Metastorm BPM 9.1 Security Target

Version 0.13

January 9, 2012

Prepared for:
Metastorm, Inc.
500 E. Pratt Street
Suite 1250
Baltimore, Maryland 21202

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
SAIC Franklin Center
6841 Benjamin Franklin Drive
Columbia, MD 21046

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Metastorm BPM 9.1 provided by Metastorm, Inc. The TOE provides the ability to view and manage information, activities, and instructions that can be used to automate a business process, for example a manager approving a staff member's form for a travel request.

The Security Target contains the following additional sections:

- TOE Description (Section 2): This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.
- Security Environment (Section 3): This section details the expectations of the environment. Describing the threat, assumptions, and the organizational security polices the TOE and its environment must adhere to.
- Security Objectives (Section 4): This section details the security objectives of the TOE and its environment.
- IT Security Requirements (Section 5): The section presents the security functional requirements (SFR) for TOE and details the requirements for EAL 4.
- TOE Summary Specification (Section 6): The section describes the security functions represented in the TOE that satisfies the security requirements.
- Protection Profile Claims (Section 7): This section identifies the Protection Profile Claim made in the ST.
- Rationale (Section 8): This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness and suitability.

1.1 *Security Target, TOE and CC Identification*

ST Title – Metastorm BPM 9.1 Security Target

ST Version – Version 0.13

ST Date – January 9, 2012

TOE Identification – Metastorm BPM 9.1.1.3

TOE Guidance Documentation –

- Metastorm BPM Release 9.1 Administration Guide,
- Metastorm BPM Release 9.1 Designer User Manual,
- Metastorm BPM Release 9.1 Web Authors Guide,
- Metastorm BPM Release 9.1 Deployment Guide,
- Metastorm BPM Release 9.1 Release Notes,
- Metastorm BPM Release 9.1 Concepts,
- Metastorm BPM Release 9.1 Installation Prerequisites,
- Metastorm BPM Release 9.1 Installation Guide,

TOE Developer – Metastorm Incorporated

Evaluation Sponsor – Metastorm Incorporated

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 2, September 2007.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 2, September 2007.
    - Part 3 Conformant
    - Package Claim/Assurance Level: EAL 4 augmented with ALC_FLR.2

1.3 Conventions, Terminology, Acronyms

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

1.3.1 Conventions

The following conventions have been applied in this document:

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

1.3.2 Terminology

Term	Definition
Actions	Metastorm BPM defines an *action* as the step or activity necessary to modify data and move a folder from one stage to the next. Actions may include activities such as: • Filling out a form • Logging a telephone call • Reviewing an attached file • Approving or denying a request. It is also possible to have actions that do not require human intervention, such as:

Term	Definition
	• Determining the routing for a folder based on information available in a folder or in a database • Raising or responding to a flag • Moving a folder after a timed event • Starting an external application Properties and formulas can be set in the MBPM Designer to accommodate a wide variety of possible actions.
Administration Forms	Administration forms can be used by the user to carry out administrative processes. Administration forms: • Do not start a process • Are not available for use in processes • Cannot access custom variables • Cannot access system folder variables • Are not associated with any folder • Cannot be renamed. Administration forms are automatically associated with a creation action that leads to an Archive stage.
Folder	A new folder, with a unique, system-generated ID, is created each time a new instance of the process is initiated. In this way, the database can track the information particular to each instance of a business process. A folder contains one or more pages (forms) of information relating to that instance of the process. This information may come from a variety of sources, such as: • Input by a user onto a form; • Data extracted from a database (internal or external); and • A file generated by another application.
Forms	Within Metastorm BPM, a user may create *forms*. These forms are used to gather and display information necessary to a business process.
Process	When a user designs a Metastorm project, it is represented through one or more *processes* (diagrams or process models), each illustrating the various steps required to complete a business process (the lifecycle of a folder). Each instance of the business process is called a *folder*, and the steps are called *actions*.
Project	Metastorm BPM views the information, activities, and instructions required to automate a business process as a *project*. In Metastorm BPM, the main component of a project is one or more processes. In addition to the processes, a project may contain forms, roles, flags, external tables, and actions. All of these components are stored in a single solution file.
Roles	Participants (users) in a process have roles assigned to them based on either their individual or group responsibilities. Assignments within a project are made based on these role designations.
Flags	Flags are used to start or continue parts of an automated business process. There are two important aspects to the concept of a flag in Metastorm BPM: the Flag itself and the Flagged Action invoked by the flag.
External Tables	External tables can be used to store configuration data, reference table data, or any other data related to the project

2. TOE Description

The Target of Evaluation (TOE) is Metastorm BPM 9.1 (MBPM).

The TOE is an IT enabled Business Process Management software product supported on Windows Server 2003, Windows Server 2008 and Windows Server 2008 R2. The TOE manages and tracks business processes flow and data in real time.

The TOE provides the core functionality of the Metastorm Product Line. Individual product components that are included in the evaluated configuration are identified and described in the Physical Boundaries section below. Other Metastorm products that are not included in the TOE provide functionality to connect the product to specialized enterprise application software that may exist in the environment (e.g., PeopleSoft) but are not required. These other products are licensed and sold separately. They do not provide any of the claimed security functions.

The remainder of this section summarizes the Metastorm BPM 9.1 product and architecture.

2.1 TOE Overview

In a traditional business environment, information may be passed between individuals in the form of memos, files, notes, and voice-mail messages. Ideally, team members receive all the information they require to complete their portion of a business process in a timely manner. This method, however, often causes problems. Pages of a document may be lost when being transferred from one team member to another. Important addresses and telephone numbers may be accidentally tossed in the garbage or buried under stacks of paperwork. Work grinds to a standstill or escalates to crisis management. The TOE provides functionality to improve upon manual and error prone business processes.

The TOE uses the concept of an electronic folder in which all information relevant to a particular task is placed. These folders are routed electronically from user to user as team members complete assigned activities. All the information necessary is gathered into a single location and made available to the participants as the project reaches their respective desktops. The chance of losing important information is minimized and business processes move smoothly toward completion.

The TOE provides a platform to automate and manage the entire life cycle of any business process. This occurs in two parts of a project's life-cycle: design and execution. The TOE views the information, activities, and instructions required to automate a business process as a project. The MBPM Designer application is used to design/create a project. The MBPM Engine subsystem supports the modelling of enterprise assets and the design, automation, control, and improvement of an organization's business processes by executing deployed projects. "Deploying" is the action that transitions a project that has been designed into a project that is executing.

Business Process Management is the process of viewing and managing the information, activities, and instructions required to automate a business process project. The main component of a project is one or more processes. Processes represent the various steps required to complete a business process. Each instance of a business process is called a folder and the steps which operate on a folder are called stages and actions. Processes are diagrams or process model logical constructs that depict business processes in the MBPM Designer application.

A folder contains one or more pages (forms) of information relating to that instance of the process. These forms are used to gather and display information necessary to the business process. A form will contain many fields each representing a specific piece of data.

When a folder reaches a user's desktop, the MBPM Engine views the folder as having reached a stage in the project. A process may also contain various system stages that do not require human interaction to move the folder to the next stage. The MBPM Engine defines an action as the step or activity necessary to modify data and move a folder from one stage to the next. Examples of actions include filling out a form, reviewing an attached file, approving/denying a request. It is also possible to have actions that do not require human intervention, such as: moving a folder based on information in the folder, or starting external applications.

