Configuration**

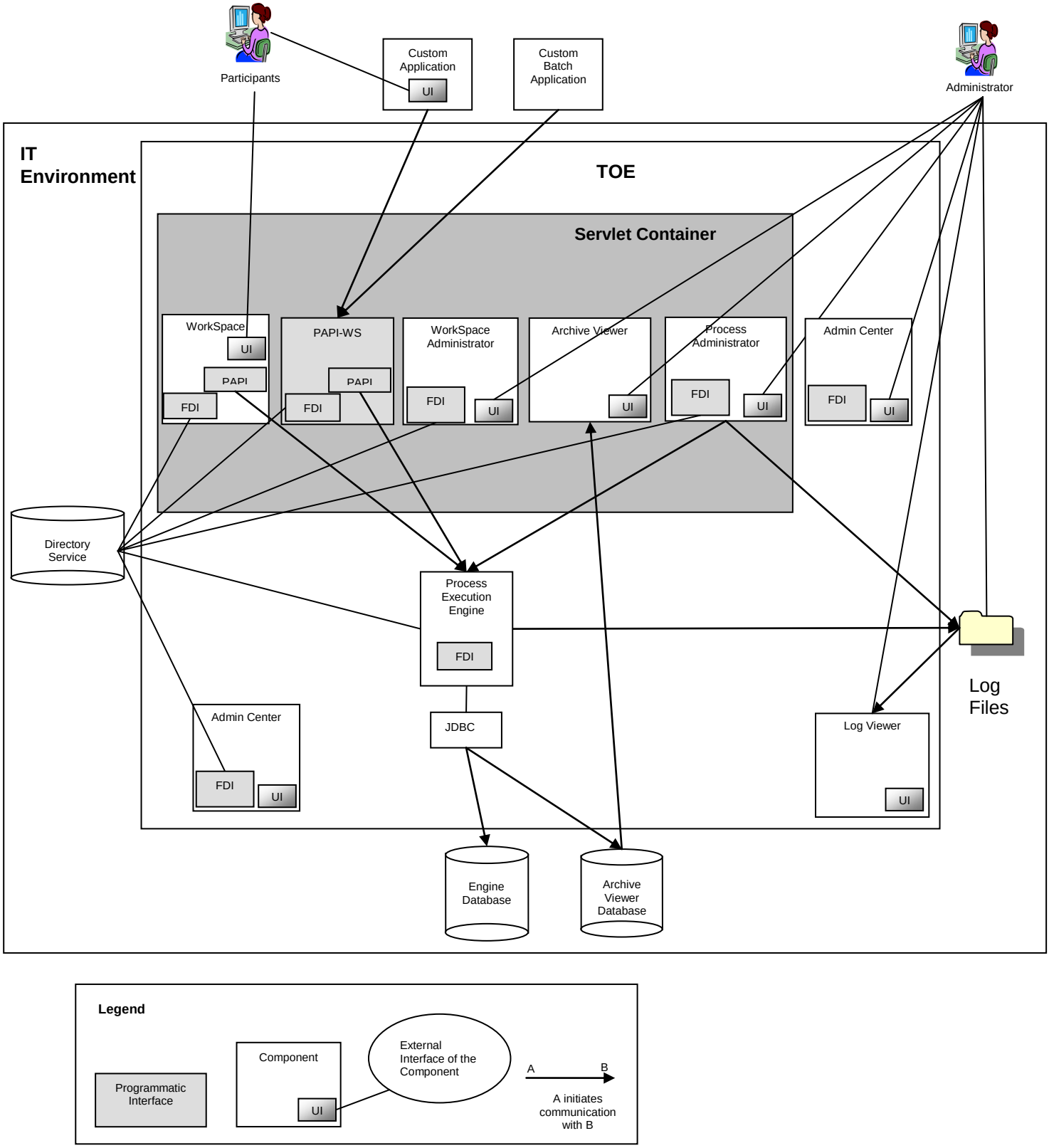
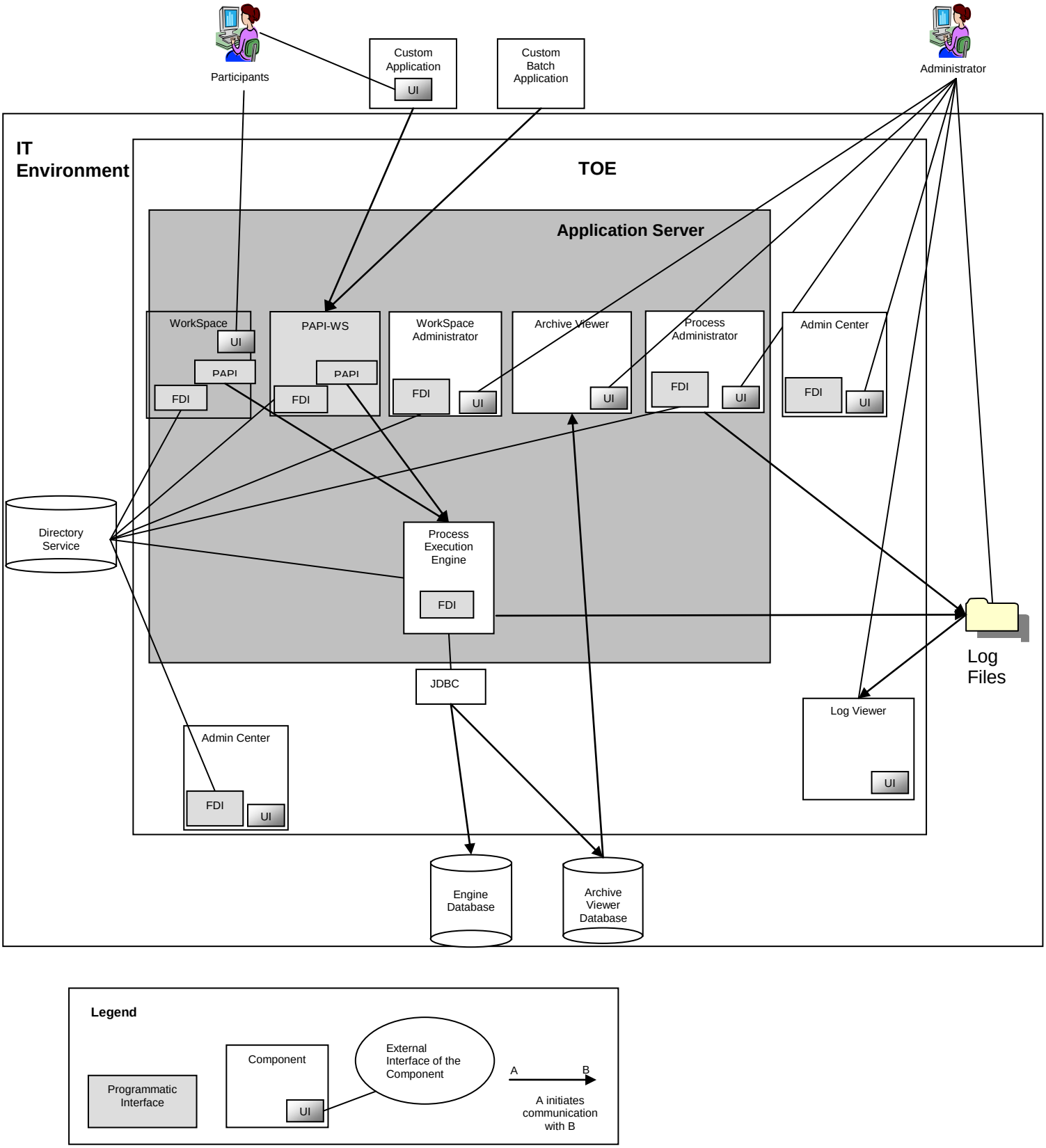


**Figure 4: Application Server Configuration**

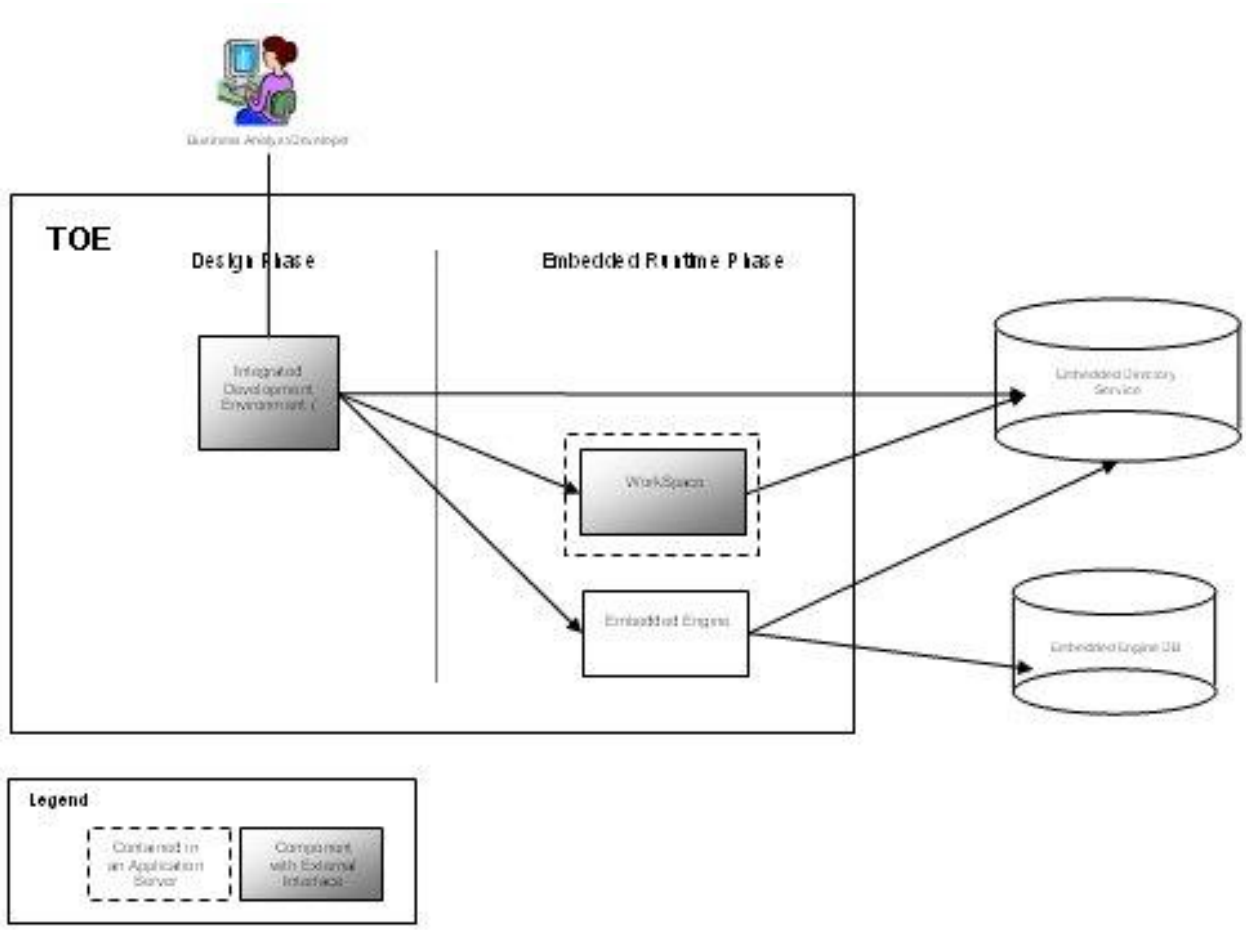


Note that in Figure 4, the Process Administrator does not communicate with the Process Execution Engine as in Figure 3. In the standalone configuration, the Process Administrator launches the Engine. In the application server configuration, the application server launches the Engine.

