Software AG webMethods Business Process Management Suite 8.2 SP2 Security Target

Version 0.6
11/08/13

Prepared for:

Software AG USA, Inc.

11700 Plaza America Drive, Suite 700
Reston, VA 20190

Prepared By:

Leidos, Inc.

Common Criteria Testing Laboratory

6841 Benjamin Franklin Drive
Columbia, MD 21046

LIST OF TABLES

DOCUMENT HISTORY

Version	Date	Contributors	Change
0.1	23 August 2011	Tim Bond Nelson de la Cruz Gary Grainger	Version submitted for CC evaluation
0.2	8 September 2011	Tim Bond Gary Grainger	Updates in response to evaluation team findings
0.3	9 September 2011	Tim Bond Gary Grainger	Updated to address residual evaluation team findings and comments
0.4	21 June 2012	Tim Bond Gary Grainger	Updated to address issues raised at Initial Validation Oversight Review
0.5	21 August 2013	Tim Bond	Updates for tVOR
0.6	8 November 2013	Tim Bond	Final updates based on tVOR

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE), and presents ST conventions, ST conformance claims, and the ST organization. The TOE type is a business process management platform.

The TOE is webMethods Business Process Management Suite 8.2 SP2 (BPMS) produced by Software AG USA, Inc. webMethods BPMS unites business processing management and service-oriented architecture capabilities to provide a comprehensive set of fully integrated tools for automating and managing processes. The webMethods BPMS TOE security features include assuring identification of users, controlling access to services, and auditing of user activity.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Software AG webMethods Business Process Management Suite 8.2 Security Target

ST Version – Version 0.6

ST Date – 11/08/2013

TOE Identification – Software AG webMethods Business Process Management Suite 8.2 SP2

TOE Developer – Software AG

Evaluation Sponsor – Software AG

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009

PP Conformance Claims – None

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009.
  - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 3, July 2009.
  - Part 3 Conformant
  - Assurance Level: EAL 2 augmented with ALC_FLR.1

1.3 Conventions and Terminology

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**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-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.

1.3.2 Acronyms

API	Application programming interface
BPEL	Business Process Execution Language
BPM	Business process management
BPMS	Business process management suite
DSP	Dynamic server page
ESB	Enterprise service bus
FTP	File Transfer Protocol
FTPS	File Transfer Protocol over TLS
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IMAP	Internet Message Access Protocol
JDK	Java Development Kit
JKS	Java Key Store
JMS	Java messaging service
JRE	Java Runtime Environment
PKCS	Public-Key Cryptography Standard

POP	Post Office Protocol
RDBMS	Relational Database Management System
SAML	Security Assertion Markup Language
SMTP	Simple Mail Transfer Protocol
SOA	Service-oriented architecture
SSL	Secure Sockets Layer
TLS	Transport Layer Security
WS	Web Services
WSDL	Web Services Description Language
W3C	World Wide Web Consortium
XSD	XML Schema Document
XSLT	Extensible Stylesheet Language Transformations

1.3.3 Terminology

Adapter	Connects an enterprise resource to the webMethods Integration Server. An adapter transforms data from resource-specific format into the format used within the webMethods product suite, and vice versa.
Business Process	A series of business activities that are performed in a specific order, by a variety of applications, systems, end users, and external businesses, according to defined business rules.
Broker	High-speed message router that provides publish-subscribe and point-to-point messaging.
Broker/local trigger	A trigger that subscribes to and processes documents published/delivered locally or to webMethods Broker.
Client Group	Represents a particular group or category of webMethods Broker client (for example, administrators, customers, or Integration Server processes). Client groups are used for access control.
Designer	A graphical tool for developing business process and developing supporting services, business rules, and tasks.
Document	A structured message (containing data) that flows through the Integration Server system
Document type	Schema-like definition that describes a document’s structure.
Element	An object in Integration Server that may have an ACL placed on it (e.g. service)
Enterprise resource	An application or system and that is managed and protected by the organization responsible for the TOE. Examples of this are a CRM system, application database, or custom HR system.
Integration Server	The webMethods Integration Server is the central run-time component of the webMethods suite and supports integration with Web services, packaged applications, mainframe and legacy applications, databases, and more. Integration Server's function in integration solutions is the execution of services.
JMS trigger	A trigger that receives messages from a destination (queue or topic) on a JMS provider and then processes those messages.

Portlet	A user interface component that generates dynamic content. My webMethods Server provides a container for portlets and aggregates dynamic content for presentation.
Security realm	A collection of server resources (such as pages and portlets) that share the same ACL.
Server verb	An operation (such as publishing, deleting, updating, subscribing, and setting permissions), which is available through the My webMethods Server API.
Service	A service is an Integration Server-resident unit of functionality that clients can invoke. (A service might be an entire application or used as part of a larger application.)
Task	A business process activity managed by Task Engine and performed by an end user.
Trigger	An object that establishes a subscription to publishable document types and specifies services that will process documents received by the subscription.
Web service	A collection of functions that are packaged as a single unit and published to a network for use by other software programs.

2. TOE Description

The Target of Evaluation (TOE) is webMethods Business Process Management Suite 8.2 SP2, which consists of Integration Server with Process Engine, webMethods Broker, My webMethods Server with Task Engine, and Designer. webMethods BPMS is part of the larger webMethods Product Suite. The BPMS product provides application services together with business process management (BPM) in a service-oriented architecture (SOA). The TOE encompasses the security functions of the BPMS. This section introduces the webMethods BPMS product and describes the parts of the product that make up the TOE. Section 2.1 provides a brief overview of relevant BPMS product components while section 2.2 describes the security functions of the TOE.

2.1 webMethods BPMS Product Overview

A *business process* is a series of business activities that are performed in a specific order, by a variety of applications, systems, employees, and external businesses, according to defined business rules. An example of a business process includes handling a purchase order from receipt through fulfillment.

BPMS provides business process management, which enables an enterprise to automate business processes. Also, business processes typically involve many variables and conditions, and the longer they run, the more likely the variables and conditions are to change. For example, a supplier might temporarily run out of parts needed to fill orders. Business process management enables users to act on running processes in response to such changes; in the example above, a user could suspend order fulfillment processes until parts are available again.

A variety of users interact with the BPMS. Administrators install and configure the BPMS. Developers and business analysts define business processes including services that each process presents. A *service* is an Integration Server-resident unit of functionality that users or processes can invoke. Service users initiate a business process through the services it presents. For example, a customer would initiate an order fulfillment process by invoking a service for submitting a purchase order, which the service would enter into the BPMS as a *document* (a structured message). The BPMS performs the activities of a business process when it receives a document through a service. Business process activities may include *tasks,* which are activities managed by Task Engine and performed by people. Hence, users also interact with running business processes. Continuing the example, a customer representative would perform a task to approve a customer's purchase order. Other users are *enterprise resources* (applications and systems managed by the organization responsible for the BPMS) that send documents and messages to a business process as well as external systems that send messages. For example, a backend system might send a document to notify the order fulfillment process when an order has been shipped. BPMS also can send information to systems and applications (such as updating a database). BPMS uses an *adapter* to transform BPMS documents into a format used by a system or application.

webMethods BPMS includes Designer, Integration Server with Process Engine, webMethods Broker, and My webMethods Server with Task Engine. Other products that are part of BPMS and the webMethods Product Suite are not included in the evaluation.

Figure 1 shows interaction with BPMS components in the context of a business process. The figure illustrates a minimal deployment of the BPMS. Integration Server presents the services that a user invokes to initiate a business process. Process Engine transitions the business process from activity to activity using services and documents. Documents are used in business process activities (and to synchronized Process Engines in deployments with multiple Integration Servers). Process Engine passes control to the Task Engine on My webMethods server for tasks. The Task Engine handles presenting information to a user as well as accepting user input. Task Engine returns control to Process Engine once a task is complete. An activity may use information from enterprise resources and external systems. Enterprise resources and external systems can send information to Integration Server as Java Message Service (JMS) messages. Enterprise resources can send JMS messages via webMethods Broker, which is a message router that transfers messages among BPMS components and enterprise resources. Integration Server can monitor enterprise resource using adapters.

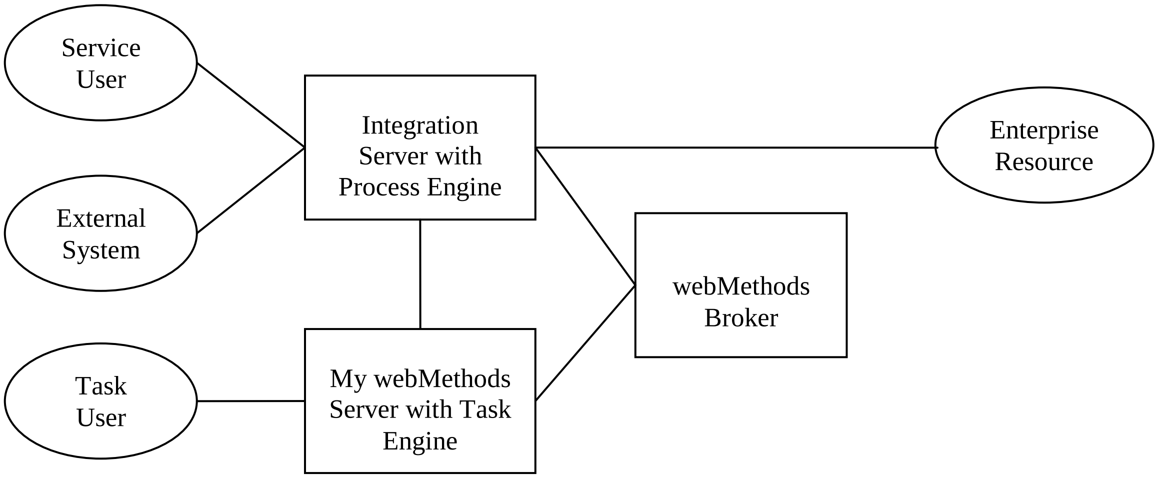
Service User
External System
Task User
Integration Server with Process Engine
My webMethods Server with Task Engine
webMethods Broker
Enterprise Resource

Figure 1 User Interactions at Run Time with BPMS Components

The BPMS components must be installed and configured before the BPMS can provide services and business processes. Figure 2 shows interactions with BPMS components at design time. An administrator can configure Integration Server using the web Administration interface built in to Integration Server. An administrator can configure My webMethods Server and webMethods Broker using My webMethods Server. A developer or business analyst uses Designer to develop services on Integration Server and to define business processes on Integration Server and My webMethods Server.

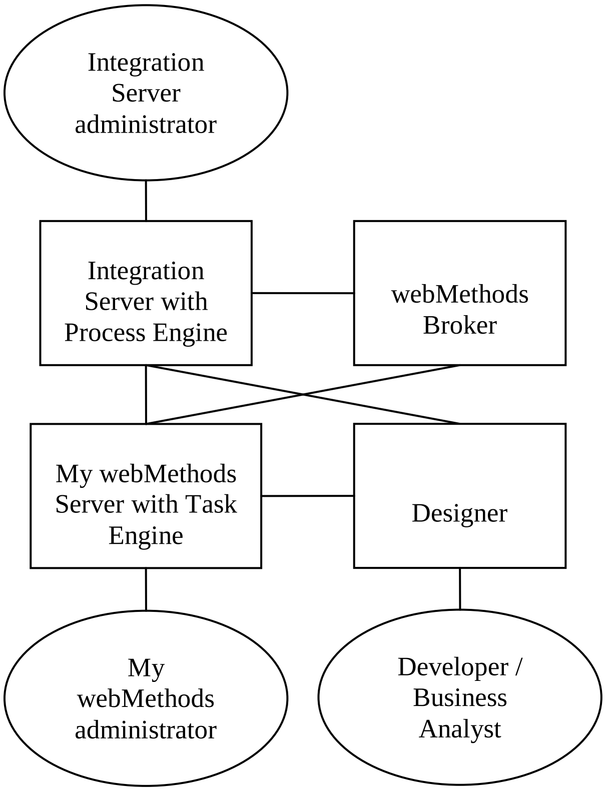
Integration Server administrator
Integration Server with Process Engine
webMethods Broker
My webMethods Server with Task Engine
Designer
My webMethods administrator
Developer / Business Analyst

Figure 2 User Interactions at Design Time with BPMS Components

2.2 TOE Overview

The webMethods BPMS TOE encompasses the security functions of the webMethods BPMS. The products that make up the TOE are:

- Integration Server with Process Engine,
- webMethods Broker, and
- My webMethods Server with Task Engine.
- Designer

2.2.1 Designer

Designer is an Eclipse-based graphical development application. It provides tools for developing services and designing business processes including tasks. A service is not available in the run-time environment until it is built and deployed to Integration Server. Similarly, a business process is not available in the run-time environment until it is published to Integration Server and a task is not available until published to My webMethods servers. Integration Server and My webMethods Server authenticate Designer users. In the run-time environment, they control access to objects as described in sections 2.2.2 and 2.2.4. Integration Server and My webMethods Server can be configured to deny Designer access in the run-time environment.

