AquaLogic® Interaction 6.1 MP1 Patch 2 with AquaLogic® Interaction Development Kit 6.0 Security Target

Version 1.0
04/23/2008

Prepared for:
BEA Systems, Inc
475 Sansome Street, 15th Floor
San Francisco, California 94111

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

1. SECURITY TARGET INTRODUCTION 4

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 4
1.2 CONFORMANCE CLAIMS 4
1.3 CONVENTIONS, ACRONYMS 5
1.3.1 Conventions 5

2. TOE DESCRIPTION 6

2.1 TOE OVERVIEW 6
2.2 TOE ARCHITECTURE 8
2.2.1 Physical Boundaries 9
2.2.2 Logical Boundaries 10
2.3 TOE DOCUMENTATION 11

3. SECURITY ENVIRONMENT 12

3.1 ORGANIZATIONAL POLICIES 12
3.2 ASSUMPTIONS 12

4. SECURITY OBJECTIVES 13

4.1 SECURITY OBJECTIVES FOR THE TOE 13
4.2 SECURITY OBJECTIVES FOR THE IT ENVIRONMENT 13
4.3 SECURITY OBJECTIVES FOR THE ENVIRONMENT 13

5. IT SECURITY REQUIREMENTS 14

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 14
5.1.1 Security audit (FAU) 14
5.1.2 User data protection (FDP) 15
5.1.3 Identification and authentication (FIA) 16
5.1.4 Security management (FMT) 16
5.1.5 Protection of the TSF (FPT) 17
5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS 17
5.2.1 Identification and authentication (FIA) 18
5.2.2 Protection of the TSF (FPT) 18
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 19
5.3.1 Configuration management (ACM) 19
5.3.2 Delivery and operation (ADO) 19
5.3.3 Development (ADV) 20
5.3.4 Guidance documents (AGD) 20
5.3.5 Life cycle support (ALC) 21
5.3.6 Tests (ATE) 22
5.3.7 Vulnerability assessment (AVA) 22

6. TOE SUMMARY SPECIFICATION 24

6.1 TOE SECURITY FUNCTIONS 24
6.1.1 Security audit 24
6.1.2 User data protection 25
6.1.3 Identification and authentication 31
6.1.4 Security management 32
6.1.5 Protection of the TSF 33
6.2 TOE SECURITY ASSURANCE MEASURES 33
6.2.1 Configuration management 34
6.2.2 Delivery and operation 34
6.2.3 Design Documentation 35
6.2.4 Guidance documents 35
6.2.5 Life cycle support 36

6.2.6 Tests 36
6.2.7 Vulnerability assessment 37

7. PROTECTION PROFILE CLAIMS 38

8. RATIONALE 39

8.1 SECURITY OBJECTIVES RATIONALE 39
8.1.1 Security Objectives Rationale for the TOE and Environment 39
8.2 SECURITY REQUIREMENTS RATIONALE 41
8.2.1 Security Functional Requirements Rationale 41
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE 44
8.4 STRENGTH OF FUNCTIONS RATIONALE 44
8.5 REQUIREMENT DEPENDENCY RATIONALE 44
8.6 EXPLICITLY STATED REQUIREMENTS RATIONALE 45
8.7 TOE SUMMARY SPECIFICATION RATIONALE 45
8.8 PP CLAIMS RATIONALE 46

APPENDIX A: ACRONYMS 47

LIST OF TABLES

Table 2-1: ALI IT Requirements for Evaluated Configurations 9
Table 2-2: IDK IT Requirements of Evaluated Configurations 10
Table 5-1 TOE Security Functional Components 14
Table 5-2 IT Environment Security Functional Components 18
Table 5-3 EAL 2 augmented with ALC_FLR.2 Assurance Components 19
Table 6-1: Default Portal Groups and Users 26
Table 6-2 Default Folders in the Administrative Objects Directory 26
Table 6-3 ALI Objects 27
Table 8-1 Environment to Objective Correspondence 39
Table 8-2 Objective to Requirement Correspondence 41
Table 8-3 Requirement Dependencies 45
Table 8-4 Security Functions vs. Requirements Mapping 46

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 AquaLogic® Interaction 6.1 MP1 Patch 2 with AquaLogic® Interaction Development Kit 6.0 provided by BEA Systems, Inc.

The Security Target contains the following additional sections:

- TOE Description (Section 2): 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.
- Security Environment (Section 3): This section describes the assumptions and organizational security policies that pertain to the TOE.
- Security Objectives (Section 4): This section details the security objectives of the TOE and its environment.
- IT Security Requirements (Section 5): This section presents the security functional requirements (SFRs) for the TOE and the IT Environment that supports the TOE. The section also details the requirements for EAL 2 augmented with ALC_FLR.2.
- TOE Summary SpecificationTOE Summary Specification (Section 6): This section describes the security functions represented in the TOE that satisfy the security requirements.
- Protection Profile Claims (Section 7): This section identifies the Protection Profile Claim made in the ST.
- Rationale (Section 8): This section presents the rationale for the security objectives, requirements, and TOE Summary Specifications (TSS) as to their consistency, completeness and suitability.
- Acronyms (Appendix A): This section provides the definition for acronyms used in the ST.

1.1 Security Target, TOE and CC Identification

ST Title – AquaLogic® Interaction 6.1 MP1Patch 2 with AquaLogic® Interaction Development Kit 6.0 Security Target

ST Version – Version 1.0

ST Date – April 23, 2008

TOE Identification – AquaLogic® Interaction 6.1 MP1 Patch 2 with AquaLogic® Interaction Development Kit 6.0

TOE Developer – BEA Systems, Inc

Evaluation Sponsor – BEA Systems, Inc (BEA)
.

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 (CC) for Information Technology (IT) Security Evaluation Part 2: Security functional requirements, Version 2.3, August 2005.
    - Part 2 Extended

- Common Criteria (CC) for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 2 augmented with ALC_FLR.2
  - Strength of Function Claim: SOF-basic

1.3 Conventions, Acronyms

This section specifies the formatting information used in the Security Target.

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, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by 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 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 …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- Explicitly stated requirements – Security Functional Requirements not defined in Part 2 of the CC are annotated with EX.

The acronyms are listed in the appendices.

2. TOE Description

The Target of Evaluation (TOE) is AquaLogic Interaction 6.1 MP1 Patch 2 with AquaLogic Interaction Development Kit 6.0, henceforth referred to as ALI and IDK respectively

ALI is the portal platform for the BEA AquaLogic User Interaction (ALUI) suite of products. A portal is a Web site that gives users a single point of access to applications and information in a single unified interface. ALI includes a portal infrastructure, a user interface (UI), a document content management system, and a search function. ALI integrates applications and ALUI components into a cohesive Web-based environment that can be customized and personalized to meet the internal and customer-oriented portal needs of large enterprise companies.

AquaLogic Interaction Development Kit 6.0 (IDK) offers Java platform and .NET client-side libraries that provide connectivity to Web services-based application programming interfaces (APIs) for ALI.

2.1 TOE Overview

AquaLogic Interaction (ALI) is the base portal application and framework for the BEA AquaLogic User Interaction (ALUI) product family. ALI integrates custom-developed applications and ALUI components into a cohesive web-based work environment that is viewed from a user's web browser. Every component of the ALI portal page is customizable, from company branding to navigation and page content. Users can also integrate content or user properties from external systems using content services and identity services products offered by BEA or built with the API provided in the IDK.

ALI's portal framework integrates applications by using portlets and also supports virtual community workspaces. Portlets are one of the mechanisms that end users use for accessing data and applications from portals. Portlets enable the integration of functionality from external systems in the portal page, thus providing a single entry point (or window) for a wide range of content and services. Portlets can be used for everything from displaying useful information to building integrated applications that combine functionality from multiple systems. The ALI portlet architecture conforms to Service-Oriented Architecture (SOA) and leverages the key SOA interfaces and protocols: HTTP and SOAP. SOA is an IT strategy that organizes the discrete functions contained in enterprise applications into interoperable, standards-based services that can be combined and reused quickly to meet business needs.

Most of the ALI portal's end-user and administrative functionality and tools are packaged and implemented as portlets. ALI organizes this functionality into categories and implements each category in a set of portlets. The categories include the following:

- User Interface,
- Web Services,
- User Management,
- Content management,
- Security,
- Search, and
- Scheduled Operations.

Separating the portlet UI from its application logic through a web service interface enables portability and supports ALI's capability to deploy on both J2EE and .NET web environments. Thus, ALI portlet architecture is characterized by the following:

- A front end that implements the portlet's UI;
- A back-end that implements the portlet's functionality; and
- An interface between the front-end and the back-end (often a web service) that generates standard HTML or text output, including but not limited to complete web pages, XML data, or snippets of HTML.

