Altair Innovation Intelligence™

Altair PBS Professional 10.1 Security Target

Version:	1.4
Status:	Released
Last Update:	2009-10-23
Classification:	Public

Trademarks

atsec is a trademark of atsec GmbH.

Linux® is registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SuSE is a trademark of Novell, Inc. in the U.S. and other countries.

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
1.0	Scott Chapman, atsec	2008-11-23	Initial Version
1.1	Scott Chapman, atsec	2009-02-11	Fixed BSI comments.
1.2	Scott Chapman, atsec	2009-04-21	Added cert ID. Fixed evaluator comments. Added review comments from Altair. Removed FMT_MSA.*-role. Added resource quotas. Added object attribute listings. Added assumptions on DNS, physical, and logical.
1.3	Scott Chapman, atsec	2009-06-15	Change PBS version number to 10.1. Added FIA_UID.2-job.
1.4	Scott Chapman, atsec	2009-10-23	Added FDP_ACx.1-job, FMT_MSA.x-job, and job access descriptions. Added FMT_MSA.x-aclres and Reservation descriptions. Added CC Admin & Usage Guide to Physical Boundary section. Added acl_root, flatuid, and $restricted comments to section 1.4.6. Split FMT_MSA.x-acl into two: FMT_MSA.x-aclque and FMT_MSA.x-aclres. Updated FDP_ACx.1-aclque and FDP_ACx.1-aclsvr. Updated FDP_ACx.1-role and User Role descriptions. Updated Queue descriptions. Updated job cookie descriptions. Updated User Role descriptions. Updated section 1.4.3's subjects. Fixed TCP/IP abbreviation in Abbreviations. Updated the software installation image names.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Altair PBS Professional 10.1 Security Target
Version:	1.4
Status:	Released
Date:	2009-10-23
Sponsor:	Altair Engineering, Inc.
Developer:	Altair Engineering, Inc.
Certification ID:	BSI-DSZ-CC-0599
Keywords:	Altair, PBS Pro, PBS Professional, PBS GridWorks, grid computing, high performance computing, workload management

1.2 TOE Identification

The TOE is Altair PBS Professional Version 10.1 (10.1.0.91350).

1.3 TOE Overview

Altair PBS Professional is a workload management software product for resource and workload management of a computer complex. Users submit batch jobs to the TOE. The TOE finds available resources for the jobs within the complex, schedules the jobs for execution, and executes the jobs on behalf of the users.

The TOE consists of a Job Server (a.k.a. Server), a Job Scheduler (a.k.a. Scheduler), and Job Executors (a.k.a. MOMs). The MOMs run on multiple host computers and present resources as virtual nodes (Vnodes) within the complex (one MOM per host computer). An authorized user submits a batch job (a.k.a. job) to the Server in the form of a shell script that contain the job's execution requirements. The job is queued by the Server on either a Job Queue (a.k.a. Queue) or on a Job Reservation Queue (a.k.a. Reservation) and then scheduled for execution on one or more MOMs. The TOE reviews the job requirements defined by the job and reviews the workload of the MOMs within the complex to determine where and when to execute a job.

The TOE also supports multi-Vnode jobs. For a multi-Vnode job, the Scheduler selects a MOM to be the "Mother Superior" for the entire job. The Mother Superior then passes various parts of the job to multiple "Subordinate MOMs" for execution and tracks the progress of the job parts to completion.

1.3.1 TOE Type

Altair PBS Professional is a distributed software application that provides a grid computing environment for workload management. It is used to schedule and execute software jobs across multiple Vnodes within the grid complex. The aspects evaluated are software components of the product along with the guidance documentation associated with the product.

1.3.2 Required Non-TOE Hardware and Software

The Operational Environment for the TOE consists of the following hardware platforms and operating systems:

- Red Hat® Enterprise Linux® 5 (x86 64-bit)
- SuSE Linux Enterprise Server 10 (x86 64-bit)

The hardware platforms and operating systems are not part of the TOE and are not shipped as part of the product.

1.3.3 Intended Method of Use

The TOE employs a distributed architecture intended to be used in a protected network environment where network eavesdropping is not allowed except by network administrative personnel (i.e. protected by policy) or where communications between networked computers is protected by some other means (e.g. IPsec). Communication with the TOE and between TOE components is not protected from modification or disclosure by the TOE.

1.3.4 Major Security Features

The TOE performs identification of users accessing the TOE, it uses multiple access control lists (ACLs) to control access to the Server, Queues, and Reservation Queues, and it provides multiple User Roles for separating administrative tasks, such as management of the security features, from non-administrative tasks, such as submitting jobs for execution.

1.4 TOE Description

1.4.1 TOE Introduction and Logical Boundary

The TOE consists of software components and guidance documentation. The software components of the TOE consist of:

- Server
- Scheduler
- MOM
- PBS Commands
- PBS Libraries

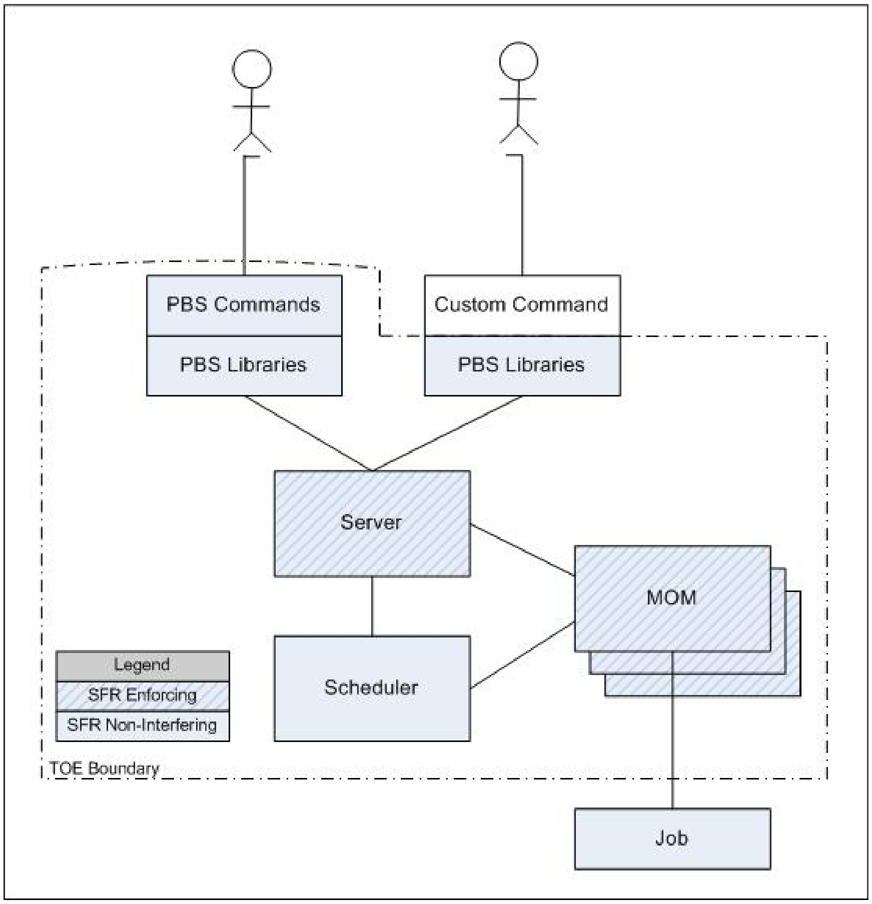

Figure 1: Logical boundary

The Server, Scheduler, and MOM are daemon processes which run continuously within the complex. There's one Server, one Scheduler, and one or more MOMs per complex. The TOE includes the PBS Commands for both administration of the TOE and for regular user interaction with the TOE. The TOE also includes the PBS Libraries which are used by the PBS Commands and which allow end-users to write custom commands/applications that access the TOE. Figure 1 shows the logical boundary of the TOE.