Designer uses the operational environment to support TLS connections to Integration Server and My webMethods Server.

2.2.2 Integration Server with Process Engine

Integration Server listens for user requests on one or more ports. A user submits a request as a file or through a web service. The Integration Server provides the following mechanisms for a user to submit a file:

- Post a file via HTTP or HTTPS,
- Transfers a file via FTP or FTPS,
- Place a file in a file system directory that the Integration Server monitors (file polling), and
- Email a file as an attachment via POP or IMAP.

Integration Server uses a FIPS 140-2 validated cryptographic module in the operational environment for cryptographic functions.

Integration Server provides web services defined through Web Service Description Language (WSDL), including WS-Security signed and encrypted messages.

A system can provide information to Integration Server through documents and triggers. A *trigger* is an object that establishes a subscription to publishable document types and specifies services that will process documents received by the subscription. An adapter translates information it receives from a system into an adapter notification document, which it publishes to Integration Server. A trigger subscribes to an adapter notification document type. The trigger invokes a service to process each document it receives.

A service developer chooses whether to implement a service with an insecure or secure protocol (for example, HTTP or HTTPS). The choice depends on an organization’s security policy, the type of service, and the operational environment. For example, an insecure protocol for a service that publishes publically-available information while a secure protocol would be appropriate for information deemed sensitive under the organization’s security policy.

Integration Server can restrict access to services by origin (that is, IP address), port, requested service, and identity of the user making the request. It can limit access to services internal to a business process, if configured to do so. Integration Server can be configured to authenticate the identity of a user requesting a service. It can maintain and store user definitions (user name, password, and other user attributes). The Integration Server can authenticate using the user database it maintains or using a user database maintained by a My webMethods Server or using an external LDAP server. A service or business process may entail documents received from enterprise resources (for example, databases). Integration Server controls the services invoked when it receives such a document.

Users develop business processes with Designer. Integration Server authenticates a Designer user before granting access to services, business processes, and other Integration Server resources. Integration Server can restrict access to services, business processes, and other Integration Server resources.

Integration Server provides an interface to manage security functions through Integration Server Administrator. Integration Server Administrator allows monitoring server activity, management of user accounts, and setting operating parameters. It is a browser-based application that uses services to accomplish its work.

The Integration Server audits security-relevant requests and events.

Process Engine controls the run-time execution of business processes on Integration Server. However, Process Engine does not provide any security functionality.

2.2.3 webMethods Broker

webMethods Broker is a message router for asynchronous publish-subscribe messaging and point-to-point messaging. For publish-subscribe messaging, a messaging client connects to webMethods Broker and publishes a document. Broker queues the document for all subscribers. It sends the document to each subscriber when a messaging client for the subscriber connects to Broker. webMethods Broker supports two messaging protocols: a webMethods-proprietary messaging protocol and the JMS protocol.

Messaging clients are enterprise resources, My webMethods Server, and Integration Server. Enterprise resources use webMethods Broker's message routing capabilities. TOE administrators manage webMethods Broker through Broker User Interface[1], a graphical user interface presented by My webMethods Server. Instances of Integration Server exchange documents through webMethods Broker (for example, to perform services or triggers).

webMethods Broker can authenticate messaging clients. It provides two authentication mechanisms: password-based authentication and two-way TLS authentication with X.509 certificates. In addition, TLS supports secure communication for sessions including sessions initiated with password-based authentication.

A messaging client connecting to webMethods Broker provides a *client group*, which represents a particular group or category of client (for example, administrators, customers, Integration Server processes). webMethods Broker uses the identity of the client and access control lists (ACLs) to determine whether the client is a member of the identified client group. If the client is a member, webMethods Broker grants the client access to documents that have a document type associated with the client group. In addition, webMethods Broker uses client groups to restrict access to its management functions.

webMethods Broker audits security relevant events. It stores audit records in files in the local file system.

2.2.4 My webMethods Server with Task Engine

webMethods BPMS components contain web applications for using and managing the components. These applications are called webMethods applications. My webMethods Server is a run-time container for functions made available by webMethods applications. The user interface for these functions is called My webMethods. My webMethods provides a ready-made environment in which users can perform functions on webMethods applications, and administrators can manage access to those functions. In particular, users access business process tasks through My webMethods. Task Engine on My webMethods Server works with Process Engine on Integration Server to control the tasks presented to users. My webMethods Server authenticates users before granting access. It maintains a database of users, which it may share with Integration Server. Like Integration Server, My webMethods Server supports authentication using an external LDAP server. My webMethods Server supports secure communication (HTTPS) with My webMethods clients and other TOE components using a FIPS 140-2 validated cryptographic module from the operational environment.

Task Engine controls the run-time execution of tasks on My webMethods Server. However, Task Engine does not provide any security functionality.

[1] webMethods Broker provides command-line utilities for some management tasks: create and configure a Broker Server, backup/restore metadata for a Broker Server, manage Broker and Broker Monitor (internal components of webMethods Broker).

2.3 TOE Architecture

The TOE is a distributed application. Figure 3 shows the general network architecture for webMethods BPMS. One or more Integration Server(s) presents services to users. One or more Integration Servers provide access to enterprise resources. Business analysts and IT developers develop services and business processes with Designer. Users perform business process tasks through My webMethods. Administrators manage a deployment, including security, through Integration Server Administrator and My webMethods graphical user interfaces. One or more webMethods Brokers provides messaging between the Integration Servers and My webMethods Server.

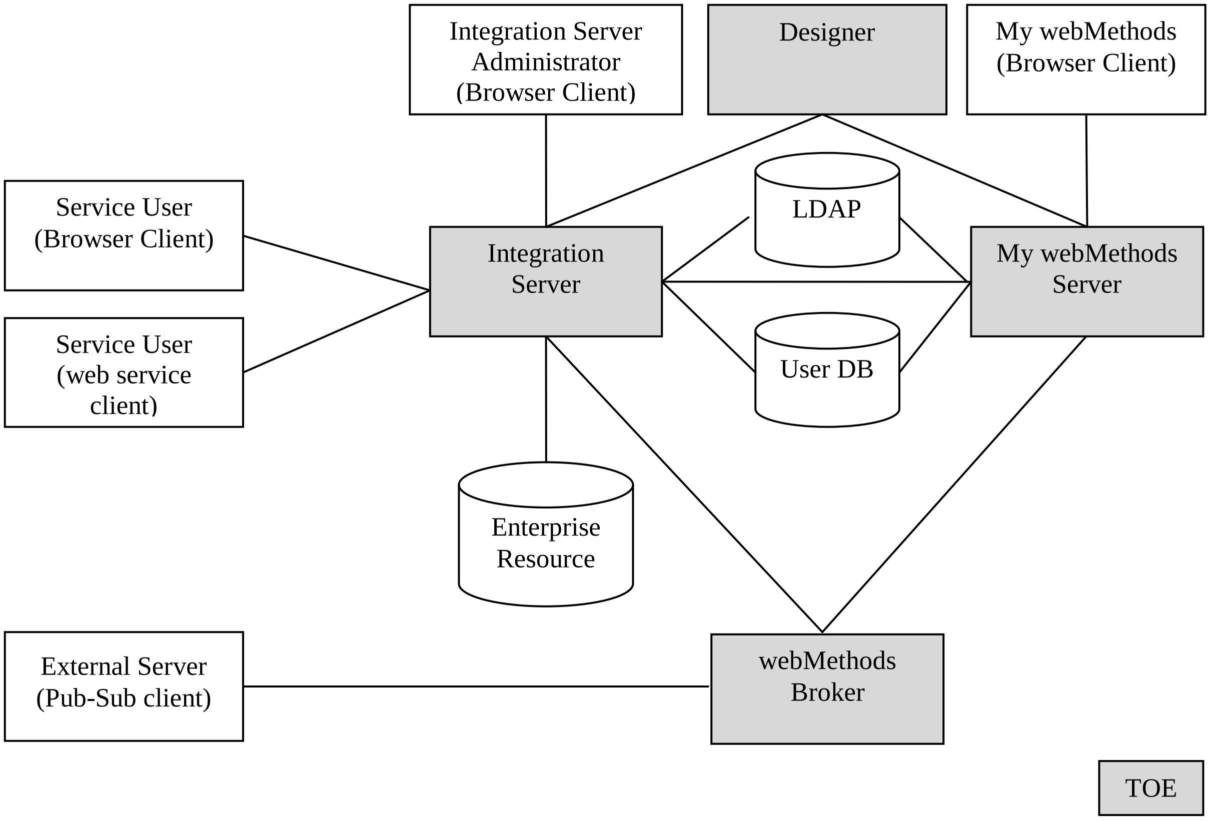
Integration Server Administrator (Browser Client)
Designer
My webMethods (Browser Client)
Service User (Browser Client)
LDAP
Integration Server
My webMethods Server
Service User (web service client)
User DB
Enterprise Resource
External Server (Pub-Sub client)
webMethods Broker
TOE

Figure 3 TOE and Its Operational Environment

Organizations deploy webMethods BPMS in a variety of operational environments. In some deployments, the operational environment strictly controls access by untrusted users to TOE components (for example, with firewalls) and protects communication between TOE components (for example, through isolated networks). In other deployments, internal users (for example, users performing tasks through My webMethods) may work on the same internal network as TOE components.

Consequently, the evaluated configuration includes a range of deployments. Some restrictions apply in all deployments. The TOE would be configured to protect its external interfaces. The TOE relies on the operational environment to restrict network access from public networks (for example, the Internet) to the TOE. The operational environment would be configured to protect TOE resources stored in the environment (that is, TOE executables and TOE data) as well the resources the TOE relies upon (that is, cryptographic modules, JVMs, LDAP server, RDBMS server, and application server).

In an environment where TOE communication and intra-TOE interfaces are visible to untrusted users (that is, end users, external servers, and non-administrative My webMethods users), the TOE would be configured to protect

TOE communication (including management sessions from Integration Server Administrator and My webMethods Server) and intra-TOE interfaces. The TOE would be configured to use cryptographic functions of the operational environment to protect communication between TOE components from disclosure and undetected modification. TOE components and the environment would be configured to protect communication between the components and the browsers and servers the TOE uses in the operational environment (for example, LDAP and RDBMS servers). In addition, each instance of webMethods Broker would be configured to authenticate all clients.

In an operational environment where TOE communication and intra-TOE interfaces are not visible to untrusted users, the TOE may rely on the operational environment for protection. TOE network communication may be in plain text where the operational environment prevents modification and disclosure. webMethods Broker need not authenticate clients, since no unauthorized clients would have access to Broker interfaces.

2.3.1 Physical Boundaries

2.3.1.1 TOE Components

The TOE consists of:

- One or more instances of Integration Server version 8.2 (including Process Engine),
- One or more instances of webMethods Broker version 8.2, and
- One or more instances of My webMethods Server 8.2 (including Task Engine), and
- One or more instances of Designer.

Each Integration Server, webMethods Broker, and My webMethods Server runs on a separate host. The physical boundary of the TOE is the boundary of the TOE software within each host platform. Figure 4, Figure 5, Figure 6, and Figure 7 illustrate the boundary between each TOE component and its operational environment.

