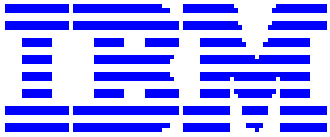

Tivoli Provisioning Manager Security Target

Version:	**1.7**
Status:	**Released**
Last Update:	**2008-12-18**
Classification:	**public**

Trademarks

IBM and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- IBM Websphere Application Server
- IBM AIX
- IBM System p
- IBM System i
- IBM System z
- IBM Tivoli Directory Server

The following terms are trademarks of Sun Microsystems:

- Sun Solaris®
- Java
- J2EE

The following terms are trademarks of Microsoft:

- Windows® 2003 Server Standard
- Windows® 2003 Server Enterprise

Other company, product, and service names may be trademarks or service marks of others.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Revision	Author(s)	Date	Changes to Previous Revision
47	david	2007-07-02	Replying to evaluator comments, etc.
48	david	2007-07-02	Added content to ST overview; updates to Terminology and Abbreviations sections; updated Figure 1; added additional information on Deployment Infrastructure; added JRE to physical boundary; corrected rationale for T.Unauthorized mapping; updated rationale for security objectives; corrected dependency analysis; editorial changes.
53	david	2007-07-03	Additional changes to retro-fit with old .doc version.
59	david	2007-07-05	Editorial changes and modifications in PDF generation.
62	david	2007-07-06	Added detailed description of assurance measures in TSS.

Revision	Author(s)	Date	Changes to Previous Revision
69	david	2007-07-19	Added clarifications on physical boundary in 2.3.1; Clarified that network communication between APDE and TPM server is prohibited in evaluated configuration; added JRE version identification.
76	david	2007-08-17	Clarified P.Accountability; revised SFRs for instance level policy; resolved inconsistencies in rationale for SFRs; editorial changes.
78	david	2007-08-22	Clarification on users in 2.1; clarification in 8.2.4 and F.MGMT.41; added puring of audit data to FMT_SMF.1; editorial changes.
81	david	2007-08-23	Further illustrations on access control throughout document, moved FMT_MSA.1 into environment.
145	amasino	2008-04-08	Correct TOE version to 5.1.1, remove F.AUD.11 and F.MGMT.43, and address Reporting component as security relevant. Minor type corrections.
150	amasino	2008-04-10	Added CD media pack as TOE delivery method.
156	amasino	2008-04-15	Remove password management functions as part of the TOE.
190	amasino	2008-07-15	Rewording of security functionality description and clarifications for role-based access control, based on BSI comments and questions dated 2008-06-26.
203	amasino	2008-08-25	Set accountability as a mandatory feature, add relationship between P.Accountability and OE.Repositories.
206	amasino	2008-09-04	Add Microsoft Active Directory as a valid LDAP server.
215	amasino	2008-10-05	Update with new TOE version.
232	amasino	2008-11-27	Add requirements for Evaluated Configuration. Remove time range criteria for audit search.
236	amasino	2008-12-08	Add patch requirement for IBM Websphere Application Server.
240	amasino	2008-12-18	Correction of patch number in Websphere Application Server

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Tivoli Provisioning Manager Security Target
Version:	1.7
Status:	Released
Date:	2008-12-18
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Certification ID:	BSI-DSZ-CC-0471
Keywords:	Tivoli, TPM, Provisioning, Provisioning Manager

1.2 TOE Identification

The TOE is Tivoli Provisioning Manager (TPM) Version 5.1.1.1 Interim Fix 6

1.3 ST Overview

Tivoli Provisioning Manager (TPM) is an automated resource management solution for corporate and Internet enterprises. TPM allows managing an organization's system life cycle by providing a centralized solution to:

- Discover existing assets (so-called endpoints) in the IT infrastructure.
- Schedule the installation of operating systems and application software on these assets.
- Determine configuration settings on the managed systems and bring them into compliance with centrally managed policies.
- Install software patches and upgrades on managed machines.

In order to limit the ability of performing central management tasks to authorized personnel, TPM provides access control functionality, as well as an auditing mechanism to provide for accountability.

1.4 CC Conformance Claim

This ST is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL3, augmented by ALC_FLR.1.

This ST does not claim conformance to any Protection Profile.

Common Criteria [CC] and Common Evaluation Methodology [CEM] version 2.3 have been taken as the basis for this conformance claim.

1.5 Strength of Function

This ST does not make any claim on strength of function.

1.6 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Automation Package Development Environment	Tool to create workflows and automation packages in the IT environment that then can be imported into the data model.
Data model	A database, containing physical and logical assets that Tivoli Provisioning Manager manages, their relationships, workflows, et al. The data model tracks IT assets, software, systems and their configuration, each asset being represented by a data object.
Data object	A data object describes a managed asset in the data model. This is the virtual representation of an endpoint in the IT environment. Users can manage these objects and are restricted in their access to them by the Instance Level Security Policy implemented by the TOE. A data object follows a pre-defined structure and is represented by an entry in one of the TPM DB's tables.
Endpoint	The system that is the final destination of a management operation, i.e., the remote resources managed with TPM.
Instance	An individual endpoint managed by TPM. Represented in the data model as a Data object.
Instance Level Security Policy	The Instance Level Security Policy is the DAC policy enforced by the TOE, mandating access of users to the individual data objects in the data model.
Provisioning Server	The system hosting the central, security-enforcing parts of the TOE, like the administrative interfaces and deployment engine, including the underlying J2EE application server, operating system and hardware.
TPM DB	The database in the IT environment that the TOE uses to store the data model.
User	Humans or machines interacting with the TOE via the provided user and programmatic interfaces. The term user in this document includes administrators of the TOE unless a specific distinction is made in the text.

1.7 Abbreviations

APDE	Automation Package Development Environment
TPM	Tivoli Provisioning Manager (the TOE)

1.8 References

CC **Common Criteria for Information Technology Security Evaluation**

Version	2.3
Date	August 2005
Location	http://www.commoncriteriaportal.org/files/ccfiles/ccpart1v2.3.pdf
Location	http://www.commoncriteriaportal.org/files/ccfiles/ccpart2v2.3.pdf
Location	http://www.commoncriteriaportal.org/files/ccfiles/ccpart3v2.3.pdf

CEM **Common Methodology for Information Technology Security Evaluation**

Version	2.3
Date	August 2005
Location	http://www.commoncriteriaportal.org/files/ccfiles/cemv2.3.pdf

2 TOE Description

2.1 Introduction

Tivoli Provisioning Manager (TPM) is an automated resource management solution for corporate and Internet enterprises. TPM allows managing an organization’s system life cycle by providing a centralized solution to:

- Discover existing assets (so-called endpoints) in the IT infrastructure.
- Schedule the installation of operating systems and application software on these assets.
- Determine configuration settings on the managed systems and bring them into compliance with centrally managed policies.
- Install software patches and upgrades on managed machines.

The users of the TOE are different types of administrators who either manage the TOE itself or are responsible for using the TOE to manage assets in the TOE's environment. In order to limit the ability of performing central management tasks to authorized personnel, TPM provides access control functionality, as well as an auditing mechanism to provide for accountability.