The back end for a portlet can be any web application that returns HTML or XML over HTTP. Portlets can be coded in any language that communicates over HTTP. The code returned by a portlet is parsed by the portal and inserted into the appropriate cell in the HTML table that makes up every portal page. Most portlets are hosted remotely. Each portlet is self-contained and executes its particular functionality in its own process.

The IDK APIs (included with both the Java and .NET versions of the IDK) provide support for portlet development, including manipulating settings, accessing user information, and managing communication with the portal. Security is enforced by the portal in exactly the same way as when the functionality is executed from within the portal. Each My Page or Community page is made up of many portlets with a range of functionality. In the ALUI web services architecture, most portlets are hosted remotely and connect to a back-end application for data or functionality. The remote portlets can access an API provided by the IDK called the Programmable Remote Client (PRC). The PRC API provides interfaces to perform object-oriented access into the portal's SOAP API, which expose elements of the portal API.

The IDK also enables developers to create remote authentication services. The IDK Authentication API provides an abstraction from the necessary SOAP calls and enables developers to simply implement an object interface for the external authentication service.

BEA ALI has an embedded, portal-specific document content management system called the Knowledge Directory. The Knowledge Directory includes facilities for importing and uploading existing documents, for storing, sharing, and managing them, and for helping users find them through search and navigation. The content model for Knowledge Directory is built on content objects of these three predefined types:

- Folders,
- Sub-folders, and
- Documents.

Folders and sub-folders organize documents and aid in their findability. Knowledge Directory supports HTML (web) documents, Microsoft Office documents, Microsoft Exchange documents (emails), Lotus Notes documents and emails, and Documentum documents. In addition, developers can integrate support for additional document types by building custom content services using the ALI IDK.

ALI also includes a Search function. Search manages an index of the terms in its sources. The index is maintained through the execution of search agent update jobs and, for many activities on Knowledge Directory objects, is maintained automatically. Administrators can control the relevance ranking of the search engine by specifying weights on the name, description, and content properties that the search engine uses to find objects that satisfy search queries. Search supports federated search of the local ALI portal and external ALI portals and external portal repositories. A search web service is needed for each external source. ALI provides search web services for the following search sources:

- Knowledge Directory,
- AquaLogic Interaction Collaboration items: project workspaces, documents, discussions, announcements, and task lists,
- AquaLogic Interaction Publisher web content,
- Portal Administrative objects: users, portlets, web services, and content services,
- External content: Documentum repositories, Lotus Notes databases, Microsoft Exchange emails, Microsoft Office documents, Microsoft Windows files, and web sites, and
- External content in sources crawled by custom search web services.

Administrators configure the sources for federated search by creating Federated Search objects in Administrative folders. These objects identify the search web services needed to perform federated searches. ALI Search supports what is called Basic Search and Advanced Search.

Basic Search queries are case insensitive text strings. Terms within search query strings may be separated by the various operators such as the AND operator, the OR operator, the quotation marks operator and others.

Advanced search gives users fine-grained control over search scope and search query terms. To control search scope in advanced search users can restrict search scope to specified items as follows:

- Document or administrative folders and subfolders,

- Object types, and
- Languages.

2.2 TOE Architecture

The TOE consists of two BEA products: AquaLogic Interaction (ALI), an enterprise portal framework useful for both intranet and extranet users and the AquaLogic Interaction Development Kit, a set of APIs enabling developers to customize and integrate their enterprise applications into ALI.

ALI includes the following core components:

- Portal server and Administrative Portal – The following components run on an application server:
  - Portal server – Hosts the dynamically-generated Web pages that users view in the portal, for example My Pages and Community pages. This is the core component of ALI.
  - Administrative Portal – This component provides a centralized management user interface for setup, configuration, security, and other administrative activities.
- Services – The following support components run as Microsoft Windows services or Unix daemon processes depending on the deployment platform:
  - Automation Service – Manages job scheduling for portal administration and maintenance activities. These jobs can include custom jobs created by portal administrators and developers that access remote content services and identity services that store and retrieve information in the portal database.
  - Content Upload Service – The Content Upload Service enables the manual upload (or automatic crawling) of document records from an internal network.
  - Document Repository Service – Stores documents uploaded by ALUI components.
  - API Service – Provides access to the ALI APIs to enable integration with other ALUI applications, as well as third-party applications.
  - Search Service – Maintains the search collection and processes search requests. Returns indexed content from the resources that are accessible though the portal. These resources can include both ALI resources and resources external to ALI.
  - ALI Logger – Provides a logging framework for debugging and diagnostic purposes.
  - Support Components:
    - Image Service – serves images and other static content for use by the ALI. The service uses an area of the local server that stores images and other static content used by the portal and remote services. The Image Service content is non-sensitive data and is stored in a system folder in the local file system. Users of the portal cannot upload content to the Image Service. This component has no role in the TSF of ALI.

The IDK is the external programming interface used for building and implementing user-centric composite applications within the TOE. The IDK provides interfaces for pagelets, portlets and integration web services, such as authentication and profile services, crawlers, and search services.

The TOE depends on the IT environment to provide the file system used by the TOE to protect and store information. Portal information is stored in databases and on disk in the local file system as follows:

- Document Repository – An area of the run-time file system that stores files that are uploaded to the portal.
- Portal database – Tables on a supported database server that store portal administrative data such as object information and security settings.

- Search Server collection – An area of the local server file system that stores indexed ALI document and object data.

2.2.1 Physical Boundaries

The TOE is a portal technology platform, ALI, and its associated development kit, the IDK. Some of the ALI components run on application servers and other components run as Windows services or UNIX daemon processes (depending on the operating system platform). The TOE components are described in section 2.2. The IDK typically runs on a separate remote server from ALI and has its own IT requirements as described in **Table 2-2222Table 2-22**.

ALI runs on a web application server. The web application server choice depends on the deployment environment: either the Java platform or Microsoft Windows .NET. The supporting services run as Windows services or as UNIX daemon processes.

The data tier components rely on the supported database servers and the local file systems of the selected operating system platform.

The IT environment of a portal technology platform includes the Web browsers that display the Web pages, the application servers and operating system platforms, and the database servers it supports. The deployment environment, either a Java application server or Windows .NET, provides run-time services such as request handling, process and thread management, memory management, and basic security.

The following table lists the elements of the IT environment for the evaluated ALI configurations.

Table 2-1111: ALI IT Requirements for Evaluated Configurations

IT Component	Supported Versions
Operating systems	Microsoft Windows Server 2003 SP1 Solaris 10 (on SPARC) Red Hat Enterprise Linux 4 Update 3 (x86)
Application servers	**Microsoft Windows:** Microsoft IIS 6.0 with .NET Framework 1.1 SP1 **Red Hat Linux:** BEA WebLogic Server 9.2 with BEA JRockit 5.0 (R26.0.0) JDK (32-bit) **Solaris:** BEA WebLogic Server 9.2 with Sun Java 2 JDK 5.0 with the Java HotSpot^{TM} Client and Server VMs (32-bit), version 1.5.0_06
Database servers	Microsoft SQL Server 2005 (with SQL Server 2000 compatibility level) Oracle 10g R2 (10.2.0.1 and above) in default or Oracle RAC configuration
Web browsers	Administrative Users: Internet Explorer 6.0, Firefox 2.0 Browsing Users: Internet Explorer 6.0; Firefox 1.5, 2.0;
External authentication sources	LDAP 2.2 MP1 Active Directory 6.3
Identity Services (optional services that used by the TOE for access to the authentication sources)	AquaLogic Interaction Identity Service – LDAP 2.2 MP1 AquaLogic Interaction Identity Service – Active Directory 6.3 Custom identity services built with the IDK.

Table 2-2222: IDK IT Requirements of Evaluated Configurations

IT Component	Description
Operating Systems	Microsoft Windows Server 2003 SP2 Solaris 10 (on SPARC) Red Hat Enterprise Linux 4 Update 3 (x86)
Application servers	**Microsoft Windows:** Microsoft IIS 6.0 **Red Hat Linux:** BEA WebLogic Server 9.2 **Solaris:** BEA WebLogic Server 9.2
Runtime Environment	Windows: IIS 6.0 with .Net 1.1 UNIX environments: Sun JDK 1.4

2.2.2 Logical Boundaries

The TOE logical boundary consists of the security functions of ALI summarized below. The full description of the security functions is in section 6.1.

2.2.2.1 Security audit

ALI provides the capability to generate audit records, determine which user actions will be audited, and display the audit records. ALI is dependent upon the IT environment to store and protect the audit records from unauthorized modifications and deletions.

2.2.2.2 User data protection

ALI enforces an access control mechanism to control users' access to ALI objects and administrative interfaces. Section 6.1.2, User Data Protection, provides a detailed description of the access control mechanism.