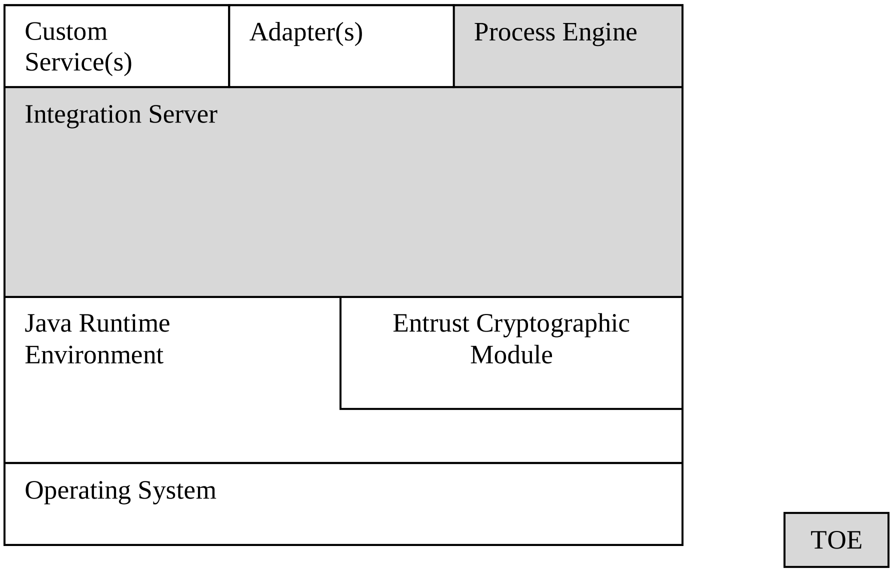

Figure 4 Integration Server Architecture

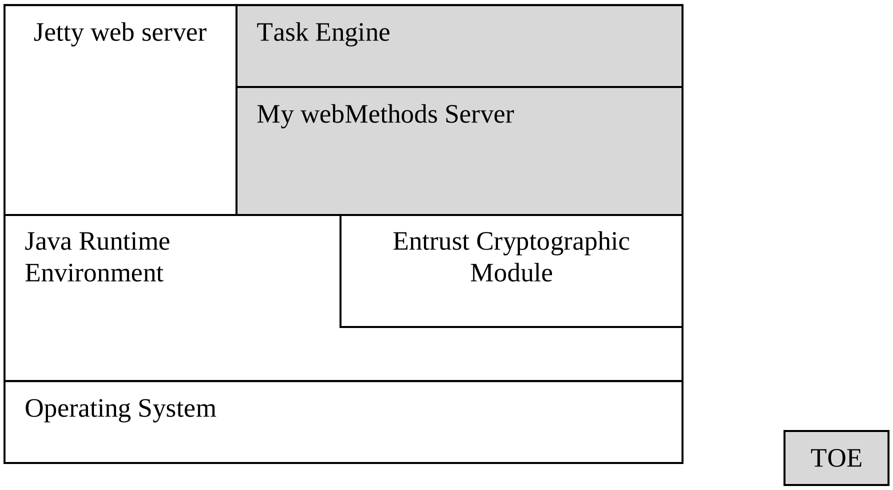

Figure 5 My webMethods Server Architecture

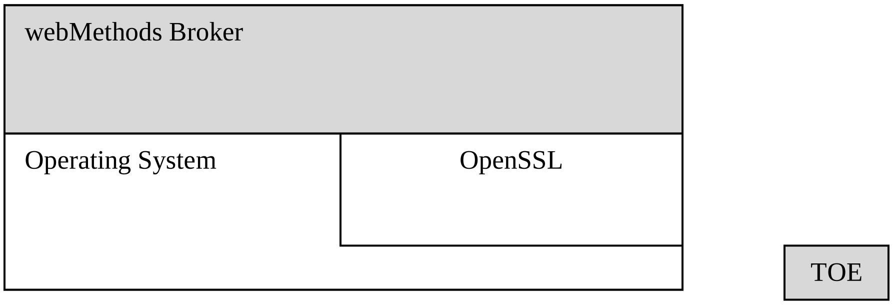

Figure 6 webMethods Broker Architecture

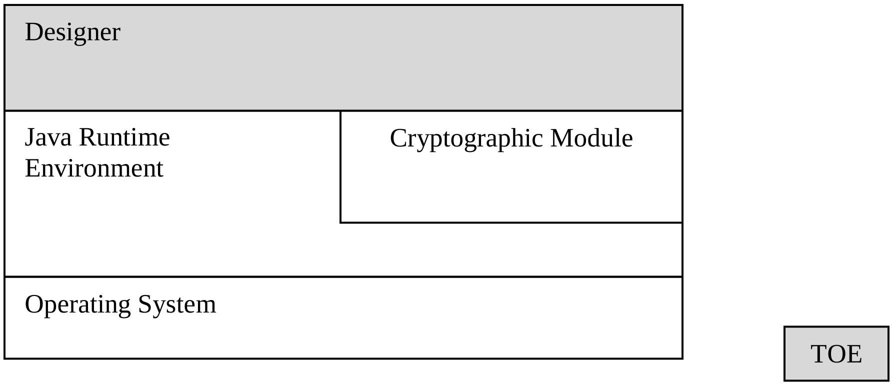

Figure 7 Designer Architecture

2.3.1.2 Operational Environment

The TOE requires host platforms including operating systems, Java Development Kit (JDK), and relational database management system (RDBMS). A JDK includes the Java Runtime Environment (JRE), in which TOE components execute. Suitable combinations of operating system, JDK, and RDBMS are described in *webMethods System Requirements 8.2*[2].While there are some restrictions on combinations, in general the TOE components run with the following operating systems, JDK, and RDBMS.

Operating systems:

- Windows Server 2003 Standard and Enterprise Edition (x86 and x86-64)
- Windows Server 2008 Standard and Enterprise Edition (x86 and x86-64)
- Windows Server 2008 R2 Standard and Enterprise Edition (x86-64)
- SUSE Linux Enterprise Server 11 SPx (x86 and x86-64)
- Red Hat Enterprise Linux Server 5.x (x86 and x86-64)
- Red Hat Enterprise Linux Server 6.x (x86-64)
- Solaris 10 (64 Ultra SPARC and x86-64)
- HP-UX 11i v3 (PA-RISC64 and IA64)
- AIX 6.1 (Power 64-bit)

JDK:

- Oracle Java 1.6.0_24 or later (Windows, Linux, Solaris)
- HP Java 1.6.0.09 or later (PA-RISC 64)
- HP Java 1.6.0.07 or later (IA64)
- IBM Java 1.6.0 SR9 or later (AIX)
- Oracle Java 1.7.0_13 or later (Windows, Linux, Solaris)

RDBMS

[2] http://documentation.softwareag.com/webmethods/wmsuites/wmsuite8-2_ga/SysReqs_Installation_and_Upgrade/8-2_System_Requirements.pdf

- Oracle 11g (R1 and R2)
- DB2 9.7 LUW
- SQL Server 2003 and 2008

The operational environment contains commodity PC and browsers for Integration Server Administrator and My webMethods. Support browsers are:

- Microsoft Internet Explorer 7.x and 8.x
- Mozilla Firefox 3.x

The TOE may use an LDAP Directory server in the operational environment.

The operational environment includes components provided by Software AG and installed with the TOE. These components are:

- Entrust Authority™ Security Toolkit 7.2 for the Java© Platform,
- OpenSSL FIPS Object Module, and
- Jetty web server.

The Integration Server and My webMethods Server TOE components rely on Entrust Authority™ Security Toolkit 7.2 for the Java® Platform for cryptographic functions it provides. This toolkit is a FIPS 140-2 validated cryptographic module (certificate #802). webMethods Broker relies on OpenSSL FIPS Object Module 1.2.3 for cryptographic functions (FIPS 140-2 certificate #1051). My webMethods Server relies on the Jetty web server to present the My webMethods graphical user interface. Integration Server uses a file system to maintain its user database as well as an Apache Derby to store non-security-relevant databases. In the evaluated configuration, Integration Server may be configured to use an external RDBMS in place of the embedded database server for the user database.

2.3.1.3 Excluded Functionality

The TOE does not include the following aspects of the webMethods BPMS:

- Custom services built on the platform (services and packages added by end consumer)
- WmTomcat package
- Broker Access Labels (not enabled by default)

Other products in the webMethods Product Suite (such as webMethods Adapters, EntireX, ApplinX, Blaze Advisor, and CentraSite) complement and supplement the capabilities of webMethods BPMS. Other products in the webMethods Product Suite are distinct from webMethods BPMS. They are not included in the evaluation, which consequently provides no information about their security features. Other products in the webMethods Product Suite may be used with an evaluated TOE.

2.3.2 Logical Boundaries

This section summarizes the security functions provided by webMethods BPMS:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Trusted path/channels

2.3.2.1 Security audit

The TOE records security-relevant events in audit records. Integration Server, webMethods Broker, and My webMethods Server generate audit records, which are stored in distinct audit trails. Security-relevant events include TOE configuration changes made by administrators, business process changes made by business analysts and IT developers, and service access by users. The set of audited events is configurable. Each audit record contains information suitable for analysis including date and time of the event, type of event, responsible identity, and event outcome.

The TOE provides tools to select Integration Server’s audit records by time or number of records and view the subset in chronological or reverse chronological order. The TOE limits access to these tools. The Integration Server stores the audit records in a database, which the operational environment protects.

The TOE webMethods Broker and My webMethods Server store their audit trails in local operating system files, which the operating system protects. The TOE relies on the operational environment (for example, using a text editor) for review of webMethods Broker and My webMethods Server audit records. My webMethods Server also provides ability to view audit trails via its web interface.

In addition, the TOE relies on the operational environment for reliable time stamps.

2.3.2.2 Cryptographic support

Integration Server built-in services include services to digitally sign documents and verify digital signatures and to encrypt and decrypt documents. Business analysts and IT developers can use the built-in cryptographic services to present end users with digital signature and secure hash functions. These services rely on the operational environment, specifically JDK cryptographic providers and cryptographic functions of the Entrust FIPS 140-2 validated cryptographic module. The JDK cryptographic providers implement World Wide Web Consortium (W3C) XML standards. The cryptographic module provides cryptographic algorithms (RSA, AES, and Triple DES). The built-in services use the operational environment to make the cryptography capabilities available for use with documents.

The TOE supports secure communication with users and between TOE components with TLS. The TOE configuration determines which ports require TLS. The TOE relies on the operational environment for key management and cryptographic functions, which are provided by the JDK (for example, TLS session establishment) and FIPS 140-2 validated cryptographic modules (for example, AES encryption and decryption). Cipher suites used by the TOE include Triple DES, AES, DSA, and RSA using common key sizes (192 bits, 128 to 256 bits, 1024 bits, and 1024 to 4096 bits, respectively).

By default, JDK limits cryptographic key sizes. The defaults can be changed with a jurisdiction policy file. In order to allow use of longer keys (for example, 256-bit AES), the JVM in the operational environment must have an unlimited strength jurisdiction policy file.

2.3.2.3 User data protection

The TOE provides services to end users and provides development tools to business analysts and IT developers. The TOE enforces policies to limit access to services, tools, information, and resource to authorized users. The policy can limit service requests based on group membership, location (that is, source IP address), and service requested. It can limit access to other resources based on user name, group membership, role membership, owner of the resource, and operation requested. Administrators can configure access control policies and user permissions. Business analysts and IT developers may set user permissions on a limited basis (for example, on resources they own).

2.3.2.4 Identification and authentication

Both access control policy enforcement and security audit depend on assured identity of users. The TOE provides password-based authentication and supports cryptographic authentication. In addition, the TOE supports use of an external LDAP server for authentication. The TOE enforces password composition rules. The TOE associates security attributes with each authenticated user. These attributes may include user name, group membership, role membership, password, and X.509 public key certificate. The TOE authentication policy is configurable so that some services and operations would require authentication while others would not.

2.3.2.5 Security management

The TOE uses roles to manage privileges. Several Integration Server, webMethods Broker, and My webMethods Server roles have privilege to manage security functions. Collectively, these roles are identified with the Authorized Administrator role defined in this ST. The TOE also uses roles to enforce its access control policy.

The TOE includes tools needed to manage its security functions and limits their use to Authorized Administrators. An Authorized Administrator can configure security audit, network port restrictions, user accounts, groups, roles, access control policy, and password policy rules.

2.3.2.6 Trusted path/channels

The TOE supports secure communication for service requests and responses. The TOE can be configured to use a FIPS 140-2 validated cryptographic module to present services through HTTPS and, for Integration Server, FTPS. In addition, the TOE can be configured to use TLS version 1.0 to protect communication among TOE components and between webMethods Broker and its clients. The TOE supports LDAPS (LDAP over TLS) for connections with LDAP directories in the operational environment. The TOE relies on the operational environment for secure communication with RDBMSs.

2.4 TOE Documentation

Software AG offers a series of documents that describe the installation of webMethods BPMS as well as guidance for subsequent use and administration of the applicable security features.

- Using the Software AG Installer
- Understanding the webMethods Product Suite
- Working with My webMethods
- Installing webMethods Products
- Administering Integration Server
- Integration Server Build-In Services Reference
- Dynamic Server Pages and Output Template Developer’s Guide
- Publish-Subscribe Developer’s Guide
- Web Services Developer’s Guide
- Administering Process Engine
- Working with BPM Tasks
- Administering webMethods Broker

TOE guidance documents are available from the Software AG documentation web site (http://documentation.softwareag.com/).

3. Security Problem Definition

The TOE security environment 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:

- Organizational Policies that the TOE and the environment of the TOE fulfill
- Threats that the TOE and the environment of the TOE counters
- Assumptions made about the operational environment and the intended method of use for the TOE

Furthermore, the TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL 2 augmented with ALC_FLR.1 as defined in the CC.