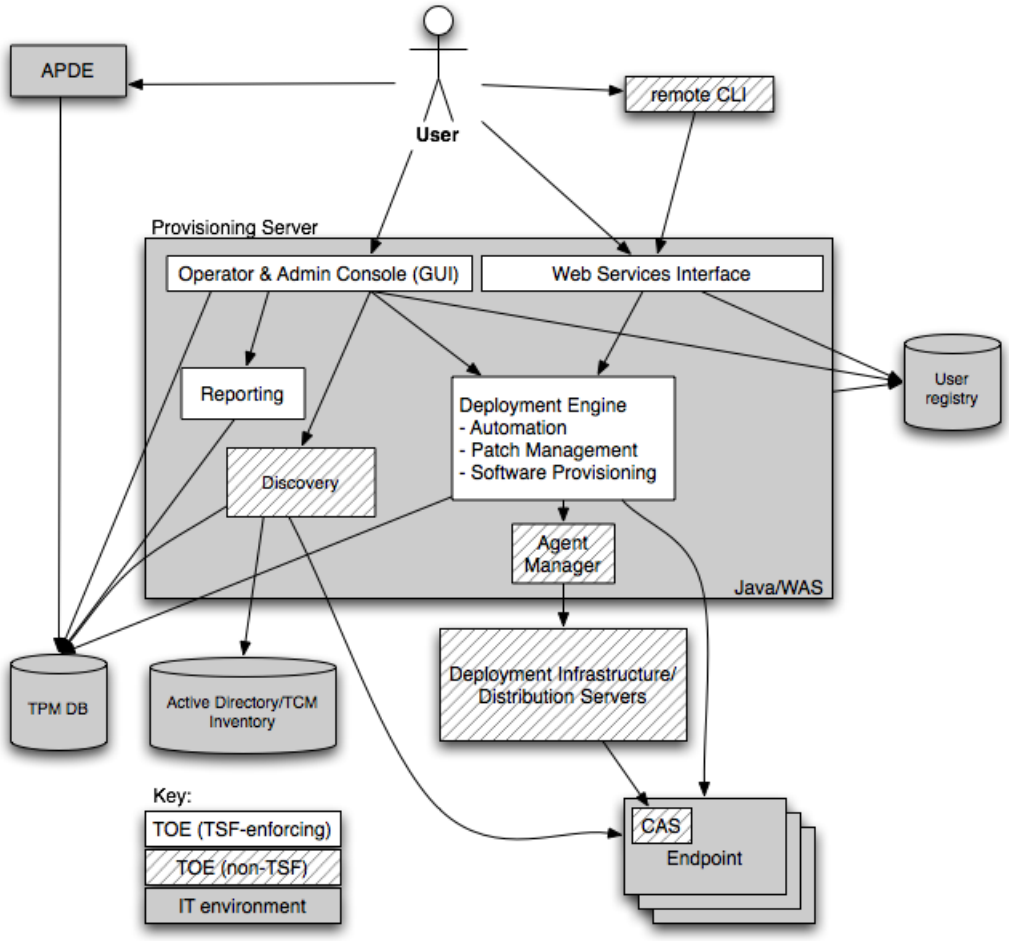

Figure 1: TOE and IT environment - architectural view

2.2 Architecture

2.2.1 The data model

Tivoli Provisioning Manager manages a virtual representation of the physical and logical assets in an enterprise's IT infrastructure in a data model, which is stored in the TPM Database (see Figure 1).

- Each asset is represented by a data object. When changes are made to an asset with Tivoli Provisioning Manager, the data object is updated in the data model. If a change is made outside of Tivoli Provisioning Manager, the external change can be identified by comparing it with the data object in the data model.
- For some assets, the data model stores data about the asset and data about deploying or provisioning the asset separately to provide a range of implementation options. For example, when a software package is added to the Tivoli Provisioning Manager software

catalog, the software package is defined as an installable file. Software definitions can then be created that describe different configuration requirements and configuration options for installing the same software package.

- Templates can be created that define standard configurations. For example, a computer template can be created that includes the routing, software, and storage configuration for a particular application tier. When a computer is added to the application tier, the defined configuration is automatically applied.
- A number of grouping options are available: Individual data objects can be members of administrator-defined groups. Application tiers can be used to identify multiple servers for management of specific applications on these servers. In addition to assigning dedicated servers to an application tier, resource pools can be used to group servers together that are available to one or more associated application tiers upon request for load management.

Tivoli Provisioning Manager records all the software defined in the data model in a software catalog. Software packages are stored in file repositories linked to the software catalog.

2.2.2 Automated provisioning

In order to manage IT assets, TPM provides the concept of automation packages to capture management procedures and execute them in an automated fashion.

Device models, also known as device drivers, are groups of workflows that can be applied to an IT asset. This means that administrators can specify device models for a specific device type that map the generic logical device operations provided by TPM to administrator-defined workflows that contain device type-specific instructions.

The logical device operations are an abstract set of generic operations that TPM's data model operates on. They are not specific to a device type and typically require instantiation for specific device types by means of workflows. Administration tasks targeting an IT asset cataloged in the data model would invoke a logical device operation, causing the deployment engine to schedule the execution of the associated device-type specific workflows on the asset.

Workflows represent the actual implementation of a specific IT process, such as the administrator-defined commands to be executed on a specific platform in order to implement a logical device operation on this device type. Workflows can automate processes from configuring and allocating servers, to installing, configuring, and patching software, and can be either large and complex or can consist of a single command. Workflows can make use of scripts and commands to be run on the target computer, Java methods that define interfaces or protocols for the interaction with the managed resources, and can query or change the data model (such as to reflect the state of a successfully executed command on the device).

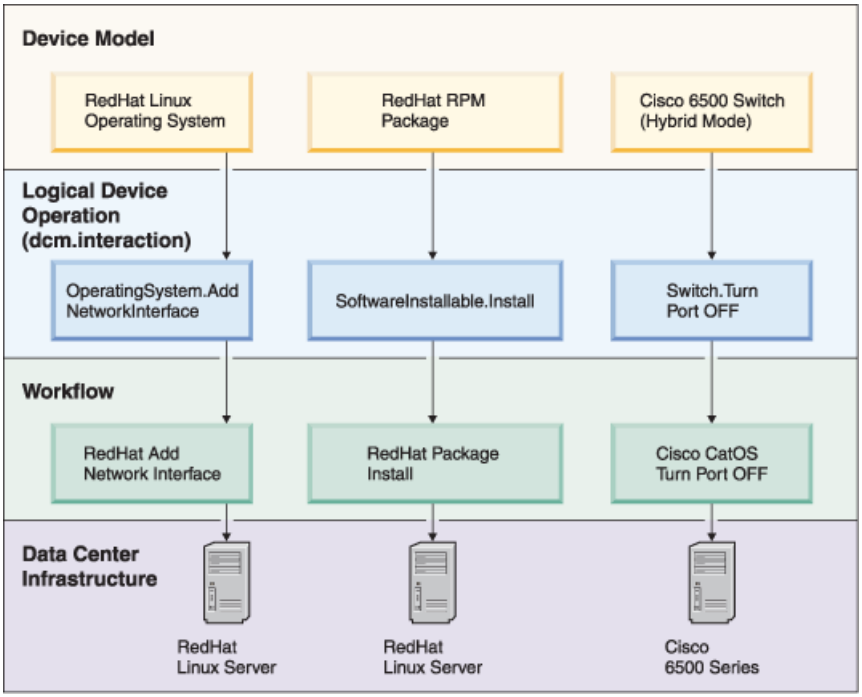

Figure 2: Automation examples

An automation package then is an installation unit that consists of the scripts, workflows, documentation and Java files for a particular device or software package.

2.2.3 Deployment engine

The deployment engine is responsible for executing the workflows that automate the configuration and allocation of IT assets. A workflow can represent either an entire reconfiguration process affecting multiple servers or a single step in a larger reconfiguration process.

The deployment engine converts generic, high-level configuration requests in device operations specified in the data model to corresponding device-specific configuration commands as specified in the workflows. Separate processes and physical servers can be involved in performing each command. Some commands are also delegated to external management systems. For example, when the deployment engine identifies that a workflow transition requires a change to an IP address on a server, the request is converted to commands that are specific to the operating system on the target server, and the commands are sent to the server. The operating system on the server is then responsible for making the actual IP address change.

2.2.4 Deployment infrastructure

TPM integrates the Tivoli Common Agent Services (CAS), a platform that provides a central agent infrastructure that can be shared by multiple distributed management services. On endpoints, a common agent is installed that communicates with the deployment infrastructure and the provisioning server and executes the tasks that have been scheduled on the TPM server. The Tivoli Content Delivery Service (CDS) provides the infrastructure for hosting (and distributing) the actual content: it allows clients to efficiently download large files by establishing an infrastructure of distributed depot servers and offering intelligent download management.

The Tivoli Device Manager Server (DMS) component is used to manage distribution jobs. In this sense, TPM is a resource manager making use of the CAS-provided agent manager to administer the endpoint infrastructure, DMS for the management of deployment jobs, and CDS-provided distribution servers for the actual content delivery.

For patch management, TPM can directly communicate with endpoints via endpoint-provided RPC mechanisms. Other communication means are supported, depending on the operation in question and administrator-specified configuration, such as ssh, scp, telnet, ftp, SMB, http, https, icmp, ldap, ldaps, Tivoli Management Framework and IPX.

The communication protocols and other data for an endpoint, such as its IP address, authentication credentials for the endpoint, etc., are configured by creating a service access point (SAP) entry for an endpoint.

2.2.5 Administration

Tivoli Provisioning Manager offers a web-based graphical user interface (GUI) centralizing all administration and management tasks for the TOE. In addition, a programmatic web services (SOAP) interface exists that can be accessed by users directly or via administrative tools provided with the TOE.

Local command line interfaces (CLI) are provided for administration tasks on the server. Remote CLIs make use of the SOAP interface.

A Java-based design environment for workflows (Automation Package Development Environment - APDE) is provided as part of the product and can be configured to communicate with the TPM database in order to create or alter workflows. Additionally, it is possible to open a network port on the TPM server for communication between APDE and the deployment engine in order to initiate the execution of workflows. APDE is not meant for use in a production environment.

2.3 TOE boundaries

2.3.1 Physical

The TOE is comprised of:

- Tivoli Provisioning Manager Version 5.1.1.1 Interim Fix 6
- Tivoli Provisioning Manager Installation Guide for AIX
- Tivoli Provisioning Manager Installation Guide for Linux
- Tivoli Provisioning Manager Installation Guide for Solaris
- Tivoli Provisioning Manager Installation Guide for Windows
- Tivoli Provisioning Manager Problem Determination Guide
- Tivoli Provisioning Manager Release Notes
- Tivoli Provisioning Manager information center

The TOE is software (and guidance) only and is available via electronic download or as a CD media pack.