2.2.2.3 Identification and authentication

ALI includes the capability to use third-party authentication sources such as Microsoft Active Directory and various LDAP 2.2 products to get user information and to perform identification and authentication. ALI enforces the identification and authentication decision received from the third-party source. ALI also includes an internal identification and authentication mechanism which is used to log in administrative users and users that are defined within the ALI. ALI includes the capability to monitor the login process and lockout user accounts that exceed the configured unsuccessful login limit.

2.2.2.4 Security management

ALI includes the Administrative Portal that is used by authorized administrators to manage the access control mechanism, TOE data, definition of roles, and security-related functions.

2.2.2.5 Protection of the TSF

ALI provides an access control mechanism that ensures the security functions are not bypassed.

2.3 TOE Documentation

ALI has a number of administrative, user, and installation guides for the TOE. These documents and others are described in section 6.2.

3. Security Environment

This section describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed. The statement of the TOE security environment defines the following:

- Assumptions related to the operational environment and the method of use intended for the TOE.
- Organizational security policies with which the TOE is designed to comply.

3.1 Organizational Policies

P.ACCESS The TOE must restrict the access to the TOE protected objects.

P.ACCOUNTABILITY Users shall be held accountable for specific security relevant actions within the TOE.

P.AUTH_USERS Only those users who have been authorized to access the information within the TOE may access the TOE.

P.MANAGE The TOE must provide authorized administrators with utilities to effectively manage the security-related functions of the TOE.

3.2 Assumptions

A.INSTALL Those responsible for the TOE must ensure the TOE is delivered, installed, managed, and operated in a manner that maintains the IT security objectives.

A.NOEVIL The administrative personnel are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided in the administrative guidance.

A.PHYSICAL The TOE software critical to security policy enforcement will be protected from unauthorized physical modification.

A.OPE_ENV The operating environment must protect the TOE and its resources from unauthorized deletions and tampering and provide a reliable timestamp for the TOE’s use.

A.TRANSMIT The operating environment will protect the data transmitted from the TOE to other IT products.

A.USER The authorized users are not negligent or malicious and will follow the guidance provided.

4. Security Objectives

This section defines the security objectives of the TOE and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified assumptions and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.ACCESS The TSF shall control access to TOE objects by identified users and groups. The TSF must allow authorized users to specify which users and groups may access the TOE objects and the operations that may be performed.

O.AUDIT The TSF shall record the user's actions in the TOE and associate the action with the user who caused the event. The TSF shall provide the capability to determine what actions will be audited and to review the audit records.

O.AUTH The TSF shall ensure that all existing users are identified and authenticated (by the TOE or by a third party) before access to the TOE is permitted.

O.MANAGE The TOE shall provide the functions for authorized administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access the functions.

4.2 Security Objectives for the IT Environment

OE.3rdAUTH The environment shall perform identification and authentication for users whose accounts are not maintained by the TOE.

OE.OPE_ENV The operating environment must protect the TOE and its resources from unauthorized deletions and tampering and provide a reliable timestamp for the TOE’s use.

4.3 Security Objectives for the Environment

OE.ADMIN The TOE administrators shall be competent, trustworthy, trained in the proper operation of the TOE, and will follow the guidance provided.

OE.INSTALL Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains the IT security objectives.

OE.PHYSICAL The TOE software critical to security policy enforcement will be protected from unauthorized physical modification.

OE.TRANSMIT The operating environment shall protect the data transmitted by the TOE from disclosure.

OE.USER The authorized users will not be negligent or malicious and will follow the guidance provided.

5. IT Security Requirements

This section defines the security functional requirements satisfied by the TOE and its IT operating environment and security assurance requirements levying against the TOE in an evaluation. The security functional requirements are a combination of the requirements drawn from the CC Part 2 and explicitly stated requirements that define functionality not modeled by the CC.

5.1 TOE Security Functional Requirements

The following table describes the SFRs satisfied by TOE.

Table 5-1111 TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.2: User identity association
	FAU_GEN_EX.1: Audit data generation
	FAU_SAR.1: Audit review
	FAU_SAR.3a: Selectable audit review
	FAU_SAR.3b: Selectable audit review
	FAU_SEL.1: Selective audit
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1a: User attribute definition
	FIA_ENF_EX.1: Enforcement of identification and authentication decision
	FIA_UAU_EX.1: Timing of authentication
	FIA_UID_EX.1: Timing of identification
FMT: Security management	FMT_MSA.1a: Management of security attributes
	FMT_MSA.1b: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_MTD.1d: Management of TSF data
	FMT_MTD.1e: Management of TSF data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1a: Non-bypassability of the TSP

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN_EX.1)

FAU_GEN_EX.1.1 The TSF shall be able to generate an audit record of the following auditable events: ALI object changes, ALI object deletion, locked accounts, user logins, and object ACL changes.

FAU_GEN_EX.1.2 The TSF shall record within each audit record at least the following information: Date and time of the event, event type, item type, item name, user identity, and the outcome of the event.

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**authorized administrators**] with the capability to read [**all audit data**] from the audit records.

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

5.1.1.4 Selectable audit review (FAU_SAR.3a)

FAU_SAR.3a.1 The TSF shall provide the ability to perform [***searches***] of audit data based on [**event type, user identity, item type, item name, date/time, and outcome of event**].

5.1.1.5 Selectable audit review (FAU_SAR.3b)

FAU_SAR.3b.1 The TSF shall provide the ability to perform [***ordering***] of audit data based on [**date/time**].

5.1.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: a) [***event type***] b) [**no other attributes**].

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 [**ALI Access Control Policy**] on [
subjects: users;
objects: Administrative folders, authentication sources, communities, community templates, content crawlers, content sources, content types, experience definitions, external operations, federated searches, filters, groups, invitations, jobs, pages, page templates, portlets, portlet bundles, portlet templates, profile sources, properties, remote servers, snapshot queries, Web services;
operations: view, modify, create, copy, move, delete].

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**ALI Access Control Policy**] to objects based on the following: [**subjects: Users: user name, group name (object creation activity rights are conferred via the group membership);**
objects: ACL].

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

- **The move operation is granted if the ACL associated to the parent folder, destination folder, and the object grants the user name/group name permission;**
- **The create operation is granted if the user name via the user's group name has the appropriate object creation activity right;**
- **The copy operation is granted if user name via the user's group name has the appropriate object creation activity right and the ACL associated to the parent folder, destination folder, and the object grants the user name/group name permission;**
- **The other requested operations are granted if the ACL associated to the parent folder and the object grants the user name/group name permission**].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [

- **Read access to authentication source, content types, filters, invitations, and properties is explicitly granted to all authorized users,**
- **The authorized administrator (built-in administrator or user in the Administrator group) is granted all access to the object**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**No explicitly deny access rules**].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [**an administrator configurable number**] unsuccessful authentication attempts occur related to [**portal login attempts within the administrator-configured time period**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [**lock the user account for a period of time that is set by the administrator or until the administrator unlocks the account**].

Application Note: The administrator's guide includes guidelines for the administrator to the set the number of allowed unsuccessful logins to 5 or less.

5.1.3.2 User attribute definition (FIA_ATD.1a)

FIA_ATD.1a.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**user name, password, groups**].

5.1.3.3 Enforcement of identification and authentication decision (FIA_ENF_EX.1)

FIA_ENF_EX.1.1 The TSF shall require each environment-defined existing user to be successfully identified and authenticated using support from the IT environment before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.4 Timing of authentication (FIA_UAU_EX.1)

FIA_UAU_EX.1.1 The TSF shall allow the creation of new user accounts by new users on behalf of the user to be performed before the user is authenticated.

FIA_UAU_EX.1.2 The TSF shall require each TOE-defined user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.5 Timing of identification (FIA_UID_EX.1)

FIA_UID_EX.1.1 The TSF shall allow the creation of new user accounts by new users on behalf of the user to be performed before the user is identified.

FIA_UID_EX.1.2 The TSF shall require each TOE-defined user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.1.4 Security management (FMT)

5.1.4.1 Management of security attributes (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the [**ALI Access Control Policy**] to restrict the ability to [***modify***] the security attributes [**ACL**] to [**authorized administrator, authorized user with admin privilege on the given object**].

5.1.4.2 Management of security attributes (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the [**ALI Access Control Policy**] to restrict the ability to [**assign to groups**] the security attributes [**activity right**] to [**authorized administrator**].

5.1.4.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**ALI Access Control Policy**] to provide [**inherited**] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized administrator, authorized user with admin privilege on the parent folder**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.4 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to [***query***] the [**audit records**] to [**member of the administrators group**].

5.1.4.5 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to [***modify***] the [**set of auditable events**] to [**member of the administrators group**].

5.1.4.6 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to [***delete, modify***, **disable, enable, unlock**] the [**TOE user account attributes other than the password**] to [**member of the administrators group**].

5.1.4.7 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1d.1 The TSF shall restrict the ability to [***modify***] the [**number of failed login attempts allowed, specified lockout period, "minutes to track failed logins"**] to [**member of the administrators group**].