The assets the TOE directly protects are the services and business processes it provides. The TOE indirectly protects resources used by services and business processes (for example, backend databases and mainframes).

3.1 Organizational Policies

P.AUDIT A product that provides security functions must be capable of producing an audit trail of security-relevant events.

3.2 Threats

T.PROCESS A user may view, read, or write business process information without permission using BPMS interfaces.

T.SERVICE A user may invoke a service, web page, or business process task without permission.

T.TAMPER A user may modify the BPMS or its configuration in order to access services or business process information without permission.

T.TRANSIT A user may capture network traffic in order to observe or alter service invocation or business process information of another user.

3.3 Assumptions

A.ADMIN Those responsible for the TOE are trusted to correctly configure and operate the TOE according to the instructions provided by the TOE documentation.

A.CRYPTO The operational environment must provide key management and underlying cryptographic functions to support TOE cryptographic operations.

A.HOST There is no untrusted software on the servers hosting the TOE.

A.PHYSICAL The TOE is protected from physical attack.

A.REVIEW The operational environment must provide text processing tools to supplement the TOE capability to review the security audit trail.

A.USERS Users will protect authentication data in their possession.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

4.1 Security Objectives for the TOE

O.ACCESS[3] The TOE must allow a user to access only those services, Integration Server resources, webMethods Broker documents, and My webMethods Server resources for which the user is authorized.

O.ADMIN The TOE must provide functions to manage the security functions of the TOE and restrict the use of these functions to authorized administrators.

O.AUDIT The TOE must provide the capability to produce audit trail of security-relevant events.

O.DOC_SEC The TOE must provide capabilities to digitally sign documents, verify digital signatures on documents, encrypt documents, and decrypt documents for documents exchanged during service invocation.

O.ID_AUTH The TOE must be able authenticate the identity of a user requesting access to security management functions or restricted services, Integration Server resources, webMethods Broker documents, or My webMethods Server resources before allowing access.

O.REVIEW The TOE must provide authorized users with the capability to review the Integration Server security audit trail.

O.TLS The TOE must provide the capability to secure communications via TLS between a user and the TOE and among TOE components.

4.2 Security Objectives for the Environment

OE.COMM The operational environment must protect communications among TOE components and between users and the TOE.

OE.CRYPTO The JDK and FIPS 140-2 validated cryptographic modules must provide key management and underlying cryptographic functions to support TOE TLS and document cryptographic operations.

OE.HOST Those responsible for the TOE must ensure that there is no untrusted software on the servers hosting the TOE.

OE.INSTALL Those responsible for the TOE must ensure that the TOE is delivered and installed in a secure manner as outlined in supplied guidance.

OE.OPERATE Those responsible for the TOE must manage and operate the TOE in a secure manner as outlined in the supplied guidance.

OE.PERSONNEL Personnel working as authorized administrators of the TOE must be carefully selected and trained for proper operation of the system.

OE.PHYSICAL Those responsible for the TOE must ensure that the TOE is protected from physical attack.

OE.REVIEW The operational environment must provide tools to review the My webMethods Server and webMethods Broker text audit trails.

OE.STORE The operational environment must provide the capability to store and protect the security audit trail produced by the TOE.

OE.TIME The operational environment must provide the TOE with reliable time stamps.

[3] See FDP_ACC.1c, and FDP_ACC.1d for the definition of resources.

OE.USERS Those responsible for the TOE must instruct users to protect authentication data in their possession.

5. IT Security Requirements

The security requirements for the TOE have all been drawn from Parts 2 and 3 of the Common Criteria. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a low to moderate degree of assurance that those security functions are properly realized.

5.1 Extended Component Definition

This ST contains no extended components. All security functional requirements are drawn from CC Part 2. All security assurance requirements are drawn from CC Part 3.

5.2 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by webMethods BPMS.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
	FAU_SEL.1: Selective audit
FCS: Cryptographic support	FCS_COP.1a: Cryptographic operation – AES, TDES
	FCS_COP.1c: Cryptographic operation – Digital Signature
	FCS_COP.1d: Cryptographic operation – Secure Hash
FDP: User data protection	FDP_ACC.1a: Subset access control – Service
	FDP_ACC.1b: Subset access control – Trigger
	FDP_ACC.1c: Subset access control – Integration Server Resources
	FDP_ACC.1d: Subset access control – My webMethods Server Resources
	FDP_ACF.1a: Security attribute based access control – Service
	FDP_ACF.1b: Security attribute based access control – Trigger
	FDP_ACF.1c: Security attribute based access control – Integration Server Resources
	FDP_ACF.1d: Security attribute based access control – My webMethods Server Resources
	FDP_ITT.1: Basic internal transfer protection
FIA: Identification and authentication	FIA_ATD.1a: User attribute definition – Integration Server
	FIA_ATD.1b: User attribute definition – My webMethods Server
	FIA_SOS.1: Verification of secrets
	FIA_UAU.1: Timing of authentication
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_UID.1: Timing of identification
FMT: Security management	FMT_MOF.1: Management of security functions behavior
	FMT_MSA.1a: Management of security attributes – Service
	FMT_MSA.1b: Management of security attributes – Trigger
	FMT_MSA.1c: Management of security attributes – Integration Server Resources

	FMT_MSA.1d: Management of security attributes – My webMethods Server Resources
	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_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted path

Table 1 TOE Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the **[*not specified*]** level of audit; and

c) **[the following auditable events:**

- **Modification of the audit configuration that occur while the audit collection functions are operating**
- **All requests to perform an operation on an Integration Server resource (that is, package, folder, service, file, specification, schema, document type, or trigger)**
- **All requests to perform an operation on a My webMethods Server resource (that is, folder, page, portlet, link, document, file, server verb, webMethods application, Task, Workspace, Content Object, Portlet Type, and Security Realm)**
- **Modification of the behaviour of the functions in the TSF**
- **Modification of the values of security attributes**
- **Modification of the default setting of permissive or restrictive rules**
- **Modification of the initial values of security attributes**

].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **[none]**.

5.2.1.2 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[Authorized Administrator]** with the capability to read **[all information]** 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.

5.2.1.3 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.2.1.4 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to apply **[selection and ordering]** of audit data based on **[**

- **Selection: Time range of audit data or number of audit records or both or**
- **Ordering: Chronological or reverse chronological order or**
- **Both selection and ordering**

].

5.2.1.5 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) **[*event type*]**;

b) **[none]**.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Cryptographic operation – TLS (FCS_COP.1a)

FCS_COP.1a.1 The TSF shall perform **[encryption, decryption]** in accordance with a specified cryptographic algorithm **[AES and TDES operating in CBC mode]** and cryptographic key sizes **[**

- **AES: 128, 256**
- **TDES: 168**

] that meet the following: **[FIPS 197 and FIPS 46]**.

5.2.2.2 Cryptographic operation – Digital Signature (FCS_COP.1c)

FCS_COP.1c.1 The TSF shall perform **[digital signature]** in accordance with a specified cryptographic algorithm **[RSA, DSA]** and cryptographic key sizes **[**

- **RSA: 1024, 2048, 3072, 4096**
- **DSA: 1024**

] that meet the following: **[FIPS 186-3 for RSA and FIPS 180-3 for DSA]**.

5.2.2.3 Cryptographic operation – Secure Hash (FCS_COP.1d)

FCS_COP.1d.1 The TSF shall perform **[cryptographic Hashing]** in accordance with a specified cryptographic algorithm **[SHA-1]** and cryptographic key sizes **[160-bits**
] that meet the following: **[FIPS 180-3]**.

5.2.3 User data protection (FDP)

5.2.3.1 Subset access control – Service (FDP_ACC.1a)

FDP_ACC.1a.1 The TSF shall enforce the **[Service SFP]** on **[**
Subjects: connections on behalf of users
Objects: services and files
Operations: execute service and request file
].

5.2.3.2 Subset access control – Trigger (FDP_ACC.1b)

FDP_ACC.1b.1 The TSF shall enforce the **[Trigger SFP]** on **[**
Subjects: triggers on behalf of enterprise resources
Objects: services
Operations: execute
].

5.2.3.3 Subset access control – Integration Server Resources (FDP_ACC.1c)

FDP_ACC.1c.1 The TSF shall enforce the **[Integration Server Resources SFP]** on **[**
Subjects: connections on behalf of users

Objects: Integration Server resource (that is, package, folder, file, service, specification, schema, document type, or trigger)
Operations: list, read, and write
].

5.2.3.4 Subset access control – My webMethods Server Resources (FDP_ACC.1d)

FDP_ACC.1d.1 The TSF shall enforce the **[My webMethods Server Resources SFP]** on **[**
Subjects: users
Objects: My webMethods Server Resources presented at the My webMethods user interface: Workspace and workspace content (including tasks
Operation: Read, write, execute
].

5.2.3.5 Security attribute based access control – Service (FDP_ACF.1a)

FDP_ACF.1a.1 The TSF shall enforce the **[Service SFP]** to objects based on the following: **[**
Subject attributes:

- **User name**
- **Group membership**
- **Presumed source IP address**
- **Destination port**

Object attributes:

- **Service URL path**
- **Execute ACL**
- **Enforce Execute ACL property**

].

FDP_ACF.1a.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[If execution is not explicitly authorized by FDP_ACF.1a.3 or explicitly denied by FDP_ACF.1a.4, then**

a) **If the user name belongs to at least one group allowed by the execute ACL and the user name does not belong to any group denied by the execute ACL, then the TSF allows the operation; else**
b) **The TSF denies execution**

].

FDP_ACF.1a.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[If execution is not explicitly denied by FDP_ACF.1a.4, then**

a) **If the service is not a top-level service and the Enforce Execute ACL property is set to "When top-level service only", then the TSF allows execution.**

].

FDP_ACF.1a.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[**

a) **If the presumed source IP address is not in the list of IP addresses allowed for the destination port, then the TSF denies execution or**
b) **If the service is a top-level service (that is, directly invoked by a client or dynamic server page (DSP)) and the service URL path is not in the list of services allowed for the destination port, then the TSF denies execution**

].

5.2.3.6 Security attribute based access control – Trigger (FDP_ACF.1b)

FDP_ACF.1b.1 The TSF shall enforce the **[Trigger SFP]** to objects based on the following: **[**
Subject attributes:

- **User name**
- **Group membership**

Object attributes:

- **Execute ACL**
- **Enforce Execute ACL property**

].

FDP_ACF.1b.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[if execution is not explicitly authorized by FDP_ACF.1b.3, then**

a) If the user name belongs to at least one group allowed by the execute ACL and the user name does not belong to any group denied by the execute ACL, then the TSF allows execution; else

b) The TSF denies execution

].

FDP_ACF.1b.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[**

a) If the service is internal (that is, invoked by another service) and the Enforce Execute ACL property is not set, then the TSF allows execution.

].

FDP_ACF.1b.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[none]**.

5.2.3.7 Security attribute based access control - Integration Server Resources (FDP_ACF.1c)

FDP_ACF.1c.1 The TSF shall enforce the **[Integration Server Resources SFP]** to objects based on the following: **[**

Subject attributes:

- **User name**
- **Group membership**

Object attributes:

- **List ACL**
- **Read ACL**
- **Write ACL**

].

FDP_ACF.1c.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

a) If the user name belongs to at least one group allowed by the ACL corresponding to the operation and the user name does not belong to any group denied by that ACL, then the TSF allows the operation; else

b) The TSF denies the operation

].

FDP_ACF.1c.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[none]**.

FDP_ACF.1c.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[none]**.

5.2.3.8 Security attribute based access control - My webMethods Server Resources (FDP_ACF.1d)

FDP_ACF.1d.1 The TSF shall enforce the **[My webMethods Server Resources SFP]** to objects based on the following: **[**

Subject attributes:

- **User name**
- **Group membership**
- **Role membership**

Object attributes:

- **Owner**
- **ACL**

].

FDP_ACF.1d.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

1. **Users are allowed to access a workspace if they are members of a role that is listed on workspace ACL**
2. **Users are allow to access workspace content if they are members of a group listed on the ACL of the workspace content**
3. **Users are allowed to execute a task if their name is listed on the ACL of the task or they are part of a group that is listed on the ACL of the task**

].

FDP_ACF.1d.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[Authorized administrators have read access to all resources accessible via the My webMethods Interface;**

The owner of a workspace content have full access to the resource

].

FDP_ACF.1d.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[No additional deny rules].**

.

5.2.3.9 Basic internal transfer protection (FDP_ITT.1)

FDP_ITT.1.1 The TSF shall enforce the **[Service SFP, Trigger SFP, Integration Server Resources SFP, and My webMethods Server Resources SFP]** to prevent the **[*disclosure* and *modification*]** of user data when it is transmitted between physically-separated parts of the TOE.