The TOE has a distributed design allowing all the TOE components to reside anywhere in a distributed environment and in almost any configuration. The components communicate with each other over TCP/IP. The only real requirements are for the Server and Scheduler to reside on the same computer and for MOMs to reside on host computers where the jobs will be executed. It's typical for the MOMs to reside on computers other than where the Server / Scheduler reside, but this is not a requirement. In the evaluated configuration, a Vnode is equivalent to a host computer containing a MOM (i.e. a host computer represents one Vnode and a Vnode represents one host computer).

The TOE supports the concept of Queues which are collections of jobs waiting to be serviced. Multiple Queues can be created within the TOE and access control lists attached to each Queue. Queues are created by TOE "Managers" and access to the Queues are controlled by the TOE "Managers". Two types of Queues exist: Execution Queues and Routing Queues. Execution

Queues contain jobs waiting to be executed. Execution Queues also have the ability to route jobs to another Server. Routing Queues contain jobs waiting to be routed somewhere else. Routing Queues can route jobs to either another Routing Queue or to an Execution Queue.

The TOE supports the concept of Reservation Queues. Reservation Queues are queue entities that reserve a set of resources for a specified time. Multiple Reservation Queues can be created within the TOE and access control lists attached to each Reservation Queue. Any authorized TOE user can create a Reservation Queue. Two main types of Reservations exist: Advance Reservations and Standing Reservations. Advance Reservations reserve a set of resources for a specified time in the future. Standing Reservations are recurring Advanced Reservations.

A typical job flow for the TOE is for an authorized user (i.e. Manager, Operator, Regular User - see section 1.4.2.4) to submit a batch job request to the Server using the PBS Commands or a custom command. The Server queues the request on an Execution Queue for execution. The Scheduler schedules where and when the job will be executed. The MOM then executes the job on the system where the MOM resides and as the user who submits the job.

The TOE also supports multi-Vnode jobs. For a multi-Vnode job, the Scheduler selects a MOM to be the Mother Superior for the entire job. The Mother Superior then passes various parts of the job to multiple Subordinate MOMs for execution and tracks the progress of the job parts to completion. The Mother Superior continues to act as a MOM for its own Vnode.

The software required by the job must already exist on the computer where the MOM resides. The TOE does not install the required software on the host computer. Thus, if the job executes a weather forecasting program, the weather forecasting program must already exist on the Vnode.

The Server, Scheduler, and MOMs run as administrative users (i.e. the root user on Linux operating systems) on their respective systems. This allows the daemons to communicate with each other over privileged ports. This also allows the MOMs to execute the jobs as the users who submit the jobs. The communications are not protected from modification or disclosure by the TOE.

The TOE relies on the trustworthiness of the operating systems for identifying users. This implies that the user community of the computers allowed to access the complex as well as the computers comprising the complex must be well managed.

1.4.2 TOE Security Features

This section describes the security features at a high level. The security functional requirement (SFR) enforcing components of the TOE are the Server and the MOMs.

1.4.2.1 Identification & Authentication

Server User Identification

The TOE uses the operating system user names to identify TOE users when they submit a request to the Server. The TOE provides a method to verify user identities through the use of privileged ports. The user identified by the Server is the user identity used by the MOM when executing a job.

Job Process Identification and Authentication

To thwart unauthorized usage of job resources, the MOM creates a pseudo-random Job Cookie for each job and passes the Job Cookie to the job. (If it's a multi-Vnode job, the Mother Superior creates the Job Cookie and passes the Job Cookie along with the job to the Subordinate MOMs which in turn pass the Job Cookie to the job.) When a Job Process requests a service from its MOM, the Job Process identifies and authenticates itself to the MOM by passing its Job Cookie to the MOM which the MOM validates. (The pseudo-random function used by the TOE to create the Job Cookie is part of the Operational Environment.)

1.4.2.2 Access Control

The TOE uses access control lists (ACLs) to manage access to the Server, Queues, and Reservation Queues. There are separate sets of ACLs for each Server, for each Queue on a Server, and for each Reservation Queue on a Server. The ACLs allow or deny access based on user names and host names. The Queue ACLs and Reservation Queue ACLs can also allow or deny access based on the user's default group (as defined by operating system where the Server is executing).

The Server enforces the access control policy for User Roles supported by the TOE. The User Roles access control policy is hardcoded within the Server.

1.4.2.3 Resource Allocation Quotas

The TOE enforces and provides a management interface for managing maximum resource allocation quotas on users, groups of users, and jobs. These quotas allow the TOE to counter denial of service issues for the TOE controlled resources. The TOE enforceable resource quotas are:

- Job slots
- Elapsed time
- Processor time
- Number of processors
- Physical memory
- Virtual memory

1.4.2.4 Management

The TOE supports the following User Roles that are effectively assignable to users. The roles are:

- Manager
- Operator
- Regular User

Managers have the most privilege of any of the User Roles. Operators have more privilege than Regular Users, but have less privilege than Managers. Authorized users that are neither Managers nor Operators are considered Regular Users.

Managers can manage the entire TOE including the security aspects of the TOE such as the management of ACLs and the assigning of users to User Roles. Operators are limited to managing the non-security aspects of the TOE such as setting and unsetting non-security attributes of Vnodes, Queues, and Servers.

1.4.3 Subjects and Objects

Users are represented as operating system processes. Operating systems use user IDs to associate users with processes. The following subjects are defined for the TOE:

Subjects:

- **Command Process -** A process that issues requests to a Server.
- **Job Process -** A process executed as part of a batch job on behalf of a user by a MOM.

Objects:

- **Job -** A job (a.k.a. batch job) is a collection of information that describes the task required to be performed by the TOE, a description of the resources required for the task, and one or more Job Processes.
- **Queue -** A Queue (a.k.a. Job Queue) is a collection of jobs waiting to be processed. Two types of Queues exist: Execution Queues and Routing Queues. For the purposes of this Security Target document, both Queue types are treated the same with respect to security. Queue attributes include:
  - Host, User, and Group ACLs
  - Host, User, and Group ACL enable/disable flags
  - Maximum number of Queue jobs
  - Maximum user resources
  - Maximum resources for a defined group of users
  - Maximum job resources
- **Reservation Queue -** A Reservation Queue (a.k.a. Reservation) reserves a set of resources for a specified time. Two main types of Reservations exist: Advance Reservations and Standing Reservations. For the purposes of this Security Target document, both Reservation types are treated the same with respect to security. Reservation Queue attributes include:
  - Host, User, and Group ACLs
  - Host, User, and Group ACL enable/disable flags
  - Maximum number of Reservation Queue jobs
  - Maximum user resources
  - Maximum resources for a defined group of users
  - Maximum job resources
- **Server -** The Server daemon (a.k.a. Job Server) is the front-end of the TOE with which users communicate to submit and control jobs. Server attributes include:
  - Host and User ACLs
  - Host and User ACL enable/disable flags
  - Maximum user resources
  - Maximum resources for a defined group of users
  - Maximum job resources
  - Maximum number of jobs per user
  - Maximum number of jobs for a defined group of users
- **Vnode -** A Vnode (a.k.a. Virtual Node) is an object representing a set of resources which form a usable part of a computer. In the evaluated configuration, a Vnode is a computer and vice versa. Vnode attributes include:
  - Maximum number of Vnode jobs
  - Maximum number of jobs per user
  - Maximum number of jobs for a defined group of users