#### 2.4.1.2 Design Time Components and the IT Environment

Figure 5 illustrates the design-time TOE components, all of which are integrated within Studio, and components of the IT environment.

**Figure 5: Design Time Environment**




As mentioned in 2.1, developers do not interact with the runtime components of the TOE directly.

Studio resides on a host that is separate from the runtime environment server. During design time, designers log into this host and design abstract business processes independent of the runtime environment. It is only when a business process project needs to be deployed that there will be an information exchange between the Studio host and the runtime environment. This interaction is initiated by developers but controlled by administrators, as is described in the TSS (see Sections 6.1.3.3 and 6.1.4).

### 2.4.2 TOE Logical Boundaries

This section identifies the security functions that the BEA Systems TOE provides.

#### 2.4.2.1 Security Audit

The TOE identifies user actions on activity instances that are relevant to the security. The TOE provides the capability to generate and record audit event for these security-relevant actions. It provides a capability for authorized applications to access the trail of audit events.

Refer to section 6.1.1Security Audit for details.

#### 2.4.2.2 User Data Protection

The TOE provides the capability for an organization to restrict access to business-process activity instances in accordance with the organization's policy. Access controls for a business-process activity instances are specified in its associated business process (abstract) model. This model is realized in the TOE via an administrator-controlled operation, and a business process instance is an instantiation of this realized model. The restrictions on a process instance are based on the organizational unit, group, and roles of the participant requesting access as well as the organizational unit and roles associated with the process activity instance being accessed.

Refer to section 6.1.2 User Data Protection for details.

#### 2.4.2.3 Identification and Authentication

The TOE requires identification and authentication of each user before providing services to the Process Administrator, Archive Viewer, WorkSpace Administrator, and WorkSpace or any custom appliction that uses the available public APIs like PAPI or PAPI-WS. In the case of Admin Center, Log Viewer, and Studio, the TOE relies on the underlying operating system for the authentication mechanisms to confirm the identity of the administrator and developer before providing services. The TOE maintains a list of attributes associated with the administrators, developers and participants that include user name, password, organizational unit, group, roles, categories within each role, and permissions. Although BEA AquaLogic BPM Suite can be configured to host services for anonymous users and single sign-on, these capabilities are not enabled in the evaluated configuration.

Refer to section 6.1.3 Identification and Authentication for details.

#### 2.4.2.4 Security Management

The TOE provides administrators with capabilities to manage the security functions of the TOE. In addition to the audit functions described above, A BEA AquaLogic BPM administrator can configure the TOE, define business processes, and define organizational information including process participants. Authorized users can manage their own authentication data.

Refer to section 6.1.4 Security Management for details.

#### 2.4.2.5 Protection of the TOE Security Functions

The TOE includes features to ensure that a user cannot circumvent the security functions of the TOE. In addition, it prevents tampering and interference with the security functions.

Refer to section 6.1.5 Protection of the TOE Security Functions for details.

## 2.5 TOE Documentation

BEA Systems offers a series of documents that describe the installation process for the TOE as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6.2 for information about these and other documentation associated with the TOE.

# 3. Security Environment

This section summarizes the threats to assets protected by the TOE and its environment and assumptions about the intended environment of the TOE. Note that while the identified threats are mitigated by the security functions implemented in the TOE, the overall assurance level (EAL 2 augmented with ALC_FLR.1) also serves as an indicator of whether the TOE would be suitable for a given environment.

## 3.1 Threats

The following are threats identified for assets protected by the TOE. The assets protected by the TOE consist of an organization's business process information as realized in the TOE (e.g. supervisory applications implementing business processes, instances of business processes, and organizational information). Threat agents include both individuals outside the organization and individuals within the organization who attempt to exceed their authorization. In either case, the assumed level of expertise of the attacker for all the threats is unsophisticated.

| | |
|---|---|
| T.ACCESS | By using a BEA AquaLogic BPM service in its intended manner, an attacker accesses (i.e., reads or modifies) user data related to business processes for which the attacker is not authorized according to the organization's policy. |
| T.MASQUERADE | An attacker masquerades as an authorized user in order to gain access to business process information for which the attacker is not authorized according to the organization's policy. |
| T.TSF_COMP | By using a BEA AquaLogic BPM service in an unintended manner, an attacker accesses business process information for which the attacker is not authorized according to the organization's policy. |
| T.TSF_RECONFIG | By using a BEA AquaLogic BPM management service, an attacker modifies the server configuration or organizational information to allow subsequent attacks to succeed. |

## 3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs.

| | |
|---|---|
| P.ACCOUNT | The authorized users of BEA AquaLogic BPM shall be held accountable for their actions within the TOE. |

## 3.3 Secure Usage Assumptions

### 3.3.1 Intended Usage Assumptions

| | |
|---|---|
| A.NO_EVIL | The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation. |
| A.SECURE_HOST | The IT infrastructure on which the BEA AquaLogic BPM security functions depend will be installed, configured, managed, and maintained in accordance with both BEA AquaLogic BPM and IT infrastructure guidance documentation. |
| A.CONFIG | The TOE will be configured by authorized administrators such that the access control policy supports the organization's security policy. |

### 3.3.2 Physical Assumptions

| | |
|---|---|
| A.LOCATE | The BEA AquaLogic BPM server and supporting IT infrastructure servers will be located within controlled access facilities, which will prevent unauthorized physical access. |

### 3.3.3 Personnel Assumptions

| | |
|---|---|
| A.COOP_USER | All users will protect their authentication information from disclosure and their IT resources (i.e. web browser and host computer) from tampering. |

# 4. Security Objectives

This section summarizes the security objectives for the TOE and its environment.

## 4.1 Security Objectives for the TOE

| | |
|---|---|
| O.AUDIT_GEN | The TOE will provide the capability to detect and create records of security-relevant events associated with users. |
| O.AUDIT_REVIEW | The TOE will provide the capability to selectively view audit information. |
| O.MANAGE | The TOE will provide all the functions and facilities necessary to support management of the security of the TOE, and restrict these functions and facilities from unauthorized use. |
| O.MEDIATE | The TOE must protect user data in accordance with its security policy. |
| O.SELF_PROTECT | The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. |
| O.TOE_ACCESS | The TOE will provide identification and authentication mechanisms for each user before providing services to the Process Administrator, Admin Center, Archive Viewer, WorkSpace Administrator and WorkSpace prior to user's logical access to the TOE. |

## 4.2 Security Objectives for the IT Environment

| | |
|---|---|
| OE.AUDIT_STORE | The IT environment will provide the capability to store audit records. |
| OE.EXTERNAL_IA | The IT environment will identify and authenticate developers prior to providing Studio services to verify identity. |
| OE.TIME_STAMPS | The IT environment will provide reliable time stamps for TSF purposes. |
| OE.TSF_PROTECT | The IT environment will maintain domains for its own execution and for TOE execution that protects the IT environment, the TOE, and their resources from interference, tampering, or unauthorized disclosure. |

## 4.3 Security Objectives for the Environment

| | |
|---|---|
| OE.NO_EVIL | Those responsible for managing the TOE shall ensure that the TOE is installed, configured, managed, and maintained in accordance with its guidance documentation. |
| OE.SECURE_HOST | Those responsible for managing the TOE and its IT environment shall ensure that the IT infrastructure on which the TOE security functions depend is installed, configured, managed, and maintained in accordance with both TOE and IT infrastructure guidance documentation. |
| OE.LOCATE | Those responsible for the TOE and its IT environment shall locate the TOE server and supporting IT infrastructure servers within controlled access facilities, which will prevent unauthorized physical access. |
| OE.COOP_USER | All users shall protect their authentication information from disclosure and their IT resources (i.e. web browser and host computer) from tampering. |
| OE.CONFIG | Those responsible for managing the TOE shall ensure that the TOE will be configured such that the access control policy supports the organization's security policy. |

# 5. IT Security Requirements

This section defines the Security Functional Requirements (SFR) and Security Assurance Requirements (SAR) for the TOE and associated environment components. Note that in addition to these requirements, the TOE also satisfies a minimum strength of function (SOF-Basic). The SOF-Basic claim is supported by the Identification and authentication security function; more specifically FIA_UAU.2a.

## 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security Audit** | FAU_GEN.1: Audit Data Generation |
| | FAU_SAR.1: Audit Review |
| **FDP: User Data Protection** | FDP_ACC.1: Subset access control |
| | FDP_ACF.1: Security attribute based access control |
| **FIA: Identification and Authentication** | FIA_ATD.1: User Attribute Definition |
| | FIA_UAU.2a: User Authentication Before any Action |
| | FIA_UID.2a: User Identification Before any Action |
| | FIA_USB.1: User Subject Binding |
| **FMT: Security Management** | FMT_MOF.1: Management of Security Functions Behavior |
| | FMT_MSA.1a: Management of security attributes: Activity security attributes |
| | FMT_MSA.1b: Management of security attributes: User security attributes |
| | FMT_MSA.3: Static attribute initialization |
| | FMT_MTD.1a: Management of TSF data: Supervisory applications |
| | FMT_MTD.1b: Management of TSF data: Roles |
| | FMT_MTD.1c: Management of TSF data: Organizational information |
| | FMT_MTD.1d: Management of TSF data: Organizational information |
| | FMT_MTD.1e: Management of TSF data: User account information |
| | FMT_SMF.1: Specification of management functions |
| | FMT_SMR.1: Security Roles |
| **FPT: Protection of the TOE Security Functions** | FPT_RVM.1a: Non-bypassability of the TSF |
| | FPT_SEP.1a: TSF domain separation |

**Table 4 TOE Security Functional Components**

### 5.1.1 Security audit (FAU)

#### 5.1.1.1 Audit data generation (FAU_GEN.1)

**FAU_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the **[*not specified*]** level of audit; and
c) **[ CREATION – Create an instance of a process**
**SELECT – Manually assign a instance to the participant for execution**
**GRAB – Manual removal of an activity from another participant.**
**TASK EXECUTION START TIME: Activities can have one or more tasks. An event with timestamp will be generated when an activity task starts.**
**TASK EXECUTION END TIME: An event with timestamp will be generated when an activity task finishes its execution.**
**IN – Enter an activity**
**EXECUTE – Perform an activity**

**OUT – Exit an activity**
**UNSELECT – Manually release an instance previously assigned for execution by a participant**
**END – instance is finished].**

**FAU_GEN.1.2** The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **[severity, application, module, and thread]**.

#### 5.1.1.2 Audit review (FAU_SAR.1)

**FAU_SAR.1.1** The TSF shall provide **[authorized participants]** with the capability to read **[all audit information related to the process instances the participant is authorized to view]** from the audit records.

**Application Note**: A participant is authorized if their assigned role(s) allow view access to the process instance audit data for a given process instance.