A user is defined as anyone in an organization who uses the Metastorm system. In most cases, this includes everyone on the staff roster, and can be expanded to include contract or temporary workers, suppliers, customers, or business partners, as required. Roles are a way of grouping people in an organization. Individuals can be grouped by skill, function, geographical location, or any other criteria relevant to an organization. User can have any number of roles, and a role can be assigned to one or many users. Roles are created during the process design and added to the database when a project is deployed.

When a project is being designed, all aspects of the business process are built into the project. This includes a representation of the business process flow, the users or roles participating in the business process, the permissions each user and role has at each stage of the project, the actions each user and role can perform, the audit data that is gathered concerning the project, and the form used to present audit data for review.

One example of a business process involves a manager approving a staff member's form for a travel request.

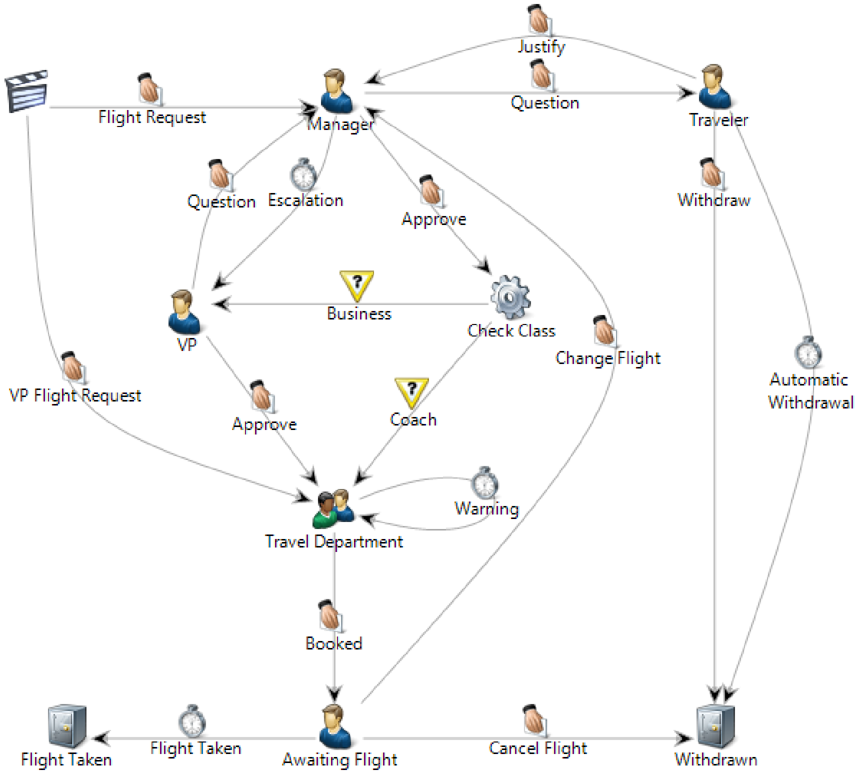

Figure 1: Sample Business Process

The above process depicts the following possible business process:

- An employee may be asked to justify their travel request.
- An employee may cancel or withdraw their travel request, ending the project.
- A request for a business-class flight is directed to a manager (VP in the above diagram stands for Vice President, simply intended as an identifier in the example, not intended to identify an additional VP role supported by the TOE) for second approval, while a request for a coach flight goes directly to the travel department after initial approval.
- Travel plans may be changed or cancelled, ending the project before the scheduled flight date.
- A request may be approved, travel scheduled, and the flight taken, ending the project.

2.1.1 Excluded Features

There are a number of features from the standard product that have been excluded from the evaluated environment. These components include

- **Server and Client Scripting**. The MBPM Designer provides a rich development environment to enable you to define your Business Processes and enabling tight integration with third party products. The

Metastorm BPM Designer enables you to generate scripts using C#, JScript and VBScript programming languages to create additional embedded functionality. The Scripts enable you to consume operating system resources, external assemblies and services to name a few. As there are very few restrictions with what you can do in the scripting it was deemed inappropriate to certify this functionality generically. Any scripts required to support your particular business processes should be considered in a case by case basis and the necessary exceptions be requested for a certified environment.

Note - this does not prevent the use of expression, conditional and visual scripting functionality within the product leveraging native functions within the designer to accomplish your general process logic/business rules.

- **"Administration Tools" native web client**. Some functionality within the native administration web client does not provide the level of logging desired for a product certified at this level for Common Criteria. To provide the equivalent functionality a series of Administration Forms built using the Metastorm BPM product have been provided for deployment to your environment. These forms are identified in 2.2 TOE Architecture.
    - As the Administration Forms were validated and delivered as part of the evaluated environment, the embedded script used by these forms is not impacted by the previous scripting exclusion statement.

2.2 TOE Architecture

A business process (project) is comprised of processes that define folders, stages and actions. A folder is a unique instance of a business process (project). A folder contains one or more forms. A form contains information relating to an instance of the business process (project). The TOE can control access to objects called forms and folders. The TOE provides users with interfaces that can view and manage business processes (projects).

The TOE has the ability to restrict user access to forms and folders. Users are assigned to a role. Forms and folders have associated with them Access Control Lists (ACLs). The ACL identifies a user and/or role and the actions that the user and/or role are permitted to perform. The ACL is used to make access control decisions for the associated object.

Users access the TOE using a web browser in the environment. The web browser utilizes the TOE's web interface. Users are required to provide a user name and password to the TOE before a session with the TOE can be established.

Administrative users access the TOE's Administration Tool using a web browser in the environment. Administrators using this application are required by the TOE to provide a user name and password to the TOE before a session with the TOE can be established.

The TOE in its intended environment can be described in terms of the following subsystems:

- MBPM ASP.NET Web Application (web server plug-in) subsystem – This subsystem is an ASP.NET application for Microsoft Internet Information Services (IIS) web server. This subsystem presents a web interface to users and administrators through IIS. User web page activity is translated into XML messages that are exchanged with the MBPM Engine subsystem. Administrators utilize MBPM Administrator Forms to manage the TOE.
    - MBPM Administrator Forms[1] – The MBPM Admin Forms provide interfaces to perform security relevant administrative operations through the use of a Metastorm provided business process model. This model contains 11 Admin Forms designed specifically to support functionality described by this Security Target. The following is a list of these forms.

        1. Assign Roles to User Form
        2. Assign Users to Role Form
        3. Audit Trail – Deployment

[1] This is typically referred to as simply "Admin Forms".

4. Audit Trail – Processes
5. Change Password Form
6. Edit Server Settings Form
7. Event Log – Metastorm Apps
8. Authentication Log Form
9. Process Log Form
10. Create, Delete and Update Users Form
11. Update Session Timeout Form

- MBPM Engine subsystem – This subsystem is a server application that evaluates and processes MBPM transaction requests from the end users. This subsystem processes Business Process Management logic that is defined by administrators and is operated upon by end users. The result is that this subsystem performs and controls work flow management functions.
- MBPM Deployment Service Subsystem – The Deployment Service subsystem is responsible for validating and then preparing process models for execution. The Deployment Service interprets the process models and writes the appropriate process metadata into the process repository, in a format that is used by the Process Engine. By acting as an intermediary between the Designer and the runtime repository, it also allows users to deploy process models without needing direct access permissions to the DBMS. It also has the secondary ability of storing copies of the Designer libraries and projects that contain the process models, for later retrieval and loading back into the Designer
- MBPM Designer application – This application provides interfaces to create and modify procedures and their components (forms, folders). This application is accessed using the interfaces provided by the application.