5.1.4.8 Management of TSF data (FMT_MTD.1e)

FMT_MTD.1e.1 The TSF shall restrict the ability to [***modify***] the [**password**] to [**member of the administrators group and authorized users associated to password**].

5.1.4.9 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [**management of object ACL as specified in FMT_MSA.1a, management of the audit function as specified in FMT_MTD.1a, and FMT_MTD.1b, management of user accounts as specified in MTD.1c, management of groups as specified in FMT_MSA.1b, management of the authentication failure mechanism as specified in FMT_MTD.1d, change of a user password, and create user accounts**].

5.1.4.10 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized administrator, authorized user**].

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

5.1.5 Protection of the TSF (FPT)

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

FPT_RVM.1a.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs satisfied by the IT environment of the TOE.

Table 5-2222 IT Environment Security Functional Components

Requirement Class	Requirement Component
FIA: Identification and authentication	FIA_ATD.1b: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
FPT: Protection of the TSF	FPT_STG._EX.1 Protected TOE data storage
	FPT_RVM.1b: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation
	FPT_STM.1: Reliable time stamps

5.2.1 Identification and authentication (FIA)

5.2.1.1 User attribute definition (FIA_ATD.1b)

FIA_ATD.1b.1 The ~~TSF~~ **IT Environment** shall maintain the following list of security attributes belonging to individual users: [**userid, password**].

5.2.1.2 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The ~~TSF~~ **IT Environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.1.3 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The ~~TSF~~ **IT Environment** shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.2.2 Protection of the TSF (FPT)

5.2.2.1 Protected TOE data storage (FPT_STG_EX.1)

FPT_STG_EX.1.1 The IT Environment shall protect the TOE data from unauthorised deletion.

FPT_STG_EX.1.2 The IT Environment shall be able to prevent unauthorised modifications to the TOE data stored in the environment.

5.2.2.2 Non-bypassability of the TSP (FPT_RVM.1b)

FPT_RVM.1b.1 The ~~TSF~~ **IT Environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.2.3 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 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.1.2 The ~~TSF~~ **IT Environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.2.4 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 **and the TOE's** use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Table 5-3333 EAL 2 augmented with ALC_FLR.2 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.2: Flaw reporting procedures
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

5.3.1 Configuration management (ACM)

5.3.1.1 Configuration items (ACM_CAP.2)

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 Delivery procedures (ADO_DEL.1)

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 Installation, generation, and start-up procedures (ADO_IGS.1)

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 Informal functional specification (ADV_FSP.1)

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 Descriptive high-level design (ADV_HLD.1)

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 Informal correspondence demonstration (ADV_RCR.1)

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 Administrator guidance (AGD_ADM.1)

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 behaviour 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 User guidance (AGD_USR.1)

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 behaviour 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 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c 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.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c 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.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.2.1e 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 Evidence of coverage (ATE_COV.1)

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 Functional testing (ATE_FUN.1)

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 Independent testing - sample (ATE_IND.2)

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 Strength of TOE security function evaluation (AVA_SOF.1)

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 Developer vulnerability analysis (AVA_VLA.1)

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

Each of the security function descriptions is organized by the security requirements corresponding to the security function.Hence, each function is described based on how it specifically satisfies each of its related requirements. This serves both to describe the security functions and to rationalize that the security functions are suitable to satisfy the necessary requirements.

6.1.1 Security audit

ALI tracks the actions of the authorized users. Audit records are generated for the following actions:

- Successful attempts to log into the portal (use of the identification and authentication mechanism)
- Account lock outs
- Modifications made to the ALI objects
- Deletion of ALI objects
- Modifications made to object security settings
- Modifications to the following global system settings:
  - *Global ACL Sync Map*
  - *Global Property Map*
  - *Global Content Type Map*
  - *Global Object Property Map*
  - *User Information Property Map*

Each audit record contains the following minimum information:

- User identity
- Date and time of the action
- message type (synonymous with event type)
- Item type
- Item name
- Outcome of the action – message that provides information about the user's actions.

The audit records are stored and protected by the IT environment. Authorized administrators who are members of the Administrator group use a TOE interface, the Audit Manager, to manage the audit function. The Audit Manager enables administrators to perform the following functions:

- Define the actions to be audited based on the message type (event type)
- Review the audit records
- Perform queries on the stored audit records based on selected search criteria. The administrator can specify queries based on the following search criteria:
  - Item type: Specifies the type of portal object to be included in the query. Values are **All** or a specific type of portal object.

    - Item Name: Name of the specific portal object.
    - User Name: Name of the user performing the action.
    - Server Name: Portal server instance to be searched.
    - Word in Message: A word to search for in the audit message.
    - Message Type: Type of user action.
    - Time Interval: Date/time range for the search.
- Sort the results of a query by date from the oldest to the most recent records and vice versa.

Auditing is disabled by default. To enable auditing, refer to the *BEA AquaLogic Interaction Administrator Guide, Version 6.1 MP1*.

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

- FAU_GEN.2: The audit records generated by ALI include the identification of the user.
- FAU_GEN_EX.1: ALI generates audit records with required information.
- FAU_SAR.1: ALI provides the authorized administrator an interface to review the audit records. The audit records are in a readable format.
- FAU_SAR.3a: ALI provides the authorized administrator the ability to query the audit logs based on applicable attributes.
- FAU_SAR.3b: ALI provides the ability to order the records from the oldest to the newest or vice versa.
- FAU_SEL.1: ALI allows the authorized administrator to determine which events will be audited.
- FMT_MTD.1a: ALI restricts the interface to query the audit records to the authorized administrator who is a member of the Administrator group.
- FMT_MTD.1b: ALI restricts the interface to determine what security events will be audited to the authorized administrator who is a member of the Administrator group.

6.1.2 User data protection

ALI implements data protection security for portal access and portal activities by managing a hierarchy of objects that determine access privileges and activity rights.

For each type of administrative portal object created and accessed within ALI, ALI enforces an access control policy based upon the creation activity rights assigned to the portal groups, the users assigned to the groups, and the access control list (ACL) associated with the object.

Activity Rights

Activity rights are portal settings that confer system-wide privileges in ALI, such as the right to create ALI objects. Activity rights are defined on a group basis. Users who do not have permission to perform a particular activity do not see the corresponding user interface elements in ALI. Activity rights are not assigned directly to individual users, rather activity rights are assigned to groups. Then users are also assigned to groups and the users acquire the associated group activity rights conferred by their specific group membership.

Access Control List

The access control list (ACL) is a list of access privileges associated with each folder or other administrative object in the portal. The ACL determines which operations a user or group can perform on an object. The access privileges controlled by the ACL are initially based on the security of the folder in which the object is created. During object creation, the object inherits the ACL of the parent folder. An option to change the object’s default ACL (inherited ACL) is presented to the user. The user creating the object is assigned Admin privileges on the object. Changes to the security of a folder may optionally be applied to all the objects within that folder for which the user has the Admin privilege.

The default portal installation creates the following group, user, and profile objects, with default access privileges and activity rights.

Table 6-1111: Default Portal Groups and Users

Installed User, Group, or Profile	Access Privileges and Activity Rights
Administrators group	Admin/All
Everyone group	Read and Edit Own Profile access
Administrator (user)	Admin/All
Default profile	None [Note: Account locked by default]
Guest profile	None
Nobody	None

The default portal installation includes the following root level folders:

- Administrative Objects Directory – This is the root folder of the Admin Objects Directory shown in the Administrative Portal
- Default Profiles – This folder contains all default profiles.
- User Profile Layout – This folder contains all user profile layout pages and sections.

Table 6-2222Table 6-22 lists the default folders created in the Administrative Objects Directory.

Table 6-2222 Default Folders in the Administrative Objects Directory

Folder	Default ACL
Administrative Resources This folder contains the following objects created at installation: users, groups, the AquaLogic Interaction Authentication Source, the WWW content source, properties, content types, and federated search objects.	Administrators Group - Admin access Everyone Group - Read access
Intrinsic Operations This folder contains external operations and intrinsic jobs, such as Search Update, Document Refresh, and Weekly Housekeeping. The folder is registered with the primary Automation Service.	Administrators Group - Admin access
Portal Resources This folder contains intrinsic portlets and Web services, as well as page, community, and portlet templates.	Administrators Group - Admin access Everyone Group - Read privilege
Default Experience Definition This folder contains the users associated with the default experience definition. Upon installation, one user is associated with the default experience definition User Profile Layout — Administrator.	Administrators Group - Admin access Everyone Group - Read access
Experience Definition Objects This folder contains the default objects related to the default Experience Definition.	Administrators Group - Admin access Everyone Group - Read access

Users in the Administrators group have full access to all portal objects.