The following components can be found in the IT environment:

- A J2EE application server. The runtime environment for the TOE is an IBM Websphere Application Server® version 6.0.2 with refresh pack 2, interim fix pack 11 and patch 5.1.1.1-TIV-TPM-IF00006-LA0001 running on:
  - IBM AIX® 5.2 or 5.3 (IBM System p™)

  - Red Hat Enterprise Linux 4.0 (IBM-compatible PC), 4.0 (IBM System p™ or IBM System i™ with LPAR support)
  - SUSE SLES 9 (IBM-compatible PC), SLES8 (Power PC), SLES9 (IBM System p™ Power5, IBM System i™ with LPAR support, or IBM System z™)
  - Windows® 2003 Server Standard or Enterprise (IBM-compatible PC)
  - Sun Solaris® 9 or 10, 64-bit (Sun SPARC.server)
- Java Runtime Environment (JRE) version 1.4.2.
- CYGWIN 1.5.10 or later **(for Windows systems only)**. CYGWIN provides some UNIX functions to Windows systems, for example, process fork and process exec.
- The user registry, an LDAP server: TPM uses a directory server to store user account data. If TPM is authorized to make changes in the LDAP repository, permissions and groups for the enforcement of the role-based security policy are stored here. TPM supports IBM Tivoli Directory Server® 6.0 Fix Pack 1 (on AIX, Linux, Solaris, or Windows) and Microsoft Active Directory® as LDAP servers in the environment.
- The TPM database, a relational database: The TPM database holds the TOE's data model, but is not in itself part of the TOE. If TPM has only read access to the LDAP server, permissions and groups for role-based security are stored in this database. TPM supports IBM DB2® (DB2 Universal Database Enterprise Server Edition 8.2, Fix Pack 11 on AIX, Linux, Solaris, or Windows) as database management system in the environment.
- Endpoints. The assets managed with the TOE are part of the IT environment.

Since the TOE's runtime environment is the Java Runtime Environment and additional services implemented in Java, there is no direct dependency of the TOE on the operating system (or, the hardware) that the runtime environment runs on. The Java-based runtime environment provides a complete abstraction layer.

2.3.2 Logical

TPM, in conjunction with its runtime environment, provides a number of security functions, as discussed in section 2.4 . Not all functionality provided by TPM is considered security functionality in terms of this evaluation, and some of the components delivered with TPM do not have any claims attributed to them that would have been examined during this evaluation:

- Discovery functionality does not contribute to TPM's security functionality, and consequently the evaluation does not make any claims regarding this function.
- Remote command line interfaces (CLI) provided for TOE management use the TOE's web services interface. Since the web services interface is considered the security-enforcing boundary for the TOE and administrators are free to use other tools than the TPM-provided ones, the CLI is not considered a contributor to the evaluated security functionality.
- The Eclipse-based workflow design environment (APDE) is not meant to be used in the production environment in the evaluated configuration.
- The deployment infrastructure does not contribute to the TOE's security policies. While affected by the management actions that are controlled by the evaluated security functionality, the evaluation currently does not make any claims on how these management actions are implemented by the deployment infrastructure.

2.3.3 Evaluated configuration

The following configuration specifics apply to the evaluated configuration of the TOE:

- Access control must be enabled.
- The Cascading rules feature must not be enabled.
- The `tioadmin` user must use the Korn or Bash shell.
- Auditing must be enabled.

2.4 Security functionality

2.4.1 TOE security functions

The TOE enforces an instance level security policy in order to implement access control for the management of individual assets.

2.4.1.1 Instance level security

TPM offers a pre-defined list of instance permissions that can be granted to users in order to protect individual data objects (assets) from unauthorized access, such as “OperatingSystem.AddUser” authorizing the addition of users to an operating system asset in the data model or “EndpointTaskRun” allowing to run a task on an endpoint. These individual permissions can be combined into permission groups.

Administrators can define access groups that comprise a set of data objects. Access groups can contain individual data objects, groups, resource pools and application tiers. Permission groups can then be assigned to access groups, and eventually users can be associated with these permission groups within an access group, defining which permissions the user has on the associated objects. The result is a tuple (access group:permission group:user).

The IT environment (via the enforcement of role-based security, see section 2.4.2.2) determines who is authorized to edit access and permission groups. By default, the authorized users are those assigned to the System Administrator role and superusers, who are not under role-based access control.

Figure 3 illustrates the above describe relationships between users, permissions, permission groups, access groups and data objects.

When a user requests an operation on a specific data object via one of the administrative interfaces provided by the TOE, the access is granted only if one of the specified permissions allow the requested operation type on the specific data object.

Instance level security can also be applied to workflows when configured by administrators. If a workflow definition requires a specific permission for a workflow parameter, TPM will verify that a user has the correct permission for the object that she or he assigned as value for the protected parameter when initiating the workflow.

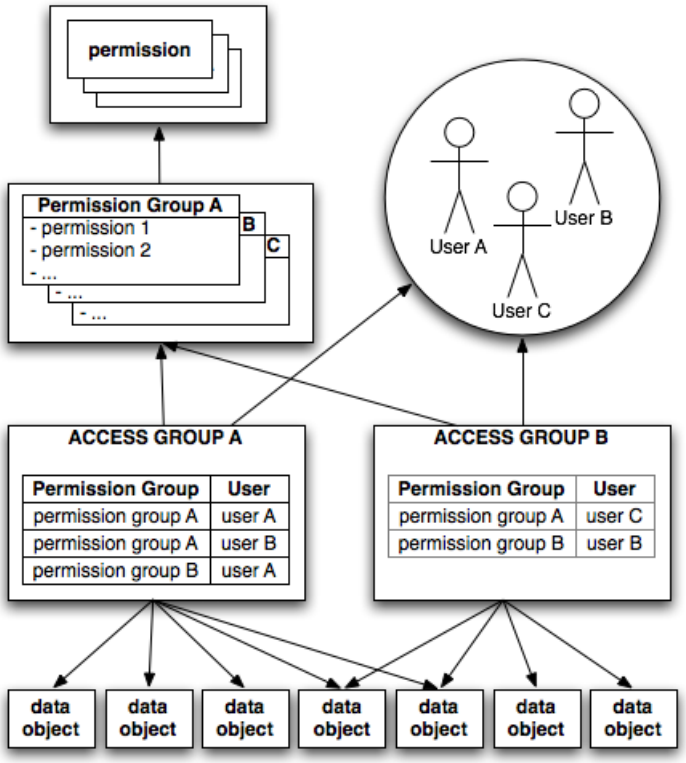

Figure 3: Instance Level Secuity Policy modeling

Users are assigned a default access group upon creation. When a user creates new data objects through the GUI, these objects are added automatically to this access group.

Users who have the superuser attribute set in their account are exempt from any access control enforcement.

TPM also provides a mechanism called "cascading rules" which allows an object to share access control attributes with other objects. Cascading rules mainly applies to the application tier and resource pools. When enabled, the access groups associated with resource pools and application tiers inherit the permissions of all of the access groups as they pertain to these devices. The cascading access control mechanism is not supported in the evaluated configuration.

2.4.1.2 Auditing

Audit records for certain security-relevant events can be generated by TPM. This includes:

- Auditing of changes to the system configuration: TPM generates a record for each change to any of the tables in the TPM DB, including all objects in the data model.
- User management and access control management: TPM creates records for the management of users and properties for the access control mechanisms that are stored in the user registry.
- User logon/logout: This is recorded by TPM, albeit logon/logout is handled by the runtime environment and not the TOE.

The GUI provides several views that administrators can use to search for and review audit records by means of the "Reporting" component; as well as to delete audit records that are older than a certain date.

Auditing can be turned on or off on a global basis by Superusers or administrators with the System Administrator role; however, for the evaluated configuration this feature must be enabled at all times.

2.4.1.3 Security management

TPM provides management capabilities for its security functions and some of the environment-provided security functionality via its user interfaces.

- user management (add, change properties, delete)
- management of access control: security roles, access groups, permission groups, etc.
- enabling/disabling of auditing

2.4.1.4 Out of scope

The TPM guidance addresses additional security concepts that a consumer can employ when using TPM to manage their infrastructure, such as role-based access control and transaction security. However, from a security evaluation point of view, these services are provided by the IT environment rather than by the TOE: TPM is a J2EE application running on an application server. As is the nature of such applications, they make use of security services provided by their runtime environment to protect themselves and their assets from interference. For example, TPM relies on the authentication service provided by its runtime environment instead of implementing its own authentication mechanisms. While from a user's point of view, users are authenticating to TPM, it is in fact the underlying application server that performs the authentication.

As mentioned in section 2.4.1.1 the cascading access control mechanism is not supported in the evaluated configuration.

The next section discusses the security functions that are provided by TPM's IT environment.