The environment the TOE runs in is composed of the following.

- Operating system – Provides runtime environment for MBPM Engine subsystem and MBPM Engine administrator console subsystem (as well as database, web server, and web browser).
- Database – Stores MBPM Engine subsystem and MBPM Engine administrator console subsystem configuration data.
- Web server – Provides runtime environment for MBPM ASP.NET Web Application subsystem. The web server is also expected to be configured such that web pages served from the MBPM ASP.NET Web Application are provided only through the HTTPS protocol for environments configured in Common Criteria mode.
- Web browser – Provides a web-based client interface to access MBPM Engine subsystem services using the MBPM ASP.NET Web Application subsystem.

2.2.1 Physical Boundaries

The components listed below are expected to operate within a single computer system. These components that make up the TOE are:

- MBPM ASP.NET Web Application subsystem
    - MBPM Administrator Forms
- MBPM Engine subsystem
- MBPM Deployment Service Subsystem
- MBPM Designer Application

When configured in Common Criteria mode the TOE depends on the following:

- Operating system –Windows Server 2003 R2, Windows Server 2003 SP2, Windows Server 2008 SP2, Windows Server 2008 R2 SP1

- Database –Microsoft SQL Server 2005 SP4, Microsoft SQL Server 2008 SP1, Microsoft SQL Server 2008 R2, Oracle 10G R2, Oracle 11G R1, Oracle 11G R2
- Web server –Microsoft IIS 6 running on Windows Server 2003, Microsoft IIS 7, Microsoft IIS 7.5
- Web browser – IE 7, IE 8, Firefox

For greater security the later versions of these components are recommended.

When installed in a configuration that does not require the Common Criteria mode (Development/Test environments for example), any of the supported configurations identified in the "Metastorm BPM 9.1 Supported Environments" guide can be used.

The TOE in its intended environment is depicted in the figure below. In the following figure, the server console represents a local console for the computer on which the MBPM Designer is running. Additionally, while the user's web browser could be on the local console as well, it is expected to be on a remote network device.

The operating system in the environment provides an execution environment for components of the TOE, a reliable clock from which the TOE obtains time, network protocol support for communicating with web browsers as well as other resource management traditional for operating systems (e.g., process isolation, timesharing of the CPU, and basic I/O).

The database provides a storage and retrieval mechanism utilized by components of the TOE.

The web server provides an execution environment for the MBPM ASP.NET Web Application component of the TOE. The web server also offers the HTTPS network protocol interface to a User's web browser. TOE guidance recommends the use of HTTPS.

The web browser acts as the user's GUI.

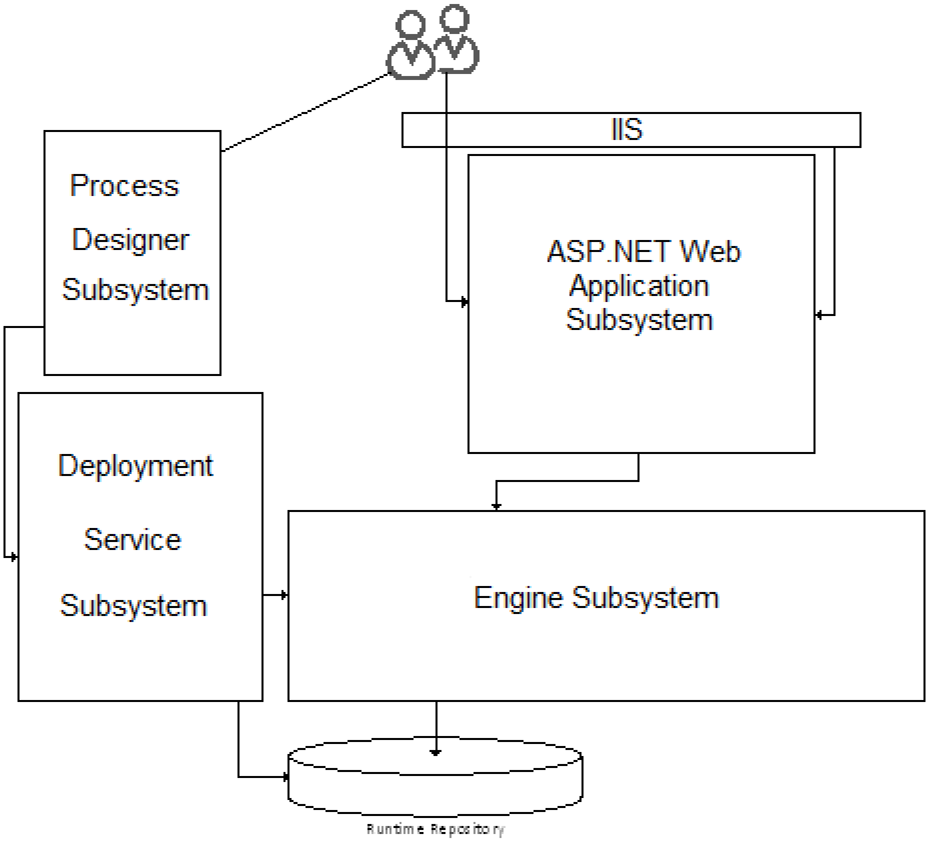
IIS
Process Designer Subsystem
ASP.NET Web Application Subsystem
Deployment Service Subsystem
Engine Subsystem
Runtime Repository

Figure 2: Metastorm BPM TOE Components

2.2.2 Logical Boundaries

The TSF provides the following security functions:

- auditing
- user data protection,
- identification and authentication,
- security management,
- protection of the TSF, and
- TOE access.

2.2.2.1 Auditing

The TOE generates audit events as each component of the TOE performs actions on deployed projects. The records of audited events are saved by the TOE in the database for later retrieval and reviewed by administrative users through an audit trail form. This audit trail form presents records in a manner determined by a project's designer and can be constructed to permit searching and/or sorting of audit records.

The MBPM Administrator Forms provide administrators with a way to view audit records created by the TOE.

See the corresponding section in the TSS for more detailed information.

2.2.2.2 User data protection

Business processes (projects) are made visible to the end user by deploying the project. This is a mechanism by which a business process (project) created inside the MBPM Designer is turned into an executable application which can be used via the MBPM ASP.NET Web Application. The deployment service transforms the business processes (projects) into the runtime metadata required by the MBPM Engine to render the defined process application at runtime. Before a business process has been deployed, it is NOT an object protected by the TOE. An un-deployed business process is data that exists in the environment that does not come under TOE control until such time as it has been deployed.

Users login and access projects using a web browser. After login the user has the following four lists available.

- Blank forms list – A list of forms that can be used by the User to start a process.
- To Do List – A list of folders on which the user (or users that share the same role) must act.
- Watch List – A list of folders the user can monitor. In some processes, the user is authorized to act on folders in the Watch list
- Administration forms – A list of administration forms to which the user has access.

The TOE can control access to objects called forms and folders using ACLs specific to each form and folder. A business processes is comprised of folders that transition between stages through actions. A folder is a unique instance of a business process. A folder contains one or more forms. A form contains fields defining specific information that pertains to an instance of the business process.

When a business process is designed, the designer chooses the users or roles that are permitted to have access to a given folder, form or field. ACLs are used to define permissions on folders and forms. Fields on a form are either visible or not, depending upon the 'Visibility Depends On' property of the field. If the field is visible the user has the ability to modify or use the field. Visibility can be restricted based upon role.