The object ACL has four hierarchical access privileges as follows:

- **Read** allows users or groups to view an object.
- **Select** allows users or groups to add this object to other objects. For example, it allows users to add portlets to their My Pages, add users to groups, or associate remote servers with Web Services. The privilege includes the Read privilege.
- **Edit** allows users or groups to modify this object. The privilege includes the Select and Read privileges.
- **Admin** allows users or groups full administrative control of this object, including deleting the object or modifying the object's ACL. The privilege includes the Edit, Read, and Select privileges.

The Everyone group (all users) has mandatory Read access to authentication sources, content types, filters, invitations, and properties.

To copy or move an object from one folder to another, the user is required to have a minimum of Read access to the originating folder and Edit access to the destination folder. In addition, to copy an object, the user must be associated to the creation activity right for the object type. The creation activity right is required for users that want to create an object.

ALI Administrative Objects

Administrative objects are created within administrative folders. The folders can contain other folders as well as objects. All objects are child objects except for the root folder, Administrative Objects Directory (created during ALI installation). When a child object is created, the child object initially inherits the ACL of the parent folder by default. During object creation, the child object’s default ACL can be modified to be different from the parent folder ACL. After object creation, other users with Admin access to the object can also modify the ACL if necessary. The user who created the object is initially assigned Admin access to the new object.

The creation activity rights and the ACLs work hand-in-hand to allow creation of new objects. For example, to create a new child object, a user must have the appropriate create activity right and Edit rights on the parent folder. After creation, the ability to modify the object is governed solely by the ACL of the object and its parent folder.

The following table describes the ALI objects (also called administrative objects).

Table 6-3333 ALI Objects

Object	Description	Types or Values
Administrative folder	A folder in the Admin Objects Directory.	n/a
Authentication source	This object enables configuration of a portal authentication source residing on a remote server.	An authentication source exists for the users in the portal database and for SSO (if used), but users can only create new authentication sources of the *remote* type.
Community	Defines a group of portal users with common objectives or interests. Communities provide content and services to a *group* of users rather than just to an individual user.	Varies depending on the specific business needs. For example, communities might be based on departments in a company.

Object	**Description**	**Types or Values**
Community template	A set of one or more *page templates*, which can include portlets, a header or footer, a branding portlet and other content that defines the minimum requirements for the community.	Varies depending on the specific business needs.
Content crawler	An extensible component (the ALI implementation of a content crawler is a Web service) used to access content from an external source and index it in the portal.	Remote – targets content on remote servers WWW – targets content from WWW locations
Content source	An object that provides access to external content repositories, allowing users and content crawlers to add document records and links in the Knowledge Directory. Content sources, like everything in the portal, have security settings that enable you to specify exactly which portal users and groups can see the content source. Users that do not have Read access to a content source cannot select it or even see it when submitting content or building a content crawler.	Remote – targets content on remote servers WWW – targets content from WWW locations
Content type	An identifier associated with a source document that determines how metadata in the source document is mapped to portal properties. Specifies options such as: source content format, whether the text of the content should be indexed for searching and how to populate values for document properties.	Varies
Experience definition	An object that specifies settings and properties to control aspects of the overall look, feel, and access features of the portal pages and resources for a group of users. Controls the branding, the header and footer, the navigation scheme, the default login page, communities, the Knowledge Directory, access to My Pages and any mandatory links.	Varies by company.
External Operation	An object that enables you to run command-line actions through the portal and schedule these actions through portal jobs.	n/a
Federated search	An object that enables portal users to search external repositories for content or enables users of other portals to search your portal for content. Federated searches provide end-users a single interface and unified result set for searches over multiple ALI portals, as well as parallel querying of external Internet and intranet-based search engines.	Incoming Outgoing

Object	Description	Types or Values
Filter	A combination of a basic fields search and statements to control what content goes into which folder when *crawling in* documents or using Smart Sort (an ALI utility program). A filter sets conditions to sort documents into associated folders in the Knowledge Directory.	n/a
Group	An object that contains other groups and users, as well as any activity rights assigned to group members. A group can have static membership and membership that changes dynamically based on the properties of users' profiles or their memberships in other groups.	Logically divided into two types: People groups – containing only users Role groups – containing only people groups and specific activity rights that apply to those groups.
Invitation	An object that enables the configuration of common text and URLs to use in sending emails to potential users of the portal, making it easy for new users to create their own user accounts.	n/a
Job	An object that enables you to schedule portal management operations. A job is a collection of related operations. Each operation is one task, such as a crawl for documents, an import of users, or one of the system maintenance tasks.	n/a
Page	A collection of content, links, and other information arranged in a Web browser window.	My Pages – personalized Web pages created by end users. Community Pages – Web pages displaying information for specific groups of users known as communities.
Page Template	Defines a particular page layout to be used in creating Web pages. The page layout determines where particular types of portlets can be displayed on the page. The page template includes portlets and layout settings.	n/a
Portlet	A portlet is any Web application that returns HTML or XML using HTTP. In the ALI Web Services architecture, most portlets are hosted remotely. Each portlet is self-contained and executes its functionality in a separate process. Most portlets connect to a back-end application for data or functionality that provides customized tools and services, as well as information.	n/a
Portlet Bundle	A group of related portlets, packaged together for easy inclusion on ALI My Pages or community pages.	n/a

Object	Description	Types or Values
Portlet Template	An object defining the basic configuration for a portlet.	n/a
Profile source	An object that defines the basic settings for a Profile Web service. The remote profile source is an external source for user properties that can be searched by portal users, forwarded to portlets to authenticate portlet access, or for other purposes.	Remote
Property	Provides information about, as well as a way to search for, documents and objects in your portal.	n/a
Remote server	Many of the objects in the portal utilize *Web services*, which are components that run on a logically separate computer from the one that runs the portal and communicate with the portal via HTTP. This separate computer is referred to as a remote server.	n/a
Snapshot query	A search query that enables you to specify conditions for searching portal objects and, optionally, display the results in a Content Snapshot Portlet and/or e-mail the results to users. You can limit the search by language, object type, folder, property, and text condition.	n/a
Web Service	Component that runs on a logically separate computer from the one that runs the portal and communicates with the portal via HTTP.	Authentication Content Intrinsic portlet Profile Remote portlet Search

The user data protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1: The TOE enforces an access control policy for access to the TOE objects. The access control policy permits operations on a TOE object based on the users and/or user's association to a group and the object ACL which defines the privileges allowed the user and/or the user's group.
- FDP_ACF.1: The TOE enforces an access control policy based on the ACL associated to an object upon creation. The access control policy permits operations on the object depending on the ACL that defines the privileges allowed the user and/or the user's group. The user will not be able to create an object if the user does not have the appropriate creation activity right.
- FMT_MSA.3: The TOE implements an inheritance policy where child objects inherit the ACL associated to the parent folder upon creation. During creation, the inherited ACL can be modified by the authorized administrator or the authorized user with admin privilege to the parent folder. Access to the new objects is restricted to the access granted by the ACL.

6.1.3 Identification and authentication

The TOE ensures that all users defined in the TOE are identified and authenticated before access to the TOE is permitted. The TOE implements an internal identification and authentication mechanism for all user accounts created within the TOE. The TOE also provides the option to utilize third-party authentication sources to maintain user accounts and to perform identification and authentication for those users. The TOE enforces the identification and authentication decisions received from the third-party sources.

Identification is required both for GUI (e.g., administrative) interfaces and also the interfaces designed to support the Interaction De3velopment Kit (IDK). In the case of the GUI, a dialog will be presented where by users can enter their login credentials which must be verified before their authenticated session is established. In the case of the IDK, a SOAP connection object is used to set up the user session and using available methods either the user login credentials can be provided directly by the IDK application or alternately a user login token can be supplied (see the Portlets discussion below). In every case, the user is identified and that identity must be authenticated (by a configured third-party method, by the TOE by requiring the correct password, or by the TOE by verifying the cryptographic hash on a login token).

The TOE provides an optional feature called self-registration that allows new users to create their own accounts by clicking “Create an account” on the login page. After creating an account, self-registered users are automatically given security privileges based on the administrator-defined default profile for new users named Default Profile. Based on this security, users can personalize their views of the portal with My Pages, portlets, and community memberships, and can view portal content. Every portal user account is based on a default profile that can include initial My Account settings, the name and number of My Pages, and the layout of the portlets on those My Pages. Portlet preferences, group memberships, and community memberships are not inherited by users created from default profiles.

The authorized administrator uses the Default Profiles utility interface to define the default profiles including those for Guest users and new self-registered users. The default profiles are used to assign settings to new users created through invitations, authentication sources, and self-registration. Guest is the identity users have before they log in or before they self register.

The TOE also monitors the unsuccessful attempts to log in by the users. When the administrator configurable unsuccessful login limit is exceeded during an administrator-specified time period set to track the failed logins, the user account is locked for the administrator-specified period of time or until the authorized administrator releases the account. The TOE maintains accounts for administrators and non-administrator users that, at a minimum, contain the username and password.