1.4.4 TSF Data and Security Attributes

The following TOE Security Functionality (TSF) data are maintained by the TOE:

- Access Control Lists
- Job Cookies
- Resource Quotas

The following user security attributes are maintained by the TOE:

- User Roles

1.4.5 Physical Boundary

The TOE is comprised of the following software installation images:

- For Red Hat® Enterprise Linux® 5 (x86 64-bit):
  - PBSPro_10.1.0-RHEL5_x86_64.tar.gz
- For SuSE Linux Enterprise Server 10 (x86 64-bit):
  - PBSPro_10.1.0-SLES10_x86_64.tar.gz

The TOE is comprised of the following guidance documents:

- PBS Professional 10.1 Administrator’s Guide
- PBS Professional 10.1 Common Criteria Administration & Usage Guide
- PBS Professional 10.1 External Reference Specification
- PBS Professional 10.1 Installation and Upgrade Guide
- PBS Professional 10.1 User’s Guide

1.4.6 Evaluated Configuration

The consumer must read, understand, and follow the guidance documentation provided as part of the TOE for the evaluated configuration.

The TOE contains a "root" ACL which is called *acl_root* by the TOE's guidance documentation. This ACL should not be used in the evaluated configuration.

Each MOM's configuration file contains a parameter named *$restricted* which allows a MOM to accept connections from non-privileged ports of hosts specified by *$restricted*. This parameter should not be used in the evaluated configuration.

The Server contains a startup attribute called *flatuid* which can be set to either **true** or **false** in the evaluated configuration. This attribute is explained in the TOE guidance documentation.

1.4.7 Operational Environment

The Operational Environment for the TOE consists of the hardware platforms and operating systems specified in section 1.3.2 under TOE Reference.

1.4.7.1 Physical

The computer hardware and networking used by the TOE are part of the Operational Environment. The computers that run the Server, Scheduler, and MOMs must be in a non-hostile environment. The security of these software components depends on the physical security of the computers where the components reside.

The Domain Name Servers (DNS) must be trustworthy.

As mentioned in section 1.3.3, the Operational Environment must provide protection for the network data against modification and disclosure. Protection mechanisms include:

- Providing physical security of both the local network and the grid resources
- Providing logical protection of grid resources through the use of firewalls and/or network isolation
- Providing encrypted VPNs (e.g. IPsec) between enterprise sites
- Providing grid resources with up-to-date anti-virus tools and applying security updates regularly
- Restricting local users of grid resources from having administrative rights

1.4.7.2 Software

The operating systems used by the TOE are part of the Operational Environment. The TOE depends on the non-bypassability of the operating systems to prevent unauthorized users from accessing, modifying, and using the TOE. The TOE also depends on the authentication of users by the operating system. The TOE retrieves user and group security attributes from the operating systems.

2 CC Conformance Claim

This ST is CC Part 2 conformant 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 3.1 have been taken as the basis for this conformance claim.

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 Operational Environment.

The **IT assets** to be protected comprise the information stored, processed or transmitted by the TOE. The term “information” is used here to refer to all data held within the product.

The TOE counters the general threat of unauthorized access to information, where “access” includes disclosure, modification, and destruction.

The **threat agents** can be categorized as either:

- Unauthorized users of the TOE, i.e. individuals who have not been granted the right to access the system; or
- Authorized users of the TOE, i.e. individuals who have been granted the right to access the system.

The threat agents are assumed to originate from a well managed user community in a non-hostile working environment. Therefore, the product protects against threats of security vulnerabilities that might be exploited in the intended environment for the TOE with medium level of expertise and effort. The TOE protects against straightforward or intentional breach of TOE security by attackers possessing a low attack potential.

3.1.1 Threats countered by the TOE

T.Access	A threat agent may gain access to protected resources of the TOE or perform TOE operations for which no access rights have been granted.
T.Authorized	Unauthorized users may gain access to the TOE.
T.DenialOfService	A TOE user, process, or job could monopolize TOE resources thereby causing a denial of service.

3.2 Assumptions

3.2.1 Environment of use of the TOE

3.2.1.1 Physical

A.Network.Protection	The Operational Environment protects network traffic from modification and disclosure by non-administrator personnel.
A.Physical.Protection	The Operational Environment protects the complex’s computer resources from unauthorized physical access.

3.2.1.2 Personnel

A.Admin.Training	The administrators of the TOE and of the Operational Environment are aware of the security policies and procedures of their organization, are trained and competent to follow the

manufacturer’s guidance and documentation, and correctly configure and operate the TOE and Operational Environment in accordance with those policies and procedures.

A.Admin.Trust The administrators of the TOE and of the Operational Environment are trustworthy and are not careless, negligent, malicious, or hostile.

A.User.Training The TOE users are aware of the security policies and procedures of their organization, and are trained and competent to follow those policies and procedures.

3.2.1.3 Connectivity

A.DNS.Trust The Domain Name Service (DNS) used by the complex provides trustworthy services.

A.Logical.Protection The Operational Environment protects the complex’s computer resources by running up-to-date anti-virus tools regularly on the complex’s computer resources, applying security updates regularly to the complex’s computer resources, using firewalls and/or network isolation to protect the complex’s computer resources, and restricting the set of people who have administrative access to the complex’s computer resources.

A.User.Authentication The Operational Environment (i.e. operating system) correctly identifies and authenticates users prior to providing them access to the TOE.

4 Security Objectives

4.1 Objectives for the TOE

O.User.Access The TOE shall ensure that users are authorized to access the protected objects of the TOE and ensure that users can perform only the actions allowed by the user's User Role in accordance to the security policy of the TOE.

O.User.Authenticated The TOE shall require identification and authentication of a Job Process to the Subordinate MOM on the host computer where the Job Process is running.

O.User.Identified The TOE shall require identification of all users that access the TOE and for a User Role to be associated with each user.

O.Resource.Availability The TOE shall provide functionality to control the use of resources such that a denial of service will not occur due to the monopolization of TOE controlled resources.

4.2 Objectives for the Operational Environment

OE.Admin.Trained The administrators of the TOE and of the Operational Environment shall be made aware of the security policies and procedures of their organization, shall be trained and competent to follow the manufacturer's guidance and documentation, and shall correctly configure and operate the TOE and Operational Environment in accordance with those policies and procedures.

OE.Admin.Trusted The administrators of the TOE and of the Operational Environment shall be trustworthy and shall not be careless, negligent, malicious, or hostile.

OE.DNS.Trusted The Domain Name Service (DNS) used by the complex shall be trustworthy.

OE.Logical.Protected The Operational Environment shall protect the complex's computer resources by running up-to-date anti-virus tools regularly on the complex's computer resources, applying security updates regularly to the complex's computer resources, using firewalls and/or network isolation to protect the complex's computer resources, and restricting the set of people who have administrative access to the complex's computer resources.

OE.Network.Protected The Operational Environment shall protect network traffic from modification and disclosure by non-administrator personnel.

OE.Physical.Protected The Operational Environment shall protect the complex's computer resources from unauthorized physical access.