See the corresponding section in the TSS for more detailed information.

2.2.2.3 Identification and authentication

The TOE defines users in terms of the security attributes user name, password, and role. The TOE provides its own username and password authentication mechanism that it uses to authenticate users. While the product supports the use of additional authentication mechanisms (e.g., LDAP, RADIUS), only the local, TOE-defined username/password mechanism is supported in the evaluated configuration. In order to access the TOE, a user account including a user name and password must be created for the user. The TOE maintains both administrator and user roles.

Administrative and non-administrative users are required to be successfully identified and authenticated by providing a valid username and password before access to the TOE and its resources is allowed. The TOE does not require a user name and password before allowing use of the Windows-based MBPM Designer application. However, it is expected the user of the MBPM Designer has been successfully identified and authenticated by the operating environment. The MBPM Designer application will obtain a username and password from the user. These credentials are submitted to the MBPM Deployment Service whenever there is an attempt to "deploy" a new business process model (a.k.a., project).

See the corresponding section in the TSS for more detailed information.

2.2.2.4 Security management

The TOE provides applications and web-based administration forms that can be used to manage the TSF. The applications and forms include those that can perform the following management functions:

- design and deploying of business process projects,
- management of subjects and authentication data,
- management of objects, and
- management of session inactivity settings.

The TOE ensures that only an administrator can login and perform administrative management function. The TOE recognizes several roles: a process designer, an administrator, a user, and designer-specified user roles.

See the corresponding section in the TSS for more detailed information.

2.2.2.5 Protection of the TSF

The TOE restricts access to both its administrative and non-administrative interfaces. The TOE ensures that only an administrator can login and perform administrative management functions. The TOE also utilizes support in the environment (e.g., the database, the web server, and the operating system) to protect data stored in the database, to communicate with network entities, and to protect communications with users. This information is provided in support of the security assurance requirements, more specifically the architecture requirements for non-bypassability.

See the corresponding section in the TSS for more detailed information.

2.2.2.6 TOE access

The TOE can terminate inactive interactive user sessions. The TOE relies on a timestamp provided by the operating system in the environment in order to determine if a session has become inactive.

See the corresponding section in the TSS for more detailed information.

3. Security Environment

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:

- Threats that the TOE is designed to counter
- Assumptions made on the operational environment and the method of use intended for the TOE
- Organizational security policies to which the TOE is designed to comply

3.1 Organizational Policies

P. AUTHORIZED_USERS Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so

P. I_AND_A All users must be identified and authenticated prior to accessing any controlled resources

P. NEED_TO_KNOW The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information

P. ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users

3.2 Threats

T. ADMIN_ERROR An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

T. MASQUERADE An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

T. TSF_COMPROMISE A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).

T. UNAUTH_ACCESS A user may gain unauthorized access (view, modify, delete) to user data.

T.AUDACC Persons may not be accountable for the actions that they conduct, thus allowing an attacker to escape detection.

3.3 Assumptions

A.LOCATE The TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

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

A.COMPROT The environment will provide protection of TSF data transmitted between the TOE and the database, as well as communication between the MBPM web application and the MBPM engine.

A.STORAGE The environment will provide a storage capability for audit records that protects audit records and makes them available for the TOE to retrieve.

4. Security Objectives

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

4.1 Security Objectives for the TOE

O.AUDITS The TOE must provide a means to review and record an audit trail of security-related events, with accurate dates and times.

O.ACCESS The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.

O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.

O.USER_AUTHENTICATION The TOE will verify the claimed identity of users.

O.USER_IDENTIFICATION The TOE will uniquely identify users.

4.2 Security Objectives for the Environment

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

OE.PHYCAL The TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

OE.COMPROT The environment will provide protection of TSF data transmitted between the TOE and the database, as well as communication between the MBPM ASP.NET Web Application and the MBPM Engine.

OE.TOE_PROTECTION The TOE will be designed to protect itself and its assets from external interference or tampering.

OE.STORAGE The environment will protect audit data stored by the TOE and allow retrieval of audit records by the TOE.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are satisfied by the TOE. All SFRs were drawn from Part 2 of the Common Criteria v3.1 Revision 2

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_SAR.1: Audit Review
	FAU_STG.4: Prevention of audit data loss
FDP: User data protection	FDP_ACC.2: Complete access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.1: Timing of authentication
	FIA_UID.1: Timing of identification
FMT: Security management	FMT_MTD.1: Management of TSF data
	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security Roles
FTA: TOE access	FTA_SSL.3: TSF-initiated termination

Table 1: TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [***not specified***] level of audit; and
c) [**the events for the "not specified" level of audit are listed in the table below**].

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, [**no additional details**].

Requirement Component	Auditable event
FAU_GEN.1	start-up and shutdown of the audit functions
FDP_ACF.1	successful requests to commit actions or projects that affect objects covered by the Work Flow Access Control Policy
FIA_UAU.1	the final decision on authentication (i.e., success or failure)
FIA_UID.1	unsuccessful use of the user identification mechanism, including the user identity provided
FTA_SSL.3	termination of an interactive session by the session locking mechanism

FMT_SMF.1	• management of user accounts, • changes to the interactive session timeout value

Table 2: Audit Events

5.1.1.2 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[authorized administrator]** with the capability to read **[all audit information stored in the database]** from the audit records.

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

5.1.1.3 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall **[*prevent audited events, except those taken by the authorised user with special rights*] and** [**no other actions**] if the audit trail is full.

5.1.2 User data protection (FDP)

5.1.2.1 Complete access control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the **[Work Flow Access Control Policy]** on **[subjects: users; objects: forms, folders]** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object within the TSC are covered by an access control SFP.

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[Work Flow Access Control Policy]** to objects based on the following: **[subject security attributes: user identifier and role; object security attributes: object owner, and access control list (ACL)]**.

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

(a) if the user identity is equal to the object owner, the requested access is allowed; or

(b) if the ACL grants the requesting user identity the requested access, the requested access is allowed; or

(c) if the user identity is a member of a role defined for the object and the ACL grants the role the requested access, the requested access is allowed].

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

[if the subject has the administrator role, the requested access is allowed].

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

5.1.3 Identification and authentication (FIA)

5.1.3.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[user identity, authentication data, and role]**.

5.1.3.2 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [**access to undeployed models**] 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.

5.1.3.3 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [**access to undeployed models**] 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.

5.1.4 Security management (FMT)

5.1.4.1 Management of TSF Data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [**manage**] the [**user account data and interactive session timeout values**] to [**administrators**]**.**

5.1.4.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Work Flow Access Control Policy**] to restrict the ability to [**modify**] the security attributes [**object owner and access control list**] to [**administrators**].

5.1.4.3 Static attribute initialization (FMT_MSA.3)

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

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

5.1.4.4 Specification of Management Functions (FMT_SMF.1)

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

a) Management of Work Flow Access Control Policy,
b) Management of user accounts,
c) Management of interactive session timeout value].

5.1.4.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[designer, administrator, user, and custom designer-specified roles]**.

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

5.1.5 TOE access (FTA)

5.1.5.1 TSF-initiated termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate an interactive session after a **[administrator configurable amount of time]**.

5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 components as specified in Part 3 of the Common Criteria v3.1 Revision 2. No operations are applied to the assurance components.