**FAU_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

### 5.1.2 User data protection (FDP)

#### 5.1.2.1 Subset access control (FDP_ACC.1)

**FDP_ACC.1.1** The TSF shall enforce the **[BEA AquaLogic BPM Activity Policy]** on **[**

- **a) subjects: participant sessions**
- **b) objects: process instances, instances of interactive activities, and tasks**
- **c) operations:**
  1. **View an instance and its tasks**
  2. **Select an instance**
  3. **Execute a task associated with an instance**
  4. **Modify instance data**
  5. **Grab an instance**
  6. **Route an instance**
  7. **Abort an instance**
  8. **Suspend an instance**
  9. **Delegate an instance to another participant**
  10. **Peer-assign an instance to another participant**
  11. **Re-assign an instance to another participant**
  12. **Escalate an instance to another participant**
  13. **Create a new process instance].**

#### 5.1.2.2 Security attribute based access control (FDP_ACF.1)

**FDP_ACF.1.1** The TSF shall enforce the **[BEA AquaLogic BPM Activity Policy]** to objects based on the following: **[**

- **a) Subject attributes**
  1. **Organizational units**
  2. **Process roles**
  3. **For each role, permissions associated with that role (Execute, Route, Select, Abort, Delegate, Suspend, Reassign, Escalate, and Peer Assignment)**
  4. **For each role, a hierarchical category value within that role (1 – 9)**
- **b) Object attributes**
  1. **(Process) Organizational units**

2. **(Interactive Activities) Process role**
3. **(Tasks) Task markings (read-only, mandatory)**
4. **(Instances of interactive Activities) Properties (Suspendable, Abortable, and Assignable)]**

**FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

1. **A subject may view an instance and its tasks if the instance's organizational units are a subset of the subject's organizational units**
2. **A subject may select an instance if**
   i. **The participant session may view the instance and its tasks and**
   ii. **The instance's role is contained in the set of the subject's roles and**
   iii. **The subject role matching the instance's role has Select permission and**
   iv. **The task's state is not Selected and**
   v. **The task's state is not Locked**
3. **A subject may execute a task associated with an instance if**
   i. **The subject may select the instance and**
   ii. **The subject role matching the instance's role has Execute permission**
4. **A subject may modify instance data if**
   i. **The subject may execute the task associated with the instance and**
   ii. **The task is not marked read-only**
5. **A subject may route an instance if**
   i. **The subject may select the instance and**
   ii. **The subject role matching the instance's role has Route permission and**
   iii. **All mandatory tasks of the instance have been successfully completed**
6. **A subject may abort an instance if**
   i. **The subject may select the instance and**
   ii. **The subject role matching the instance's role has Abort permission and**
   iii. **The instance has the property Abortable**
7. **A subject may suspend an instance if**
   i. **The subject may select the instance and**
   ii. **The subject has the Suspend permission and**
   iii. **The instance has the property Suspendable**
8. **A subject may delegate an instance if**
   i. **The subject has selected the instance and**
   ii. **The subject role matching the instance's role has Delegate permission and**
   iii. **The instance has the property Assignable and**
   iv. **The target participant may select the instance and**
   v. **The subject category is greater than the category of the target participant in the role matching the instance's role**
9. **A subject may peer-assign a instance if**
   i. **The subject has selected the instance and**
   ii. **The subject role matching the instance's role has Peer Assignment permission and**
   iii. **The instance has the property Assignable and**
   iv. **The target participant may select the instance and**
   v. **The subject category is equal to the category of the target participant in the role matching the instance's role**
10. **A subject may re-assign an instance if**
    i. **Another participant has selected the instance and**
    ii. **The subject role matching the instance's role has Re-assign permission and**

**iii. The instance has the property Assignable and**
**vi. The target participant may select the instance and**
**iv. The subject category is higher than the category of both the original and target participants in the role matching the instance's role**

**11. A subject may escalate an instance if**
**i. The subject has selected the instance and**
**ii. The subject role matching the instance's role has Escalate permission and**
**iii. The instance has the property Assignable and**
**iv. The target participant may select the instance and**
**v. The subject category is less than the category of the target participant in the role matching the instance's role**

**12. A subject may create a new instance of a process if**
**i. The subject may view a Global Creation activity for the process and**
**ii. The subject may perform the task associated with the Global Creation activity of the process and**
**iii. When programmatically, the subject possesses at least one role in the process].**

**FDP_ACF.1.3** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[none]**.

**FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the **[none]**.

## 5.1.3 Identification and Authentication (FIA)

### 5.1.3.1 User Attribute Definition (FIA_ATD.1)

**FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: **[**
**a) User name,**
**b) Password,**
**c) Organizational units,**
**d) Organizational groups,**
**e) Organizational roles,**
**f) For each role, category within the role and**
**g) For each role, permissions associated with that role (Execute, Route, Select, Abort, Delegate, Re-assign, Escalate, and Peer Assignment)].**

### 5.1.3.2 User authentication before any action (FIA_UAU.2a)

**FIA_UAU.2.1a** The TSF shall require each **BEA AquaLogic BPM administrator, End user administrator, and BEA AquaLogic BPM participant** user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

**Application Note:** The TOE requires authentication prior to accessing services to the Process Administrator, Archive Viewer, WorkSpace Administrator, and WorkSpace or any custom appliction that uses the available public APIs like PAPI or PAPI-WS.].

### 5.1.3.3 User Identification before any Action (FIA_UID.2a)

**FIA_UID.2.1a** The TSF shall require each **BEA AquaLogic BPM administrator, End user administrator, and BEA AquaLogic BPM participant** user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

#### 5.1.3.4 User Subject Binding (FIA_USB.1)

**FIA_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[**

**a) Organizational units**
**b) Process roles**
**c) For each role, permissions associated with that role**
**d) For each role, a hierarchical category value within that role].**

**FIA_USB.1.2** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[a subject's security attributes are initialized to all the values associated with the corresponding participant attributes identified in FIA_ATD.1]**.

**FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[**

a) **after initialization, participant session attributes are static for the lifetime of the session;**
b) **changes to a participant's security attributes during a participant session do not take effect until the next participant session is created]**.

### 5.1.4 Security Management (FMT)

#### 5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

**FMT_MOF.1.1** The TSF shall restrict the ability to **[*disable and enable*]** the functions **[Process Execution Engine operation]** to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.2 Management of security attributes: Activity security attributes (FMT_MSA.1a)

**FMT_MSA.1a.1** The TSF shall enforce the **[BEA AquaLogic BPM Activity Policy]** to restrict the ability to **[*[specify]*]** the security attributes **[organizational units, process role, and properties]** to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.3 Management of security attributes: User security attributes (FMT_MSA.1b)

**FMT_MSA.1b.1** The TSF shall enforce the **[BEA AquaLogic BPM Activity Policy]** to restrict the ability to **[*query and modify*]** the security attributes **[organizational units, process role**, **role permissions, and role categories]** to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.4 Static attribute initialisation (FMT_MSA.3)

**FMT_MSA.3.1** The TSF shall enforce the **[BEA AquaLogic BPM Activity Policy]** to provide **[*[configured]*]** default values for security attributes that are used to enforce the SFP.

**FMT_MSA.3.2** The TSF shall allow the **[no identified roles]** to specify alternative initial values to override the default values when an object or information is created.

#### 5.1.4.5 Management of TSF data: Supervisory applications (FMT_MTD.1a)

**FMT_MTD.1a.1** The TSF shall restrict the ability to **[*query, [publish, deploy, and undeploy]*]** the **[supervisory applications]** to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.6 Management of TSF data: Process Roles (FMT_MTD.1b)

**FMT_MTD.1b.1** The TSF shall restrict the ability to **[*query, modify, delete and [create]*]** the **[abstract process roles]** to **[BEA AquaLogic BPM developer]**.

#### 5.1.4.7 Management of TSF data: Organizational information (FMT_MTD.1c)

**FMT_MTD.1c.1** The TSF shall restrict the ability to **[*query, modify, delete, and [create]*]** the **[organizational units, organizational groups, organizational roles, and participants]** to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.8 Management of TSF data: Organizational information (FMT_MTD.1d)

**FMT_MTD.1d.1** The TSF shall restrict the ability to **[*modify and [create]*]** the **[participants]** to **[End user administrator]**.

#### 5.1.4.9 Management of TSF data: User account information (FMT_MTD.1e)

**FMT_MTD.1e.1** The TSF shall restrict the ability to **[*query, modify, delete and [create]*]** the **[**

**a) User name,**
**b) Password,**
**c) Organizational units**
**d) Organizational groups,**
**e) Organizational roles,**
**f) Category within a role, and**
**g) Permission associated with a role]**

to **[BEA AquaLogic BPM administrator]**.

#### 5.1.4.10 Specification of Management Functions (FMT_SMF.1)

**FMT_SMF.1.1** The TSF shall be capable of performing the following security management functions: **[**

**a) Enable (i.e. start) and disable (i.e. stop) the Process Execution Engine**
**b) Object security attribute management,**
**c) User attribute management,**
**d) Supervisory application management,**
**e) Role management,**
**f) Organizational information management, and**
**g) Process instance assignment].**

#### 5.1.4.11 Security roles (FMT_SMR.1)

**FMT_SMR.1.1** The TSF shall maintain the roles **[**

**a) BEA AquaLogic BPM administrator**
**b) End user administrator**
**c) BEA AquaLogic BPM participant].**

**FMT_SMR.1.2** The TSF shall be able to associate users with roles.

### 5.1.5 Protection of the TOE Security Functions (FPT)

#### 5.1.5.1 Non-bypassability of the TSP (FPT_RVM.1a)

**FPT_RVM.1.1a** The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

#### 5.1.5.2 TSF domain separation (FPT_SEP.1a)

**FPT_SEP.1.1a** The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

**FPT_SEP.1.2a** The TSF shall enforce separation between the security domains of subjects in the TSC.

## 5.2 IT Environment Security Functional Requirements

The following table identifies the security functional requirements (SFRs) that are satisfied by the IT Environment. All of these SFRs were drawn from the CC (Part 2) and adapted by refinement for application to the IT environment.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security Audit** | FAU_STG.1: Protected audit trail storage |
| **FIA: Identification and Authentication** | FIA_UID.2b: User identification before any action |
| | FIA_UAU.2b: User authentication before any action |
| **FPT: Protection of the TOE Security Functions** | FPT_RVM.1b: Non-bypassability of the ~~TSP~~ **IT environment's security policy** |
| | FPT_SEP.1b: ~~TSF~~ **IT environment** domain separation |
| | FPT_STM.1: Reliable time stamps |

**Table 5 IT Environment Security Functional Components**

### 5.2.1 Security audit (FAU)

#### 5.2.1.1 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~ **IT environment** shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The ~~TSF~~ **IT environment** shall be able to [***prevent***] unauthorised modifications to the stored audit records in the audit trail.

### 5.2.2 Identification and Authentication (FIA)

#### 5.2.2.1 User identification before any action (FIA_UID.2b)

FIA_UID.2.1b The ~~TSF~~ **IT environment** shall require each **BEA AquaLogic BPM administrator and BEA AquaLogic BPM developer** user to identify itself before allowing any other ~~TSF~~**IT-environment**-mediated actions on behalf of that user.

#### 5.2.2.2 User authentication before any action (FIA_UAU.2b)

FIA_UAU.2.1b The ~~TSF~~ **IT environment** shall require each **BEA AquaLogic BPM administrator and BEA AquaLogic BPM developer** user to be successfully authenticated before allowing any other ~~TSF~~**IT-environment** mediated actions on behalf of that user.

**Application Note:** The underlying operating system is responsible for confirming the identity and verifying that identity of developers prior to accessing Studio resources and that of administratrors prior to accessing Admin Center and Log Viewer.

### 5.2.3 Protection of the TOE Security Functions (FPT)

#### 5.2.3.1 Non-bypassability of the ~~TSP~~ IT environment's security policy (FPT_RVM.1b)

**FPT_RVM.1.1b** The ~~TSF~~ **IT environment** shall ensure that ~~TSP~~ **IT environment's security policy** enforcement functions are invoked and succeed before each function within the ~~TSC~~ **IT environment's scope of control** is allowed to proceed.