OE.User.Authenticated The Operational Environment (i.e. operating system) shall correctly identify and authenticate users prior to providing them access to the TOE.

OE.User.Trained The TOE users shall be aware of the security policies and procedures of their organization, and shall be trained and competent to follow those policies and procedures.

4.3 Security Objectives Rationale

4.3.1 Security Objectives 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.User.Access	T.Access
O.User.Authenticated	T.Authorized
O.User.Identified	T.Authorized
O.Resource.Availability	T.DenialOfService

Table 1: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational 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.Admin.Trained	A.Admin.Training
OE.Admin.Trusted	A.Admin.Trust
OE.DNS.Trusted	A.DNS.Trust
OE.Logical.Protected	A.Logical.Protection
OE.Network.Protected	A.Network.Protection
OE.Physical.Protected	A.Physical.Protection
OE.User.Authenticated	A.User.Authentication
OE.User.Trained	A.User.Training

Table 2: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Security Objectives 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.Access	The threat that a threat agent may gain access to protected resources of the TOE or perform TOE operations for which no access rights have been granted is diminished by: • O.User.Access which ensures that users are authorized to access the protected object of the TOE and ensures that users can perform only the actions allowed by the user's User Role in accordance to the security policy of the TOE.
T.Authorized	The threat that unauthorized users may gain access to the TOE is diminished by: • O.User.Authenticated which requires the TOE to identify and authenticate a Job Process to its Subordinate MOM. • O.User.Identified which requires the TOE to identify all users that access the TOE and for a User Role to be associated with each user.
T.DenialOfService	The threat that a TOE user, process, or job could monopolize TOE resources thereby causing a denial of service is diminished by: • O.Resource.Availability which requires that the TOE provide functionality to control the use of resources such that a denial of service will not occur due to the monopolization of TOE controlled resources.

Table 3: 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.Network.Protection	The assumption that the Operational Environment protects network traffic from modification and disclosure by non-administrator personnel is covered by: • OE.Network.Protected which requires the Operational Environment to protect network traffic from modification and disclosure by non-administrator personnel.
A.Physical.Protection	The assumption that the Operational Environment protects the complex's computer resources from unauthorized physical access is covered by: • OE.Physical.Protected which requires the Operational Environment to protect the complex's computer resources from unauthorized physical access.

Assumption	Rationale for security objectives
A.Admin.Training	The assumption that the administrators of the TOE and of the Operational Environment are aware of the security policies and procedures of their organization, are trained and competent to follow the manufacturer's guidance and documentation, and correctly configure and operate the TOE and Operational Environment in accordance with those policies and procedures is covered by: • OE.Admin.Trained which requires the administrators of the TOE and of the Operational Environment shall be made aware of the security policies and procedures of their organization, to be trained and competent to follow the manufacturer's guidance and documentation, and to correctly configure and operate the TOE and Operational Environment in accordance with those policies and procedures.
A.Admin.Trust	The assumption that the administrators of the TOE and of the Operational Environment are trustworthy and are not careless, negligent, malicious, or hostile is covered by: • OE.Admin.Trusted which requires the administrators of the TOE and of the Operational Environment to be trustworthy and to not be careless, negligent, malicious, or hostile.
A.User.Training	The assumption that the TOE users are aware of the security policies and procedures of their organization, and are trained and competent to follow those policies and procedures is covered by: • OE.User.Trained which requires the TOE users to be aware of the security policies and procedures of their organization, and to be trained and competent to follow those policies and procedures.
A.DNS.Trust	The assumption that the Domain Name Service used by the complex provides trustworthy services is covered by: • OE.DNS.Trusted which requires the Domain Name Service used by the complex to be trustworthy.
A.Logical.Protection	The assumption that the Operational Environment protects the complex's computer resources by running up-to-date anti-virus tools regularly on the complex's computer resources, applying security updates regularly to the complex's computer resources, using firewalls and/or network isolation to protect the complex's computer resources, and restricting the set of people who have administrative access to the complex's computer resources is covered by: • OE.Logical.Protected which requires the Operational Environment to protect the complex's computer resources by running up-to-date anti-virus tools regularly on the complex's computer resources, applying security updates regularly to the complex's computer resources, using firewalls and/or network isolation to protect the complex's computer resources, and restricting the set of people who have administrative access to the complex's computer resources.

Assumption	Rationale for security objectives
A.User.Authentication	The assumption that the Operational Environment (i.e. operating system) correctly identifies and authenticates users prior to providing them access to the TOE is covered by: • OE.User.Authenticated which requires the Operational Environment (i.e. operating system) to correctly identify and authenticate users prior to providing them access to the TOE.

Table 4: Sufficiency of objectives holding assumptions

5 Extended Components Definition

There are no extended component definitions.

6 Security Requirements

6.1 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.
FDP User data protection	FDP_ACC.1-aclque Queue and Reservation Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1-aclsrv Server Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1-job Job Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1-role Role Access Control Policy	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-aclque Queue and Reservation Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-aclsrv Server Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-job Job Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-role Role Access Control Functions	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
FIA Identification and authentication	FIA_ATD.1 User Attribute Definition	FIA_ATD.1	CC Part 2	No	No	Yes	No
	FIA_SOS.2 TSF Generation of Secrets	FIA_SOS.2	CC Part 2	No	No	Yes	No
	FIA_UAU.2 Job Process Authentication before Any Action	FIA_UAU.2	CC Part 2	No	Yes	No	No
	FIA_UID.2-job Job Process Identification before Any Action	FIA_UID.2	CC Part 2	Yes	Yes	No	No
	FIA_UID.2-usr Server User Identification before Any Action	FIA_UID.2	CC Part 2	Yes	Yes	No	No
	FIA_USB.1 User-Subject Binding	FIA_USB.1	CC Part 2	No	No	Yes	No
FMT Security management	FMT_MSA.1-aclque Management of Queue ACL Security Attributes	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes

Security functional class	Security functional requirement	Security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FMT_MSA.1-aclres1 Management of Reservation ACL Security Attributes	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1-aclres2 Management of Reservation ACL Security Attributes	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1-aclsrv Management of Server ACL Security Attributes	FMT_MSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FMT_MSA.1-job Management of Job Security Attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-aclque Static Queue ACL Attribute Initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-aclres Static Reservation ACL Attribute Initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-aclsrv Static Server ACL Attribute Initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-job Static Job Attribute Initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1-role Management of User Role Data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MTD.1-rsrc Management of Resource Quota Data	FMT_MTD.1	CC Part 2	Yes	No	Yes	Yes
	FMT_SMF.1 Specification of Management Functions	FMT_SMF.1	CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security Roles	FMT_SMR.1	CC Part 2	No	No	Yes	No
FRU Resource utilisation	FRU_RSA.1-env Environment Resources	FRU_RSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FRU_RSA.1-time Time Resources	FRU_RSA.1	CC Part 2	Yes	Yes	Yes	Yes
	FRU_RSA.1-toe TOE Resources	FRU_RSA.1	CC Part 2	Yes	Yes	Yes	Yes

Table 5: Security functional requirements for the TOE

6.1.1 User data protection (FDP)

6.1.1.1 Queue and Reservation Access Control Policy (FDP_ACC.1-aclque)

FDP_ACC.1.1-aclque The TSF shall enforce the [**Queue Access Control Policy and Reservation Access Control Policy**] on [

- **Subjects:**
  - **Command Processes representing Authorized Users**
- **Objects:**
  - **Queues**
- **Operations:**
  - **Enqueue jobs (i.e. submit to this Queue, move to this Queue, route to this Queue)**

].