The assurance requirements (including those defined for EAL 4 and the ALC_FLR.2 requirement) were selected as the assurance level because the TOE is a commercial product whose users require a low to moderate degree of independently assured security. In addition, augmentation was chosen to provide the added assurances that result from having flaw remediation procedures and correcting security flaws as they are reported. The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential. As such, is the assurance requirements are appropriate to provide the assurance necessary to counter the limited potential for attack.

The ASE requirements are not copied into this document as they are intended to define the requirements upon which this document is evaluated. Assurance requirements in this document are those used to evaluate the product.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design
AGD: Guidance Documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-Cycle Support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.2: Testing: security enforcing modules
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability Assessment	AVA_VAN.3: Focused vulnerability analysis
ASE: Security Target Evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification

Table 3: EAL 4 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 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.

5.2.1.2 Complete functional specification (ADV_FSP.4)

ADV_FSP.4.1d The developer shall provide a functional specification.

ADV_FSP.4.2d The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.4.1c The functional specification shall completely represent the TSF.

ADV_FSP.4.2c The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3c The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4c The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from security enforcing effects and exceptions associated with an invocation of each TSFI.

ADV_FSP.4.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.4.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.1.3 Implementation representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2d The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

ADV_IMP.1.1c The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2c The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3c The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

ADV_IMP.1.1e The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

5.2.1.4 Basic modular design (ADV_TDS.3)

ADV_TDS.3.1d The developer shall provide the design of the TOE.

ADV_TDS.3.2d 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.3.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2c The design shall describe the TSF in terms of modules.

ADV_TDS.3.3c The design shall identify all subsystems of the TSF.

ADV_TDS.3.4c The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5c The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6c The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose.

ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, and called interfaces to other modules.

ADV_TDS.3.9c The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10c The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

ADV_TDS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.2.2 Guidance documents (AGD)

5.2.2.1 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 fulfil 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.

5.2.2.2 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.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Production support, acceptance procedures and automation (ALC_CMC.4)

ALC_CMC.4.1d The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.4.2d The developer shall provide the CM documentation.
ALC_CMC.4.1c The TOE shall be labelled with its unique reference.
ALC_CMC.4.2c The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.4.3c The CM system shall uniquely identify all configuration items.
ALC_CMC.4.4c The CM system shall provide automated measures such that only authorised changes are made to the configuration items.
ALC_CMC.4.5c The CM system shall support the production of the TOE by automated means.
ALC_CMC.4.6c The CM documentation shall include a CM plan.
ALC_CMC.4.7c The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.4.8c The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ALC_CMC.4.9c The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.4.10c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.
ALC_CMC.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 Problem tracking CM coverage (ALC_CMS.4)

ALC_CMS.4.1d The developer shall provide a configuration list for the TOE.
ALC_CMS.4.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.
ALC_CMS.4.2c The configuration list shall uniquely identify the configuration items.
ALC_CMS.4.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.3 Delivery procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document 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.

5.2.3.4 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.
ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.2.3.5 Flaw reporting procedures (ALC_FLR.2)

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

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

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

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

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

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

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

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

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

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

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

5.2.3.6 Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

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

5.2.3.7 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of each development tool.

ALC_TAT.1.1c Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3c The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

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

5.2.4 Tests (ATE)

5.2.4.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

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

5.2.4.2 Testing: security enforcing modules (ATE_DPT.2)

ATE_DPT.2.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and modules in the TOE design.

ATE_DPT.2.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.2.3c The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.

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

5.2.4.3 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.

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.

5.2.4.4 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 interfaces to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Focused vulnerability analysis (AVA_VAN.3)

AVA_VAN.3.1d The developer shall provide the TOE for testing.

AVA_VAN.3.1c The TOE shall be suitable for testing.

AVA_VAN.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4e 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 Enhanced-Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Auditing

The TOE generates audit events for the "not specified" level of audit. The TOE generates an audit record that includes event details such as event type, subject identity, the time the event occurred, and an indicator of the outcome of the event (success or failure). The time stamp is provided by the environment. The following events are audited by the TOE:

- start-up and shutdown of the audit functions[2]
- successful requests to commit an action covered by the Work Flow Access Control Policy,
- the final decision on authentication,
- unsuccessful use of the user identification mechanism, including the user identity provided,
- termination of an interactive session by the session locking mechanism,
- management of user accounts, and
- changes to the interactive session timeout value.

The TOE audits each time the MBPM Engine is started and stopped by writing a log record into the Windows Application Log. The TOE always generates and stores audit data when it is running. Thus, auditing the starting and stopping of the engine audits the starting and stopping of the audit function.

Audit data is automatically collected as each component of the TOE perform actions on deployed projects. The audit data is stored in database tables. The event history is recorded in the database eEvent table automatically (out of the box functionality). The eLog table contains one record for every time a project is deployed as well as for records of failures in process execution[3]. Because the TOE offers to users only those actions for which they are permitted, users cannot attempt to perform actions for which they do not have permission. Thus failures in process execution occur as a result of environmental conditions (e.g., network connectivity problems) or constraints imposed by the environment (e.g., database consistency checks).

Once any audit data is written to the database or the Windows Application log, the database and Windows are then responsible for storing and protecting the audit data. The audit trail is stored as a database table. If the database becomes full the MBPM Engine will fail thus stopping auditable events until the administrator makes space.

Metastorm BPM provides a complete record for every action that is performed on a Metastorm record (a.k.a. folder) within the system. This information is placed in the eEvent table. This audit trail information includes: User information, date and time of each record creation, modification, or deletion of folder information, and a description of the action performed. The information that is saved in the database (i.e., the eLog and the eEvent tables) is also referred to as the Process Repository. Information from the Process Repository can be reported on as required.

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

- FAU_GEN.1: The TOE generates audit events for the "not specified" level of audit. The audit data is stored in the database tables. The event history is recorded in the database eEvent table automatically. The eLog table contains a record of deployed projects and process execution failure. The TOE writes a message in the Windows Application log for the starting and stopping of the Engine.

[2] The TOE writes a message in the Windows Application log for the starting and stopping of the Engine.

[3] Because forms present to users only those actions and fields for which the user has access, failures in process execution do not represent security violations.

- FAU_SAR.1: The TOE provides the administrator a GUI interface to view the audit records in text format. The "Audit Trail form" presents audit records to the administrator for review.
- FAU_STG.4: When the database table that the TOE uses to store audit records becomes full, the TOE stops performing auditable events until the administrator makes space for more audit records. The only non-auditable events are viewing requests already assigned to a user if they are already authenticated against the system since logging in generates an audit event. The system administrator will need to log in to the database environment and make space to re-enable normal operations.

6.1.2 User data protection

The TOE implements a Work Flow Access Control Policy for object access based on:

- user identities,
- object ownership,
- assigned roles, and
- Access Control Lists (ACLs).

The TOE objects that are subject to this policy are forms, folders and actions (or operations). Forms are used to define business process information in objects. Folders are collections of forms that represent logical constructs of business process model processes. Combinations of forms and folders represent business processes that the TOE can provide users interfaces with in order to view and manage. Users manipulate the forms and the routing of the process by performing operations, or actions, supported by the business process (e.g., fill out fields of a form, submit the form, delete the form, and approve a request).

Once the user completes an operation on a form, the TOE may make the form available to another user (e.g., once a user submits a travel request form it is available to a manager for approval). Users continue performing actions on forms until the form has reached the "end" of the defined business process.