#### 5.2.3.2 ~~TSF~~ IT Environment domain separation (FPT_SEP.1b)

**FPT_SEP.1.1b** The ~~TSF~~ **IT environment** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

**FPT_SEP.1.2b** The ~~TSF~~ **IT environment** shall enforce separation between the security domains of subjects in the ~~TSC~~ **IT environment's scope of control**.

#### 5.2.3.3 Reliable time stamps (FPT_STM.1)

**FPT_STM.1.1** The ~~TSF~~ **IT environment** shall be able to provide reliable time stamps for its own use **and for use by the TOE**.

## 5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL2 augmented with ALC_FLR.1 component as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

| Requirement Class | Requirement Component |
|---|---|
| **ACM: Configuration management** | ACM_CAP.2: Configuration items |
| **ADO: Delivery and operation** | ADO_DEL.1: Delivery procedures |
| | ADO_IGS.1: Installation, generation, and start-up procedures |
| **ADV: Development** | ADV_FSP.1: Informal functional specification |
| | ADV_HLD.1: Descriptive high-level design |
| | ADV_RCR.1: Informal correspondence demonstration |
| **AGD: Guidance documents** | AGD_ADM.1: Administrator guidance |
| | AGD_USR.1: User guidance |
| **ALC: Life cycle support** | ALC_FLR.1: Basic flaw remediation |
| **ATE: Tests** | ATE_COV.1: Evidence of coverage |
| | ATE_FUN.1: Functional testing |
| | ATE_IND.2: Independent testing - sample |
| **AVA: Vulnerability assessment** | AVA_SOF.1: Strength of TOE security function evaluation |
| | AVA_VLA.1: Developer vulnerability analysis |

**Table 6 EAL 2 augmented with ALC_FLR.1Assurance Components**

### 5.3.1 Configuration management (AC)

#### 5.3.1.1 ACM_CAP.2 - Configuration items

ACM_CAP.2.1d The developer shall provide a reference for the TOE.

ACM_CAP.2.2d The developer shall use a CM system.

ACM_CAP.2.3d The developer shall provide CM documentation.

ACM_CAP.2.1c The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2c The TOE shall be labeled with its reference.

ACM_CAP.2.3c The CM documentation shall include a configuration list.

ACM_CAP.2.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.5c The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.6c The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.

ACM_CAP.2.7c The CM system shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

### 5.3.2 Delivery and operation (ADO

#### 5.3.2.1 ADO_DEL.1 - Delivery procedures

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2d The developer shall use the delivery procedures.

ADO_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.3.2.2 ADO_IGS.1 - Installation, generation, and start-up procedures

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

### 5.3.3 Development (ADV)

#### 5.3.3.1 ADV_FSP.1 - Informal functional specification

ADV_FSP.1.1d The developer shall provide a functional specification.

ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2c The functional specification shall be internally consistent.

ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4c The functional specification shall completely represent the TSF.

ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### 5.3.3.2 ADV_HLD.1 - Descriptive high-level design

ADV_HLD.1.1d The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1c The presentation of the high-level design shall be informal.

ADV_HLD.1.2c The high-level design shall be internally consistent.

ADV_HLD.1.3c The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6c The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

#### 5.3.3.3 ADV_RCR.1 - Informal correspondence demonstration

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

### 5.3.4 Guidance documents (AGD)

#### 5.3.4.1 AGD_ADM.1 - Administrator guidance

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.3.4.2 AGD_USR.1 - User guidance

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

## 5.3.5 Life cycle support (ALC)

### 5.3.5.1 ALC_FLR.1 - Basic flaw remediation

ALC_FLR.1.1D The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

## 5.3.6 Tests (ATE)

### 5.3.6.1 ATE_COV.1 - Evidence of coverage

ATE_COV.1.1d The developer shall provide evidence of the test coverage.

ATE_COV.1.1c The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

### 5.3.6.2 ATE_FUN.1 - Functional testing

ATE_FUN.1.1d The developer shall test the TSF and document the results.

ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.3.6.3 ATE_IND.2 - Independent testing - sample

ATE_IND.2.1d The developer shall provide the TOE for testing.

ATE_IND.2.1c The TOE shall be suitable for testing.

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

### 5.3.7 Vulnerability assessment (AVA)

#### 5.3.7.1 AVA_SOF.1 - Strength of TOE security function evaluation

AVA_SOF.1.1d The developer shall perform a strength-of-TOE-security function analysis for each mechanism identified in the ST as having a strength-of-TOE-security function claim.

AVA_SOF.1.1c For each mechanism with a strength-of-TOE-security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength-of-TOE-security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

#### 5.3.7.2 AVA_VLA.1 - Developer vulnerability analysis

AVA_VLA.1.1d The developer shall perform a vulnerability analysis.

AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

## 6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

## 6.1 TOE Security Functions

### 6.1.1 Security Audit

#### 6.1.1.1 Audit generation

The Process Execution Engine can be configured to generate audit records of all, or a subset of, activities in a business process associated to a particular process instance. The records generated correspond to the following events:

- start-up and shutdown of the audit functions;
- process instance creation;
- manual assignment of an activity to a participant for execution;
- manual removal of a process instance from another participant;
- activity task start and end events;
- entering and exiting an activity;
- performance (execution of) an activity task by a participant;
- manual release of an activity previously assigned for execution by a participant; and
- the completion of a activity.

Audit records will contain the following data:
- Date and time of the event;
- type of event;
- subject identity;
- process id;
- activity id; and
- outcome (success or failure) of the event

The Log File will contain for each audit event type, based on the auditable event definitions, the event's
- severity
- application
- module
- thread

The audit function is active when the Process Execution Engine is running. Hence, startup and shutdown of the Process Execution Engine indicates the startup and shutdown of the audit function.

Audit records are generated on a per-process basis. The BEA AquaLogic BPM developer who designs a process can, for purposes of testing the process while it is being developed, configure audit records to be generated for that process or what activities within the process should generate the events. Audit records may include messages from the Process Execution Engine.

A logging function is related to Process Business Language (PBL),[4] but information generated by this function is sent to the general system log, and is not relevant to the TOE audit security function. Only data generated via the Process Execution Engine (and recorded in its backend database table, PPROCINSTEVENTS) and the Process instance-related events are relevant to the TOE audit function.

An administrator can override the default auditing event level defined by the developer. When a process is published and deployed, there are options to: (a) accept the existing auditing rules specified in the business process design, (b) to avoid any event auditing, or (c) to force the auditing of all process activity events. Because only administrators can publish and deploy processes, the auditing options they select ultimately determine the audit-generation behavior of the TOE for that process instance.

[4] PBL refers to a high-level scripting language used to define the business rules and the logic of activity types and certain transitions within a process. Sometimes, as is done here, the term is used to refer to a script written in PBL and associated with an activity.

All activities generate the same default events: IN, OUT. Every time an activity task executes, the Process Execution Engine records the EXECUTE event. Instances can also be manually selected and unselected for execution, and these are audited as SELECT and UNSELECT events. For Tasks within an activity, the Process Execution Engine can generate TASK EXECUTION START TIME and TASK EXECUTION END TIME events. There are two additional events for process instances: CREATION and END. The engine can generate a GRAB event when one participant (with the right permissions and successfully authenticated) manually removes an activity from another participant. Finally, a BEA AquaLogic BPM developer can define and create custom events embedded in the implementation of a business process activity task implementation, although this capability is outside the scope of the evaluation.

The Process Execution Engine records all the events that relate to a process instance that take place during the execution to its PPROCINSTEVENTS table, which is part of the Process Execution Engine database. This database is an external, backend database that is provided by the IT environment.

In addition, the Process Execution Engine can log execution events in a log file. The events logged in that log file are assigned with a severity that is one of: Fatal, Severe, Warning, Info, and Debug (from greatest to least severe). A BEA AquaLogic BPM administrator can change the type of information being recorded in the log. These logs are not associated with the main audit generation function described above for the TOE; however, these logs can contain records related to the same security-relevant events that are recorded in the audit trail. These log files are not accessible to non-administrative users.

Once the process instance has completed its work in the business process and reached the END node of the business process and has remained in the END node for a configurable amount of time, the engine can purge the instance information and related audit data to an archive database. This allows the consolidation of instance data and related audit information from multiple processes to be centralized for archive and auditing reasons. The archive database can be queried using the Archive Viewer. The archive database is completely self-contained and is able to keep history for process instance data and audit information as long as desired for compliance policies.

#### 6.1.1.2 Audit review

Process instance-related events can be accessed through the WorkSpace only for those instances that are visible to the authorized participant based on the user's role assignment and organizational unit. Access to the per-process audit information is controlled by the TOE's access control policy and enforcement mechanism. Process instance audit information can also be retrieved while the Process Execution Engine is running using the PAPI API. The events and information that authorized users can review are described in Section 6.1.1.1.

Log Viewer enables an administrator to read these logged events. A set of log files is created for each project defined. Log Viewer reads the files and displays them to help the administrator monitor and trace Process Execution Engine execution. An administrator can use the Log Viewer to view logs regardless of whether the Process Execution Engine is running. The Log Viewer application is restricted to administrators of the TOE.

In the evaluated configuration, the Log Viewer resides on the Administrator's host. If the Administrator's host is the same as the host where the Process Execution Engine runs, then access to the Process Execution Engine log files will be local and there is no transport protocol required. If the Administrator's host is remote to the host where the Process Execution Engine resides, the Log Viewer needs to connect to the Process Administrator using HTTPS and needs to identify and authenticate with the Process Administrator and validate he/she is an administrator.

Similarly, the Archive Viewer can also review and query process instance information and related audit information for those instances that were purged from the Process Execution Engine database after their grace period in the END process activity. Only administrators have access to this web application. The Archive Viewer allows only retrieval of data and it does not support any update/remove/insert operations; it is just for read-only access.

It is important understand the difference between the information in the Process Execution Engine log files and the audit data associated to a process instance. The Process Execution Engine log files contain log information about the engine behavior and also how it is processing instances. If there are errors, these will be found in the Process Execution Engine log files. The information in the Process Execution Engine log files can be viewed by an authorized administrator within the Log Viewer. On the other hand, the process instance events that are stored as process instance associated data for auditing purposes. These can be programmatically accessed using ALBPM's

PAPI API. They can also be viewed by authorized participants[5] via WorkSpace. For archived instances, it is also possible to find audit events associated to instances through the Archiving Viewer component.

Thus, log file data is only viewable by administrators, but audit data is viewable by authorized participants.

#### 6.1.1.3 Requirements tracing

The Security Audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1
- FAU_SAR.1

#### 6.1.1.4 IT Environment support

This TOE security function is supported by security-relevant functionality located in the IT environment. FPT_STM.1 provides reliable time stamps for audit records. FIA_UID.2b and FIA_UAU.2 provide authenticated user identities for use the TOE user identity function (FIA_UID.2a) in audit records. The Process Execution Engine records all the events that relate to a process instance that take place during the execution to its PPROCINSTEVENTS backend database table (which is an external database provided by the IT environment).

The PPROCINSTEVENTS table is a part of the Process Execution Engine database, which is provided by the IT environment. This database can be either local or remote to the Process Execution Engine itself. After a process instance completes, the (per-instance) audit events in this table are archived periodically as configured by an administrator. When archived, the local audit events are moved to a similar table in the Archiving Database and deleted from the local PPROCINSTEVENTS table. The Archiving database can also be either local or remote. The TOE uses JDBC to access these databases; if the databases are remote, security of the communications from TOE to the remote database is provided by protection of the IT environment.