2.4.2 IT environment support

The TOE runs in a J2EE environment, i.e., on a Websphere Application Server (WAS). This application server does not only provide an abstraction layer that makes the TOE independent from support of any underlying operating system or hardware, but also implements a number of security functions that the TOE directly relies on for providing secure operations and the safeguarding of TSF and user data.

2.4.2.1 Authentication

TPM makes use of the authentication service provided by the application server to authenticate its users. While TPM-specific code exists to customize the retrieval of stored authentication credentials, this technically means that the underlying IT environment provides the authentication security functionality.

2.4.2.2 Role-based security for user interfaces

The user interfaces (UIs) exposed by the TOE are subject to the enforcement of the role-based security policy. Pre-defined permissions are granted to groups of users and determine whether a user is authorized to access individual UI pages or features on these pages.

Mapping J2EE roles to the administrator-defined permission/group relationships and using the WAS-provided security framework to enforce these implement role-based security.

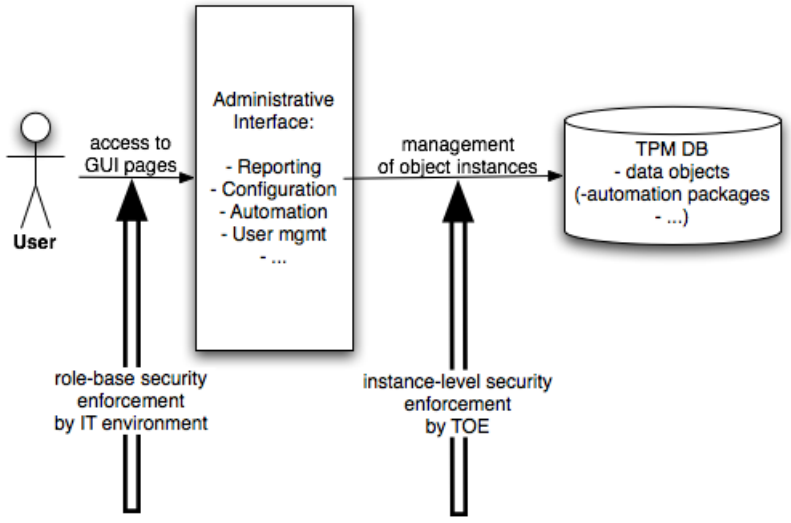

Figure 4: Schematic relationship between Role-based and Instance level security policies

Figure 4 illustrates again the relationship between the two access control policies relevant for the TOE:

- Users access one of the administrative interfaces provided by the TOE. The IT environment implements the Role-based security policy to decide whether a user is granted access to specific management functions (e.g., by restricting access to specific pages of the GUI). Attributes for the security policy are the user ID (stored in the user registry) and the security roles associated with a user ID that grant specific access permissions (either stored in the user registry or TPM database).
- If an administrative interface is being used to maintain assets (i.e., data objects in the TPM database), or if a workflow is invoked that accesses such data objects, the TOE uses the instance level security policy to decide whether a user can perform operations on the requested data objects. Attributes for the security policy are the user ID (stored in the user registry) and the permission groups and access groups associated with the individual user (stored in the TPM database).

2.4.2.3 Transaction security

On the server side, the application server provides the implementation of protocols (such as, SSL/TLS or OpenSSH) that are used to protect communication sessions between the TOE parts as well as the TOE and its environment, namely network connections between the server, the deployment infrastructure and the endpoints.

2.5 Security Policy Model

The security policy for TPM is defined by the security functional requirements in section 5.2 and refined into a security policy model by the TSF in section 6.1. The following is a list of the subjects and objects participating in the policy.

Subjects:

- users

Objects:

- data objects

TSF data:

- user accounts, including the following security attributes:
  - user ID
  - default access group
  - password
  - superuser attribute
  - association with access groups and permission groups
- Instance Level Security Policy attributes, including:
  - access groups
  - permission groups
- workflow definitions (specifically, the assignment of required permissions to workflow parameters)
- audit records

User data:

- user data is mainly constituted by attributes of the data objects that are not relevant for the TSP enforcements, such as parameters for remote management of the instance represented by a data object

3 Security Problem Definition

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE are essentially comprised of the data model that the TOE maintains to store configuration and other data about the IT assets in the TOE-managed infrastructure. The information in the data model determines the TOE's actions towards applying specific configurations and software to the remotely managed IT assets, and being able to manipulate this data and – as a result – to influence the setup of the assets could lead to compromise of integrity, confidentiality and availability in an organization's infrastructure.

The **threat agents** having an interest in manipulating the data model can be categorized as either:

- Unauthorized individuals ("attackers") which are unknown to the TOE and its runtime environment.
- Authorized users of the TOE (i.e., administrators) who try to manipulate data that they are not authorized to access.

Threat agents originate from a well managed user community within an organizations internal network. Hence, only inadvertent or casual attempts to breach system security are expected from this community.

TOE administrators, including administrators of the TOE environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

3.1.1 Threats countered by the TOE

T.Unauthorized	A user or attacker gains access to TSF or user data without proper authorization.

3.1.2 Threats countered by the TOE environment

TE.UnknownUsers	An attacker is able to gain logical (network) access to the TOE without being properly authenticated.
TE.GUIAccess	A user or attacker is able to gain access to administrative interfaces of the TOE without proper authorization.
TE.Repositories	An attacker is able to gain access to TSF or user data stored in the IT environment.
TE.Remote	An attacker is able to compromise data exchanged between the Provisioning Server and systems in the deployment infrastructure.
TE.System	An attacker is able to circumvent TSP enforcement functions by penetrating or manipulating the runtime environment of the TOE.

3.2 Assumptions

3.2.1 Intended usage of the TOE

A.Configuration It is assumed that the TOE is configured and operated in its evaluated configuration as defined in this Security Target and the TOE guidance.

3.2.2 Environment of use of the TOE

3.2.2.1 Physical

A.Physical It is assumed that the machine(s) providing the runtime environment for the TOE are protected against unauthorized physical access and modification.

3.2.2.2 Personnel

A.Administrators The administrators of the TOE, of the TOE's underlying systems, and of the systems in the TOE's IT environment who are involved in safeguarding TSF data or providing functionality that the TOE depends on are assumed not to be careless, willfully negligent, or hostile. They will follow and abide by the instructions provided in the administrator guidance that is part of the TOE. They are well trained to securely and trustworthy administer all aspects of the TOE operation in accordance with this Security Target.

3.2.2.3 Connectivity

A.Runtime The machines providing the runtime environment for the Provisioning Server are assumed to be used solely for this purpose and not to run other application software except as required for the support of the TOE and for the management and maintenance of the underlying system and hardware. Especially it is assumed that the underlying systems are configured in a way that prevents unauthorized access to security functions provided by or protected by the runtime environment either locally or via any network based connections.

3.3 Organizational Security Policies

P.Accountability Administrators shall be accountable for security-relevant configuration changes.

4 Security Objectives

4.1 Objectives for the TOE

O.Audit The TOE shall provide accounting information for security-relevant configuration changes to the TOE.

O.ObjectAccess The TOE shall enforce an Instance Level Security policy in order to allow administrators to restrict access to managed objects to authorized users.

4.2 Objectives for the IT Environment

OE.Authentication The runtime environment for the TOE shall authenticate TOE users and provide user identities to the TOE for the use in Instance Level Security policy enforcement.

OE.GUIAccess The runtime environment for the TOE shall restrict access to GUI interfaces for TOE administration to authorized users.

OE.TimeSource The runtime environment for the Provisioning Server shall provide a reliable time source for the TOE's use.

4.3 Objectives for the Non-IT Environment

OE.Administrators Those responsible for the operation of the TOE must ensure that administrators are not careless, willfully negligent, or hostile, and that they are well trained and will follow the provided administrator guidance to operate the TOE in its evaluated configuration.

OE.Remote Those responsible for the operation of the TOE must ensure that communication of the Provisioning Server with instances in the deployment infrastructure is protected against loss of confidentiality and integrity.

OE.Repositories Those responsible for the operation of the TOE must ensure that user and TSF data stored in repositories in the IT environment is only available to the TOE.

OE.Runtime Those responsible for the operation of the TOE must ensure that the systems hosting the Provisioning Server are used solely for this purpose and configured in a way that prevents unauthorized access to the TOE.
This includes preventive measures to ensure that all systems that are hosting parts of the TOE are protected against unauthorized physical access and network-based attacks.

5 Security Requirements

5.1 Extended Components Definition

This ST defines one extended component to reflect the fact that the TOE does not provide for auditing of the startup and shutdown of its audit functionality as required in the CC Part 2-defined FAU_GEN.1.1 a). The extended component is therefore derived from FAU_GEN.1 by removing this specific requirement and belong into the same family.

5.1.1 FAU_GEN_TPM.1 - Audit data generation for TPM

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

Management:

There are no management activities foreseen.

Audit:

There are no auditable activities foreseen.