6.1.1.2 Server Access Control Policy (FDP_ACC.1-aclsrv)

FDP_ACC.1.1-aclsrv The TSF shall enforce the [**Server Access Control Policy**] on [

- **Subjects:**
  - **Command Processes representing Authorized Users**
- **Objects:**
  - **Servers**
- **Operations:**
  - **Request services**

].

6.1.1.3 Job Access Control Policy (FDP_ACC.1-job)

FDP_ACC.1.1-job The TSF shall enforce the [**Job Access Control Policy**] on [

- **Subjects:**
  - **Command Processes representing Authorized Users**
- **Objects:**
  - **Jobs**
- **Operations:**
  - **All Regular User operations on Jobs**

].

6.1.1.4 Role Access Control Policy (FDP_ACC.1-role)

FDP_ACC.1.1-role The TSF shall enforce the [**Role Access Control Policy**] on [

- **Subjects:**
  - **Command Processes representing Authorized Users**
- **Objects:**

  - **Jobs**
  - **Queues**
  - **Reservation Queues**
  - **Servers**
  - **Vnodes**
- **Operations:**
  - **All operations among subjects and objects**
  - **All operations among subject and object attributes**

].

6.1.1.5 Queue and Reservation Access Control Functions (FDP_ACF.1-aclque)

FDP_ACF.1.1-aclque The TSF shall enforce the [**Queue Access Control Policy and Reservation Access Control Policy**] to objects based on the following: [

- **Subjects and objects defined in FDP_ACC.1-aclque**
- **Subjects attributes:**
  - **User name**
  - **Host name**
  - **User's default operating system group on the Server**
  - **User Role**
- **Object attributes:**
  - **Host ACL**
  - **User ACL**
  - **Group ACL**
  - **Host ACL enable/disable flag**
  - **User ACL enable/disable flag**
  - **Group ACL enable/disable flag**

].

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

- **If all enabled Queue ACLs (i.e., Host ACL, User ACL, and/or Group ACL) allow the user access, then the user can perform all enqueue operations (i.e., submit to this Queue, move to this Queue, route to this Queue) on this Queue; otherwise, all enqueue operations on this Queue are denied.**
- **If all Queue ACLs (i.e., Host ACL, User ACL, and Group ACL) are disabled, then the user can perform all enqueue operations on this Queue.**

].

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

- **If the Host ACL is enabled and allows access or is disabled AND if at least one of the User and Group ACLs is enabled and denies access, then a Manager or Operator is allowed to use the move enqueue operation to move an existing job to this Queue.**

].

FDP_ACF.1.4-aclque The TSF shall explicitly deny access of subjects to objects based on the [**following additional rules: none**].

6.1.1.6 Server Access Control Functions (FDP_ACF.1-aclsrv)

FDP_ACF.1.1-aclsrv The TSF shall enforce the [**Server Access Control Policy**] to objects based on the following: [

- **Subjects and objects defined in FDP_ACC.1-aclsrv**
- **Subjects attributes:**
  - **User name**
  - **Host name**
  - **User Role**
- **Object attributes:**
  - **Host ACL**
  - **User ACL**
  - **Host ACL enable/disable flag**
  - **User ACL enable/disable flag**

].

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

- **If all enabled Server ACLs (i.e., Host ACL and/or User ACL) allow access, then the operation is allowed; otherwise, the operation is denied.**
- **If all Server ACLs (i.e., Host ACL and User ACL) are disabled, then the operation is allowed.**

].

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

- **If the Host ACL is enabled and allows access or is disabled AND if the user is a Manager or Operator, then the operation is allowed (i.e., the User ACL is ignored when the user is a Manager or Operator).**

].

FDP_ACF.1.4-aclsrv The TSF shall explicitly deny access of subjects to objects based on the [**following additional rules: none**].

6.1.1.7 Job Access Control Functions (FDP_ACF.1-job)

FDP_ACF.1.1-job The TSF shall enforce the [**Job Access Control Policy**] to objects based on the following: [

- **Subjects and objects defined in FDP_ACC.1-job**
- **Subjects attributes:**
  - **User Role**
- **Object attributes:**
  - **Attributes of Jobs**

].

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

- **Users can only perform the operations specified in FDP_ACC.1-job on their own jobs; otherwise, access is denied.**

].

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

- **When query_other_jobs is set to true, Regular Users can query another user's job.**
- **All Managers and Operators can perform all operations defined in FDP_ACC.1-job on any user's job.**

].

FDP_ACF.1.4-job The TSF shall explicitly deny access of subjects to objects based on the [**following additional rules: none**].

6.1.1.8 Role Access Control Functions (FDP_ACF.1-role)

FDP_ACF.1.1-role The TSF shall enforce the [**Role Access Control Policy**] to objects based on the following: [

- **Subjects and objects defined in FDP_ACC.1-role**
- **Subjects attributes:**
  - **User Role**
- **Object attributes:**
  - **Attributes of all objects listed in FDP_ACC.1-role**

].

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

- **If the subject invoking the operation on an object is assigned to a User Role that allows the operation, then the operation is allowed; otherwise, the operation is denied.**
- **If the subject invoking the operation on an attribute of the object is assigned to a User Role that allows the operation, then the operation is allowed; otherwise, the operation is denied.**

].

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

FDP_ACF.1.4-role The TSF shall explicitly deny access of subjects to objects based on the [**following additional rules: none**].

6.1.2 Identification and authentication (FIA)

6.1.2.1 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 Role**

].

6.1.2.2 TSF Generation of Secrets (FIA_SOS.2)

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet [**128-bits of pseudo-random data using a 32-bit pseudo-random generator that is part of Operational Environment**].

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for [**Job Process authentication**].

6.1.2.3 Job Process Authentication before Any Action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each ~~user~~ *Job Process* to be successfully authenticated *to the Subordinate MOM* before allowing any other TSF-mediated actions on behalf of that ~~user~~ *Job Process*.

6.1.2.4 Job Process Identification before Any Action (FIA_UID.2-job)

FIA_UID.2.1 The TSF shall require each ~~user~~ *Job Process* to be successfully identified before allowing any other TSF-mediated actions on behalf of that ~~user~~ *Job Process*.

6.1.2.5 Server User Identification before Any Action (FIA_UID.2-usr)

FIA_UID.2.1 The TSF shall require each *Server* user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.2.6 User-Subject Binding (FIA_USB.1)

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

- **User ID**
- **User name**
- **User Role**

].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**none**].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**none**].

6.1.3 Security management (FMT)

6.1.3.1 Management of Queue ACL Security Attributes (FMT_MSA.1-aclque)

FMT_MSA.1.1-aclque The TSF shall enforce the [**Queue Access Control Policy**] to restrict the ability to [**modify**] the security attributes [**Host ACL, User ACL, and Group ACL**] *and [enable, disable] the security attributes [Host ACL, User ACL, and Group ACL]* to [**Operator and Manager**].

6.1.3.2 Management of Reservation ACL Security Attributes (FMT_MSA.1-aclres1)

FMT_MSA.1.1-aclres1 The TSF shall enforce the [**Reservation Access Control Policy**] to restrict the ability to [**modify**] the security attributes [**Host ACL, User ACL, and Group ACL**] *and [enable, disable] the security attributes [Host ACL, User ACL, and Group ACL]* to [**Operator and Manager**] *after the Reservation Queue has been created*.