The ST does not identify audit storage functionality as a TOE capability – this capability is provided by the IT environment. Neither the TOE nor the IT environments provide audit storage failure protection. In both cases, this must be handled manually by system administrators in the IT environment. Because database operations are transactional, if space is consumed by the PPROCINSTEVENTS table of the audit trail database, the transaction will rollback and the database will revert to its previous state. However, there is no claim with respect to lost records or TOE behavior should this event occur. The IT environment protects the audit data from unauthorized deletion and modification, but can not protect from storage failure. This functionality is identified by FAU_STG.1.

### 6.1.2 User Data Protection

BEA AquaLogic BPM constrains a participant's ability to:

- View an instance and its tasks
- Select an instance
- Execute a task associated with an instance
- Modify the instance variables through the implementation of an activity task using Studio
- Route an instance
- Abort an instance
- Suspend an instance
- Delegate an instance to another participant
- Peer-assign an instance to another participant
- Re-assign an instance from one participant to another participant
- Escalate an instance to another participant

[5] Participants are authorized if their assigned roles are allowed view access to the audit data.

- Create a new process instance.

The constraints above describe the logical access control for the TOE in both configurations. The logical processing behaves exactly the same regardless of whether the TOE is in a standalone or applications server configuration. When in the applications server configuration, there are some internal communications and protection differences, because J2EE container mechanisms can be extended to the Process Execution Engine (e.g., security descriptors, Enterprise Java Bean communications and transaction management). Analogous capabilities and protection is provided in the standalone configuration, but via different mechanisms. In addition, the processes of publishing and deploying a process instance are internally different. However, from an external perspective the security infrastructure and behavior are exactly the same for both configurations, by intent.

The constraints are based on attributes associated with the end user's session and the instance of an interactive activity. The detailed rules for implementing the constraints are specified in FDP_ACF.1.2. This section does not repeat these rules but rather provides additional implementation details. There are certain actions that are enabled at modeling time as part of the activity definition such as whether the instance is ABORTABLE, SUSPENDABLE or AUTOCOMPLETE (automatic routing to the next activity in the process) and are inherited. In addition, task markings (read-only, mandatory) are defined aspects of the task set at modeling time. These permissions are cross checked with those associated with a person through Role assignment (ABORT, EXECUTE, ROUTE, SELECT, DELEGATE, REASSIGN, ESCALATE, and PEER ASSIGNMENT). The intersection of the subject and object permissions determines the ability to execute these actions.

At publishing and deployment time, supervisory applications are associated with an organizational units and groups. Process instances are invoked instances of the supervisory application. Publishing and deploying is an administrative operation, but creating a process instance is controlled via the access control rules specified in FDP_ACF.1.2. A process instance is created within the scope of the organizational unit and group of the deployed supervisory application.

As the process instance executes on a Process Execution Engine, tasks that are available to participants are constrained by the participant's organizational unit and the process instances organizational unit. The Process Execution Engine can control logically separate process instances separately; in other words, the tasks associated with a given process in one organizational unit will not be accessible to participants in a different organizational unit, even if the participant's role(s) would allow the access. Organizational units bound the scope for role and group identifiers.

The Process Execution Engine maintains a session to represent an interaction with an end user. Session information includes the organizational unit and roles associated with the user as well as permissions assigned to the user for each role. The Process Execution Engine controls access by end users to tasks associated with instances of interactive activities. It maintains the organizational units associated with each process and the role and instance data associated with each process instance. Tasks are functions that can be performed within an activity instance (for example, functions that are pending execution).

Instance variables are variables associated with a process instance. They contain information that is relevant to the activity instances of the process. Instance variables for a given process instance can be accessed from different process activities as the processing flows through the process. A task may be marked *read-only* meaning that the task can not modify instance data. That is, any change to a process instance and its variables (instance variables, predefined variables, etc.) within the BP-method (business logic) will be ignored.

The Process Execution Engine makes each activity instance of an activity visible in WorkSpace or through available APIs such as PAPI or PAPI-WS, but only to the participant(s) assigned to the role associated with the activity.

Activities have a main task and can have optional tasks. Tasks appear in a list in a user's workspace. A session may use a task to modify instance data (i.e., instance variables) for an activity only when the task is not marked read-only.

A BEA AquaLogic BPM administrator may configure participant attributes so that a participant can assign an interactive activity instance to another participant in the same role. Assigning an activity instance is only available for a participant if the activity was defined as Assignable during modeling time in Studio, and depending on the participant's permissions defined at runtime by the administrator. The assignment to another participant depends on the category that each participant has for the role.

Specifically, each participant is assigned to a role with a category. This category represents the hierarchical level of the participant within the role. Possible category values are from 1-9. The higher a participant's category is, the higher the hierarchical level of the participant in the role. A BEA AquaLogic BPM administrator may grant the following capabilities to a participant by assigning permissions and/or rank attributes:

1. Delegate: enables the participant to assign instances to participants with a lower category (or "rank") in the Role.
2. Escalate: enables the participant to assign instances to participants with a higher category (or "rank") in the Role.
3. Peer Assignment: enables the participant to assign instances to participants with the same category (or "rank") in the Role.
4. Re-assign: enables the participant to re-assign an instance from one participant in the role to any other participant in the role, if the re-assigning participant's category (or "rank") is higher than both of the target participant's ranks.[6]

A participant can create new process instances of a process model if their roles include access to the Global Creation activity of the process instance. The Global Creation and Begin activities are default activities created automatically for every process instance. If a participant's roles allow access, the user can run the activity by selecting it in WorkSpace. When the Global Creation activity executes, a process instance begins creation and initialization. The Begin activity finishes the process instance creation. Every process contains a "Begin" activity that is automatically executed as the first activity for all process instances. When using the provided APIs (PAPI and PAPI-WS) any participant with at least one of the process roles assigned can create new instances of process.

Global activities are used to allow end users to run applications or database queries only when needed. These applications are not an integral part of the process, but they contain information that can be accessed on an "as needed" basis. Global activities can be assigned to any user-defined role. Global activities can contain code that programmatically creates a process instance.

At development time, the developer may decide that the process needs to be accessible through SOAP so Web Service clients can enact and create a process instance in a business process. The interface of this process will be exposed through a Web Service interface definition (WSDL file). The web service interface provides the full set of the operations available through the WorkSpace. Two types of web service operations may be defined by Process Designers:

1. Process Creation: Allows creation of new process instances of a process model. Process creation is the ability to create a process instance externally (from outside the process).
2. Process Notification: Allows applications to send notifications to the process. Process instances waiting in Notification Wait activities can be notified by external Web applications or Web services via the WSDL file.

Additionally, there is another Web Service API known as PAPI-WS that, apart from enabling the creation of process instances and send notifications, also allows operations for creating a session with a Process Execution Engine and gets the status of a given created instance and is also able to retrieve process instances.

The TOE and its IT support environment cooperate to provide internal transaction controls to protect the integrity of TOE data. Internal transactions are protected with a "rollback" capability in the case that data transfer or modification operations are disrupted before the operations are completed. In the case of the TOE, the transaction manager mechanism is integrated with the Process Execution Engine.

#### 6.1.2.1 Requirements tracing

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1
- FDP_ACF.1

[6] The Re-assign permission allows an unconstrained form of the Peer Assignment operation if the subject's rank is high, which is why it utilizes the same permission. The ranks of the participants involved modify what is allowed with this permission.

#### 6.1.2.2 IT Environment support

This TOE security function is supported by security functionality located in the IT environment. FIA_UID.2b and FIA_UAU.2b support this security function by providing authenticated user identities that are used by the TOE for developer access control to Studio. Transaction management in the IT environment is provided by the Enterprise Java Beans mechanism.

### 6.1.3 Identification and Authentication

#### 6.1.3.1 User identification and authenication

The Process Execution Engine requires a user to log in through the login dialog using username and password when using the WorkSpace. User identification and authentication is performed by an internal framework called FDI (Fuego Directory Interface), which is the internal interface to the directory service that is configured for the TOE. The directory server stores security credentials of the TOE users (administrators and participants).

Users accessing the TOE as participants through the WorkSpace are authenticated via an instantiation of the FDI within that component. Users accessing the TOE as administrators are authenticated via an instantiation of the FDI implemented through the Process Administrator component.

In other words, the TOE requires users to provide a unique user name and password prior to accessing services to the Process Administrator, Archive View, WorkSpace Administrator, and WorkSpace or any custom application using any of the available public APIs (PAPI or PAPI-WS) prior to users being granted logical access. [FIA_UAU.2a, FIA_UID.2a]

Users in the developer role have access to the Studio component, which are hosted on a separate workstation. Authentication for these users is performed via the local operating system authentication mechanism. Developers typically do not interact with the TOE's runtime environment directly. Also, the IT environment identifies and authenticates the administrators' access Log Viewer and Admin Center.

#### 6.1.3.2 User attributes

An end user of BEA AquaLogic BPM is called a participant. The TOE maintains information about participants including the user name and security attributes used to enforce the TSP. The following security-relevant attributes are maintained for participants [FIA_ATD.1]. Creating, modifiying, deleting and querying user accounts is restricted to BEA AquaLogic BPM administrator (FMT_MTD.1e.1).

- User name – a user name to associate an external user name with participant sessions
- Password – a unique password to associate with a unique user name
- Organizational units – the highest organizational identifier for the participant
- Organizational groups – a set of groups within the organizational unit to which the participant belongs
- Organizational roles – a set of roles (which define capabilities) to which the participant belongs.
- For each organizational role, category within the role (which define delegation, re-assignment, escalation, and peer assignment limits within the role)
- For each role, permissions associated with that role.

#### 6.1.3.3 User subject binding

The Process Execution Engine maintains a session to represent an interaction with an end user. Session information includes session identification, participant identification, and the organizational unit and roles associated with the user. The same information is maintained on the API Client side.

User attributes are bound to TOE subjects after the user completes I&A and establishes a participant session. When the participant session is created, organizational units, process roles, permissions associated with each role, and a hierarchical category associated with each role are assigned to the subject. These attributes are static for the lifetime of the subject. Changes to user security attributes do not take effect until the user logs in again and creates a new participant session.

Each participant belongs to an Organization or Organizational Unit. He or she can only perform tasks on processes deployed in that organizational unit or any of the lower levels within the organizational unit's hierarchy.

Each participant has zero or more assigned organizational roles. A participant's roles determine the participant's capabilities.

Each participant belongs to zero or more groups. A participant inherits all the roles defined for groups to which he or she belongs.

Each participant is assigned a role category and permissions for process instance assignment. BEA AquaLogic BPM uses participants' categories and assignment permissions to determine whether a participant may delegate, escalate, or re-assign a process instance. [FIA_USB.1]

#### 6.1.3.4 Requirements tracing

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_ATD.1
- FIA_UAU.2a
- FIA_UID.2a
- FIA_USB.1

#### 6.1.3.5 IT Environment support

The IT environment identifies and authenticates (FIA_UID.2b and FIA_UAU.2b) BEA AquaLogic BPM developers using the Studio component and BPM administrators prior to accessing the Admin Center and Log Viewer.