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

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

b) [assignment: **other specifically defined auditable event**]

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: **other audit relevant information**]

5.2 TOE Security Functional Requirements

The following table shows the Security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU Security audit	FAU_GEN_TPM.1 Audit data generation for TPM		ECD	No	No	Yes	Yes
	FAU_GEN.2 User identity association		CC Part 2	No	No	No	No
	FAU_SAR.1 Audit review		CC Part 2	No	No	Yes	No
	FAU_SAR.3 Selectable audit review		CC Part 2	No	No	Yes	Yes

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FDP User data protection	FDP_ACC.1 Subset access control		CC Part 2	No	No	Yes	No
	FDP_ACF.1 Security attribute based access control		CC Part 2	No	No	Yes	No
FMT Security management	FMT_MSA.3 Static attribute initialisation		CC Part 2	No	No	Yes	Yes
	FMT_SMF.1 Specification of Management Functions		CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security roles		CC Part 2	No	No	Yes	No

Table 1: Security functional requirements for the TOE

5.2.1 Security audit (FAU)

5.2.1.1 Audit data generation for TPM (FAU_GEN_TPM.1)

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

a) All auditable events for the **not specified** level of audit; and

b) **changes to the data model, user and role management**

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **no other audit relevant information**

5.2.1.2 User identity association (FAU_GEN.2)

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

5.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **authorized users** with the capability to read **all available audit information** from the audit records.

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

Application note:

*The term **authorized users** in this context applies to any user that has assigned the ReportAuditView permission, or belong to one of the following pre-defined roles (which have this permission granted by default): System administrator, Inventory specialist, Storage administrator, Network administrator, Configuration administrator and Configuration operator.*

5.2.1.4 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform **searches** of audit data based on **user ID** .

5.2.2 User data protection (FDP)

5.2.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **Instance Level Security Policy** on

- **subjects: users;**
- **objects: data objects and workflow attributes;**
- **operations: access requests from subjects to objects**

.

5.2.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **Instance Level Security Policy** to objects based on the following:

- **subjects: users;**
- **objects: data objects;**
- **security attributes:**
  - **access groups**
  - **permission groups**
  - **permissions**
  - **user identities**

.

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

- **Members of access groups are members of the access groups' parents.**
- **Members of permission groups are members of the permission groups' parents.**
- **For management requests to protected data objects, a user is granted the requested access if:**
  - **An access group contains both a) the requested object and b) a permission group that contains the permission required to perform the requested operation and that is associated with the user's user identity.**

- **When assigning protected data objects as values to workflow parameters, the requested workflow is only executed if, for each of the parameters that has instance level checking enabled:**
  - **An access group contains both a) the object which has been assigned as value to the respective parameter and b) a permission group that contains the permission required in the workflow's definition of the parameter and that is associated with the user's user identity.**

.

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

- **Access to all objects is granted to superusers.**

.

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

5.2.3 Security management (FMT)

5.2.3.1 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **Instance Level Security Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

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

5.2.3.2 Specification of Management Functions (FMT_SMF.1)

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

- **en- and disabling of auditing**
- **purging of audit data**
- **user management**
- **management of access groups and permission groups for enforcement of the Instance Level Security Policy**

.

5.2.3.3 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles

- **pre-defined roles**
- **roles defined by an authorized administrator**

.

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

Application note:

*The term **authorized administrator** in this context applies to any user that has assigned the SystemManagementEdit and SystemManagementView permissions, belongs to the System administrator pre-defined role (which has these permissions granted by default), or is a superuser. The superuser is not considered a role as the IT environment does not enforce role-based access control to users with this attribute on.*

Application note: *The TOE provides the following pre-defined roles: System Administrator, Software Operator, Change Approver, Inventory Specialist, Storage Administrator, Network Administrator, Configuration Administrator, Configuration Operator.*

5.3 Security Requirements for the IT Environment

The runtime environment for the TOE is WebSphere Application Server (WAS). The following security functional requirements (SFRs) have been intentionally defined so that they can be mapped onto the SFRs in the Security Target for WAS. Generalizations (in cases where the WAS ST encompasses more functionality than required to support the TOE or is more specific as to mechanisms that is uses than is relevant for the TOE) and refinements (in cases where the WAS ST expects the applications that WAS supports to do so) have been made, which shall not be construed as introducing inconsistencies between the two Security Targets. At the time of development of this Security Target, the current evaluated configuration of WAS supports all of the following SFRs (either directly or through requirements on its IT environment) with the exception of FPT_STM.1-e.

The following table shows the Security functional requirements for the IT environment, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FDP User data protection	FDP_ACC.1-e Subset access control	FDP_ACC.1	CC Part 2	No	Yes	Yes	No
	FDP_ACF.1-e Security attribute based access control	FDP_ACF.1	CC Part 2	No	Yes	Yes	No
FIA Identification and authentication	FIA_UAU.1-e Timing of authentication	FIA_UAU.1	CC Part 2	No	Yes	Yes	No
	FIA_UID.1-e Timing of identification	FIA_UID.1	CC Part 2	No	Yes	Yes	No
FMT Security management	FMT_MSA.1-e Management of security attributes	FMT_MSA.1	CC Part 2	No	Yes	Yes	Yes
	FMT_MSA.3-e Static attribute initialisation	FMT_MSA.3	CC Part 2	No	Yes	Yes	Yes
	FMT_SMF.1-e Specification of Management Functions	FMT_SMF.1	CC Part 2	No	Yes	Yes	No
	FMT_SMR.1-e Security roles	FMT_SMR.1	CC Part 2	No	Yes	Yes	No

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FPT Protection of the TSF	FPT_STM.1-e Reliable time stamps	FPT_STM.1	CC Part 2	No	Yes	No	No

Table 2: Security functional requirements for the IT environment

5.3.1 User data protection (FDP)

5.3.1.1 Subset access control (FDP_ACC.1-e)

FDP_ACC.1.1-e The ~~TSF~~*IT environment* shall enforce the **Role-based Security Policy** on

- **subjects: remote callers;**
- **objects: protected methods of the TPM application;**
- **operations: access requests from remote callers to TPM protected methods.**

.

5.3.1.2 Security attribute based access control (FDP_ACF.1-e)

FDP_ACF.1.1-e The ~~TSF~~ *IT environment* shall enforce the **Role-based Security Policy** to objects based on the following:

- **subjects: remote caller;**
- **objects: protected methods of TPM application;**
- **operations: access requests to administrative interfaces of the TPM application;**
- **security attributes: user ID identifying the subject, access permissions related to specific interfaces, TPM security roles representing a list of access permissions and being associated with the user ID requesting access to the object**

.

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

- **The user ID of the caller is mapped to a TPM-specific security role; or**
- **A group ID of the caller is mapped to a TPM-specific security role; and**
- **The TPM-specific security role has permission to access the protected resources.**

.

FDP_ACF.1.3-e The ~~TSF~~ *IT environment* shall explicitly authorise access of subjects to objects based on the following additional rules: **no additional rules** .

FDP_ACF.1.4-e The ~~TSF~~ *IT environment* shall explicitly deny access of subjects to objects based on the **no additional rules** .

5.3.2 Identification and authentication (FIA)

5.3.2.1 Timing of authentication (FIA_UAU.1-e)

FIA_UAU.1.1-e The ~~TSF~~ *IT environment* shall allow **non-TSF mediated actions** on behalf of the user to be performed before the user is authenticated.

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

5.3.2.2 Timing of identification (FIA_UID.1-e)

FIA_UID.1.1-e The ~~TSF~~*IT environment* shall allow **non-TSF mediated actions** on behalf of the user to be performed before the user is identified.

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

5.3.3 Security management (FMT)

5.3.3.1 Management of security attributes (FMT_MSA.1-e)

FMT_MSA.1.1-e The ~~TSF~~*IT environment* shall enforce the **Role-based Security Policy** to restrict the ability to **delete , write** the security attributes

- **Mappings of user/group IDs to pre-defined roles and roles defined by authorized administrators,**
- **access groups, permission groups for the TSF-enforced Instance Level Security Policy**

to **authorized administrators** .

5.3.3.2 Static attribute initialisation (FMT_MSA.3-e)

FMT_MSA.3.1-e The ~~TSF~~*IT environment* shall enforce the **Role-based Security Policy and other runtime environment-defined policies** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2-e The ~~TSF~~*IT environment* shall allow the **authorized administrators** to specify alternative initial values to override the default values when an object or information is created.

5.3.3.3 Specification of Management Functions (FMT_SMF.1-e)

FMT_SMF.1.1-e The ~~TSF~~*IT environment* shall be capable of performing the following security management functions:

- **Management of Role-based Security Policy attributes.**
- **User and role management.**
- **Other runtime environment-defined security management functions.**

.

5.3.3.4 Security roles (FMT_SMR.1-e)

FMT_SMR.1.1-e The ~~TSF~~*IT environment* shall maintain the roles

- **pre-defined roles**
- **roles defined by an authorized administrator**

.