6.1.3.3 Management of Reservation ACL Security Attributes (FMT_MSA.1-aclres2)

FMT_MSA.1.1-aclres2 The TSF shall enforce the [**Reservation Access Control Policy**] to restrict the ability to [**disable**] the security attributes [**User ACL**] to [**none**] *when creating a Reservation Queue*.

6.1.3.4 Management of Server ACL Security Attributes (FMT_MSA.1-aclsrv)

FMT_MSA.1.1-aclsrv The TSF shall enforce the [**Server Access Control Policy**] to restrict the ability to [**modify**] the security attributes [**Host ACL and User ACL**] *and [enable, disable] the security attributes [Host ACL and User ACL]* to [**Manager**].

6.1.3.5 Management of Job Security Attributes (FMT_MSA.1-job)

FMT_MSA.1.1-job The TSF shall enforce the [**Job Access Control Policy**] to restrict the ability to [**modify**] the security attributes [**query_other_jobs**] to [**Manager**].

6.1.3.6 Static Queue ACL Attribute Initialisation (FMT_MSA.3-aclque)

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

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

6.1.3.7 Static Reservation ACL Attribute Initialisation (FMT_MSA.3-aclres)

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

FMT_MSA.3.2-aclres The TSF shall allow the [**authorized user who creates the Reservation Queue**] to specify alternative initial values to override the default values when an object or information is created.

6.1.3.8 Static Server ACL Attribute Initialisation (FMT_MSA.3-aclsrv)

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

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

6.1.3.9 Static Job Attribute Initialisation (FMT_MSA.3-job)

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

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

6.1.3.10 Management of User Role Data (FMT_MTD.1-role)

FMT_MTD.1.1-role The TSF shall restrict the ability to [**modify**] the [**User Role of an authorized user**] to [**Manager**].

6.1.3.11 Management of Resource Quota Data (FMT_MTD.1-rsrc)

FMT_MTD.1.1-rsrc The TSF shall restrict the ability to [**modify**] the [**Resource Quotas**] to [**Manager, Operator**].

6.1.3.12 Specification of Management Functions (FMT_SMF.1)

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

- **Management of ACLs**
- **Management of Resource Allocation Quotas**
- **Management of User Roles**

].

6.1.3.13 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**Manager, Operator, and Regular User**].

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

6.1.4 Resource utilisation (FRU)

6.1.4.1 Environment Resources (FRU_RSA.1-env)

FRU_RSA1.1-env The TSF shall enforce maximum quotas of the following resources: [**Number of processors, Physical memory size, Virtual memory size**] that [***an* individual user, defined group of users, subjects *(individual jobs only)***] can use [**simultaneously**].

6.1.4.2 Time Resources (FRU_RSA.1-time)

FRU_RSA1.1-timeThe TSF shall enforce maximum quotas of the following resources: [**Elapsed time, Processor time**] that [***an* individual user, defined group of users, subjects *(individual jobs only)***] can use [**over a specified period of time**].

6.1.4.3 TOE Resources (FRU_RSA.1-toe)

FRU_RSA1.1-toe The TSF shall enforce maximum quotas of the following resources: [**Job slots**] that [***an* individual user, defined group of users**] can use [**simultaneously**].

6.2 Security Functional Requirements Rationale

6.2.1 Security Requirements 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
FDP_ACC.1-aclque	O.User.Access
FDP_ACC.1-aclsrv	O.User.Access
FDP_ACC.1-job	O.User.Access
FDP_ACC.1-role	O.User.Access
FDP_ACF.1-aclque	O.User.Access
FDP_ACF.1-aclsrv	O.User.Access
FDP_ACF.1-job	O.User.Access
FDP_ACF.1-role	O.User.Access
FIA_ATD.1	O.User.Identified
FIA_SOS.2	O.User.Authenticated
FIA_UAU.2	O.User.Authenticated
FIA_UID.2-job	O.User.Authenticated
FIA_UID.2-usr	O.User.Identified

Security Functional Requirements	Objectives
FIA_USB.1	O.User.Authenticated O.User.Identified
FMT_MSA.1-aclque	O.User.Access
FMT_MSA.1-aclres1	O.User.Access
FMT_MSA.1-aclres2	O.User.Access
FMT_MSA.1-aclsrv	O.User.Access
FMT_MSA.1-job	O.User.Access
FMT_MSA.3-aclque	O.User.Access
FMT_MSA.3-aclres	O.User.Access
FMT_MSA.3-aclsrv	O.User.Access
FMT_MSA.3-job	O.User.Access
FMT_MTD.1-role	O.User.Access
FMT_MTD.1-rsrc	O.Resource.Availability
FMT_SMF.1	O.User.Access
FMT_SMR.1	O.User.Identified
FRU_RSA.1-env	O.Resource.Availability
FRU_RSA.1-time	O.Resource.Availability
FRU_RSA.1-toe	O.Resource.Availability

Table 6: Mapping of security functional requirements to security objectives

6.2.2 Security Requirements 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.User.Access	The objective that • the TOE shall ensure that users are authorized to access the protected objects of the TOE and ensure that users can perform only the actions allowed by the user's User Role in accordance to the security policy of the TOE is met by: • [FDP_ACC.1-aclque] which specifies the Queue and Reservation ACL policy

Security objectives	Rationale
	• [FDP_ACF.1-aclque] which specifies the Queue and Reservation ACL rules • [FDP_ACC.1-aclsrv] which specifies the Server ACL policy • [FDP_ACF.1-aclsrv] which specifies the Server ACL rules • [FDP_ACC.1-job] which specifies the Job access policy • [FDP_ACF.1-job] which specifies the Job access rules • [FDP_ACC.1-role] which specifies the Server's hardcoded access policy • [FDP_ACF.1-role] which specifies the Server's hardcoded access rules • [FMT_MSA.1-aclque] and [FMT_MSA.3-aclque] which specify how the Queue ACLs are managed by the appropriate users • [FMT_MSA.1-aclres1], [FMT_MSA.1-aclres2], and [FMT_MSA.3-aclres] which specify how the Reservation Queue ACLs are managed by the appropriate users • [FMT_MSA.1-aclsrv] and [FMT_MSA.3-aclsrv] which specify how the Server ACLs are managed by the appropriate users • [FMT_MSA.1-job] and [FMT_MSA.3-job] which specify how user access to Jobs is managed by the appropriate users • [FMT_MTD.1-role] which specifies the set of User Roles that can manage the User Roles • [FMT_SMF.1] which specifies that the Server, Queue, and Reservation Queue ACLs and the User Roles can be managed by the TOE
O.User.Authenticated	The objective that • the TOE shall require identification and authentication of a Job Process to its Subordinate MOM is met by: • [FIA_UID.2-job] which identifies a Job Process to its MOM via a Job Cookie • [FIA_UAU.2] which authenticates a Job Process to its MOM via a Job Cookie • [FIA_USB.1] which binds the identified user to a process • [FIA_SOS.2] which states that the MOM or Mother Superior creates a Job Cookie that's used as a shared secret between a Job Process and its MOM
O.User.Identified	The objective that • the TOE shall require identification of all users that access the TOE and for a User Role to be associated with each user is met by: • [FIA_ATD.1] which specifies the user security attributes associated with a TOE user • [FIA_UID.2-usr] which identifies TOE users • [FIA_USB.1] which binds the identified user to a process