### 6.1.4 Security Management

The TOE provides two tpes of administrators: there are two types of administrators: a system administrator, called simply an "administrator", and a more restricted type of administrator, called an "end user administrator". A system administrator has access to the entire project, whereas an end user administrator has privileges limited to creating and modifying participants and monitoring processes". The TOE also supports Developer and the BPM participant roles. A BEA AquaLogic BPM administrator performs most of the typical administrative tasks (as described below). A BEA AquaLogic BPM developer (sometimes referred to as a business analyst and business architect) designs and implements business processes with the aid of the Studio component. Part of this design task is to assign abstract roles used for access control to the processes' activity instances, which are the top-level decomposition units of the business process.[7] [FMT_SMR.1] [FMT_SMF.1e]

There are two ways for a BEA AquaLogic BPM developer to create a new abstract process role.[8] Using the organization window within Studio, a BEA AquaLogic BPM developer can create an abstract role to be used in the entire project. The second way is for the BEA AquaLogic BPM developer to create an abstract role while designing a process. From that moment on, that role can be referred to anywhere in the project. A developer has the ability to query, modify, and delete abstract process roles within the Developer and Studio applications. [FMT_MTD.1b]

Developers typically do not integrate with the Runtime part directly. Developers work locally with their projects and then they can (locally) publish/export the project to the directory service where the embedded Process Execution Engine can get the information from. The actual publication/deployment phase is administrator-controlled and can be done in one of three different ways. Using the Process Administrator component, an administrator can publish and deploy a project: a) from an export file or from a local filesystem directory. [FMT_MTD.1a]

Publishing a project has two main purposes:

- Each process is prepared for deployment to the end users' real-time environment.
- BP-method (business logic) is compiled to Java classes.

During deployment:

- The process is associated to a certain Organizational Unit
- The Engine is notified that a new process (or a new version of an already deployed process) is available to users associated to that Organizational Unit, so that they can begin working with it.

[7] An activity instance is a grouping of individual tasks, which are defined at the next decomposition level.
[8] All references to "abstract roles" in the ST should be interpreted as "abstract process roles."

Notwithstanding process design and development activities, administrators perform all other administrative functions through one of the following four components: Admin Center, Process Administrator, Log Viewer, or Archive Viewer. In the evaluated configuration, all administrator communication with the latter three components is performed via HTTPS. In the case of the Admin Center, the administrator must login to the host of the runtime components of the TOE, where the Admin Center is hosted. In this case, access to the TOE component is local, so no transport security is necessary.

The published and deployed processes start to execute in a runtime environment after starting the Process Execution Engine where they have been deployed. After being deployed, an administrator can query the system to examine which supervisory applications are deployed, and selectively "undeploy" any supervisory application if for any reason (e.g., to replace it with an updated version of the business process). [FMT_MTD.1a] [FMT_SMF.1d] [FMT_SMF.1g]

A BEA AquaLogic BPM administrator can start and stop a Process Execution Engine from the Process Administrator. When the Process Execution Engine is stopped, all the processes stop executing. This means that no task is executed. Participants will not be able to connect to their WorkSpaces or custom programs using available APIs to the processes deployed to the stopped Process Execution Engine. [FMT_MOF.1] [FMT_SMF.1a]

A BEA AquaLogic BPM developer creates business processes which are translated into supervisory applications by publishing and deploying the project file. Publishing and deploying can only be done by BEA AquaLogic BPM Administrators. Each business process represented by a supervisory application is broken down into logical steps called activities. The BEA AquaLogic BPM developer (during the development process) assigns each activity an abstract role, which indicates who may perform the specified activity. Administrators control the mapping of abstract roles to operational roles during the publication and deployment process. [FMT_MTD.1b]

When a process is deployed in one of the organizational units, it becomes visible for the participants belonging to that organizational unit and for those belonging to any of the organizational units located in lower levels of the same branch within the organization hierarchy. It is also necessary to have the right roles assigned to have visibility to different areas of the process as defined in the business process at modeling time. Process Administrator validates the fact that a single process is deployed only once in the same branch of the organization tree. When a process is deployed in the root Organization, it becomes visible for all the organizational units. [FMT_SMF.1g]

A BEA AquaLogic BPM administrator is able to specify security attributes for participants in Process Administrator. At that time, abstract roles defined during the development process are mapped to actual roles that are relevant in the runtime environment. Similarly, user accounts can be created and the security attributes of participants queried or modified by assigning roles (with specific permissions and category attributes), and/or changing the organizational unit to which the individual belongs. [FMT_SMF.1b] [FMT_MSA.1a] [FMT_MSA.3]

The Participant view in the Organization window contained in the Process Administrator allows a BEA AquaLogic BPM administrator to create, modify, and delete participants in the organization. Organizational units are, typically, departments or divisions within an organization, which may be organized in a hierarchy. A BEA AquaLogic BPM administrator deploys processes and assigns participants to organizational units. A process can be deployed for one of the organizational units so that only participants in that organizational unit and in lower levels within the hierarchy are able to perform tasks in that process. [FMT_MTD.1c] [FMT_MTD.1d] [FMT_MTD.1e] [FMT_MSA.1b] [FMT_MSA.3] [FMT_SMF.1c] [FMT_SMF.1f]

BEA AquaLogic BPM maintains both organizational and process roles. A role in a process (abstract role) defines a job function for work. Roles defined in the process are stored independently of the role information defined in the organization (organizational role). This separation allows designers to develop processes as templates making them reusable in different organizations. However, in order to ease process deployment, the Process Administrator ensures that an organizational role always exists for each abstract role of the process and handles the mapping between them automatically using the role names. It is also possible to create any abstract role/organizational role mapping as the deployment requires for a given organization structure. It is not mandatory to have a 1-to-1 relationship between abstract and organizational roles. [FIA_ATD.1] [FMT_MTD.1e] [FMT_MSA.1b] [FMT_MSA.3] [FMT_SMF.1c]

A group is a profile. Groups can be defined in Studio (development time) as well as through the Process Administrator (production deployment time). The TOE restricts the capability to assign participants to groups to the BEA AquaLogic BPM administrator. Participants in a group to provide them with the abilities defined for the group. A set of roles is assigned to the group. When participants log in to WorkSpace, the groups to which the participants

belong are checked in order to determine the final set of roles they play within the organization. This means that the participant inherits all the roles defined for the groups he or she belongs to. [FMT_SMF.1e]

The Admin Center is used to perform the following: starting and stopping the applications server, configuring the directory service, configuring individual web applications, configuring the Process Administrator component, configuring WorkSpace settings, viewing various (performance-related) logs, and applying maintenance packs. [FMT_SMF.1]

#### 6.1.4.1 Requirements tracing

The Security Management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1
- FMT_MSA.1a
- FMT_MSA.1b
- FMT_MSA.3
- FMT_MTD.1a
- FMT_MTD.1b
- FMT_MTD.1c
- FMT_MTD.1d
- FMT_MTD.1e
- FMT_SMF.1
- FMT_SMR.1

#### 6.1.4.2 IT Environment support

No support is provided for this security function by the IT environment.

### 6.1.5 Protection of the TOE Security Functions

All changes to the Business Processes data and state of the system are centralized through the Process Execution Engine. The Engine authenticates the subject and validates authorization for every request to perform an operation. No user interface or API allows a user to do any of these operations without going to the Engine. [FPT_SEP.1a]

The Engine maintains a session to represent an interaction with an end user. This includes interaction with the user through WorkSpace and for each web service application that performs web service operations. The Process Execution Engine prevents one session from interfering with other sessions. The session mechanism also helps ensure that the security functions are invoked and succeed before the BEA AquaLogic BPM provides service to an end user. [FPT_RVM.1a]

In addition, to ensure the TOE data is adequately protected, the TOE must be installed, configured, and maintined using the Installation and Administrative guides and documents. The Installation Guide identifies the installation prerequisites, installation procedures, and the configuration settings that ensure the TOE is installed in its evaluated configuration. The Administrators Guide describes the interfaces that are used to manage the TOE and its functions in a secure manner. The Users Guide identifies the functions that are available to the non-administrative user. All of the guides are available at http://e-docs.bea.com/albsi/docs60/.

#### 6.1.5.1 Requirements tracing

The Protection of the TOE Security Functions function is designed to satisfy the following security functional requirements:

- FPT_RVM.1a
- FPT_SEP.1a

#### 6.1.5.2 IT Environment support

This TOE security function is supported by security functionality located in the IT environment as provided by FPT_SEP.1b and FPT_RVM.1b ensure, respectively, that the TSF (as well as other processes in the IT environment) are provided protected domains for their own execution and that the IT environment's security policy is enforced effectively.

IT environment security is about the "physical and logical" protection of the system, outside of the product scope. That is, the protections provided by the IT environment include:

- The underlying operating system and supporting applications to be installed, configured, and maintained following the product guidance documents. The TOE relies on the IT environment to provide protection of the TOE and its data and resources. The TOE also relies on the IT environment to enforce users are properly identified and authenticated and all security policy functions are invoked and succeed before each function within the IT environment's scope of control is allowed to proceed. The prerequsites, configuration, and requirements can be found at http://e-docs.bea.com/albsi/docs60/installguide/index.html
- Preventing external connections to the database servers. For example, the Engine must be the only entity allows to change the state of the Engine database
- Preventing unauthorized access to the servers' file systems and config files.
- Preventing some users from accessing the client apps completely—for example, preventing them from even seeing the log-in screens. For example, solutions for internal employees should not be publicly accessible
- Preventing tampering with network links and equipment

## 6.2 TOE Security Assurance Measures

### 6.2.1 Configuration management

The TOE's Configuration Management Plan (CMP) identifies the measures applied by BEA Systems to ensure that configuration items are uniquely identified and that documented procedures are used to control and track changes that are made to the TOE. BEA Systems uses the Subversion source control system to ensure that changes to the implementation representation are controlled. BEA Systems performs configuration management on the TOE implementation representation (source code), design documentation, tests, user and administrator guidance, vulnerability analysis documentation, and configuration management documentation (the CMP itself).

These activities are documented in:

- BEA AquaLogic BPM Suite Configuration Management Plan

The Configuration management assurance measure will satisfy the following assurance requirements:

- ACM_CAP.2

### 6.2.2 Delivery and operation

The TOE's delivery documentation identifies the TOE and explains how the TOE is delivered, the carriers utilized, and the procedures and processes implemented to maintain security when the TOE is distributed to the user's site. It documents their system control and distribution facilities and the procedures that provide assurance that the recipient receives the TOE that the sender intended to send, without any modifications. It allows the receiver to verify that what was received corresponds precisely to the TOE, thus detecting any tampering with the actual version, or substitution of a false version.

BEA AquaLogic BPM Suite Installation, generation, and start-up procedures explain how the TOE is installed, generated, and started up in a secure manner, as intended by BEA Systems. They show a secure transition from the TOE's implementation representation, under configuration control, to its initial operation in the customer environment.

These activities are documented in:

- BEA AquaLogic BPM Suite Version 6.0 Delivery and Operation
- Installation Guide for AquaLogic BPM

The Delivery and operation assurance measure will satisfy the following assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

### 6.2.3 Development

BEA Systems is preparing documents describing the design of the TOE. This will include a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; and correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the subsystems.

These activities will be documented in:

- BEA AquaLogic BPM Suite Functional Specification
- BEA AquaLogic BPM Suite High-level Design
- BEA AquaLogic BPM Suite Design Correspondence Analysis

The Development assurance measure will satisfy the following assurance requirements:

- ADV_FSP.1
- ADV_HLD.1
- ADV_RCR.1

### 6.2.4 Guidance documents

The TOE's guidance documentation describes the product in a way that ensures that administrators and users understand the operation environment. They provide the administrators with detailed, accurate information on how to administer the TOE in a secure manner and how to make effective use of the TSF privileges and protection functions.