The TOE presents users with the folders (and the forms within the folders) for which the users have access. Fields within a form have a property called "visibility depends on" which allows a designer to setup whether a field within a form is displayed. Although fields do not natively have access controls, this feature can also be used to control visibility based upon the user's role.

The TOE has the ability to restrict access to forms, folders and actions (or operations) to their owners, administrators and other standard and custom defined roles during the design of the business processes. This is accomplished by associating these roles with the project components dictating which groups of users have access to it. These associations may help define who can initiate a particular process, take a specific action at a stage within the process, or see a form and its associated fields for example. Roles may be calculated, in which case data provided in the business process may help identify the user(s) fulfilling this role, or they may be managed through assignment of users to these roles as part of process administration once deployed. A user can have any number of roles, and a role can be assigned to one or many users.

Forms and actions have settings to "Only show action if" and "Only show form if" respectively. Through defining conditional logic in these settings you can further restrict access to them in addition to any role based restrictions also defined.

Because a business process project typically undergoes a thorough design prior to its deployment throughout an enterprise network, the flexibility for a designer to specify specific default permissions or to permit unrestricted permission simply allows a designer to choose a desired behaviour after deployment. Once a project has been deployed, the ACLs on newly created objects will match the constraints defined by the business process designer.

It is the combination of all of these settings and role memberships that translate to the ACL for a running process.

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

- FDP_ACC.2, FDP_ACF.1: All users are subject to the Work Flow Access Control Policy for all available operations on forms and folders.

6.1.3 Identification and authentication

The TOE defines users in terms of:

- user identity,
- authentication data, and
- roles.

The TOE provides its own username and password authentication mechanism that it uses to authenticate users. While the product supports additional authentication mechanisms, only the local, TOE-defined username/password mechanism is supported in the evaluated configuration. In order to access the TOE, user account data including a user name and password must be created for the user. Administrators can change their own password or a user's password through one of the MBPM Admin Forms called the "Change Password Form". The TOE also allows a business process to be created and deployed to provide users with the ability to change their own password.

Administrative users access the TOE to perform administrative tasks using the MBPM admin forms. Administrators are required to provide a user name and password before a session with the TOE can be established. Administrators are required to provide a user name and password before deploying a business process project (deployment was previously referred to as publishing[4]) when using the MBPM Designer application. Prior to deployment, the MBPM Designer can be used to design a project; however, the MBPM Designer does not control access to undeployed projects. The BPM Designer stores undeployed projects in files maintained and protected by the environment.

Note that the TOE does not implement any password composition rules or minimum password lengths. Administrative guidance is relied on to ensure that when user accounts are created, a minimum password length of eight printable characters is used. Guidance also provides recommendations on good password practices such as not sharing passwords, not writing down passwords, avoiding passwords with repeating characters, dictionary words or other easily guessed passwords.

When a user, without the necessary role, requests communication with the TOE, access is denied. Users cannot proceed to use their TOE role until they have supplied a user name and password that corresponds to the TOE access list. The TOE security functions are invoked and succeed before each function within the TSC is allowed to proceed.

The TOE relies on a timestamp provided by the operating system in the environment in order to determine if a session has become inactive. When a threshold has been exceeded, the user must re-authenticate.

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

- FIA_ATD.1: The TOE defines users in terms of security attributes that include user name, authentication data (password), and role.
- FIA_UAU.1: The TOE offers no TSF-mediated functions, except those available through the MBPM Designer application prior to the login required to deploy a model. The TOE authenticates users using its username/password mechanism.
- FIA_UID.1: The TOE offers no TSF-mediated functions, except those available through the MBPM Designer application prior to the login required to deploy a model.

6.1.4 Security management

The TOE supports several roles that are responsible for creating, managing and using a business process (project). The users in the designer role are responsible for creating a definition of process flows, defining security defaults, available actions, and the set of collected data associated with a project. The administrator role is held by users that deploy projects, manage the execution of deployed projects, manage users, manage the access control policy and define inactivity thresholds. The 'user' role is held by users that participate in the business process by following the project established by the designer. The custom, designer-specified roles are a set of roles that are created by a

[4] Deploying a process is the act of saving the process into the database, thus making it possible for the TOE to execute the project.

designer for use as an access control grouping within the context of a specific business process. An example of a custom, designer-specified role would be a manager that has access to folders, forms and/or actions that the manager's subordinates create.

The designer and administrator roles are both trusted with the responsibility of creating and operating a working business process that accomplishes a business goal.

The TOE provides designers and administrators with either a Windows application graphical user interfaces (GUI) or a web-based GUI. These GUIs are used to create and manage projects, and to manage the security functions of the TOE. The administrative interfaces support the following management functions:

- creation and deploying business process projects
- management of subjects and authentication data
- management of objects
- management of session inactivity settings

Designers use the MBPM Designer application to create business process projects. Designers are not required to login to the MBPM Designer. The MBPM Designer is expected to utilize access controls and version controls (i.e., tools to manage changes to multiple versions of a piece of a business process project while it is being designed) available in the environment as needed during the development of a business process project.

Administrators are required to provide a user name and password (i.e., to login) before a Web-based session with the TOE can be established.

The administrative interfaces also enable an administrator to manage the static roles assigned in an organization. These roles are defined in the process of designing a process. Users are then assigned to roles. Role management tasks include:

- Adding and deleting users
- Granting a role to a user or group of users
- Granting multiple roles to a user
- Viewing a user's roles
- Validating role allocations
- Viewing the holders of roles
- Removing a role from a user or group of users
- Removing a user from a role

The MBPM Admin Forms provide the following core functions:

- Changing user passwords
- View audit records

The MBPM Admin Forms ensure that only an administrator can login. The MBPM Designer application ensures that only administrators are allowed to deploy a model by requiring a login prior to deploying. Users that have been assigned the administrator role, are considered authorized administrators, all others are simply users.

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

- FMT_MTD.1: Only administrators can login to the MBPM Admin forms and thus only administrators can manage user account data (including authentication data) and session inactivity settings.
- FMT_MSA.1: Only administrators can login to the MBPM Admin forms and thus only administrators can manage object owners and access control lists.
- FMT_MSA.3: By default, every object is created with the creator as the owner and access permitted to everyone. Subsequently, access can be changed for other users. The designer can specify alternative values.
- FMT_SMF.1: The TOE provided administrator console interfaces to manage the Work Flow Access Control Policy, to manage user accounts, and to manage inactive session threshold values.

- FMT_SMR.1: Users that create business process projects are considered to be holding the designer role. Users that have been assigned the administrator role are considered authorized administrators. Users can also be assigned to custom, designer-specified roles. All others are simply users.

6.1.5 Protection of the TSF

As described in section 2.2.1, the TOE consists of the following subsystems.

- MBPM ASP.NET Web Application subsystem
    - MBPM Administrator Forms
- MBPM Engine subsystem
- MBPM Deployment Service Subsystem
- MBPM Designer Application

In addition to these subsystems, the TOE makes extensive use of a database and a web server that are part of the environment to provide a base layer of security to protect the TSF. Physical access to the equipment used to host the environment and TOE is controlled by the system administrator and/or IT department for the organization. Access control lists defined in the environment ensure that only authorized personnel may access or modify the binary files, configuration files, and registry settings that make up the TOE.