5.2.4 Identification and authentication (FIA)

5.2.4.1 User attribute definition – Integration Server (FIA_ATD.1a)

FIA_ATD.1a.1 The TSF shall maintain the following list of security attributes belonging to individual **Integration Server** users:

[User name,
Group membership,
Password, and
X.509 certificate].

Application Note: The TSF maintains the attributes specified in FIA_ATD.1a for internally-defined users. For users defined in an external LDAP directory, the LDAP directory maintains the same security attributes.

5.2.4.2 User attribute definition – My webMethods Server (FIA_ATD.1b)

FIA_ATD.1b.1 The TSF shall maintain the following list of security attributes belonging to individual **My webMethods Server** users:

[User name,
Group membership,
Role membership, and
Password].

Application Note: The TSF maintains the attributes specified in FIA_ATD.1b for internally-defined users. For users defined in an external LDAP directory, the LDAP directory maintains the same security attributes.

Application Note: webMethods Broker does not maintain user attributes. For users defined in an external LDAP directory, the LDAP directory maintains the user name and password attributes. For users defined using X.509 certificates, webMethods broker uses key stores and trust stores maintained with OpenSSL.

5.2.4.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **[the following minimums for local Integration Server passwords as specified by an Authorized Administrator:**

a) Number of characters (alphabetic characters, digits, and special characters combined),
b) Number of upper case alphabetic characters,
c) Number of lower case alphabetic characters,
d) Number of digits, and
e) Number of special characters (neither alphabetic nor digits)
].

5.2.4.4 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow **[triggers, anonymous services, scheduled services, and anonymous Broker client sessions]** on behalf of the user to be performed before the user is authenticated.

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

5.2.4.5 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide **[password-based, X.509 certificate-based, and external LDAP mechanisms]** to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **[component providing authentication:**

a) Integration Server applies one or more of the following mechanisms as configured and in the order configured by an Authorized Administrator: password-based, X.509 certificate-based, and external LDAP,
b) My webMethods Server applies either password-based or external LDAP mechanism as configured by an Authorized Administrator;
c) **When the TSF applies the external LDAP mechanism, the TSF presents user name and password to an external LDAP directory and uses the directory's decision;**
d) **When the TSF applies the external LDAP mechanism with multiple directories, the TSF queries each LDAP directory until authentication succeeds or all attempts fail**
].

5.2.4.6 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow **[triggers, anonymous services, scheduled services, and anonymous Broker client sessions]** on behalf of the user to be performed before the user is identified.

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

5.2.5 Security management (FMT)

5.2.5.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to **[*enable, disable, and determine the behaviour of*]** the functions **[**

Running Integration Server, webMethods Broker, and My webMethods Server and Security audit log function

] to **[Authorized Administrator]**.

Application Note: The TSF restricts the ability to start (that is, enable), shut down (that is, disable), and query the status of (that is, determine the behavior of) TOE components to Authorized Administrators.

5.2.5.2 Management of security attributes – Service (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the **[Service SFP]** to restrict the ability to **[*query and modify*]** the security attributes **[service URL path, execute ACL, and Enforce Execute ACL property]** to **[Authorized Administrator] and users authorized under the Integration Server Resources SFP**.

5.2.5.3 Management of security attributes – Trigger (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the **[Trigger SFP]** to restrict the ability to **[*query and modify*]** the security attributes **[service URL path, execute ACL, and Enforce Execute ACL property]** to **[Authorized Administrator] and users authorized under the Integration Server Resources SFP**.

5.2.5.4 Management of security attributes – Integration Server Resources (FMT_MSA.1c)

FMT_MSA.1c.1 The TSF shall enforce the **[Integration Server Resource SFP]** to restrict the ability to **[*query and modify*]** the security attributes **[list ACL, read ACL, and write ACL]** to **[Authorized Administrator] and users authorized under the Integration Server Resources SFP**.

5.2.5.5 Management of security attributes – My webMethods Server Resources (FMT_MSA.1d)

FMT_MSA.1d.1 The TSF shall enforce the **[My webMethods Server Resource SFP]** to restrict the ability to **[*query and modify*]** the security attributes **[ACL]** to **[Authorized Administrators] and users authorized under the My webMethods Server Resources SFP**.

5.2.5.6 ** Removed **

5.2.5.7 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **[Service SFP, Trigger SFP, Integration Server Resources SFP, My webMethods Server Resources SFP, and Broker Message SFP]** to provide **[*[constraining (allow access only by authenticated users)]*]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[Authorized Administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.5.8 Management of TSF data (FMT_MTD.1a)

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

a) User accounts,
b) Groups, and
c) Roles,
] to **[Authorized Administrator]**.

5.2.5.9 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to **[*query and modify*]** the **[**

a) Audit event type selection,
b) User attributes except My webMethods Server passwords,
c) Port configuration, and
d) Password policy
] to **[Authorized Administrator]**.

5.2.5.10 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to **[*modify*]** the **[**

a) My webMethods Server user's password

] to **[Authorized Administrator and the My webMethods Server user associated with the password]**.

5.2.5.11 Specification of Management Functions (FMT_SMF.1)

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

a) Review security audit data,

b) Enable, disable, and determine the behavior of the running of TOE components and security audit log,

c) Query and modify object attributes defined in the Service SFP,

d) Query and modify object attributes defined in the Trigger SFP,

e) Query and modify object attributes defined in the Integration Server Resources SFP,

f) Query and modify object attributes defined in the My webMethods Server Resources SFP,

g) Query, delete, and create user accounts, groups, and roles, and

h) Query and modify audit event type selection, user attributes, port configuration, and password policy

].

5.2.5.12 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[Authorized Administrator]**.

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

5.2.6 Protection of the TSF (FPT)

5.2.6.1 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1 The TSF shall protect TSF data from **[*disclosure* and *modification*]** when it is transmitted between separate parts of the TOE.

5.2.7 Trusted path/channels (FTP)

5.2.7.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit **[*the TSF and another trusted IT product*]** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **[sessions with external LDAP Directories as configured by an Authorized Administrator]**.

5.2.7.2 Trusted path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall provide a communication path between itself and **[*remote*] administrative** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **[*modification and disclosure*]**.

FTP_TRP.1.2 The TSF shall permit **[*remote users*]** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for **[*[administrative requests to HTTPS ports]*]**.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2: Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.1: Basic flaw remediation
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

Table 2 EAL 2 augmented with ALC_FLR.1 Assurance Components

5.3.1 Development (ADV)

5.3.1.1 Security architecture description (ADV_ARC.1)

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

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

5.3.1.2 Security-enforcing functional specification (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.

ADV_FSP.2.2d The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1c The functional specification shall completely represent the TSF.

ADV_FSP.2.2c The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3c The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4c For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5c For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

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

ADV_FSP.2.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 Basic design (ADV_TDS.1)

ADV_TDS.1.1d The developer shall provide the design of the TOE.

ADV_TDS.1.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2c The design shall identify all subsystems of the TSF.

ADV_TDS.1.3c The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4c The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5c The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6c The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

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

ADV_TDS.1.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

5.3.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d The developer shall provide operational user guidance.

AGD_OPE.1.1c The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c The operational user guidance shall be clear and reasonable.

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

5.3.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle support (ALC)

5.3.3.1 Use of a CM system (ALC_CMC.2)

ALC_CMC.2.1d The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2d The developer shall provide the CM documentation.
ALC_CMC.2.3d The developer shall use a CM system.
ALC_CMC.2.1c The TOE shall be labeled with its unique reference.
ALC_CMC.2.2c The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3c The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.2 Parts of the TOE CM coverage (ALC_CMS.2)

ALC_CMS.2.1d The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2c The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.3 Delivery procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2d The developer shall use the delivery procedures.
ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.4 Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1d The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.1.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Tests (ATE)

5.3.4.1 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 in the test documentation and the TSFIs 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.

5.3.4.2 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, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.

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

5.3.4.3 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 execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3e The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.5 Vulnerability assessment (AVA)

5.3.5.1 Vulnerability analysis (AVA_VAN.2)

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

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

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

AVA_VAN.2.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Trusted path/channels

6.1 TOE Security Functions

6.1.1 Security audit

Integration Server, webMethods Broker, and My webMethods Server generate audit records, which are stored in distinct audit trails. Integration Server builds on JDK primitives to implement logging. It maintains a separate log for security events. Integration Server stores audit records in a database in the operational environment. Audit records from multiple instances of Integration Server may be consolidated in a single database. webMethods Broker and My webMethods Server save their security audit logs to text files in their local file system. Each instance of webMethods Broker and My webMethods Server will have its own audit trail, since the audit trails are stored locally. The operational environment (database management system and operating systems) protect the stored audit trails.

The TSF relies on the operation environment for persistent protected storage of the security audit. It relies on the operational environment for text tools (such as a text editor) to review the webMethods Broker and My webMethods Server audit trails. An administrator accesses these audit trails through the operating system hosting the webMethods Broker or My webMethods Server.

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

- FAU_GEN.1: The TSF implements security audit as specified in this requirement. The logging functions were implemented so as to provide the log record content required by the CC. The operations were completed to match the events audited by the Integration Server, webMethods Broker, and My webMethods Server. Each component generates events related to management of its configuration and to access to resources it manages. See section 6.1.3 for descriptions of objects identified in the requirement.
- FAU_SAR.1, FAU_SAR.2, and FAU_SAR.3: Integration Server Administrator provides an interface for viewing the Integration Server security audit logs. My webMethods Server provides a log viewer interface to view My webMethods Server audit logs. TSF role definitions limit these functions to Authorized Administrators, as identified in section 6.1.5. The Integration Server provides capabilities for selecting and ordering the audit records as specified in FAU_SAR.3. It presents the selected audit records in tabular form with columns: Timestamp, Message, Server ID, Client ID, User ID, and Security Event Type.
- FAU_SEL.1: Each type of TOE component groups security audit events into types. An Authorized Administrator may select audited events by event type for each TOE component (for example, Integration Server records all Authentication events or none).
    - Integration Server groups security logs into 14 types: Authentication, Authorization, Certificates, Configuration, JDBC Pools, Keystore, Packages, Passwords, Ports, Proxy Servers, Remote Servers, Services, SSL, and Web Services.
    - webMethods Broker groups logs into 11 categories: Connection, Session, Broker Server Modification, Broker Server Security, Broker Server Start/Stop, Broker Modification, Documentation Type Modification, Client Group Modification, Client Group Security, Gateway Modification, and Territory

Modification. All categories contain security audit events, except for Gateway Modification and Territory Modification.

  - My webMethods Server groups security logs into 4 broad types: Portal object add/create/update/delete, Directory Principal create/update/delete, Authentication Events including Login and Logout, and permission changes.

6.1.2 Cryptographic support

The TSF meets the cryptographic requirements by using cryptographic functions provided in the operational environment. The TSF implements each of the specified protocols using lower-level functions provided by the JDK and FIPS 140-2 validated cryptographic modules. In order to allow use of longer keys (for example, 256-bit AES), the JVM in the operational environment must be configured with an unlimited strength jurisdiction policy file. The FIPS cryptographic modules are:

- Entrust Authority™ Security Toolkit 7.2 for the Java® Platform (FIPS 140-2 certificate #802)
- OpenSSL FIPS Object Module, Software Version 1.2.3 (FIPS 140-2 certificate #1051)

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

- FCS_COP.1a: The TSF supports AES and TDES encryption on the Integration Server, My webMethods Server, and webMethods Broker. The TOE uses cryptographic algorithms: AES and TDES operating in CBC mode. The algorithms, key sizes, and standards in FCS_COP.1a reflect the implementation.
  - Integration Server and My webMethods Server use the JDK to access key stores (for server keys and client certificates) and trust stores (for root certificates). Key stores and trust stores for a server exist in the file system of the operating system hosting the server. Integration Server support key stores in PKCS#12 and JKS (Java Key Store) formats. They support trust stores in JKS format. The TSF relies on the operating system to protect the key stores and trust stores. The servers use the Entrust cryptographic module as well as the JDK packages for cryptographic functions.
  - webMethods Broker accesses key stores and trust stores directly. Broker uses PKCS#12 or PEM files for key stores. It uses concatenated PEM files for a trust store. It uses OpenSSL for cryptographic functions and to manage key stores and trust stores.
- FCS_COP.1c: The TSF includes built-in services that provide digital signature support on the Integration Server. Integration Server manages the message format. It uses the Entrust cryptographic module to digitally sign, encrypt, and decrypt documents as well as to verify digital signatures on documents. The algorithms, key sizes, and standards in FCS_COP.1c reflect the implementation. TSF includes built in services that provide RSA.
- FCS_COP.1d: The TSF includes built-in services that provide secure hash functions on the Integration Server. Integration Server manages documents and applies the appropriate transformations. It calls on providers in the Entrust cryptographic module to construct and verify message digests. The algorithms, key sizes, and standards in FCS_COP.1d reflect the implementation.