These activities are documented in the following documents, which can be obtained or accessed at the following location: http://edocs.bea.com/albsi/docs60/ :

- Release Notes for AquaLogic BPM 6.0
- Installation Guide
- Upgrade Guide
- ALBPM Process API
- API Reference
- API Changes from Version 5.7
- Configuration Guide (Standalone Edition)
- Configuration Guide (WebLogic Edition)
- Configuration Guide (WebSphere Edition)
- Process Administrator
- Admin Center
- Log Viewer

- Archive Viewer
- WorkSpace
- WorkSpace Administrator
- Studio

The BEA AquaLogic BPM Suite guidance documents will satisfy the following assurance requirements:

- AGD_ADM.1
- AGD_USR.1

### 6.2.5 Life cycle support

BEA Systems will provide documents describing their established procedures and methods for tracking security flaws, identifying corrective actions, and distributing corrective action information to TOE users. TOE users, developers and engineers can report flaws at any time during the TOE's life cycle.

These activities will be documented in:

- BEA AquaLogic BPM Suite Life-cycle Plan

The Life cycle support assurance measure will satisfy the following assurance requirements:

- ALC_FLR.1

### 6.2.6 Tests

BEA Systems has test suites to test the functions of the TOE, although exhaustive specification testing of the interfaces is not required. It shows that each security function has been sufficiently tested against the behavioral claims in the functional specification and high-level design.

These activities will be documented in:

- BEA AquaLogic BPM Suite Test Plan
- BEA AquaLogic BPM Suite Test Coverage Analysis
- BEA AquaLogic BPM Suite version 6.0 Test Results

The Tests assurance measure will satisfy the following assurance requirements:

- ATE_COV.1
- ATE_FUN.1
- ATE_IND.2

### 6.2.7 Vulnerability assessment

The TOE's Vulnerability Assessment will show that the existence and exploitability of flaws or weaknesses in the TOE in the intended environment, documenting the systematic search for vulnerabilities in the TOE, and providing an assessment of vulnerabilities found to determine their relevance to the intended environment for the TOE.

The TOE's Strength of Function (SOF) analysis may be addressed in the Vulnerability Assessment will shows that the SOF claims made in the ST for all probabilistic or permutational mechanisms are supported by an analysis.

The strength of function claim is: SOF-Basic. The SOF claim is mapped to Identification and authentication security function, more specifically FIA_UAU.2a security functional requirement.

These activities will be documented in:

- BEA AquaLogic BPM Suite Vulnerability Assessment

The Vulnerability assurance measure will satisfy the following assurance requirements:

- AVA_SOF.1
- AVA_VLA.1

---

# 7. Protection Profile Claims

The TOE does not make a PP conformance claim.

# 8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

## 8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

### 8.1.1 Complete Coverage – Environmental Assumptions

This section provides shows coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

| Assumption | Objective |
|---|---|
| A.CONFIG | OE.CONFIG |
| A.COOP_USER | OE.COOP_USER |
| A.LOCATE | OE.LOCATE |
| A.NO_EVIL | OE.NO_EVIL |
| A.SECURE_HOST | OE.SECURE_HOST |

**Table 7 Tracing between Assumptions and Objectives**

#### 8.1.1.1 A.CONFIG

*The TOE will be configured by authorized administrators such that the access control policy supports the organization's security policy.*

This Assumption is met by ensuring that:

- OE.CONFIG: Those responsible for the TOE confirm A.CONFIG holds in their operational environment. OE.CONFIG requires authorized administrators to configure the TOE's access control policy and enforcement mechanism to support the organization's security policy, which satisfies A.CONFIG.

#### 8.1.1.2 A.NO_EVIL

*The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation.*

This Assumption is met by ensuring that:

- OE.NO_EVIL: Those responsible for the TOE confirm A.NO_EVIL holds in their operational environment. OE.NO_EVIL requires that those responsible for managing the TOE shall ensure that the TOE is installed, configured, managed and maintained in accordance with its guidance documentation, which satisfies OE.NO_EVIL.

#### 8.1.1.3 A.SECURE_HOST

*The IT infrastructure on which the BEA AquaLogic BPM security functions depend will be installed, configured, managed, and maintained in accordance with both BEA AquaLogic BPM and IT infrastructure guidance documentation.*

This Assumption is met by ensuring that:

- OE.SECURE_HOST: Those responsible for the TOE and its IT environment confirm A.SECURE_HOST holds in their operational environment. OE.SECURE_HOST requires that those responsible for managing the TOE and its IT environment shall ensure that the IT infrastructure on which the TOE security functions depend is installed, configured, managed, and maintained in accordance with both TOE and IT infrastructure guidance documentation, which satisfies A.SECURE_HOST.

#### 8.1.1.4 A.LOCATE

*The BEA AquaLogic BPM server and supporting IT infrastructure servers will be located within controlled access facilities, which will prevent unauthorized physical access.*

This Assumption is met by ensuring that:

- OE.LOCATE: Those responsible for the TOE and its IT environment confirm A.LOCATE holds in their operational environment. OE.LOCATE requires that those responsible for the TOE and its IT environment shall locate the TOE server and supporting IT infrastructure servers within controlled access facilities, which will prevent unauthorized physical access, which satisfies A.LOCATE.

#### 8.1.1.5 A.COOP_USER

*All users will protect their authentication information from disclosure and their IT resources (i.e. web browser and host computer) from tampering.*

This Assumption is met by ensuring that:

- OE.COOP_USER: Those responsible for the TOE confirm A.NO_EVIL holds in their operational environment. OE.COOP_USER requires that all users shall protect their authentication information from disclosure and their IT resources (i.e. web browser and host computer) from tampering, which satisfies A.COOP_USER.

### 8.1.2 Complete Coverage – Organizational Security Policies

This section shows that all organizational security policies are completely covered by the TOE security objectives and that each objective counters or addresses at least one policy.

| Policy | Objective |
|---|---|
| P.ACCOUNT | O.AUDIT_GEN |
| P.ACCOUNT | O.AUDIT_REVIEW |

| Policy | Objective |
|---|---|
| P.ACCOUNT | OE.AUDIT_STORE |
| P.ACCOUNT | OE.EXTERNAL_IA |
| P.ACCOUNT | OE.TIME_STAMPS |

**Table 8 Tracing between Policies and Objectives**

#### 8.1.2.1 P.ACCOUNT

*The authorized users of BEA AquaLogic BPM shall be held accountable for their actions within the TOE.*

This Policy is satisfied by ensuring that:

- O.AUDIT_GEN: The TSF generates records of security-relevant actions by users, which serves as the basis for accountability.
- O.AUDIT_REVIEW: The TSF presents the record of security-relevant actions by users in a manner that BEA AquaLogic BPM administrators can understand, so they can act on inappropriate actions by users.
- OE.AUDIT_STORE: The IT environment provides audit storage to store audit records generated by the TSF and as a repository for use by the TOE audit review function.
- OE.EXTERNAL_IA: The IT environment identifies and authenticates developers prior to providing Studio services to verify identity.
- OE.TIME_STAMPS: The IT environment includes time information in the record of security-relevant actions to facilitate analyzing the record.

### 8.1.3 Complete Coverage – Threats

This section shows that all threats are completely covered by the TOE security objectives and that each objective counters or addresses at least one threat.

| Threat name | Objective name |
|---|---|
| T.ACCESS | O.MEDIATE |
| T.ACCESS | OE.CONFIG |
| T.ACCESS | OE.EXTERNAL_IA |
| T.MASQUERADE | O.TOE_ACCESS |
| T.MASQUERADE | OE.COOP_USER |
| T.MASQUERADE | OE.EXTERNAL_IA |
| T.TSF_COMP | O.MANAGE |
| T.TSF_COMP | O.SELF_PROTECT |
| T.TSF_COMP | OE.TSF_PROTECT |
| T.TSF_RECONFIG | O.MANAGE |

**Table 9 Tracing between Threats and Objectives**

#### 8.1.3.1 T.ACCESS

*By using a BEA AquaLogic BPM service in its intended manner, an attacker accesses (i.e., reads or modifies) user data related to business processes for which the attacker is not authorized according to the organization's policy*

This Threat is countered by ensuring that:

- O.MEDIATE: The TSF prevents T.ACCESS by eliminating the attack method. The TSF restricts access by users to data in accordance with the organizations security policy.
- OE.CONFIG: Those responsible for the TOE confirm A.CONFIG holds in their operational environment. The access control policy and enforcement mechanism are configured such that the access control enforced by the TOE represents the controls intended by the organization's security policy.
- OE.EXTERNAL_IA: The IT environment identifies and authenticates developers prior to providing Studio services to verify identity.

#### 8.1.3.2 T.MASQUERADE

*An attacker masquerades as an authorized user in order to gain access to business process information for which the attacker is not authorized according to the organization's policy.*

This Threat is countered by ensuring that:

- OE.COOP_USER: All users prevent T.MASQUERADE by eliminating an attack method. Each user protects his or her authentication data so that others may not use that data to access the TSF.
- O.TOE_ACCESS: The TSF prevents T.MASQUERADE by eliminating attack methods. The TSF requires identification and authentication of each user before providing services to the Process Administrator, Admin Center, Archive Viewer, WorkSpace Administrator and Workspace prior to user's logical access to the TOE.
- OE.EXTERNAL_IA: The IT environment identifies and authenticates developers prior to providing Studio services to verify identity.

#### 8.1.3.3 T.TSF_COMP

*By using a BEA AquaLogic BPM service in an unintended manner, an attacker accesses business process information for which the attacker is not authorized according to the organization's policy*

This Threat is countered by ensuring that:

- O.MANAGE: The TSF prevents T.TSF_COMP by preventing attack methods. The TSF management functions and facilities provide a BEA AquaLogic BPM administrator with adequate tools to correctly configure the behavior of the TSF services.
- O.SELF_PROTECT: The TSF prevents T.TSF_COMP by preventing attack methods. The TSF ensures that its security functions are invoked. The TSF ensures that an attacker cannot use TSF services to tamper or interfere with the TSF itself or other users. Also, the TOE protects data transferred between TOE components.
- OE.TSF_PROTECT: The IT environment prevents T.TSF_COMP by prevents attack methods. The IT environment supports the TSF's self-protection mechanism. The IT environment prevents an attacker using the IT infrastructure to bypass the TSF or to tamper with the TSF configuration and executables.

#### 8.1.3.4 T.TSF_RECONFIG

*By using a BEA AquaLogic BPM management service, an attacker modifies the server configuration or organizational information to allow subsequent attacks to succeed.*

This Threat is satisfied by ensuring that:

- O.MANAGE: The TSF prevents T.TSF_RECONFIG by eliminating the attack method. The TSF restricts management services to BEA AquaLogic BPM administrators and authorized users.

## 8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 9** indicates the requirements that effectively satisfy the individual objectives.