FMT_SMR.1.2-e The ~~TSF~~*IT environment* shall be able to associate users with roles.

5.3.4 Protection of the TSF (FPT)

5.3.4.1 Reliable time stamps (FPT_STM.1-e)

FPT_STM.1.1-e The ~~TSF~~*IT environment* shall be able to provide reliable time stamps for its own use.

5.4 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 3 components, augmented by ALC_FLR.1, as specified in [CC] part 3. No operations are applied to the assurance components.

6 TOE Summary Specification

6.1 TOE Security Functions

The individual TSF are described by providing a general overview/introduction as necessary for the reader's understanding, and eventually by listing numbered claims that specify testable security functionality or properties. For example, testable claims for the security function F.MGMT are identified as F.MGMT.01, F.MGMT.02, etc.

6.1.1 Instance Level Security (F.DAC)

The TOE enforces the Instance Level Security DAC policy for administrators using the TOE interfaces to manage data objects and to execute workflows.

The permissions that can be assigned to permission groups are hard-coded in the TOE:

F.DAC.01 The individual permissions that can be a member of permission groups and that can be assigned to workflow parameters are:

- Application.Deploy
- Application.Undeploy
- BootServer.CaptureBackupImage
- BootServer.CaptureImage
- BootServer.DeleteImage
- BootServer.InstallGoldenMasterImage
- BootServer.InstallImage
- BootServer.InstallScriptedImage
- BootServer.ReplicateImage
- BootServer.RestoreBackupImage
- Cluster.AddRepairedServer
- Cluster.AddServer
- Cluster.RemoveServer
- ClusterDomain.AddNode
- ClusterDomain.Config
- ClusterDomain.CreateResource
- ClusterDomain.CreateResourceGroup
- ClusterDomain.Remove
- ClusterDomain.RemoveNode
- ClusterDomain.Start
- ClusterDomain.StartNode
- ClusterDomain.Stop
- ClusterDomain.StopNode
- ClusterResource.CreateDependency
- ClusterResource.Start
- ClusterResource.Stop
- ClusterResource.Update
- DCM.Update

- DCM.View
- Device.CreateServiceAccessPoint
- Device.Discover
- Device.DiscoverDrift
- Device.ExecuteCommand
- Device.GetAttribute
- Device.HardwareReboot
- Device.Initialize
- Device.ManagerSoftware
- Device.Ping
- Device.PowerOff
- Device.PowerOn
- Device.Reboot
- Device.Rebuild
- Device.RemoveServiceAccessPoint
- Device.SetAttribute
- Discovery.CheckStatus
- Discovery.Discover
- Discovery.Drift
- Discovery.OnDevice
- Discovery.ServerUpdateDcm
- Discovery.Start
- Discovery.Stop
- Discovery.Update
- Discovery.UpdateDcm
- EndPointTask.Run
- FileRepository.GetFile
- FileRepository.PutFile
- FileRepository.RemoveFile
- Firewall.AddACL
- Firewall.DisableACL
- Firewall.EnableACL
- Firewall.RemoveACL
- HostPlatform.AllocateVirtualResource
- HostPlatform.CreateVirtualServer
- HostPlatform.DeallocateVirtualResource
- HostPlatform.DestroyVirtualServer
- HostPlatform.ExpandResourceAllocation
- HostPlatform.ReduceResourceAllocation
- IPSystem.AddIPAddress
- IPSystem.AddNetworkInterface
- IPSystem.ApplyRoutingTable

- IPSystem.RemoveIPAddress
- IPSystem.RemoveNetworkInterface
- LoadBalancer.AddRealIPToVirtualIP
- LoadBalancer.CreateVirtualIP
- LoadBalancer.RemoveRealIPFromVirtualIP
- LoadBalancer.RemoveVirtualIP
- MonitoringApplication.ConfigMonitoring
- MonitoringApplication.RemoveMonitoring
- MonitoringApplication.StartMonitoring
- MonitoringApplication.StopMonitoring
- OperatingSystem.AddGroup
- OperatingSystem.AddNetworkInterface
- OperatingSystem.AddUser
- OperatingSystem.AddUserToGroup
- OperatingSystem.ApplyRoutingTable
- OperatingSystem.CaptureUserProfile
- OperatingSystem.CreateDASDPhysicalVolu
- OperatingSystem.CreateSANPhysicalVolum
- OperatingSystem.RemoveGroup
- OperatingSystem.RemoveNetworkInterface
- OperatingSystem.RemovePhysicalVolume
- OperatingSystem.RemoveUser
- OperatingSystem.RemoveUserFromGroup
- OperatingSystem.RemoveUserProfile
- OperatingSystem.RestoreUserProfile
-
- PowerUnit.CycleOutlet
- PowerUnit.TurnOutletOFF
- PowerUnit.TurnOutletON
- Router.CreateRoute
- Router.RemoveRoute
- SANFabric.AddZoneMembers
- SANFabric.CreateZone
- SANFabric.RemoveZone
- SANFabric.RemoveZoneMembers
- ServiceAccessPoint
- Software.CheckStatus
- Software.Install
- Software.Start
- Software.Stop
- Software.Uninstall
- SoftwareDistributionApplication.RegisterSoftwarePackage

- SoftwareDistributionApplication.UnregisterSoftwarePackage
- SparePool.CleanupServer
- SparePool.InitializeServer
- StorageManager.Change
- StoragePool.CreateStorageVolume
- StoragePool.GetStorageVolumes
- StoragePool.RemoveStorageVolume
- StorageSubsystem.CreateStorageVolume
- StorageSubsystem.GetStorageVolumes
- StorageSubsystem.LunMapping
- StorageSubsystem.LunMasking
- StorageSubsystem.LunUnmapping
- StorageSubsystem.LunUnmasking
- StorageSubsystem.MapVolumeArrayToFA
- StorageSubsystem.MapVolumeToFA
- StorageSubsystem.RemoveStorageVolume
- StorageSubsystem.UnmapVolumeArrayFromFA
- StorageSubsystem.UnmapVolumeFromFA
- Switch.CarryVLANThroughSwitchEndpoint
- Switch.ChangeSwitchEndpointAttributes
- Switch.ClearVLANFromSwitchEndpoint
- Switch.CreateVLAN
- Switch.MovePort
- Switch.RemoveVLAN
- Switch.TurnPortOFF
- Switch.TurnPortON

Access requests are evaluated as follows:

F.DAC.10 The TOE uses the user ID provided by the IT environment for the evaluation of access requests.

F.DAC.11 Data model members of an access group's children are considered members of the access group when evaluating access requests.

F.DAC.12 Permission members of a permission group's children are considered members of the permission group when evaluating access requests.

F.DAC.13 The access request of a user to a data object in the data model is granted if the conditions under 1. or 2. are true:

1. The object is a member of an access group that the user is associated with by means of an access group:permission group:user tuple.
2. The user is a superuser.

F.DAC.14 A workflow execution requested by a user is granted only if for each workflow parameter that requires a permission the conditions under 1. or 2. are true:

1. The object that is passed as value for the parameter is a member of an access group that the user is associated with by means of an access group:permission group:user tuple.

2. The user is a superuser.

As a result of evaluating the access permissions for a request, the following TOE behavior occurs:

F.DAC.20 Objects that are not assigned to any access group are only visible to the superuser.

F.DAC.21 If a user tries to access a specific object in a fashion that he is not authorized to, the TOE will return an error message.

F.DAC.22 If a user performs a general query on the data model (e.g., selects a list of objects for display in the GUI), the TOE will only return those objects that the user has access to and will not issue an error message.

6.1.2 Auditing (F.AUD)

The TOE generates audit records for certain transactions:

F.AUD.01 Auditing is performed for each database table in the TPM DB holding the data model. The TOE will generate an audit record whenever an entry in a table is being modified.

F.AUD.02 Audit records will be generated for user management actions, including adding/removing users and altering their access control settings (security role and access group memberships).

Facilities to search and review audit records are offered by the TOE's GUI:

F.AUD.10 Administrators can search for audit records pertaining to changes to individual tables in the TPM DB (i.e., the data model) based on table name and user name. The search results contain the time, user name, operation performed on the database table, object ID, and object-dependent properties.

F.AUD.12 Administrators can search for audit records pertaining to role management by user name originating the action. The search results contain the time, originating user name, operation type, operation ID, the role affected, and the permissions associated with the role after the action was completed.

F.AUD.13 Administrators can search for audit records pertaining to user management by user name originating the action. The search results contain the time, originating user name, operation type, operation ID, target user name, and the security roles that the target user had after the operation was performed.

F.AUD.14 Administrators can generate administrator-defined or pre-defined audit reports based on views defined in the underlying data model.

6.1.3 Management (F.MGMT)

The TOE offers a web-based GUI for management of the TSF. The management options offered by the GUI depend on the security roles that have been defined for the user – note that this is part of the Role-Based Security provided by the IT environment. In principle, the GUI is organized into several menus and sub-menus to manage deployment tasks and activities, software packages, the inventory (i.e., endpoints in the data model), applications (the combination of specific software configurations on specific servers), reporting (including auditing reports), the TOE itself (system management) and automation packages.