Portlets

The TOE is designed to facilitate user access to remote portlets outside the direct control of the TOE. However, a given portlet may require access to portal functions on behalf of a given TOE user in order to complete its function. As such, the TOE object that represents the portlet must be configured to allow the TOE to send the user’s login token to that portlet. Subsequently, when the TOE sends a request to the portlet on behalf of a TOE user, it will create and provide a login token to the portlet. This token is comprised of the user identifier, the current time, a lifetime value for the token, and a keyed-HMAC hash[1]. Once the portlet (e.g., a remote server) gets the login token, it can use it, during its lifetime, to perform TOE operations on behalf of the user originally requesting the portlet service.

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

- FIA_AFL.1: The TOE locks out user accounts when the administrator-configured unsuccessful login attempts threshold is exceeded during an administrator-specified time period. The administrator is able to change the threshold - a limit of 5 or less is recommended. The users will be locked out for a specified period of time or until the authorized administrator unlocks the account.
- FIA_ATD.1a: The TOE maintains the username and password for all TOE defined accounts.

[1] The hashing function – Keyed-HMAC using SHA-1 – is delegated to the underlying Java/.NET platform.

- FIA_ENF_EX.1: The TOE interfaces with third-party authentication sources that maintain user accounts and enforces the identification and authentication decisions received from the third-party authentication sources before allowing the non-administrator user access to the TOE.
- FIA_UAU_EX.1: The TOE enforces authentication of all TOE-defined user accounts.
- FIA_UID_EX.1: The TOE enforces identification of all TOE-defined user accounts.

6.1.4 Security management

The TOE provides web-based administrative interfaces to manage the security functions of the TOE. The administrative interfaces are collectively called the Administrative Portal. Access to the Administrative Portal is restricted to groups that have been assigned activity rights for administrative functions. The activity rights are listed and described in the Administrator guide under Delegating Activity Rights section.
ALI has two types of authorized administrators:

- Users in the Administrators group. This group is created during installation and initially contains just the default administrative user, Administrator. The user account for the user Administrator is established during installation of ALI. Other user accounts can be created and added to the group by this Administrator.
- Users in groups which have been assigned a specific subset of portal administrative activity rights and corresponding admin privilege on the object.

All users in the Administrators group have access to all areas and features of the Administrative Portal. For all other users, the ability to perform administrative tasks is controlled by granting specific activity rights and corresponding access privileges. Users who do not have rights to perform a particular activity do not see the corresponding user interface elements in the portal. Users who have activity rights for specific functions also need the corresponding access privileges to portal folders where those functions can be performed.

The Administrative Portal provides the authorized administrators the portal utilities used to manage the security functions. The portal utilities are accessed by a drop-down menu. Using the sub-interfaces the authorized administrators are able to manage the following functions:

- The users of the Administrators group are able to manage the TOE user accounts using the Administrative Objects Directory, the Create User, and the Release Disable Logins utilities. The authorized administrator is able to delete a user account by selecting the user object. Using the Create User utility, the authorized administrator is able to create new user accounts, modify the user information such as the password and the user's group membership and disable/enable the account. In instances where the user accounts are disabled because of excessive failed login attempts, the authorized administrator is able to delete or enable the accounts using the Release Disable Logins utility.
- Using the Portal Settings: User Settings Management interface to manage the identification and authentication mechanism, the users of the Administrators group are able to set the parameters that control the number of unsuccessful logins allowed by the TOE established users before the user's account locks, how long the TOE account will stay locked before automatically unlocking and allowing the user to attempt to login again, and how long the failed logins are tracked.
- Using the Create objects utilities, the authorized administrators are able to modify the object ACL. The object editors allow the authorized administrators to add and delete users and groups and change the access privilege of the users and groups associated with the ACL with the exception of the Administrators group. The authorized administrator uses the Portal Group utility to assign activity rights to the group object.

The authorized users with TOE user accounts are able to change their own passwords using the My Account: Change Password interface.

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

- FMT_MSA.1a: Using the Create objects utilities, the members of the administrator groups and the authorized user with the Admin privilege is able to modify the object ACL.
- FMT_MSA.1b: Using the Activity Manager and Create Group utility, the authorized administrator can assign activity rights to a group object.
- FMT_MTD.1c: Using the Administrative Objects Directory interface, the authorized administrator is able to delete user accounts. Using the Create User utility, the authorized administrator is able to create, modify, enable, and disable user accounts, and the Release Disabled Logins interface to unlock existing user accounts and to delete locked accounts.
- FMT_MTD.1d: Using the Portal Settings: User Settings Manager interface, the authorized administrator is able to modify the parameters of the identification and authentication mechanism. The parameters include the numbers of allowed failed login attempts, the specified lockout period, and the minutes to track failed logins.
- FMT_MTD.1e: Authorized users of TOE established accounts can modify their own password using the My Account: Change Password interface. The My Account is not part of the Administrative Portal, but part of the user's main portal page. The authorized administrator, using the Create User Utility is able to change another user's password.
- FMT_SMF.1: The TOE provides a web-based interface that allows the authorized administrator to manage the security functions and configure the TOE, the authorized user to change their password, and new users to establish user accounts within the TOE. Once established, the TOE accounts are managed by the authorized administrator.
- FMT_SMR.1: The TOE defines an authorized administrator and the authorized user of the TOE.

6.1.5 Protection of the TSF

The interfaces offered by ALI, including GUI and IDK interfaces, have all been carefully designed, implemented, and tested to ensure that they do not offer opportunities to tamper with or interfere with the operation of the security functions and also to ensure that they do not offer any access to protected resources that is not subject to user authentication requirements and access control policies and activity rights.

ALI is designed to operate in security domains provided by the underlying Java Platform and .NET run-time environments. It relies on the environment to protect components from interference and tampering by untrusted subjects. It also relies on application server sessions to store data associated with each active user. ALI enforces separation between the objects associated with each user's session and stores them on separate application server sessions.

Only authorized administrators who are users with the Access Administration activity right or are members of the Administrators group are allowed access to the Administrative Portal.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_RVM.1a: The TOE enforces access control mechanisms to ensure that the TOE security functions are not bypassed.

6.2 TOE Security Assurance Measures

The following assurance measures are applied to satisfy the Common Criteria EAL 2 augmented with ALC_FLR.2 assurance requirements:

- Configuration Management;

- Delivery and operation;
- Guidance; documents;
- Design Documentation;
- Lifecycle Support;
- Tests; and
- Vulnerability Assessment.

6.2.1 Configuration management

The configuration management measures applied by BEA 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 performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, delivery and operations, life cycle support, vulnerability assessment, and the CM documentation.

These activities are documented in:

- AquaLogic User Interaction Configuration Management

The Configuration management assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

- ACM_CAP.2

6.2.2 Delivery and operation

BEA provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. The BEA delivery procedures describe all applicable procedures to be used to prevent inappropriate access to the TOE. BEA also provides documentation that describes the steps necessary to install ALI in accordance with the evaluated configuration.

These activities are documented in:

- BEA AquaLogic User Interaction Delivery and Operation
- BEA Common Criteria Configuration Notice for AquaLogic Interaction 6.1 MP1 with AquaLogic Interaction Development Kit 6.0

ALI 6.1 MP1

- Release Notes for AquaLogic® Interaction 6.1
- BEA AquaLogic® Interaction Installation and Upgrade Guide, Version 6.1 MP1, September 17, 2007
- Installation Worksheet for AquaLogic® Interaction 6.1 MP1 – Windows
- Installation Worksheet for AquaLogic® Interaction 6.1 MP1 - UNIX
- Deployment Guide for BEA AquaLogic®™ User Interaction – this information is divided into 5 separate guides as follows:
  - BEA AquaLogic® User Interaction Deployment Overview, September 19, 2007
  - BEA AquaLogic® User Interaction Deployment Planning, September 19, 2007
  - BEA AquaLogic® User Interaction Customization Overview, September 19, 2007
  - BEA AquaLogic® User Interaction Deployment Maintenance Guide, September 19, 2007
  - BEA AquaLogic® User Interaction Networking and Authentication Guide, September 19, 2007

IDK 6.0

- Release Notes for AquaLogic® Interaction Development Kit 6.0
- BEA AquaLogic® Interaction Development Kit Installation Guide, version 6.0, July 2007

The Delivery and operation assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

6.2.3 Design Documentation

BEA has documents describing all facets of the design of the TOE. These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

These activities are documented in:

- BEA AquaLogic Interaction 6.1 MP1 with AquaLogic Interaction Development Kit 6.0 Functional Specification (ADV_FSP)
- FSP Addendum
- AquaLogic Interaction 6.1 with AquaLogic Interaction Development Kit High Level Design (ADV_HLD)
- AquaLogic Interaction 6.1 with AquaLogic Interaction Development Kit 6.0 Representation Correspondence (ADV_RCR)
- AquaLogic User Interaction (ALUI) Development Documentation Online