Security objectives	Rationale
	• [FMT_SMR.1] which specifies the User Roles associated with a user
O.Resource.Availability	The objective that • the TOE shall provide functionality to control the use of resources such that a denial of service will not occur due to the monopolization of TOE controlled resources is met by: • [FMT_MTD.1-rsrc] which specifies the set of User Roles that can manage resource quotas • [FRU_RSA.1-env] which specifies the TOE controlled environment resources • [FRU_RSA.1-time] which specifies the TOE controlled time resources • [FRU_RSA.1-toe] which specifies the TOE controlled TOE resources

Table 7: Security objectives for the TOE rationale

6.2.3 Security Requirements Dependency Analysis

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
FDP_ACC.1-aclque	FDP_ACF.1	FDP_ACF.1-aclque
FDP_ACC.1-aclsrv	FDP_ACF.1	FDP_ACF.1-aclsrv
FDP_ACC.1-job	FDP_ACF.1	FDP_ACF.1-job
FDP_ACC.1-role	FDP_ACF.1	FDP_ACF.1-role
FDP_ACF.1-aclque	FDP_ACC.1	FDP_ACC.1-aclque
	FMT_MSA.3	FMT_MSA.3-aclque FMT_MSA.3-aclres
FDP_ACF.1-aclsrv	FDP_ACC.1	FDP_ACC.1-aclsrv
	FMT_MSA.3	FMT_MSA.3-aclsrv
FDP_ACF.1-job	FDP_ACC.1	FDP_ACC.1-job
	FMT_MSA.3	FMT_MSA.3-job
FDP_ACF.1-role	FDP_ACC.1	FDP_ACC.1-role

Security Functional Requirement	Dependencies	Resolution
	FMT_MSA.3	The Role Access Control Policy is hardcoded in the Server and contains no manageable attributes.
FIA_ATD.1	No dependencies.	
FIA_SOS.2	No dependencies.	
FIA_UAU.2	FIA_UID.1	FIA_UID.2-job
FIA_UID.2-job	No dependencies.	
FIA_UID.2-usr	No dependencies.	
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1-aclque	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-aclque
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1-aclres1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-aclque
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1-aclres2	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-aclque
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1-aclsrv	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-aclsrv
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1-job	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-job
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3-aclque	FMT_MSA.1	FMT_MSA.1-aclque
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3-aclres	FMT_MSA.1	FMT_MSA.1-aclres1 FMT_MSA.1-aclres2
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3-aclsrv	FMT_MSA.1	FMT_MSA.1-aclsrv

Security Functional Requirement	Dependencies	Resolution
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3-job	FMT_MSA.1	FMT_MSA.1-job
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1-role	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1-rsrc	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.2-usr
FRU_RSA.1-env	No dependencies.	
FRU_RSA.1-time	No dependencies.	
FRU_RSA.1-toe	No dependencies.	

Table 8: TOE SFR dependency analysis

6.3 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.

6.4 Security Assurance Requirements Rationale

The evaluation assurance level has been chosen commensurate with the threat environment that is experienced by typical consumers of the TOE. In addition, the evaluation assurance level has been augmented with ALC_FLR.1 commensurate with the augmented flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7 TOE Summary Specification

7.1 TOE Security Functionality

The following section explains how the security functions are implemented. The different TOE security functions cover the various SFR classes.

The primary security features of the TOE are:

- Identification and Authentication
- Access Control
- Resource Allocation Quotas
- Management

7.1.1 Identification and Authentication (I&A)

The TOE supports an identification mechanism and an authentication mechanism. This section describes the supported mechanisms.

7.1.1.1 Identification of Server Users

When a user's process (i.e., Command Process) contacts the Server, the process does it through an unprivileged port. The process also executes a local privileged (setuid root) process that connects to the Server via a privileged port. The user's process passes to the privileged process the unprivileged port that it is using to contact the Server. The privileged process then sends the following data:

- the user name associated with the real user ID of the process that invoked it from the local password file
- the host name of the privileged process
- the unprivileged port number used by the process that invoked it to connect to the Server

The Server compares the data sent via the privileged port to its list of unauthenticated connections to identify the unprivileged port user.

This section maps to the following SFRs:

- FIA_ATD.1
- FIA_UID.2-usr
- FIA_USB.1

7.1.1.2 Identification and Authentication of Job Processes

A MOM (or, in a multi-Vnode case, the Mother Superior) creates a token known as a Job Cookie for every job that it receives and adds the Job Cookie to the job. When the job executes, it executes as one or more Job Processes. When a Job Process sends a request to its MOM, the process uses an unprivileged port to connect to the MOM because the MOM executes the job as the user who submitted the job, who is not necessarily a privileged user (i.e. not necessarily root). To prove to the MOM that it is authorized to make a request, the Job Process passes the Job Cookie as part of the request to the MOM; thus, the Job Cookie is used as both an identifier and as a shared secret between the Job Process and its MOM.

A Job Cookie is a 128-bit pseudo-random number. The 32-bit pseudo-random generator used to create the Job Cookie is part of the Operational Environment.

This section maps to the following SFRs:

- FIA_SOS.2
- FIA_UAU.2
- FIA_UID.2-job

7.1.2 Access Control

7.1.2.1 Role Access Control

The Server enforces the following access control policy between processes acting on behalf of a user (i.e., Command Processes) and the following objects:

- Jobs
- Queues
- Reservation Queues
- Servers
- Vnodes

Each TOE user has one of the User Roles defined in section 7.1.4 associated with it. When the user performs an operation on an object listed above or on an attribute of one of these objects, the TOE uses the user's role to help determine the user's access rights. If the role does not allow the user access, then access is denied.

The Server does not allow this policy to be actively managed (i.e., this policy is hardcoded in the Server); therefore, new roles cannot be added to the TOE and the policy cannot be modified. The Server does allow Managers to assign users to the different User Roles. See section 7.1.4 for more information on User Role management.

This section maps to the following SFRs:

- FDP_ACC.1-role
- FDP_ACF.1-role

7.1.2.2 Queue ACLs

Queue ACLs specify who can perform enqueue job operations (i.e. submit to this Queue, move to this Queue, route to this Queue) on a Queue. Each Queue has the following ACLs associated with it:

- Host ACL
- User ACL
- Group ACL

Managers can create Queues and specify the initial ACL values. Both Managers and Operators can manage Queue ACLs.

Each ACL can be individually enabled or disabled by a Manager or Operator on an existing Queue. Also, the contents of each ACL can be managed by a Manager or Operator on an existing Queue. If an ACL is enabled, the contents of the ACL are evaluated. If an ACL is disabled, the ACL is ignored.

All enabled ACLs on the Queue must allow the subject access in order for an enqueue operation to be allowed; otherwise, the operation is denied. There exists an exception to this rule that allows either a Manager or an Operator to move an existing job to this Queue when this Queue's Host ACL allows access (i.e., is enabled and allows access or is disabled) and simultaneously at least one of this Queue's other two ACLs (i.e., User ACL and Group ACL) is enabled and denies access. If all ACLs are disabled, access is granted and the operation is allowed. By default, all ACLs are disabled allowing all users access to the Queue.

Host ACLs contain host names. The subject's host name is compared to the entries in the Host ACL.

User ACLs contain a user name / host name combination. The subject's user name / host name combination is compared to the entries in the User ACL.

Group ACLs contain Server operating system group names. The subject's default group name on the Server is compared to the entries in the Group ACL.