It should be noted that the authorization of administrators to perform any of the following management functions is performed by the IT environment as part of the Role-Based Security policy (the administrator must have the “System Administrator” role).

The functional aspects that are related to management of F.MGMT are the following:

F.MGMT.01 Superusers or administrators with the System administrator role can en- and disable auditing.

F.MGMT.02 Superusers or administrators with the System administrator role can purge (delete) all audit records from the TPM DB that occurred before a specific date and time.

To support the application of the Instance Level Security policy to workflows, the TOE provides the following management functionality:

F.MGMT.10 Administrators can define permission requirements for individual workflow parameters within workflow definitions.

Access group management for the enforcement of the Instance Level Security policy is offered as follows:

F.MGMT.20 Administrators can create, modify and delete access groups. The TOE enforces unique identifiers for access groups.

F.MGMT.21 Administrators can (dis-)associate individual data objects, groups, resource pools, application tiers and report definitions with access groups.

F.MGMT.22 Objects can be a member of none, one or several access groups.

F.MGMT.23 An access group can only have no or one parent access group, but no, one, or multiple access group children. Administrators can change the association of existing groups with a parent group.

F.MGMT.24 Administrators can associate access groups with (permission group:user) tuples. When managing access groups, the administrator is presented with the permission groups that have been associated with the access group, and for each permission group with the users that are authorized to execute these permissions for the objects associated with the access group.

F.MGMT.25 Upon creation, the administrator has to associate the new user with a default access group.

F.MGMT.26 When creating a new data object, the default access group of the user creating the object is assigned to the object automatically.

Management functionality for permission groups for the Instance Level Security policy is provided as follows:

F.MGMT.30 Administrators can create, modify and delete permission groups. The TOE enforces unique identifiers for permission groups.

F.MGMT.31 Administrators can (dis-)associate individual permissions with permission groups.

F.MGMT.32 A permission can be a member of none, one or several permission groups.

F.MGMT.33 A permission group can only have no or one parent permission group, but no, one, or multiple permission group children (“nested groups”). Nested groups can only be assigned to a parent group upon their creation. It is not possible to re-associate a group with another parent group or dis-associate a group with its parent without deleting it.

User management is provided as follows:

F.MGMT.40 Administrators can add, modify and delete users with unique user IDs.

F.MGMT.41 In support of F.DAC, authorized administrators can manage the following properties of users: the designation whether a user is a superuser or not, the definition of a default access group for the user, and the assignment/removal of access permissions (i.e., combinations of specific access groups and permission groups).

F.MGMT.42 For management of IT environment security functions, the TOE's interface allows administrators to associate users with pre-defined or administrator-defined security roles.

6.2 TOE Assurance Measures

The assurance measures provided by the developer to meet the security assurance requirements for the TOE are based on the developer action elements and the requirements on content and presentation of evidence elements defined for the individual assurance requirements in CC Part 3:

SAR	Assurance Measures
ACM_CAP.3	The developer uses configuration management systems to provide version and access control for configuration items.
ACM_SCP.1	Configuration management is provided for the implementation representation of the TOE, the design documentation, test documentation, and all other evidence relevant for the TOE evaluation.
ADO_DEL.1	The developer employs protection mechanisms to provide for verifiable integrity of the TOE delivered to consumers.
ADO_IGS.1	Installation, generation and start-up procedures necessary for the secure operation of the TOE will be provided.
ADV_FSP.1	The Functional Specification for all externally visible interfaces of the TOE and the security functions implemented by the TOE will be provided.
ADV_HLD.2	A security enforcing High-level Design describing the TOE architecture will be provided for evaluation.
ADV_RCR.1	The developer will provide a correspondence analysis, demonstrating how to trace the TOE Security Functions identified in this Security Target to the Functional Specification, and the description of the security functions and interfaces in the Functional Specification to the subsystems detailed in the High-level Design.
AGD_ADM.1	Administrator guidance is provided to allow the secure operation of the TOE in its intended environment in compliance with the evaluated configuration.
AGD_USR.1	To the extent that the TOE is available to end-users, the relevant security functionality and assumptions on user behavior is documented for end-users.
ALC_DVS.1	The TOE is developed in an environment that provides for its integrity and for the confidentiality of restricted design and other information related to the TOE development.

SAR	Assurance Measures
ALC_FLR.1	Documentation on the developer's flaw remediation processes will be provided. The appropriate procedures to address consumer- and otherwise reported security flaws are integrated into the developer's product life-cycle.
ATE_COV.2	The developer will provide a test coverage analysis, demonstrating how the security functions and interfaces documented in the Functional Specification are covered by functional tests.
ATE_DPT.1	Also provided will be a test depth analysis that demonstrates which functional tests map to the subsystems and subsystem interfaces documented in the High-level Design.
ATE_FUN.1	The developer will provide a test plan, test procedure and test case documentation, as well as the actual test results for the tests that will be run against the evaluated version of the TOE as identified in the test coverage and depth analyses.
ATE_IND.2	A copy of the TOE will be provided to the evaluation lab for independent testing.
AVA_MSU.1	The developer will provide the guidance mentioned above for ADO_IGS.1, AGD_ADM.1 and AGD_USR.1.
AVA_SOF.1	This Security Target does not identify any mechanisms that would require a strength of function analysis to be provided for the evaluation.
AVA_VLA.1	A vulnerability analysis for the TOE will be performed and documented.

Table 3: Assurance measures meeting the TOE security assurance requirements

7 Protection Profile Claims

This ST does not claim conformance to any Protection Profile.

8 Rationale

8.1 Security Objectives Rationale

8.1.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.Audit	P.Accountability
O.ObjectAccess	T.Unauthorized

Table 4: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the IT environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.Authentication	TE.UnknownUsers
OE.GUIAccess	TE.GUIAccess
OE.TimeSource	P.Accountability

Table 5: Mapping of security objectives for the IT environment to assumptions, threats and policies

The following table provides a mapping of the objectives for the non-IT environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.Administrators	A.Configuration A.Administrators
OE.Remote	TE.Remote
OE.Repositories	TE.Repositories P.Accountability
OE.Runtime	A.Physical A.Runtime TE.System

Table 6: Mapping of security objectives for the non-IT environment to assumptions, threats and policies

8.1.2 Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.Unauthorized	The threat of unauthorized access to TSF or user data is removed by the objective O.ObjectAccess to implement access control mechanisms in the TOE.
TE.UnknownUsers	The threat of unidentified users accessing the TOE is removed by requiring authentication services from the IT environment in OE.Authentication.
TE.GUIAccess	The threat of unauthorized users having access to administrators interfaces of the TOE is removed by requiring the IT environment to provide access control for these interfaces in OE.GUIAccess.
TE.Repositories	The threat of gaining unauthorized access to the repositories in the IT environment that are used by the TOE is removed by the objective OE.Repositories requiring the implementation of countermeasures.
TE.Remote	The threat of interception of data transferred between different entities of the TOE is countered by objective OE.Remote that requires administrators to establish countermeasures in the IT or non-IT environment.
TE.System	The threat of exploitation of the Provisioning Server's underlying system in order to circumvent the TSP is diminished by the objective OE.Runtime to prevent unauthorized access to the system.

Table 7: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
A.Physical	The assumption on physical protection of the TOE is achieved by the objective OE.Runtime to provide such protection.
A.Administrators	The assumptions that administrators are trustworthy and well trained is achieved by the objective OE.Administrators to ensure these properties of administrators.

Assumption	Rationale for security objectives
A.Runtime	The assumption on exclusive TOE use of the underlying machines for the TOE and preventing unauthorized access is achieved by the objective OE.Runtime to implement corresponding measures for the Provisioning Manager's runtime environment.
A.Configuration	The assumption that the TOE will be configured and operated in its evaluated configuration is achieved by the objective OE.Administrators to ensure that administrators obey by the guidance.

Table 8: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.Accountability	The policy to provide accountability for the action of TOE administrators is implemented by the objective O.Audit to provide an auditing mechanism and supported by objectives OE.TimeSource to provide a reliable time source, and OE.Repositories to provide protection of the audit logs, in the runtime environment.

Table 9: Sufficiency of objectives enforcing Organizational Security Policies

8.2 Security Functional Requirements Rationale

8.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objectives
FAU_GEN_TPM.1	O.Audit
FAU_GEN.2	O.Audit
FAU_SAR.1	O.Audit
FAU_SAR.3	O.Audit
FDP_ACC.1	O.ObjectAccess
FDP_ACF.1	O.ObjectAccess
FMT_MSA.3	O.ObjectAccess

Security Functional Requirements	Objectives
FMT_SMF.1	O.Audit O.ObjectAccess
FMT_SMR.1	O.ObjectAccess

Table 10: Mapping of security functional requirements to security objectives

The following table provides a mapping of SFRs for the environment to security objectives, showing that each security functional requirement for the environment addresses at least one security objective. Non IT objectives for the TOE environment must be accomplished by procedural or administrative measures such that no SFRs are formulated for them in this ST.