The Development assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

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

6.2.4 Guidance documents

BEA provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

ALI 6.1 MP1

- BEA AquaLogic® Interaction Administrator Guide, version 6.1 MP1, September, 17 2007
- BEA AquaLogic® Interaction Installation and Upgrade Guide, version 6.1 MP1, September, 17 2007
- BEA AquaLogic® User Interaction Deployment Maintenance Guide, September, 19 2007
- BEA AquaLogic® User Interaction Networking and Authentication Guide, September, 19 2007
- BEA AquaLogic® User Interaction Deployment Overview, September, 19 2007
- BEA AquaLogic® User Interaction Deployment Planning, September, 19 2007

- Release Notes for AquaLogic® Interaction 6.1, September, 17 2007
- AquaLogic® Interaction Online Help

IDK 6.0:

- Release Notes for AquaLogic® Interaction Development Kit 6.0
- BEA AquaLogic® Interaction Development Kit Installation Guide, version 6.0, July 2007
- Development Tutorials:
    - Introduction to Pagelet Development
    - IDK QuickStart: Hello World Pagelet (Java | .NET)
    - IDK QuickStart: Hello World Portlet (Java | .NET)
    - Setting Up a Custom IDK Project (Java | .NET)
- API Documentation:
    - Javadoc
    - NDoc

The Guidance documents assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

BEA has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws, how all security flaws and the status of fixes for each security flaw is tracked, and how corrective measures are made available.

These activities are documented in:

- BEA AquaLogic User Interaction Flaw Remediation

The Life cycle support assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

- ALC_FLR.2

6.2.6 Tests

The test documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

These activities are documented in:

- BEA AquaLogic Interaction 6.1 MP1 with AquaLogic Interaction Development Kit 6.0 Testing Documentation (ATE), Submission to Common Criteria Process
- Installation Worksheet for AquaLogic® Interaction 6.1 – Windows
- Test Audit Logs
- BEA Systems Object Locks Test Plan

- BEA Portal Smoke Test Plan
- BEA Systems Audit Manager Functional Test Plan
- BEA Systems Object Security Test Plan

The Tests assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

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

6.2.7 Vulnerability assessment

BEA has conducted a strength-of-function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength-of-function claim, SOF-basic.

BEA performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE.

These activities are documented in:

- AquaLogic Interaction 6.1 MP1 with AquaLogic Interaction Development Kit 6.0 Vulnerability Assessment

The Vulnerability assessment assurance measure satisfies the following EAL 2 augmented with ALC_FLR.2 assurance requirements:

- AVA_SOF.1
- AVA_VLA.1

7. Protection Profile Claims

The ST does not claim compliance to a Protection Profile.

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 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

Table 8-1111 Environment to Objective Correspondence

	P.ACCESS	**P.ACCOUNTABILITY**	**P.AUTH_USERS**	**P.MANAGE**	**A.INSTALL**	**A.NOEVIL**	**A.PHYSICAL**	**A.OPE_ENV**	**A.TRANSMIT**	**A.USER**
O.ACCESS	X			X						
O.AUDIT		X								
O.AUTH			X							
O.MANAGE				X						
OE.3rdAUTH			X							
OE.OPE_ENV								X		
OE.ADMIN						X				
OE.INSTALL					X					
OE.PHYSICAL							X			
OE.TRANSMIT									X	
OE.USER										X

8.1.1.1 P.ACCESS

The TOE must restrict the access to the TOE protected objects.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The objective ensures that the TOE restricts access to the TOE objects to the authorized users.

8.1.1.2 P.ACCOUNTABILITY

Users shall be held accountable for specific security relevant actions within the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.AUDIT: The objective ensures that the user's actions on the TOE security-relevant actions on the TOE are captured.

8.1.1.3 P.AUTH_USERS

Only those users who have been authorized to access the information within the TOE may access the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.AUTH: The objective ensures that users defined in the TOE are identified and authenticated before access to the TOE and its functions is allowed.
- OE.3rdAuth: The objective ensures that users defined in the IT environment are identified and authenticated before allowed access to the TOE and its functions.

8.1.1.4 P.MANAGE

The TOE must provide authorized administrators with utilities to effectively manage the security-related functions of the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The objective ensures that the TOE provides the tools to allow an authorized user to determine who may have access to the objects.
- O.MANAGE: This objective ensures that the TOE provides the tools necessary for the authorized administrator to manage the security-related functions; audit function, access control function, and other administrative functions.

8.1.1.5 A.INSTALL

Those responsible for the TOE must ensure the TOE is delivered, installed, managed, and operated in a manner that maintains the IT security objectives.

This Assumption is satisfied by ensuring that:

- OE.INSTALL: The objective ensures that the TOE will be delivered, installed, managed and operated in a manner that maintains the security objectives.

8.1.1.6 A.NOEVIL

The administrative personnel are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided in the administrative guidance.

This Assumption is satisfied by ensuring that:

- OE.ADMIN: This objective ensures that the TOE is managed and administered in a secure manner by competent and security aware personnel in accordance with the administrator documentation.

8.1.1.7 A.PHYSICAL

The TOE software critical to security policy enforcement will be protected from unauthorized physical modification.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: The objective ensures that the TOE is protected from physical attacks.

8.1.1.8 A.OPE_ENV

The TOE operating environment shall provide the mechanism to isolate the TOE components and resources, ensure the TOE cannot be tampered with or bypassed, the TOE data is protected from unauthorized deletions, and generates a reliable timestamp of the TOE's use.

This Assumption is satisfied by ensuring that:

- OE.OPE_ENV: The objective ensures that the TOE's operating environment protects the TOE and it associated data and provides a reliable timestamp.

8.1.1.9 A.TRANSMIT

The operating environment will protect the data transmitted from the TOE to other IT products.

This Assumption is satisfied by ensuring that:

- OE.TRANSMIT: The objective ensures that the TSF data exported from the TOE is protected from disclosure.

8.1.1.10 A.USER

The authorized users are not negligent or malicious and will follow the guidance provided.

This Assumption is satisfied by ensuring that:

- OE.USER: The objective ensures that the user of the TOE will work co-operatively and will follow the provided guidance.

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 5** indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

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

Table 8-2222 Objective to Requirement Correspondence

	O.ACCESS	**O.AUDIT**	**O.AUTH**	**O.MANAGE**	**OE.3rdAUTH**	**OE.OPE_ENV**
FAU_GEN.2		X				
FAU_GEN_EX.1		X				
FAU_SAR.1		X				
FAU_SAR.3a		X				
FAU_SAR.3b		X				
FAU_SEL.1		X				
FDP_ACC.1	X					

	O.ACCESS	**O.AUDIT**	**O.AUTH**	**O.MANAGE**	**OE.3rdAUTH**	**OE.OPE_ENV**
FDP_ACF.1	X					
FIA_AFL.1			X			
FIA_ATD.1a			X			
FIA_ENF_EX.1			X			
FIA_UAU_EX.1			X			
FIA_UID_EX.1			X			
FMT_MSA.1a	X			X		
FMT_MSA.1b				X		
FMT_MSA.3	X					
FMT_MTD.1a				X		
FMT_MTD.1b				X		
FMT_MTD.1c				X		
FMT_MTD.1d				X		
FMT_MTD.1e				X		
FMT_SMF.1				X		
FMT_SMR.1				X		
FPT_RVM.1a	X		X	X		
FPT_STG_EX.1						X
FIA_ATD.1b					X	
FIA_UAU.2					X	
FIA_UID.2					X	
FPT_RVM.1b						X
FPT_SEP.1						X
FPT_STM.1						X

8.2.1.1 O.ACCESS

> *The TSF shall control access to TOE objects by identified users and groups. The TSF must allow authorized users to specify which users and groups may access the TOE objects and the operations that may be performed.*

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: The requirement helps meets the objective by identifying the user objects subject to the access control policy.
- FDP_ACF.1: The requirement meets this objective by ensuring the TOE only allows access to user objects based on the defined access control policy.
- FMT_MSA.1a: The TOE allows the authorized administrator and authorized users with admin privilege to determine who will have access to the objects and the actions the users will be to perform.
- FMT_MSA.3: The TOE restrict enforces a restrictive access when a new object is created. Newly-created objects inherit the ACL from the parent folder.
- FPT_RVM.1a: The TOE always enforces the access control policy to ensure objects are accessed by authorized users and only permitted actions are allowed.

8.2.1.2 O.AUDIT

The TSF shall record the user's actions in the TOE and associate the action with the user who caused the event. The TSF shall provide the capability to determine what actions will be audited and to review the audit records.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.2: Each audit records will contain the identity of the user that caused the action.
- FAU_GEN_EX.1: The TOE will generate audit records based on the user's actions.
- FAU_SAR.1: The TOE will provide the authorized administrator the ability review the audit records.
- FAU_SAR.3a: The authorized administrator is able to query the audit logs based on contents of the audit records.
- FAU_SAR.3b: The results of the audit query can be order from the oldest to the newest or vice versa.
- FAU_SEL.1: The TOE provides the authorized administrator the capability to determine what actions will be audited.