This section maps to the following SFRs:

- FDP_ACC.1-aclque
- FDP_ACF.1-aclque
- FMT_MSA.1-aclque
- FMT_MSA.3-aclque

7.1.2.3 Reservation Queue ACLs

Reservation Queue ACLs specify who can perform enqueue job operations (i.e. submit to this Queue, move to this Queue, route to this Queue) on a Reservation Queue. Each Reservation Queue has the following ACLs associated with it:

- Host ACL
- User ACL
- Group ACL

All user roles can create Reservation Queues and specify the initial ACL values. A User ACL will always be enabled on a Reservation Queue when it is first created. By default, the TOE will set the User ACL to allow only the creator access, but the creator can specify different User ACL values at the time of creation. Also by default, the Host ACL and Group ACL are disabled, but the creator can specify a Host ACL and/or Group ACL at the time of creation. Once created, if the creator is a Regular User, the creator will not be able to modify the ACLs or enable/disable the ACLs.

Each ACL can be individually enabled or disabled by a Manager or Operator on an existing Reservation Queue. Also, the contents of each ACL can be managed by a Manager or Operator on an existing Reservation Queue. If an ACL is enabled, the contents of the ACL are evaluated. If an ACL is disabled, the ACL is ignored.

All enabled ACLs on the Reservation Queue must allow the subject access in order for an enqueue operation to be allowed; otherwise, the operation is denied. There exists an exception to this rule that allows either a Manager or an Operator to move an existing job to this Reservation Queue when this Queue's Host ACL allows access (i.e., is enabled and allows access or is disabled) and simultaneously at least one of this Queue's other two ACLs (i.e., User ACL and Group ACL) is enabled and denies access. If all ACLs are disabled, access is granted and the operation is allowed.

Host ACLs contain host names. The subject's host name is compared to the entries in the Host ACL.

User ACLs contain a user name / host name combination. The subject's user name / host name combination is compared to the entries in the User ACL.

Group ACLs contain Server operating system group names. The subject’s default group name on the Server is compared to the entries in the Group ACL.

This section maps to the following SFRs:

- FDP_ACC.1-aclque
- FDP_ACF.1-aclque
- FMT_MSA.1-aclres1
- FMT_MSA.1-aclres2
- FMT_MSA.3-aclres

7.1.2.4 Server ACLs

Server ACLs specify who can access the Server. Each Server has the following ACLs associated with it:

- Host ACL
- User ACL

Each ACL can be individually enabled or disabled by a Manager. Also, the contents of each ACL can be managed by a Manager. If an ACL is enabled, the contents of the ACL are evaluated. If an ACL is disabled, the ACL is ignored.

All enabled ACLs must allow the subject access in order for the requested operation to be allowed; otherwise, the operation is denied. If no ACLs are enabled, access is granted and the operation is allowed. By default, both ACLs are disabled allowing all users access to the Server. The Server allows for the use of alternate default ACL values (both ACL contents and ACL enable /disable flags) upon creation / instantiation of the Server daemon. These alternate values are configurable by a Manager.

There exists one special case. If the User ACL is enabled and the user is a Manager or Operator, then the User ACL always allows the user access.

Host ACLs contain host names. The subject’s host name is compared to the entries in the Host ACL.

User ACLs contain a user name / host name combination. The subject’s user name / host name combination is compared to the entries in the User ACL.

This section maps to the following SFRs:

- FDP_ACC.1-aclsrv
- FDP_ACF.1-aclsrv
- FMT_MSA.1-aclsrv
- FMT_MSA.3-aclsrv

7.1.2.5 Job Access Control

By default, users can only access their own jobs. The TOE can modify user access to jobs through the *query_other_jobs* attribute. When this attribute is set to **false**, Regular Users can only perform operations on their own jobs. They cannot access another user's job. When this attribute is set to **true**, Regular Users can perform operations on their own jobs and they can query the status of another user's job.

Only Managers can set the value of this attribute. Managers and Operators are also allowed to perform Regular User operations on any user's job.

This section maps to the following SFRs:

- FDP_ACC.1-job
- FDP_ACF.1-job
- FMT_MSA.1-job
- FMT_MSA.3-job

7.1.3 Resource Allocation Quotas

The TOE enforces maximum resource quotas on users, groups of users, and jobs to counter denial of service issues. Below is the list of enforced quotas. The terms in parentheses are the resource attribute names used in the TOE guidance.

- Number of processors used by a user, group of users, or job at any given time (**ncpus**)
- Amount of physical memory used by a user, group of users, or job at any given time (**mem**)
- Amount of virtual memory used by a user, group of users, or job at any given time (**vmem**)
- Cumulative elapsed time used by a user, group of users, or job (**walltime**)
- Cumulative processor time used by a user, group of users, or job (**cput**)
- Number of executing jobs for a user or group of users at any given time (**max_user_run, max_group_run**)

The TOE allows Managers and Operators to modify these resource quota values.

This section maps to the following SFRs:

- FMT_MTD.1-rsrc
- FRU_RSA.1-env
- FRU_RSA.1-time
- FRU_RSA.1-toe

7.1.4 Management

The TOE supports the following authorized User Roles:

- Managers
- Operators
- Regular Users

Managers are administrators that have the greatest power for managing and configuring the TOE. Operators have less power than Managers, but have some power to manage and configure the TOE. Regular Users have no power to manage and configure the TOE, but they can submit jobs to the TOE for processing.

Only Managers can manage (modify) the User Role security attribute of a user. Through this attribute, Managers can control which users are Managers, Operators, and Regular Users.

In addition, the TOE provides functions for the management of ACLs and users.

This section maps to the following SFRs:

- FMT_MTD.1-role
- FMT_SMF.1
- FMT_SMR.1

8 Abbreviations, Terminology and References

8.1 Abbreviations

ACL	Access Control List
DNS	Domain Name Service
IPsec	Internet Protocol Security
MOM	Machine Oriented Miniserver
PBS	Portable Batch System
SFR	Security Functional Requirement
SSL	Secure Socket Layer
ST	Security Target
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality

8.2 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.

Authorized Users Any authorized User Role (i.e. Manager, Operator, and Regular User).

Manager A user authorized to use all capabilities of the product including administrative capabilities.

Operator A user authorized to use most of the capabilities of the product including some administrative capabilities (i.e. a superset of the Regular User capabilities, but a subset of the Manager capabilities).

Product The term product is used to define software components that comprise PBS Professional.

Regular User A user authorized to use the product, but does not have product administrative capabilities (i.e. does not have the administrative capabilities of a Manager nor Operator).

Subject A subject is a process representing a user.

User Security Attributes Defined by functional requirement FIA_ATD.1, every user is associated with one or more security attributes which allow the TOE to enforce its security functions on this user.

Vnode A Virtual Node (Vnode) is an object representing a set of resources which form a usable part of a computer. Architecturally speaking, a Vnode can be an entire host computer or it can be a nodeboard or a blade. In the evaluated configuration, a Vnode is an entire host computer.

8.3 References

CC **Common Criteria for Information Technology Security Evaluation, CCMB-2006-09-001 Version 3.1 Revision 1 September 2006, CCMB-2007-09-002 Version 3.1 Revision 2 September 2007, CCMB-2007-09-003 Version 3.1 Revision 2 September 2007**
Version (specified above)
Date (specified above)
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R1.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R2.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R2.pdf

CEM **Common Methodology for Information Technology Security Evaluation**
Version 3.1R2
Date September 2007
Location http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R2.pdf