Security Functional Requirements	Objectives
FDP_ACC.1-e	OE.GUIAccess
FDP_ACF.1-e	OE.GUIAccess
FIA_UAU.1-e	OE.Authentication
FIA_UID.1-e	OE.Authentication
FMT_MSA.1-e	OE.GUIAccess
FMT_MSA.3-e	OE.GUIAccess
FMT_SMF.1-e	OE.GUIAccess
FMT_SMR.1-e	OE.GUIAccess
FPT_STM.1-e	OE.TimeSource

Table 11: Mapping of security functional requirements for the environment to security objectives

8.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.Audit	The objective to provide means to audit changes to configuration data is met by requirements for audit record generation (FAU_GEN_TPM.1) and association of audited events with the originating user ID (FAU_GEN.2). Administrators have the ability to review and search audit data (FAU_SAR.1 and FAU_SAR.3). Supportive management functions have been specified in FMT_SMF.1.

Security objectives	Rationale
O.ObjectAccess	The objective to allow the restriction of access to managed objects is implemented by a discretionary access control policy as specified in FDP_ACC.1 and FDP_ACF.1. It is supported by requirements pertaining to the management of the access control enforcement (FMT_MSA.3, FMT_SMF.1, FMT_SMR.1).

Table 12: Security objectives for the TOE rationale

The following rationale provides justification for each security objective for the IT environment, showing that the security functional requirements for the IT environment are suitable to meet and achieve the security objectives:

Security objectives	Rationale
OE.Authentication	The objective for authentication (and implied, identification) of users is met by FIA_UAU.1-e and FIA_UID.1-e.
OE.GUIAccess	The objective to provide access control for the (administrative) interfaces of the TOE is met by a DAC policy as defined in FDP_ACC.1-e and FDP_ACF.1-e. Supportive management activities have been specified in FMT_MSA.1-e, FMT_MSA.3-e, FMT_SMF.1-e and FMT_SMR.1-e.
OE.TimeSource	The objective to provide a reliable time source for the TOE's use is met by the requirement for such a time source in FPT_STM.1-e.

Table 13: Security objectives for the TOE environment rationale

8.2.3 Security Requirements Dependency Analysis

Dependencies within the EAL3 package selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed here again. The included component on flaw remediation, ALC_FLR.1, has no dependencies on other requirements.

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN_TPM.1	FPT_STM.1	FPT_STM.1-e
FAU_GEN.2	FAU_GEN.1	FAU_GEN_TPM.1

Security Functional Requirement	Dependencies	Resolution
	FIA_UID.1	FIA_UID.1-e
FAU_SAR.1	FAU_GEN.1	FAU_GEN_TPM.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1
	FMT_MSA.3	FMT_MSA.3
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1-e
	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.1-e

Table 14: TOE SFR dependency analysis

The dependencies on FAU_GEN.1 are resolved by FAU_GEN_TPM.1. For a rationale why this extended component has been chosen please consult the extended component definition in section 5.1. While FAU_GEN_TPM.1 cannot be considered hierarchically higher than FAU_GEN.1, the ST authors determined that neither FAU_GEN.2 nor FAU_SAR.1 depend on the specific functionality that FAU_GEN_TPM.1 lacks (compared to FAU_GEN.1), but rather on the availability of audit records in general.

The following table contains the dependency analysis for the IT environment SFRs:

Security Functional Requirement	Dependencies	Resolution
FDP_ACC.1-e	FDP_ACF.1	FDP_ACF.1-e
FDP_ACF.1-e	FDP_ACC.1	FDP_ACC.1-e
	FMT_MSA.3	FMT_MSA.3-e
FIA_UAU.1-e	FIA_UID.1	FIA_UID.1-e
FIA_UID.1-e	No dependencies.	
FMT_MSA.1-e	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-e FDP_ACC.1
	FMT_SMR.1	FMT_SMR.1-e
	FMT_SMF.1	FMT_SMF.1-e
FMT_MSA.3-e	FMT_MSA.1	FMT_MSA.1-e

Security Functional Requirement	Dependencies	Resolution
	FMT_SMR.1	FMT_SMR.1-e
FMT_SMF.1-e	No dependencies.	
FMT_SMR.1-e	FIA_UID.1	FIA_UID.1-e
FPT_STM.1-e	No dependencies.	

Table 15: IT environment SFR dependency analysis

8.2.4 Internal consistency and mutual support of SFRs

Section 8.2.2 has already demonstrated how the IT security requirements work together to implement the individual objectives for the TOE and the IT environment. This section will elaborate on the internal consistency and mutual support of the IT security requirements.

The TOE's purpose is to maintain a database of managed objects, and to perform management functions on these objects as defined by the users of the TOE, who may be administrators of either the TOE itself or the administrators of the managed endpoints in the IT Environment.

As discussed in the TOE description, the provision of security functionality by the TOE in this context focuses on control of the TOE users and the changes they are able to make to the data model that is used by the TOE for endpoint administration. Since the TOE is running within a J2EE-based application server, it makes use of the IT environment's security functionality to achieve this goal where appropriate.

Users who connect to the Provisioning Server have to authenticate themselves in order to prove their identity, as modeled by FIA_UID.1-e and FIA_UAU.1-e. They are then subjected to two different access control mechanisms. The application server implements a DAC policy that restricts access based on the requested interfaces and the actions that an interface allows to initiate. This Role-Based Security Policy is defined by FDA_ACC.1-e and FDP_ACF.1-e, and is supported by management requirements (FMT_MSA.1-e, FMT_MSA.3-e, FMT_SMF.1-e, FMT_SMR.1-e).

The Instance Level Security Policy implemented by FDP_ACC.1 and FDP_ACF.1 restricts user access to individual objects in the TOE's data model. It is again supported by requirements for management facilities (FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FMT_SMR.1). Note that the TOE provides an administrative interface for the management of both policies, and that at the same time, the access control mechanism provided by the IT environment is responsible for restricting access to these administrative interfaces.

In addition, the TOE implements an organizational security policy to provide accountability for configuration changes in its data model. This is implemented by requirements to generate audit records (FAU_GEN_TPM.1, FAU_GEN.2) and to offer facilities for their review (FAU_SAR.1, FAU_SAR.3). Again, requirements for the management of the audit mechanism are provided (FMT_SMF.1). The IT environment supports the generation of audit records by providing a reliable time source, as modeled in FPT_STM.1-e.

8.3 Security Assurance Requirements Rationale

8.4 TOE Summary Specification Rationale

8.4.1 Security Functions justification

The following table shows that the IT security functions specified in the TOE summary specification meet all the security functional requirements for the TOE and work together to satisfy the TOE security functional requirements:

SFR	Security Functions	Rationale
FAU_GEN_TPM.1	F.AUD	The implementation of the audit systems for the TOE is described in F.AUD.
FAU_GEN.2	F.AUD	F.AUD includes the implementation of user identity association for audit records.
FAU_SAR.1	F.AUD	Selective review of audit records is provided by F.AUD.
FAU_SAR.3	F.AUD	Selective review of audit records is provided by F.AUD.
FDP_ACC.1	F.DAC	F.DAC implements the Instance Level Security policy.
FDP_ACF.1	F.DAC	F.DAC implements the Instance Level Security policy.
FMT_MSA.3	F.MGMT	F.MGMT describes that only default access groups are assigned upon user creation.
FMT_SMF.1	F.MGMT	F.MGMT provides the management functionality for F.DAC and F.AUD.
FMT_SMR.1	F.DAC F.MGMT	F.MGMT and F.DAC describe the management and significance of the superuser role and access and permission groups.

Table 16: Mapping of TOE SFRs to TSF

8.4.2 Mutual support of Security Functions

The TOE seeks to protect the data objects that it is being used to manage. This is achieved by the access control policy in F.DAC. In addition, accountability is established for changes to the data model by the auditing functionality defined in F.AUD. Management support for these security functions is provided by F.MGMT.

8.5 Evaluation Assurance Level and Strength of Function

The evaluation assurance level (EAL) 3 was chosen as a medium level of assurance reflecting the expected assurance requirements of commercial customers using the target of evaluation (TOE) to manage their infrastructures. The TOE is intended to provide a reasonable level of protection for the protected assets comparable to the protection provided by most

commercial-off-the-shelf operating system products. This is reflected as well in the definition of the TOE environment in chapter 2, the security problem definition in chapter 3 and the security objectives for the TOE in chapter 4 of this ST. The assurance level EAL3 was augmented with ALC_FLR.1 to address the flaw remediation process used for the product.

The ST claims that the functions provided by the TOE do not contain probabilistic or permutational mechanisms, and as a result no SOF rating is applicable.