8.2.1.3 O.AUTH

The TSF shall ensure that all existing users are identified and authenticated (by the TOE or by a third party) before access to the TOE is permitted.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: The TOE will lock user's account when the administrator configured unsuccessful login attempts threshold is reached.
- FIA_ATD.1a: The TOE will maintain, at the minimum, the username and the password of TOE defined user accounts.
- FIA_ENF_EX.1: The TOE can enforce the identification and authentication decision received from the environment for users who are not defined in the TOE.
- FIA_UAU_EX.1: The TOE ensures that TOE-defined users are authenticated before access to the TOE is permitted. New users are able to create a new account before access is granted.
- FIA_UID_EX.1: The TOE ensures that TOE-defined users are identified before access to the TOE is permitted. New users are able to create a new account before access is granted.
- FPT_RVM.1a: The TOE allows enforces identification and authentication before access to the TOE is granted.

8.2.1.4 O.MANAGE

The TOE shall provide the functions for authorized administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access the functions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MSA.1a: The TOE will restrict the ability to modify the object ACL to authorized administrators and authorized users with Admin privilege.
- FMT_MSA.1b: The TOE will restrict the assignment of the activity right to the authorized administrator.
- FMT_MTD.1a: The TOE restricts the capability to query the audit records to the authorized administrator.
- FMT_MTD.1b: The TOE restricts the capability to determine what actions will be audited to the authorized administrator.
- FMT_MTD.1c, FMT_MTD.1e: The TOE restricts the ability manage existing user accounts to the authorized administrator.
- FMT_MTD.1d: The TOE restricts ability to configure the authentication failure mechanism to the administrator.
- FMT_MTD.1e: The TOE allows users to change their own password.
- FMT_SMF.1: The TOE will provide the interfaces to administrator the TOE.
- FMT_SMR.1: The TOE defines the authorized user and authorized administrator.
- FPT_RVM.1a: The TOE enforces the restrictions on the security functions to users with appropriate administrative activity rights.

8.2.1.5 OE.3rdAUTH

The environment shall perform identification and authentication for users whose accounts are not maintained by the TOE.

This IT Environment Security Objective is satisfied by ensuring that:

- FIA_ATD.1b: The IT environment will maintain non-administrator user accounts not defined within the TOE. The accounts will contain the username and password.
- FIA_UAU.2: The IT environment will perform authentication for non-administrative users whose accounts are maintained in a third-party authentication source.
- FIA_UID.2: The IT environment will perform identification for non-administrative users whose accounts are maintained in a third-party authentication source.

8.2.1.6 OE.OPE_ENV

The operating environment must protect the TOE and its resources from unauthorized deletions and tampering and provide a reliable timestamp for the TOE's use.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STG_EX.1: The IT environment will enforce mechanisms to store and protect the TOE data from unauthorized deletions.
- FPT_RVM.1b: The IT environment will enforce mechanisms to ensure that the TOE security functions can not be bypassed.
- FPT_SEP.1: The IT environment will protect the TOE and its data to ensure that the TOE and its data are not deleted, or modified by untrusted subject.
- FPT_STM.1: The IT environment will provide the timing mechanism the TOE will utilize to supply the timestamp for the audit records and for the authentication failure mechanism.

8.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL 2 augmented with ALC_FLR.2 assurance package. The EAL chosen is based on the statement of the security environment and the security objectives defined in this ST. The sufficiency of the EAL chosen is justified based on those aspects of the environment that have impact upon the assurance needed in the TOE. The users are conscientious, non-hostile, trusted, and will follow the guidance (A.NOEVIL, A.USER, OE.ADMIN, and OE.USER). The TOE is physically protected and properly and securely configured (A.INSTALL, OE.INSTALL). Given these aspects, a TOE based on good commercial development practices is sufficient. EAL 2 augmented with ALC_FLR.2 is an appropriate level of assurance for the TOE described in this ST.

8.4 Strength of Functions Rationale

The TOE claims a SOF of SOF-Basic because the claim is sufficient for the defined environment and to meet the objectives associated with access to the TOE. The SOF-Basic claim only applies to authentication method (passwords) described in the Identification and Authentication function, which supports FIA_UAU_EX.1.

8.5 Requirement Dependency Rationale

The ST satisfies all the requirement dependencies of the Common Criteria, except as noted below. Table 5 Requirement Dependency Rationale lists the requirement from Sections 5.1 and 5.2 with a dependency and indicates which requirement was included to satisfy the dependency, if any. Note, however, that FAU_GEN.1 has been effectively replaced with an explicit requirement, FAU_GEN_EX.1 that fulfills the need to generate audit records to be protected and reviewed.

Note: The table assumes that requirement iteration have the same dependences, thus the iterations are not individually identified in the table (e.g. FMT_MTD.1a).

Table 8-3333 Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN_EX.1, FIA_UID_EX.1
FAU_GEN_EX.1	FPT_STM.1	FPT_STM.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN_EX.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN_EX.1, FMT_MTD.1b
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1 and FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	FIA_UAU_EX.1
FIA_ENF_EX.1	FIA_UID.1b and FIA_UAU.1b	FIA_UID.1b and FIA_UAU.1b
FIA_UAU_EX.1	FIA_UID_EX.1	FIA_UID_EX.1
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.1
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMR.1	FIA_UID.1	FIA_UID_EX.1
FPT_STG_EX.1	FAU_GEN_EX.1	FAU_GEN_EX.1
FIA_UAU.2	FIA_UID.1	FIA_UID.2

8.6 Explicitly Stated Requirements Rationale

This Security Target defines several explicit requirements, FAU_GEN_EX.1, FIA_ENF_EX.1, FIA_UID_EX.1, FIA_UAU_EX.1, and FPT_STG_EX.1. FAU_GEN_EX.1, FIA_UID_EX.1, FIA_UAU_EX.1, FPT_STG_EX.1 are based on the CC version of FAU_GEN.1, FIA_UAU.1, FIA_UID.1, FAU_STG.1 except the explicitly stated requirements exhibit the correct behavior of the TOE. In the case of FAU_GEN_EX.1, the TOE only audits administrator-specified events related to user's actions on the TOE. The TOE does not audit the start up or shut down of the audit function. FIA_UID_EX.1 and FIA_UAU_EX.1clarifies the users identified and authenticated by the TOE.

In the case of FIA_ENF_EX.1, the requirement is necessary to model the TOE behavior of enforcing the identification and authentication results received from a third party authentication source for the non-administrative users defined in the third party authentication source.

The requirements specify straight-forward functions as they exist in the TOE and are subject to evaluation using the entire set of security assurance requirements.

The FPT_STG_EX.1 is necessary to model the requirement that the IT environment provides storage and protects the TOE data which is not limited to the audit record, user account, and TOE configuration data. The CC only defines the requirements to handle the storage and protection of audit records.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary SpecificationTOE 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 8-4 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

Table 8-4444 Security Functions vs. Requirements Mapping

	Security audit	User data protection	Identification and authentication	Security management	Protection of the TSF
FAU_GEN.2	X				
FAU_GEN_EX.1	X				
FAU_SAR.1	X				
FAU_SAR.3a	X				
FAU_SAR.3b	X				
FAU_SEL.1	X				
FDP_ACC.1		X			
FDP_ACF.1		X			
FIA_AFL.1			X		
FIA_ATD.1a			X		
FIA_ENF_EX.1			X		
FIA_UAU_EX.1			X		
FIA_UID_EX.1			X		
FMT_MSA.1a				X	
FMT_MSA.1b				X	
FMT_MSA.3		X			
FMT_MTD.1a	X				
FMT_MTD.1b	X				
FMT_MTD.1c				X	
FMT_MTD.1d				X	
FMT_MTD.1e				X	
FMT_SMF.1				X	
FMT_SMR.1				X	
FPT_RVM.1a					X

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.

Appendix A: Acronyms

ACL	Access Control List
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
EAL	Evaluation Assurance Level
FDP	User Data Protection CC Class
FIA	Identification and Authentication CC Class
FMT	Security Management CC Class
FSP	Functional Specification
HLD	High Level Design
ISO 15408	Common Criteria 2.1 ISO Standard
IT	Information Technology
J2EE	Java2 Platform, Enterprise Edition
JDK	Java Development Kit
LDAP	Lightweight Directory Access Protcol
MOF	Management of Functions
MTD	Management of TSF Data
OSP	Organization Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SM	Security Management
SMR	Security Management Roles
SOAP	Simple Object Access Protocol
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UAU	User Authentication
UDP	User Data Protection
UI	User Interface