The MBPM Engine defines the business process model configuration information, including the role definitions and how they apply to various aspects of each process. A Business Process Management user running the MBPM Engine must provide a user name and password in order to deploy a project or save configuration data. The user name and password must be valid credentials belonging to the user that has been granted the ability to deploy. The MBPM Deployment Service subsystem confirms the credentials and authorization by calling the MBPM Engine subsystem.

The MBPM Engine reads the configuration information and project information from the database and performs operations (e.g., running projects). The MBPM ASP.NET Web Application (a.k.a., MBPM web application) presents a web interface to users of the project. This web interface passes user requests for actions to the MBPM engine for processing. The MBPM engine enforces the Work Flow Access Control Policy on all requests passed from the MBPM web application. The MBPM web application requires users to provide an MBPM user name and password which is passed to the MBPM Engine for verification. The MBPM Engine utilizes configuration information stored within the database to determine if the MBPM user name and password provided by the user is valid.

Only those users who have authenticated their credentials, as described above, may access the MBPM Admin Forms provided that the user has been granted the role permissions that authorize access to the MBPM Admin Forms module. The MBPM Admin Forms provide the functionality for creating, maintaining and deleting user accounts, login credentials and roles assigned to the user accounts, and other additional security settings that are specific to the TOE. Therefore, no unauthorized modification to the security settings is possible.

Figure 3 shows communication paths between subsystems of the TOE and software in the environment. Also shown are the TSFI that represent each communication pathway.

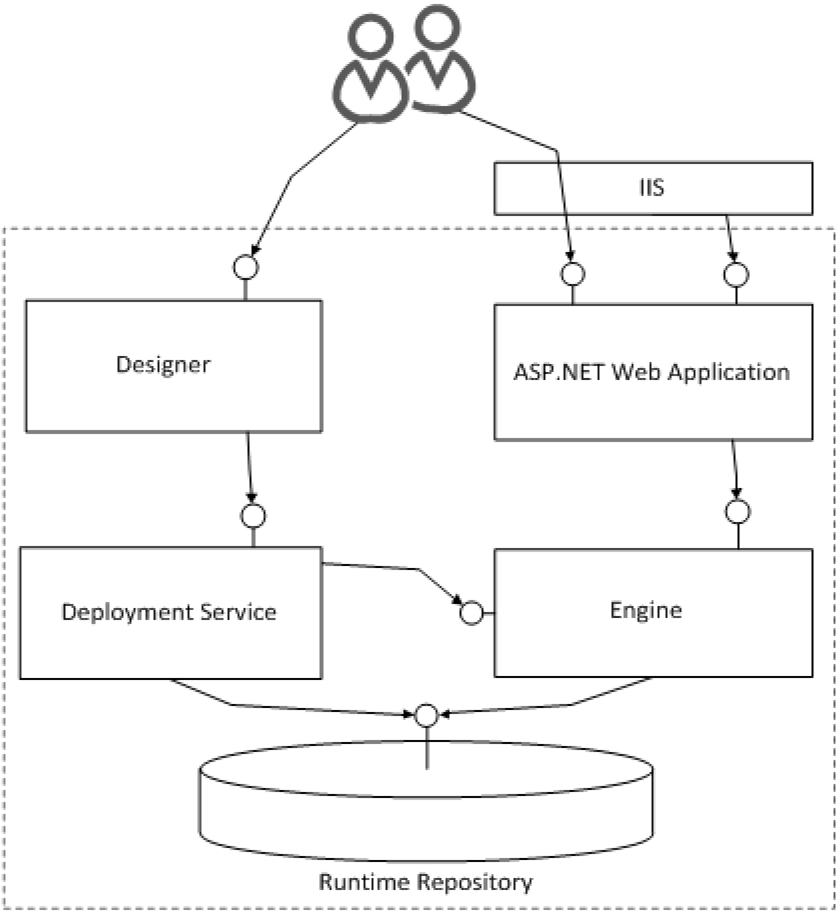

Figure 3: Communication Pathways

The database must be configured such that the ODBC, OLEDB, and ADO.NET communication pathways utilize all necessary protection mechanisms to protect the integrity and confidentiality of the information passing between the subsystems of the TOE and the database. Since database credentials may pass between the TOE and the database, mechanisms provided by and configured in the environment must be used that protect the integrity and confidentiality of these credentials.

The MBPM ASP.NET Web application utilizes .NET Remoting communication mechanisms provided by the environment to facilitate invocation of MBPM Engine functionality. These mechanisms offer features that protect the integrity and confidentiality of the information passing between the subsystems. To mitigate spoofing attack, the ASP.NET Web Application's application pool identity must be configured as a 'Client' of the 'Metastorm Process Engine' COM+ application.

The web server makes the web interfaces offered by the MBPM ASP.NET web application available to users with web browsers. Since the web server is in the environment, it is the environment that determines whether HTTP or HTTPS is used to protect communications between the user and the MBPM ASP.NET web application. TOE guidance recommends the use of HTTPS.

The communications between the MBPM Designer and the MBPM Deployment Service; as well as the communications between the MBPM Deployment Service and the MBPM Engine also utilize the default transport layer security that is provided by WCF, and ultimately implemented in the operating system. These communications are encrypted using the Windows credentials under which the programs are executing.

Note, this information is provided in support of the security assurance requirements, more specifically the architecture requirements for non-bypassability.

6.1.6 TOE access

TOE Administrators set the session time-out period for users accessing the TOE. The default for this setting is 60 minutes. If a user attempts to access a service after the period of inactivity indicated by the session time-out value, the TOE displays a message telling the user that their session has timed out and that they must log out and log back in again. The session times out if the user does not interact with a service for a set period. The following are examples of interacting with a service:

- Refreshing a list
- Opening a folder
- Submitting a folder
- Entering data and moving to another field when there are dependencies

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

- FTA_SSL.3: The TOE can terminate a user's interactive session after an administrator set time period that must be greater than zero.

7. Protection Profile Claims

There is no Protection Profile claim 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;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

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

8.1.1 Security Objectives Rationale for the TOE and Environment

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

	O.AUDITS	**O.ACCESS**	**O.ADMIN_ROLE**	**O.MANAGE**	**OE.TOE_PROTECTION**	**O.USER_AUTHENTICATION**	**O.USER_IDENTIFICATION**	**OE.CONFIG**	**OE.PHYCAL**	**OE.COMPROT**	**OE.STORAGE**
P.AUTHORIZED_USERS		X									
P.I_AND_A						X	X				
P.NEED_TO_KNOW		X					X				
P.ROLES			X								
T.AUDREC	X										
T.ADMIN_ERROR				X							
T.MASQUERADE						X	X				
T.TSF_COMPROMISE					X						
T.UNAUTH_ACCESS		X	X								
A.LOCATE									X		
A.NO_EVIL								X			
A.COMPROT										X	
A.STORAGE											X

Table 4: Environment to Objective Correspondence

8.1.1.1 P. AUTHORIZED_USERS

Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

8.1.1.2 P. I_AND_A

All users must be identified and authenticated prior to accessing any controlled resources.

This Organizational Policy is satisfied by ensuring that:

- O.USER_AUTHENTICATION: The TOE will verify the claimed identity of users.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.3 P. NEED_TO_KNOW

The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.USER_IDENTIFICATION: The TOE will uniquely identify users.

8.1.1.4 P. ROLES

The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

This Organizational Policy is satisfied by ensuring that:

- O.ADMIN_ROLE: The TOE will provide authorized administrator roles to isolate administrative actions.

8.1.1.5 T.AUDREC

Persons may not be accountable for the actions that they conduct, thus allowing an attacker to escape detection.