6.1.3 User data protection

A variety of users access the TSF:

- Business analysts and developers implement services and business processes on Integration Server;
- Users access services provided by Integration Server;
- My webMethods users perform tasks as part of business processes;
- Broker clients connect to webMethods Broker to publish and subscribe to documents;
- Enterprise resources provide documents in response to service requests;
- Administrators manage the security functions of the TOE.

The user data protection requirement are written to describe the access control policy the TSF enforces on non-administrative data. (See section 6.1.5 for restrictions on security management functions.) Separate requirements are used to break the access control policy into manageable portions. Each portion of the policy addresses a distinct set of users or user data. The TSF can control access by individual user or by groups of users (using groups in Integration Server, client groups in webMethods Broker, and groups and roles in My webMethods Server). A My webMethods Server role is a collection of groups.

The Service SFP specifies the access control policy for user requests to execute services or request files. A user connects to an Integration Server port using the URL of a specific service or file. Integration Server uses attributes of the connection (represented as a Java object) to determine whether access is allowed. All connections provide a presumed source IP address and destination port. If authentication is configured for the service URL, then the connection provides the user name and groups for the user. If authentication is not configured, then Integration Server associates an anonymous user with the connection (user “Default” and groups Everybody and Anonymous).

A service request may invoke a sequence of services, where the first is invoked directly by the user’s request and others are invoked by an Integration Server service preceding them in the sequence. A service invoked by the user’s request is called a top-level service. The other services in the sequence are called internal services.

Services can be scheduled to run in the future using the service scheduler. Services can be run periodically on behalf of any user in the system. Scheduling services is only permitted by Administrative users.

Integration Server can serve files (for example, dynamic server pages and HTML files) that reside in the pub directory for a package (described below) or subdirectory of the pub directory.

Integration Server associates four access control lists with objects:

- List. Allows a user to see that an object exists. The object will be displayed on screens in Designer and Integration Server Administrator. List access also allows a user to view an object's metadata (for example, a service’s inputs, output’s, settings, and ACL permissions).
- Read. Allows a user to view the source of an object through Designer and Integration Server Administrator.
- Write. Allows a user to edit an object. This access also allows a user to delete or lock an object or to assign an ACL to it.
- Execute Allows a user to execute a service. This access also gives the user access to files the server serves, such as DSP and .htm files.

Each ACL identifies groups that are allowed to access an object (Allowed Groups) or groups that are not allowed to access an object (Denied Groups). An ACL may contain both Allow Groups and Deny Groups. If an authorized administrator explicitly sets an ACL on an object, Integration Server checks the designated ACL. If an authorized administrator has *not* explicitly set an ACL on an object, the following happens:

- For objects (other than files), if the parent folder is protected by an ACL, the object inherits the folder's protection. If the folder has no explicit protection, the object inherits the protection of the folder's parent. Integration server assigns ACLs when it creates a package or top-level folder, which provides a basis for inheritance. Top-level folders never inherit List access from the parent package.
- For files, if the parent folder is protected by an ACL, the file inherits the folder's protection. However, if the file resides in a subfolder that is not explicitly protected by an ACL, the server assigns the Default ACL to the file. The Default ACL allows access to all authenticated users and users with the My webMethods Users role.

Each service has the property “Enforce Execute ACL.” The property can be set to “When top-level service only” and “Always.” Of the four types of ACL, only the Execute ACL applies to invoking services or requesting files. An authorized administrator configures the “Enforce Execute ACL” property and Execute ACL for a service. An authorized administrator configures Execute ACL for a file.

Integration server enforces the Services SFP by applying the access control rules specified in FDP_ACF.1a using the attributes described above as it invokes each service.

The Trigger SFP specifies the access control policy for enterprise resources (for example, backend servers) and external systems providing information to Integration Server. Integration Server implements Broker/local triggers and JMS triggers. A Broker/local trigger acts on behalf of an enterprise resource (that is, an application or server managed by the same organization responsible for the TOE). A JMS trigger acts on behalf of an external system or enterprise resource.

For Broker/local triggers, an authorized administrator associates an enterprise resource with Integration Server using an adapter, which is a BPMS component outside the TOE. An adapter either listens for information from the enterprise resource or polls the resource for information. The adapter translates information it receives from a enterprise resource into an adapter notification document, which it publishes to Integration Server. Integration Server sends an adapter notification document to each trigger that subscribes to the adapter notification's document type. The trigger identifies the service for processing the document and invokes that service. Integration Server applies the Trigger SFP before executing the requested service.

For JMS triggers, an authorized administrator creates and configures a JMS trigger. The configuration identifies the JMS provider for the trigger and the message queues or topics to which the trigger subscribes. The configuration defines the service or web service the trigger invokes to process each message. Integration Server applies the Trigger SFP before executing the requested service.

Integration Server associates a user with triggers. All Broker/local triggers are associated with the same user. Each JMS trigger can be associated with a user independently. An authorized administrator can replace the default association with another pre-defined user (Default, Developer, or Replicator) or with a user the administrator defines. The pre-defined user Administrator is the default for Broker/local triggers (with groups Everybody, Administrators, and Replicators) as well as JMS triggers.

The Enforce Execute ACL property and Execute ACL attributes are the same as for the Services SFP.

Integration server enforces the Trigger SFP by applying the access control rules specified in FDP_ACF.1b using the attributes described above as it invokes each service.

The Integration Server Resources SFP specifies the access control policy for user requests to access objects that define services and business processes. A user connects to an Integration Server using Designer[4]. Integration Server authenticates the identity of the user. The connection provides the user name and groups for the user. If the user does not belong to the Developer group, then Integration Server denies the connection.

Objects that define services and business processes are:

- Package: A package is a set of services and related files such as specifications and document types. A package is managed as a unit (for example, the contents of a package are enabled or disabled as a whole). Every service on an Integration Server must belong to a package.
- Folder: Integration Server organizes services in a hierarchical structure using folders. A folder may contain services or other folders, which are called subfolders. A fully qualified service name identifies the location of a service within the hierarchy. A fully qualified service name consists of a folder identifier and service name (like *folder.subfolder1.subfolder2:service*).
- Service: A service is an Integration Server-resident unit of functionality that clients can invoke.
- Specification: A specification is a description of a service, such as a Web Services Description Language (WSDL) file.
- Schema: A schema is a description of an XML document expressed as an XML Schema Document (XSD). In addition, flat files present complex hierarchical data in a record-based storage format. A flat file schema is the metadata that describes the structure of flat file.
- Document type: A document type is a schema-like definition that describes the structure of a particular kind of document. An adapter notification document type is a document type associated with an adapter notification. Adapter notifications determine whether an event has occurred on the adapter's resource and then sends the notification data to Integration Server in the form of a published document.

[4] A user could connect to Integration Server using webMethods Developer. However, Software AG has deprecated Developer, which should not be used in the evaluated configuration.

- Trigger: A trigger establishes a subscription to a publishable document type and specifies how to process instances of that publishable document type (that is, specifies a trigger service).

For the Integration Server Resources SFP, List ACL, Read ACL, and Write ACL access control lists are as described above. Users in the Developer group use Designer to configure ACLs for packages, specifications, document types, schema, and triggers. Users in the Administrator group use Integration Server Administrator to configure ACLs for folders and services. However, Integration Server enforces the rules independent of the ultimate source of requests (for example, Designer).

Integration Server enforces the Integration Server Resources SFP by applying the access control rules specified in FDP_ACF.1c using the attributes described above as it invokes each service. In particular, a user cannot assign an ACL to object unless the user is a member of that ACL. When changing an ACL for an object, a user must be a member of the existing ACL and the ACL being assigned.

The My webMethods Server SFP specifies the access control policy for user requests to access objects through the My webMethods interface. The policy covers interfaces objects (such as pages and portlets) as well as objects that implement the interface (such as tasks and webMethods applications). A user connects to My webMethods, which is a browser-based web interface that My webMethods Server defines and presented through the Jetty web server. If authentication is configured for a particular URL, then the connection provides the user name, groups, and roles for the user. If authentication is not configured, the My webMethods Server associates an anonymous user with the connection (user “Guest” with no group and no role).

The objects that define My webMethods are:

- Folder: An object that is used to hold one or more other objects
- Page: A page is a web page defined in My webMethods server and presented through the Jetty web server.
- Portlet: A portlet is an interface component My webMethods server displays within a web page.
- Link: A object representing a URL (e.g. http://www.cnn.com)
- Document: An object representing a portal document
- File: An object representing an arbitrary file
- Server verb: A server verb is an operation (such as publishing, deleting, updating, subscribing, and setting permissions), which is available through the My webMethods Server API. My webMethods server typically have two levels of security checks for server verbs, performed in this order: 1) does the user have access to the server verb itself and 2) does the user have the rights to the resource upon which the server verb is trying to act.
- webMethods application: An set of objects (portlets and pages) that is used to manage a particular product (e.g. Broker)
- Task: A task is a business process activity managed by Task Engine and performed by a person.
- Workspace: A workspace is a page that a user creates and uses as work areas for some specific purpose. Users build the content of the workspace by dragging portlets on to the workspace.
- Content Object: Any resource on My webMethods Server, including files, folders, and pages.
- Portlet type: This object represents the types of portlets that can be created (legacy, CAF, etc.).
- Security realm: A security realm is a collection of server resources (such as pages and portlets) that share the same ACL.

My webMethods Server associates with each object a user name, which is called the object’s *owner*. My webMethods Server also may associate with an object a *lock*, which prevents more than one user from editing the object simultaneously. The lock identifies the user name that has exclusive access to the object. My webMethods Server itself may hold a lock for an object, which makes the object read only for all users. A *Capability* is an operation a type of object supports. For example, pages have the Capability “Add Portlet to Page” while folders have Capabilities “Create Sub Folder” and “Can Read Items in the Folder.” As the example illustrates, My webMethods Server associates different types of Capabilities with distinct types of objects.

A My webMethods Server ACL is a list of access control entries. An *access control entry* (ACE) defines rights to a set of Capabilities. An ACE consists of a Principal and a Rights Set. A *Principal* is a user, group, or role. A *Right Set* is a set of Capability-Settings pairs. Capabilities are described above. A Setting is one of the following:

- DENY: Denies the access to perform the capability.
- GRANT: Explicitly grants access to perform the capability.
- DELEGATE: Explicitly grants access and gives the right to assign the capability to another Principal.
- NONE: Does not provide an explicit Setting. Authorization for this server resource will be determined from another source (that is, another Capability-Setting pair or another ACE).

My webMethods Server uses *static propagation*, which assigns a newly created child object the access rights of its parent object (that is, the ACL of the child is a copy of the ACL of the parent). The assignment is static in that subsequent changes to the access rights of parent do not affect the access rights of the child (that is, changing the parent's ACL does not automatically change the child's ACL).

My webMethods Server enforces the My webMethods Server Resources SFP by applying the access control rules specified in FDP_ACF.1d using the attributes described above as it invokes each service. FDP_ACF.1d.2 encapsulates the precedence rules for a Right Set. The precedence rules are 1) DENY takes precedence over allow (that is, GRANT and DELEGATE) and 2) users always take precedence over groups and roles.

The TSF stores its resources (TOE executables and TOE data) in the operational environment. Hence, the TSF relies on the operational environment to prevent unmediated access to its resources. Integration Server stores services and related objects in packages in its local file system. My webMethods Server stores its objects in a RDBMS. webMethods Broker manages objects in memory and in the local file system.

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

- FDP_ACC.1a and FDP_ACF.1a: The Service SFP specifies user access to services and files. FDP_ACC.1a limits the scope of the policy to users executing services or accessing files at an Integration Server. FDP_ACF.1a defines the rules implemented in the Integration Server to enforce the policy. The rules reflect the scope of the Service SFP, since they are based on user attributes and Execute ACL.
- FDP_ACC.1b and FDP_ACF.1b: The Trigger SFP specifies controls the TSF places on resource responses to service requests. FDP_ACC.1b limits the scope of the policy to triggers, which are invoked when an adapter presents a document to the Integration Server from an enterprise resource or when a JMS trigger receives a message from a JMS provider. FDP_ACF.1b defines the rules implemented in the Integration Server to enforce the policy. Fewer attributes are needed, since enterprise resources are more constrained than users in general.
- FDP_ACC.1c and FDP_ACF.1c: The Integration Server Resources SFP specifies access controls related to implementing services and business processes on Integration Server. FDP_ACC.1c limits the scope of the policy by specifying the applicable Integration Server resources and operations. This SFP complements the Service SFP in that the operations are list, read, and write resources but not execute. FDP_ACF.1c defines the rules implemented in the Integration Server. The rules are enforced by Integration Server independent of the ultimate source of requests (for example, Designer).
- FDP_ACC.1d: The My webMethods Server SFP specifies access controls on My webMethods users. FDP_ACC.1d and FDP_ACF.1d reflect the implementation of capabilities My webMethods Server presents to users.