### 8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFRs) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective(s) that it is intended to satisfy.

| Objective | Security Functional Requirement | |
|---|---|---|
| O.AUDIT_GEN | FAU_GEN.1 | Audit data generation |
| O.AUDIT_REVIEW | FAU_SAR.1 | Audit review |
| O.AUDIT_REVIEW | FAU_SAR.3 | Selectable audit review |
| O.MANAGE | FMT_MOF.1 | Management of Security Functions Behavior |
| O.MANAGE | FMT_MSA.1a | Management of security attributes: Activity security attributes |
| O.MANAGE | FMT_MSA.1b | Management of security attributes: User security attributes |
| O.MANAGE | FMT_MSA.3 | Static attribute initialization |
| O.MANAGE | FMT_MTD.1a | Management of TSF data: Supervisory applications |
| O.MANAGE | FMT_MTD.1b | Management of TSF data: Roles |
| O.MANAGE | FMT_MTD.1c | Management of TSF data: Organizational information |
| O.MANAGE | FMT_MTD.1d | Management of TSF data: Organizational information |
| O.MANAGE | FMT_MTD.1e | Management of TSF data: User account information |
| O.MANAGE | FMT_SMF.1 | Specification of management functions |
| O.MANAGE | FMT_SMR.1 | Security Roles |
| O.MEDIATE | FDP_ACC.1 | Subset access control |
| O.MEDIATE | FDP_ACF.1 | Security attribute based access control |
| O.SELF_PROTECT | FPT_RVM.1a | Non-bypassability of the TSF |
| O.SELF_PROTECT | FPT_SEP.1a | TSF domain separation |
| O.TOE_ACCESS | FIA_ATD.1 | User Attribute Definition |
| O.TOE_ACCESS | FIA_UAU.2a | User Authentication Before any Action |
| O.TOE_ACCESS | FIA_UID.2a | User Identification Before any Action |
| O.TOE_ACCESS | FIA_USB.1 | User Subject Binding |
| OE.TIME_STAMPS | FPT_STM.1 | Reliable time stamps |
| OE.TSF_PROTECT | FPT_RVM.1b | Non-bypassability of the ~~TSP~~ **IT environment's security policy** |
| OE.TSF_PROTECT | FPT_SEP.1b | ~~TSF~~ **IT environment** domain separation |
| OE.EXTERNAL_IA | FIA_UID.2b | User identification before any action |

| Objective | Security Functional Requirement | |
|---|---|---|
| OE.EXTERNAL_IA | FIA_UAU.2b | User authentication before any action |
| OE.AUDIT_STORE | FAU_STG.1 | Protected audit trail storage |

**Table 10 Tracing between Objectives and Requirements**

#### 8.2.1.1 O.AUDIT_GEN

*The TOE will provide the capability to detect and create records of security-relevant events associated with users*

This TOE security objective is satisfied by ensuring that:

- FAU_GEN.1 identifies the security-relevant events for the TSF. It specifies the capability to detect these events and create corresponding audit records, which include information needed to associate the events with users.

#### 8.2.1.2 O.AUDIT_REVIEW

*The TOE will provide the capability to selectively view audit information,*

This TOE security objective is satisfied by ensuring that:

- FAU_SAR.1 provides a BEA AquaLogic BPM administrator with the capability to view audit information in an understandable form.
- FAU_SAR.3 provides the capability to select and view meaningful subsets of the audit information.

#### 8.2.1.3 O.MANAGE

*The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.*

This TOE security objective is satisfied by ensuring that:

- FMT_SMF.1 identifies the functions needed to manage the security of the TOE.
- FMT_SMR.1 identifies the roles used to specification restriction on the TOE security management functions.
- FMT_MOF.1 specifies the capability to start and stop the TSF and restricts that capability to BEA AquaLogic BPM administrators.
- FMT_MSA.1a and FMT_MSA.1b provide a BEA AquaLogic BPM administrator with the capabilities to specify business process information and user information, respectively, within the TSC.
- FMT_MSA.3 specifies the default values for subject and object attributes within the scope of the TSF access control policy. These attribute values are configured by a BEA AquaLogic BPM administrator and cannot be overridden at run time.
- FMT_MTD.1a, FMT_MTD.1b, FMT_MTD.1c, and FMT_MTD.1d specify the functions available to a BEA AquaLogic BPM administrator and End user administrator for managing business process and organizational information.
- FMT_MTD.1e specifies the functions available to authorized BEA AquaLogic BPM users for managing user account information.

#### 8.2.1.4 O.MEDIATE

*The TOE must protect user data in accordance with its security policy.*

This TOE security objective is satisfied by ensuring that:

- FDP_ACC.1 identifies the TOE policy for restricting access to business process information in the TSC. It specifies the scope of the policy in terms of active agents (subjects), information containers (objects), and operations between them within the TSC.
- FDP_ACF.1 specifies the rules that the TSF enforces to restrict access to business process information.

#### 8.2.1.5 O.SELF_PROTECT

*The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces; as well as, protects data transferred between TOE components.*

This TOE security objective is satisfied by ensuring that:

- FPT_RVM.1a specifies the property that the TSF security functions cannot be bypassed within the TSC.
- FPT_SEP.1a specifies internal protections for the TSF itself and for subjects within the TSC.

#### 8.2.1.6 O.TOE_ACCESS

*The TOE will provide identification and authentication mechanisms for each user before providing services to the Process Administrator, Admin Center, Archve Viewer, WorkSpace Administrator and WorkSpace prior to user's logical access to the TOE.*

This TOE security objective is satisfied by ensuring that:

- FIA_UAU.2a specifies that every user be authenticated prior to allowing access to TSF services.
- FIA_UID.2a specifies that every user provide an identity before the TSF allows access to TSF services.
- FIA_ATD.1 identifies the attributes that the TSF associates with each user and uses to enforce its security policies.
- FIA_USB.1 binds a user's security attributes to TOE subjects so that access controls can be applied to those subjects.

#### 8.2.1.7 OE.TIME_STAMPS

*The IT environment will provide reliable time stamps for TSF purposes.*

This IT environment security objective is satisfied by ensuring that:

- FPT_STM.1 specifies the capability of the IT environment to provide time stamps.

#### 8.2.1.8 OE.TSF_PROTECT

*The IT environment will maintain domains for its own execution and for TOE execution that protects the IT environment, the TOE, and their resources from interference, tampering, or unauthorized disclosure.*

This IT environment security objective is satisfied by ensuring that:

- FPT_RVM.1b specifies the property that the TSF security functions cannot be bypassed within the scope of control of the IT infrastructure.
- FPT_SEP.1b specifies self protection for each component of the IT infrastructure. Moreover, each component contributes to protecting the TSF.

#### 8.2.1.9 OE.EXTERNAL_IA

*The IT environment will identify and authenticate developers prior to providing Studio services to verify identity.*

This IT environment security objective is satisfied by ensuring that:

- FIA_UID.2b specifies the capability that users will be identified externally.
- FIA_UAU.2b specifies the capability that user identities will be authenticated externally.

#### 8.2.1.10 OE.AUDIT_STORE

*The IT environment will provide the capability to store audit records.*

This IT environment security objective is satisfied by ensuring that:

- FAU_STG.1 specifies the capability that the IT environment will provides storage capability for TOE audit records.

## 8.3 Security Assurance Requirements Rationale

The selected security assurance level is EAL2 augmented with ALC_FLR.1.

EAL2 was selected as the base assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. BEA AquaLogic BPM is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments, it is assumed that attackers will have little attack potential. As such, EAL2 is appropriate to provide the assurance necessary to counter the limited potential for attack.

The base assurance level was augmented with ALC_FLR.1 to reflect quality assurance measures that BEA System employs.

## 8.4 Strength of Functions Rationale

The TOE is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments, it is assumed that attackers will have a low attack potential. As such, strength-of-function of 'SOF-Basic' is appropriate for the intended environment. The SOF-Basic claim is supported by the Identification and authentication security function, more specifically FIA_UAU.2a.

## 8.5 Requirement Dependency Rationale

The following table demonstrates that all dependencies among the claimed security requirements are satisfied and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

| ST Requirement | | Dependency | ST Requirement |
|---|---|---|---|
| FAU_GEN.1 | Audit data generation | FPT_STM.1 | FPT_STM.1 |
| FAU_SAR.1 | Audit review | FAU_GEN.1 | FAU_GEN.1 |
| FAU_SAR.3 | Selectable audit review | FAU_SAR.1 | FAU_SAR.1 |
| FAU_STG.1 | Protected audit trail storage | FAU_GEN.1 | FAU_GEN.1 |
| FDP_ACC.1 | Subset access control | FDP_ACF.1 | FDP_ACF.1 |
| FDP_ACF.1 | Security attribute based access control | FDP_ACC.1 | FDP_ACC.1 |
| FDP_ACF.1 | Security attribute based access control | FMT_MSA.3 | FMT_MSA.3 |
| FIA_UAU.2a | User authentication before any action | FIA_UID.1 | FIA_UID.2a |
| FIA_USB.1 | User-subject binding | FIA_ATD.1 | FIA_ATD.1 |
| FMT_MOF.1 | Management of Security Functions Behavior | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MOF.1 | Management of Security Functions Behavior | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MSA.1a | Management of security attributes: Activity security | FDP_ACC.1 | FDP_ACC.1 |

| ST Requirement | | Dependency | ST Requirement |
|---|---|---|---|
| | attributes | | |
| FMT_MSA.1a | Management of security attributes: Activity security attributes | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MSA.1a | Management of security attributes: Activity security attributes | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MSA.1b | Management of security attributes: User security attributes | FDP_ACC.1 | FDP_ACC.1 |
| FMT_MSA.1b | Management of security attributes: User security attributes | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MSA.1b | Management of security attributes: User security attributes | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MSA.3 | Static attribute initialization | FMT_MSA.1 | FMT_MSA.1a–b |
| FMT_MSA.3 | Static attribute initialization | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1a | Management of TSF data: Supervisory applications | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1a | Management of TSF data: Supervisory applications | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MTD.1b | Management of TSF data: Roles | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1b | Management of TSF data: Roles | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MTD.1c | Management of TSF data: Organizational information | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1c | Management of TSF data: Organizational information | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MTD.1d | Management of TSF data: Organizational information | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1d | Management of TSF data: Organizational information | FMT_SMF.1 | FMT_SMF.1 |
| FMT_MTD.1e | Management of TSF data: User account information | FMT_SMR.1 | FMT_SMR.1 |
| FMT_MTD.1e | Management of TSF data: User account information | FMT_SMF.1 | FMT_SMF.1 |
| FMT_SMR.1 | Security Roles | FIA_UID.1 | FIA_UID.2a |

**Table 11 Dependency Analysis**

The assurance dependency rationale first invokes the internal consistency of the EAL2 CC SAR package. The assurance dependency rationale for the augmentation ALC_FLR.1 included in this ST is that ALC_FLR.1 has no dependencies.

## 8.6 Explicitly Stated Requirements Rationale

There are no explicitly stated requirements specified in this document.

## 8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to

provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 12 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

| Security Functional Requirement | | Security Function |
|---|---|---|
| FAU_GEN.1 | Audit data generation | Security Audit |
| FAU_SAR.1 | Audit review | Security Audit |
| FAU_SAR.3 | Selectable audit review | Security Audit |
| FMT_MOF.1 | Management of Security Functions Behavior | Security Management |
| FMT_MSA.1a | Management of security attributes: Activity security attributes | Security Management |
| FMT_MSA.1b | Management of security attributes: User security attributes | Security Management |
| FMT_MSA.3 | Static attribute initialization | Security Management |
| FMT_MTD.1a | Management of TSF data: Supervisory applications | Security Management |
| FMT_MTD.1b | Management of TSF data: Roles | Security Management |
| FMT_MTD.1c | Management of TSF data: Organizational information | Security Management |
| FMT_MTD.1d | Management of TSF data: Organizational information | Security Management |
| FMT_MTD.1e | Management of TSF data: User account information | Security Management |
| FMT_SMF.1 | Specification of management functions | Security Management |
| FMT_SMR.1 | Security Roles | Security Management |
| FDP_ACC.1 | Subset access control | User Data Protection |
| FDP_ACF.1 | Security attribute based access control | User Data Protection |
| FPT_RVM.1a | Non-bypassability of the TSF | Protection of the TOE Security Functions |
| FPT_SEP.1a | TSF domain separation | Protection of the TOE Security Functions |
| FIA_ATD.1 | User Attribute Definition | Identification and Authentication |
| FIA_UAU.2a | User Authentication Before any Action | Identification and Authentication |
| FIA_UID.2a | User Identification Before any Action | Identification and Authentication |
| FIA_USB.1 | User Subject Binding | Identification and Authentication |

**Table 12 Security Functions vs. Requirements Mapping**

## 8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.