This Threat is countered by ensuring that:

- O.AUDITS: This security objective is necessary to counter the threat: T.AUDACC by requiring an audit trail of security relevant events and the ability to view the audit trail.

8.1.1.6 T. ADMIN_ERROR

An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is countered by ensuring that:

- O.MANAGE: The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.

8.1.1.7 T. MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

This Threat is countered by ensuring that:

- O.USER_IDENTIFICATION: The TOE will uniquely identify users.
- O.USER_AUTHENTICATION: The TOE will verify the claimed identity of users.

8.1.1.8 T. TSF_COMPROMISE

A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).

This Threat is countered by ensuring that:

- OE.TOE_PROTECTION: The TOE will be designed to protect itself and its assets from external interference or tampering**.**

8.1.1.9 T. UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

This Threat is countered by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
- O.ADMIN_ROLE: The TOE will provide authorized administrator roles to isolate administrative actions.

8.1.1.10 A. LOCATE

The TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

This Assumption is satisfied by ensuring that:

- OE.PHYCAL: The TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

8.1.1.11 A. NO_EVIL

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

This Assumption is satisfied by ensuring that:

- OE.CONFIG: The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation

8.1.1.12 A.COMPROT

The environment will provide protection of TSF data transmitted between the TOE the database, as well as communication between the ME web application and the ME engine.

This Assumption is satisfied by ensuring that:

- OE.COMPROT: The environment will provide protection of TSF data transmitted between the TOE and the database, as well as communication between the ME web application and the ME engine.

8.1.1.13 A.STORAGE

The environment will provide a storage capability for audit records that protects audit records and makes them available for the TOE to retrieve.

This Assumption is satisfied by ensuring that:

- OE.STORAGE: The environment will protect audit data stored by the TOE and allow retrieval of audit records by the TOESecurity Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that Table 5 indicates the requirements that effectively satisfy the security objectives.

8.1.2 Security Functional Requirements Rationale

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

	O.AUDITS	**O.ACCESS**	**O.ADMIN_ROLE**	**O.MANAGE**	**OE.TOE_PROTECTION**	**O.USER_AUTHENTICATION**	**O.USER_IDENTIFICATION**
FAU_GEN.1	X						
FAU_SAR.1	X						
FAU_STG.4	X						
FDP_ACC.2		X					
FDP_ACF.1		X					
FIA_ATD.1							X
FIA_UAU.1		X				X	
FIA_UID.1		X					X
FMT_MTD.1				X			
FMT_MSA.1				X			
FMT_MSA.3				X			
FMT_SMF.1				X			
FMT_SMR.1			X	X			
FTA_SSL.3		X					
ADV_ARC.1					X		

Table 5: Objective to Requirement Correspondence

8.1.2.1 O.AUDITS

The TOE must provide a means to review and record an audit trail of security-related events, with accurate dates and times.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: Security-relevant events must be defined and auditable by the TOE.

- FAU_SAR.1: The TOE provides the capabilities for the administrator to review the audit records.
- FAU_STG.4: The TOE stops performing auditable events for users when the database table that holds audit records becomes full. An administrator must make space available for audit records before normal operation can continue.

8.1.2.2 O.ACCESS

The TOE will ensure that users gain only authorized access to it and to the resources that it controls

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2, FDP_ACF.1: All users are subject to the Work Flow Access Control Policy for all available operations on forms and folders.
- FIA_UID.1 and FIA_UAU.1: The TOE requires all users to be successfully identified and authenticated before gaining access to TSF mediating actions.
- FTA_SSL.3: The TOE terminates a user's interactive session after an administrator set time period that must be greater than zero, thus minimizing exposure of an unattended terminal.

8.1.2.3 O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: Users that have been assigned the administrator role are considered authorized administrators, all others are simply users.

8.1.2.4 O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MTD.1: Administrators can manage the configuration of user account data and of interactive session timeout values.
- FMT_MSA.3: By default every object is created with the creator as the owner. Subsequently, access can be granted to other users. The administrator can specify alternative values.
- FMT_SMF.1: The TOE provided administrator console interfaces to manage the Work Flow Access Control Policy, to manage user accounts, and to manage inactive session threshold values.
- FMT_SMR.1: Users that have been assigned the administrator role are considered authorized administrators, all others are simply users.

8.1.2.5 O.TOE_PROTECTION

The TOE will be designed to protect itself and its assets from external interference or tampering.

This TOE Security Objective is satisfied by ensuring that:

- ADV_ARC.1 requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed. Without this objective, there could be no assurance that users could not view or modify TSF executables, TSF data, or TSF-protected data. The TOE security functions are invoked and succeed before each function within the TSC is allowed to proceed.

8.1.2.6 O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.1: The TOE offers no TSF-mediated functions, except those available through the MBPM Designer application prior to the login required to deploy a model. The TOE authenticates users using its username/password mechanism.

8.1.2.7 O.USER_IDENTIFICATION

The TOE will uniquely identify users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1: The TOE defines users in terms of security attributes that include user name, password, and role.
- FIA_UID.1: The TOE offers no TSF-mediated functions, except those available through the MBPM Designer application prior to the login required to deploy a model.

8.2 Requirement Dependency Rationale

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

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	none[5]
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.4	FAU_STG.1	none
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2 and FMT_MSA.3
FIA_ATD.1	none	none
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	none	none
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FTA_SSL.3	none	none
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.4 and ADV_TDS.3
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3
ADV_IMP.1	ADV_TDS.3 and ALC_TAT.1	ADV_TDS.3 and ALC_TAT.1
ADV_TDS.3	ADV_FSP.4	ADV_FSP.4
AGD_OPE.1	ADV_FSP.1	ADV_FSP.4
AGD_PRE.1	none	none
ALC_CMC.4	ALC_CMS.1 and ALC_DVS.1 and ALC_LCD.1	ALC_CMS.4 and ALC_DVS.1 and ALC_LCD.1
ALC_CMS.4	none	none
ALC_DEL.1	none	none
ALC_DVS.1	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1

Table 6: Dependency Table

[5] The TOE uses a time stamp provided by its operating environment to place a date and time into audit records.

Because the TOE stores its audit data in a database table that is controlled by a DBMS in the environment, the TOE does not perform the functions to protect the audit trail. Instead the environment (i.e., the DBMS) is responsible for protecting the TOE's audit trail. Therefore, the dependency for FAU_STG.1 that exists on FAU_STG.4 is actually satisfied by the environment and not by the TOE.

8.3 Extended Requirements Rationale

There are no extended requirements in this Security Target.

8.4 TOE Summary Specification Rationale

Each subsection in Section 6.1, the TOE Security Functions, 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 7 demonstrates the relationship between security requirements and security functions.

	Auditing	User data protection	Identification and Authentication	Security Management	Protection of the TSF	TOE access
FAU_GEN.1	X					
FAU_SAR.1	X					
FAU_STG.4	X					
FDP_ACC.2		X				
FDP_ACF.1		X				
FIA_ATD.1			X			
FIA_UAU.1			X			
FIA_UID.1			X			
FMT_MTD.1				X		
FMT_MSA.1				X		
FMT_MSA.3				X		
FMT_SMF.1				X		
FMT_SMR.1				X		
FTA_SSL.3						X
ADV_ARC.1					X	

Table 7: Security Functions vs. Requirements Mapping

8.5 PP Claims Rationale

See Section 7, Protection Profile Claims.