6.1.4 Identification and authentication

The access control policy, management restrictions, and auditing all depend on assurance in the identity a user presents. The TSF implements functions to obtain an identity from a user and authenticate it. The TSF provides several authentication mechanisms to accommodate the different ways users access the TSF.

Integration Server can authenticate users with a password-based mechanism. It can maintain the password information in a database it maintains itself or retrieve password information from a database shared with My webMethods Server. The shared user database feature is called Central User Management. The TOE uses RDBMS in the operational environment for both databases. Integration Server implements a certificate-based authentication mechanism. Integration Server supports LDAP authentication, with one or more LDAP directories. When configured for LDAP authentication, Integration Server sends a user's name and password to each LDAP directory in the configured order until authentication succeeds or all attempts fail. An Authorized Administrator can configure authentication mechanisms by port, with different ports using distinct authentication mechanisms.

My webMethods Server provides a password-based authentication mechanism. It stores user names and passwords in a database, which it may share with Integration Server. The shared user database feature is called Central User Management. Central User Management supports single sign on through Security Assertion Markup Language (SAML). When a user attempts to access Integration Server through My webMethods, My webMethods sends a SAML artifact to Integration Server. Integration Server uses the SAML artifact and the shared database to authenticate the user. The TSF uses a salted hash to store passwords. My webMethods Server also supports external LDAP directory authentication. An Authorized Administrator configures authentication mechanization by connection type.

webMethods Broker implements a X.509 certificate-based authentication mechanism (called SSL authentication) and supports external LDAP directory authentication (called Basic Authentication). A webMethods Broker presents a port for each type of authentication mechanism and for unauthenticated access. The ACLs an Authorized Administrator assigns to a Client Group determines which authentication mechanism webMethods Broker applies for users connecting with that group.

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

- FIA_ATD.1a and FIA_ATD.1b: Integration Server and My webMethods Server each maintain user attributes. Integration Server and My webMethods Server may share a database containing common user attributes. All three components may rely on external LDAP directory to store user attributes. Note that webMethods Broker does not maintain attributes for clients. Rather, it relies on external LDAP servers and trust stores.

- FIA_SOS.1: Integration Server implements password policy rules for the passwords it maintains. The password policy rules determine the composition of passwords the TSF accepts. There are no programmatic restrictions on the minimum values for password length or the number of characters from each character class; each may be set to zero. However, Integration Server sets default minimums appropriate for the anticipated operational environment. These default minimums are: eight characters for password length, two upper case characters, two lower case characters, one special character, and one digit. For My webMethods Server, the local user password stores are meant for administrator and system users only and not for general application use. Hence, password composition is enforced by administrative policy. If desired, a custom password policy module that enforces arbitrary complexity rules can be developed and deployed webMethods Broker does not maintain passwords.

- FIA_UID.1 and FIA_UAU.1: Integration Server can be configured to allow anonymous access to services. Integration associates users with triggers and scheduled services based on its configuration and not information provided by external entities. Hence, Integration server allows these services without identification or authentication. It requires users to authenticate their identities before allowing any other actions. Similarly, webMethods Broker can be configured to allow anonymous Broker client sessions.

- FIA_UAU.5: This requirement reflects the authentication mechanisms implemented by the TSF as described above.

6.1.5 Security management

The TSF can control management function access by individual user or by groups of users (using groups in Integration Server, client groups in webMethods Broker, and groups and roles in My webMethods Server). The Authorized Administrator role is made up of groups that may manage the TSF. Integration Server includes the following pre-defined groups:

- Administrators	This group identifies users that have administrator privileges. A user must have administrator privileges to configure and manage the server.
- Anonymous	This group identifies users that have not authenticated.

- Developers This group identifies users that have developer privileges. A user must have developer privileges to connect to the server from the Designer or Developer.
- Everybody All users are a member of this group. Every new user is automatically added to the Everybody group.
- Replicators This group identifies users that have replicator privileges. The Replicators group gives its members the authority to perform package replication. (By default, the server uses members of the Replicators group for package replication.)

The pre-defined Integration Server groups Administrators, Developers, and Replicators correspond to the Authorized Administrator role defined in this ST.

Broker includes the following pre-defined client groups:

- Admin Members of the admin client group have full permissions and administrative privileges for a given Broker.
- accessLabelAdapter The accessLabelAdapter client group corresponds to an advanced security feature called the Access Label Adapter (ALA). The Broker Access Labels feature is excluded from the evaluated configuration.
- Adapters This system-defined client group is for the 4.x Adapter Developer Kit (ADK), used in previous versions of Broker. The adapter developer kit is deprecated for this release of Broker.
- eventLog This client group can subscribe to all document types defined within the Broker. The eventLog client group is used by a Broker's dead letter queue facility, which receives messages for which there are no subscribers. The eventLog client group is deprecated for this release of Broker.

The pre-defined webMethods Broker admin client group corresponds to the Authorized Administrator role defined in this ST.

My webMethods Server includes roles:

- Admin Role Provides access to all My webMethods Server resources. By default, the SysAdmin and Designer users are members of this role.
- My webMethods Administrators Provides access to user management and other functions needed by the My webMethods Administrator, who is a default member of this role.
- My webMethods Users Provides access to the My webMethods user interface for all users of My webMethods applications. By default, the My webMethods Server Administrator is a member of this role, but an authorized administrator must add all other users to it.

The pre-defined My webMethods server roles Admin Role and My webMethods Administrators correspond to the Authorized Administrator role defined in this ST.

Integration Server and My webMethods Server both provide pre-defined users corresponding to the Authorized Administrator role.

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

- FMT_MOF.1: For the TSF, ‘enable’ means to start and ‘disable’ means to shut down. Authorized Administrators can enable and disable Integration Server, My webMethods Server, and webMethods individually. Administrators also can determine the running state of TOE components. Security audit logging can be turned off and on, but only by Authorized Administrators.
- FMT_MSA.1a, FMT_MSA.1b, FMT_MSA.1c, and FMT_MSA.1d: Integration Server Administrator and My webMethods Server provide the Authorized Administrators with the capabilities to view and change TSF configuration. This includes settings that determine the access control policy. Administrators can grant

other groups and users the capability to manage access control attributes. The attributes chosen in the FMT_MSA.1 requirements match settings implemented to define access control. See section 6.1.3.

- FMT_MSA.3: Integration Server supports explicit and implicit protection through ACL as described in 6.1.3 above. If an element is assigned an ACL, the TSF enforces the ACL. If an element is not assigned an ACL, the element inherits an ACL, which the TSF enforces. The Default ACL allows access to authenticated users and users with the My webMethods User role. An Authorized Administrator can specify alternative to the Default ACL explicitly or through inheritance.
- FMT_MTD.1a, FMT_MTD.1b, and FMT_MTD.1c: The TSF provides functions to manage not only access control policy but also other TSF data used for security policy enforcement. These requirements list the remaining TSF data. Integration Server users may change their own passwords, but only if they belong to the pre-defined Developer group. This means that only Authorized Administrators may change their own Integration Server passwords. Other Integration Server users are assigned passwords as needed by an Authorized Administrator. My webMethods Server users can change their own passwords. webMethods Broker does not maintain passwords.
- FMT_SMF.1: This requirement lists the management functions implemented by the TSF. The other management requirements specify restrictions on those functions. Integration Server provides functions to manage its audit functions; start up/shutdown Integration Server; manage attributes for the Service SFP, Trigger SFP, and Integration Server Resources SFP; manage Integration Server users and groups; manage port configuration; and manage password policy. The Integration Server Administrator application provides the interface for Integration Server security management functions. My webMethods Server provides functions to manage its audit functions, start up/shutdown My webMethods Server, manage attributes for the My webMethods Server SFP, manage My webMethods Server users and groups, and manage password policy. In addition, My webMethods Server provides functions to manage webMethods Broker audit functions, manage start up/shutdown webMethods broker, and manage attributes for the Broker Documents SFP. My webMethods provides the interface for My webMethods Server and webMethods Broker security management functions.
- FMT_SMR.1: The Authorized Administrator role is used in the Security Target to encompass the groups and roles with privilege to manage the TSF. Each TOE component provides pre-defined groups or roles. The Authorized Administrator role as used in this ST consists of Integration Server groups Administrators, Developers and Replicators; webMethods Broker group admin Client Group; and My webMethods Server roles Admin Role and My webMethods Administrator. None of the other roles can administer TOE servers.

6.1.6 Trusted path/channels

The TSF uses TLS as the primary mechanism for protecting user data (such as documents and business process configuration) and TSF data (such as authentication data) in transit. In particular, TLS is used in HTTPS and FTPS for secure communication with users. Integration Server and My webMethods Server implement TLS using lower-level services provided by the JDK and the Entrust Authority Security Toolkit. webMethods Broker uses OpenSSL FIPS Object Module.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FDP_ITT.1 and FPT_ITT.1: The TSF can be configured to use TLS to protect communication among Integration Server, My webMethods Server, and webMethods Broker. TLS protects the confidentiality and integrity of both user data and TSF data in transit between TOE components. TLS need not be configured when the operational environment ensures secure communication (for example, when TOE components are on a dedicated, protected network).
- FTP_ITC.1: The TSF supports TLS communication with webMethods Broker Clients and external LDAP directories. A Broker Client initiates secure communication with the TSF by connecting to a port designated for TLS. Integration Server, My webMethods Server, and webMethods Broker initiate secure communication with servers in the operational environment as configured. RDBMS support for TLS varies. The TSF relies on RDBMS client drivers provided with each RDBMS and can use TLS communication when a RDBMS client and server support it.

- FTP_TRP.1: Integration Server supports HTTPS and FTPS for secure communication with users. Users initiate secure communication with the TSF by connecting to designated ports for services. Integration Server provides cryptographic credentials for the client to validate as part of TLS session establishment. Integration server authenticates the client either using a password-mechanism or client certificates. An authorized administrator can configure Integration Server to require client certificates. My webMethods Server support TLS communication for My webMethods.

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

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;
- Requirement Dependencies;
- TOE Summary Specification.

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.

	P.AUDIT	T.PROCESS	T.SERVICE	T.TAMPER	T.TRANSIT	A.ADMIN	A.CRYPTO	A.HOST	A.PHYSICAL	A.REVIEW	A.USERS
O.ACCESS		X	X	X							
O.ADMIN				X							
O.AUDIT	X										
O.DOC_SEC					X						
O.ID_AUTH		X	X								
O.REVIEW	X										
O.TLS					X						
OE.COMM					X						
OE.CRYPTO					X		X				
OE.HOST				X				X			
OE.INSTALL				X		X					
OE.OPERATE				X		X					
OE.PERSONNEL						X					
OE.PHYSICAL				X					X		
OE.REVIEW	X									X	
OE.STORE	X										
OE.TIME	X										
OE.USERS		X	X								X

Table 3 Environment to Objective Correspondence

8.1.1.1 P.AUDIT

A product that provides security functions must be capable of producing a audit trail of security-relevant events.

This Organizational Policy is satisfied by ensuring that:

- O.AUDIT: Supports the policy through the generation of security-relevant audit events.
- O.REVIEW: Gives authorized users capabilities to analyze the audit trail produced by Integration Server and webMethods Broker
- OE.REVIEW: Provides the capability to analyze the audit trail produced by My webMethods Server.
- OE.STORE: Ensures the audit trail would be available for analysis.
- OE.TIME: Provides reliable time stamps for sequencing events in the audit trail.

8.1.1.2 T.PROCESS

A user may view, read, or write business process information without permission using BPMS interfaces.

This Threat is satisfied by ensuring that:

- O.ACCESS: Business process information is encapsulated in services, Integration Server resources, webMethods Broker documents, and My webMethods Server resources. This objective restricts access to those resources based on user's authorization.
- O.ID_AUTH: Provides the means to determine a user's authorization to services, Integration Server resources, webMethods Broker documents, and My webMethods Server resources based an assured identity.
- OE.USERS: Mitigates one user impersonating another.

8.1.1.3 T.SERVICE

A user may invoke a service, web page, or business process task without permission.

This Threat is satisfied by ensuring that:

- O.ACCESS: Users invoke services through Integration Server. They execute business process tasks through My webMethods Server. Users may view web pages through either component. This objective restricts access based on user's authorization.
- O.ID_AUTH: Provides the means to determine a user's authorization to invoke restricted services, view restricted web pages, and perform restricted business process tasks based an assured identity.
- OE.USERS: Mitigates one user impersonating another.

8.1.1.4 T.TAMPER

A user may modify the BPMS or its configuration in order to access services or business process information without permission.

An attacker might attempt to modify the BPMS or its configuration via the operational environment, a TOE management interface, or a TOE user interface. This Threat is satisfied by ensuring that:

- OE.HOST: Reduces the tools available to realize the threat.
- OE.INSTALL: Ensures installers apply safeguards described in the installation guidance.
- OE.OPERATE: Ensures operators apply safeguards described in operating guidance.
- OE.PHYSICAL: Mitigates modification of the BPMS and its configuration through the operational environment.
- O.ADMIN: Restricts access to the BPMS configuration at TOE security management interfaces.
- O.ACCESS: Restricts potential access to the BPMS configuration TOE user interfaces.

8.1.1.5 T.TRANSIT

A user may capture network traffic in order to observe or alter service invocation or business process information of another user.

This Threat is satisfied by ensuring that:

- O.DOC_SEC: Provides capabilities to protect the integrity and confidentiality of individual documents exchanged during service invocation. These services may be used in addition to or in place of securing the communication between a user and the TOE.
- O.TLS: Provides capability for TLS-protected communication between users and the TOE, which may be used when the operational environment alone does not provide suitable protection.
- OE.COMM: Provides protection appropriate for operational environment.
- OE.CRYPTO: Complements TOE objectives O.DOC_SEC and O.TLS by providing key management and cryptographic operations.

8.1.1.6 A.ADMIN

Those responsible for the TOE are trusted to correctly configure and operate the TOE according to the instructions provided by the TOE documentation.

This Assumption is satisfied by ensuring that:

- OE.INSTALL: Upholds the assumption during delivery and installation, before the TOE is operational.
- OE.OPERATE: Upholds the assumption during normal operation of the TOE.
- OE.PERSONNEL: A careful selection process provides confidence that a TOE administrator is not malicious. Training adds confidence in the competence of a TOE administrator.

8.1.1.7 A.CRYPTO

The operational environment must provide key management and underlying cryptographic functions to support TOE cryptographic operations.

This Assumption is satisfied by ensuring that:

- OE.CRYPTO: Directly upholds the assumption.

8.1.1.8 A.HOST

There is no untrusted software on the servers hosting the TOE.

This Assumption is satisfied by ensuring that:

- OE.HOST: Directly upholds the assumption.

8.1.1.9 A.PHYSICAL

The TOE is protected from physical attack.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: Directly upholds the assumption.

8.1.1.10 A.REVIEW

The operational environment must provide text processing tools to supplement the TOE capability to review the security audit trail.

This Assumption is satisfied by ensuring that:

- OE.REVIEW: Directly upholds the assumption.

8.1.1.11 A.USERS

Users will protect authentication data in their possession.

This Assumption is satisfied by ensuring that:

- OE.USERS: Directly upholds the assumption.

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

8.2.1 Security Functional Requirements Rationale

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

	O.ACCESS	**O.ADMIN**	**O.AUDIT**	**O.DOC_SEC**	**O.ID_AUTH**	**O.REVIEW**	**O.TLS**
FAU_GEN.1			X				
FAU_SAR.1						X	
FAU_SAR.2						X	
FAU_SAR.3						X	
FAU_SEL.1			X				
FCS_COP.1a							X
FCS_COP.1c				X			
FCS_COP.1d				X			
FDP_ACC.1a	X						
FDP_ACC.1b	X						
FDP_ACC.1c	X						
FDP_ACC.1d	X						
FDP_ACF.1a	X						
FDP_ACF.1b	X						
FDP_ACF.1c	X						
FDP_ACF.1d	X						
FDP_ITT.1							X
FIA_ATD.1a					X		
FIA_ATD.1b					X		
FIA_SOS.1					X		
FIA_UAU.1					X		
FIA_UAU.5					X		
FIA_UID.1					X		
FMT_MOF.1		X					
FMT_MSA.1a		X					
FMT_MSA.1b		X					
FMT_MSA.1c		X					
FMT_MSA.1d		X					
FMT_MSA.3		X					
FMT_MTD.1a		X					
FMT_MTD.1b		X					
FMT_MTD.1c		X					
FMT_SMF.1		X					
FMT_SMR.1		X					
FPT_ITT.1							X
FTP_ITC.1							X
FTP_TRP.1							X

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE must allow a user to access only those services, Integration Server resources, webMethods Broker documents, and My webMethods Server resources for which the user is authorized.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1a and FDP_ACF.1a: Integration Server, My webMethods Server and webMethods Broker each play a role in enforcing access controls. The FDP_ACC.1* and FDP_ACF.1* specify aspects of the access control policy that each component provides. FDP_ACC.1a and FDP_ACF.1a specify restrictions Integration Server enforces on users executing services.
- FDP_ACC.1b and FDP_ACF.1b: Specify restrictions Integration Server enforces on resources that provide responses to service requests.
- FDP_ACC.1c and FDP_ACF.1c: Specify restrictions Integration Server enforces on users reading or writing services, web pages, and business process tasks.
- FDP_ACC.1d and FDP_ACF.1d: Specify restrictions My webMethods Server enforces on users accessing business process tasks and business process information.

8.2.1.2 O.ADMIN

The TOE must provide functions to manage the security functions of the TOE and restrict the use of these functions to authorized administrators.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: Restricts changing the operating state of the TOE and auditing to Authorized Administrators.
- FMT_MSA.1a, FMT_MSA.1b, FMT_MSA.1c, and FMT_MSA.1d: Each of these requirements corresponds to an access control SFP. The management requirement restricts the capability to manage security attributes for the corresponding SFP to Authorized Administrators.
- FMT_MSA.3: Complements each of the FMT_MSA.1 requirements by addressing default security attributes used when user data is created.
- FMT_MTD.1a, FMT_MTD.1b, and FMT_MTD.1c: Provide restrictions on functions for managing security functions other than access control. Only Authorized Administrators can mange users (accounts with attributes, groups, and roles), audit, password composition rules, and port configuration (such as allowed services and mandating TLS). My webMethods Server users may change their own passwords.
- FMT_SMF.1: Lists the security management functions the TOE provides.
- FMT_SMR.1: Defines a distinct role for authorized administrators.

8.2.1.3 O.AUDIT

The TOE must provide the capability to produce audit trail of security-relevant events.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: Specifies the TOE’s capability to generate audit records.
- FAU_SEL.1: Allows Authorized administrators to tailor the audit record generation.

8.2.1.4 O.DOC_SEC

The TOE must provide capabilities to digitally sign documents, verify digital signatures on documents, encrypt documents, and decrypt documents for documents exchanged during service invocation.

This TOE Security Objective is satisfied by ensuring that:

- FCS_COP.1c, and FCS_COP.1d: Specify cryptographic operations to meet O.DOC_SEC directly. The requirements specify support for multiple protocols.

8.2.1.5 O.ID_AUTH

The TOE must be able authenticate the identity of a user requesting access to security management functions or restricted services, Integration Server resources, webMethods Broker documents, or My webMethods Server resources before allowing access.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1a and, FIA_ATD.1b: Define attributes the TSF can use to identify (user name and X.509 certificate) and authenticate (password and X.509 certificate) a user. In addition, the requirement defines attributes to support the access control policy.
- FIA_SOS.1: Provides assurance in claimed identities by enforcing a password composition policy.
- FIA_UID.1 and FIA_UAU.1: Specify the capability to identify and authenticate a user before allowing access to services and information protected by the TSF. The requirements accommodate limited anonymous access.
- FIA_UAU.5: Lists mechanisms the TSF can use to authenticate an identity and defines when the TSF uses each mechanisms.

8.2.1.6 O.REVIEW

The TOE must provide authorized users with the capability to review the security audit trail.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1, FAU_SAR.2, and FAU_SAR.3: Specify the functions the TSF provides for audit review and restricts those functions to Authorized Administrators.

8.2.1.7 O.TLS

The TOE must provide the capability to secure communications via TLS between a user and the TOE and among TOE components.

This TOE Security Objective is satisfied by ensuring that:

- FCS_COP.1a: Specifies support for TLS as a mechanism for secure communications.
- FDP_ITT.1 and FPT_ITT.1: Provide for secure communication among TOE components for both user data and TSF data.
- FTP_ITC.1: Provides for secure communication between the TSF and external IT entities (Broker clients, LDAP directory servers, etc.).
- FTP_TRP.1: Provides for secure communication between the TSF and human users.

8.3 Security Assurance Requirements Rationale

The TOE is intended for an environment requiring a low to moderate level of assurance in the security functionality of conventional commodity TOEs, as presented in the statement of security environment (Section 3). The target assurance level of EAL 2 augmented with ALC_FLR.1 is appropriate for such an environment.

8.4 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST. As indicated in the table, dependencies are satisfied by the TOE and the operational environment.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	Satisfied by the operational environment through OE.TIME
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1b

FCS_COP.1a	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	Satisfied by operational environment through OE.CRYPTO
FCS_COP.1c	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	Satisfied by operational environment through OE.CRYPTO
FCS_COP.1d	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	Satisfied by operational environment through OE.CRYPTO
FDP_ACC.1a	FDP_ACF.1	FDP_ACF.1a
FDP_ACC.1b	FDP_ACF.1	FDP_ACF.1b
FDP_ACC.1c	FDP_ACF.1	FDP_ACF.1c
FDP_ACC.1d	FDP_ACF.1	FDP_ACF.1d
FDP_ACF.1a	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1a and FMT_MSA.3
FDP_ACF.1b	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1b and FMT_MSA.3
FDP_ACF.1c	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1c and FMT_MSA.3
FDP_ACF.1d	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1d and FMT_MSA.3
FDP_ITT.1	FDP_ACF.1 or FDP_IFC.1	FDP_ACC.1a, FDP_ACC.1b, FDP_ACC.1c, and FDP_ACC.1d
FIA_ATD.1a	none	none
FIA_ATD.1b	none	none
FIA_SOS.1	none	none
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.5	none	none
FIA_UID.1	none	none
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1a	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.1a
FMT_MSA.1b	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.1b
FMT_MSA.1c	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.1c
FMT_MSA.1d	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.1d
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1a, FMT_MSA.1b, FMT_MSA.1c, FMT_MSA.1d, and FMT_SMR.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FTP_TRP.1	none	none
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.2 and ADV_TDS.1
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1
ADV_TDS.1	ADV_FSP.2	ADV_FSP.2
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2
AGD_PRE.1	none	none
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2
ALC_CMS.2	none	none
ALC_DEL.1	none	none
ALC_FLR.1	none	none
ATE_COV.1	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.1
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1
AVA_VAN.2	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.1	ADV_ARC.1 and ADV_FSP.2 and

	and AGD_OPE.1 and AGD_PRE.1	ADV_TDS.1 and AGD_OPE.1 and AGD_PRE.1

Table 5 Requirement Dependencies

8.5 TOE Summary Specification Rationale

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

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. Table 6 Security Functions vs. Requirements Mapping demonstrates the relationship between security requirements and security functions.

	Security audit	**Cryptographic Support**	**User data protection**	**Identification and authentication**	**Security management**	**Trusted path/channels**
FAU_GEN.1	X					
FAU_SAR.1	X					
FAU_SAR.2	X					
FAU_SAR.3	X					
FAU_SEL.1	X					
FCS_COP.1a		X				
FCS_COP.1c		X				
FCS_COP.1d		X				
FDP_ACC.1a			X			
FDP_ACC.1b			X			
FDP_ACC.1c			X			
FDP_ACC.1d			X			
FDP_ACF.1a			X			
FDP_ACF.1b			X			
FDP_ACF.1c			X			
FDP_ACF.1d			X			
FDP_ITT.1						X
FIA_ATD.1a				X		
FIA_ATD.1b				X		
FIA_SOS.1				X		
FIA_UAU.1				X		
FIA_UAU.5				X		
FIA_UID.1				X		
FMT_MOF.1					X	
FMT_MSA.1a					X	
FMT_MSA.1b					X	
FMT_MSA.1c					X	
FMT_MSA.1d					X	

FMT_MSA.3					X	
FMT_MTD.1a					X	
FMT_MTD.1b					X	
FMT_MTD.1c					X	
FMT_SMF.1					X	
FMT_SMR.1					X	
FPT_ITT.1						X
FTP_ITC.1						X
FTP_TRP.1						X

Table 6 Security Functions vs. Requirements Mapping