IBM
Logical Partition Architecture
for Power7
Security Target

Version 0.33
March 8, 2013

Prepared for:
International Business Machines Corporation

Rochester, MN 55901

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

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Logical Partition Architecture for Power7 provided by International Business Machines Corporation. The Logical Partition Architecture for Power7 (LPAR) is a product that facilitates the sharing of hardware resources by disparate applications (e.g., AIX, Linux). The product is based on the concept of a 'hypervisor' that is designed to instantiate 'partitions', each with its own distinct resources, that each appear to their hosted applications as a completely functional underlying platform. These partitions are implemented to prevent interference among partitions and to prevent simultaneous sharing of storage and other device resources.

The Security Target contains the following additional sections:

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

1.1 Security Target, TOE and CC Identification

ST Title – IBM Logical Partition Architecture for Power7 Security Target

ST Version – Version 0.33

ST Date – 08 March 2013

TOE Identification – IBM Logical Partition Architecture for Power7 operating on IBM Power Systems hardware with AH730_087 or AM740_088

TOE Developer – IBM

Evaluation Sponsor – IBM

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

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 3, July 2009.
    - Part 3 Conformant
    - Assurance Level: EAL 4 augmented with ALC_FLR.2

-

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is Logical Partition Architecture for Power7.

While the TOE is designed to generally support the entire line of IBM Power Systems products, it has been evaluated and tested in the context models 770 (AM740_088 firmware)_ and 795 (AH730_087).

2.1 TOE Overview

The TOE firmware designed to abstract and virtualize physical hardware resources to provide the underlying platform for one or more concurrent operating systems. Each virtual platform is known as a partition. The operating systems executing in the available partitions are treated as subjects of the TOE, where the TOE not only provides the necessary operational support for the hosted operating systems, but also serves to separate them from each other to ensure mutual non-interference.

While not included as part of the TOE, the TOE is configured using a connected Hardware Management Console (HMC) that provides access to the functions necessary to enable administrative personnel to effectively manage the allocation of resources (i.e., processors, memory, and I/O devices) to the configured partitions. Once the TOE is configured, the HMC is expected to be disconnected so that it offers no interfaces while the TOE is operating in its evaluated configuration.

2.2 TOE Architecture

a. The TOE consists of the PowerVM Hypervisor **(PHYP):** which provides virtualization and other advanced server

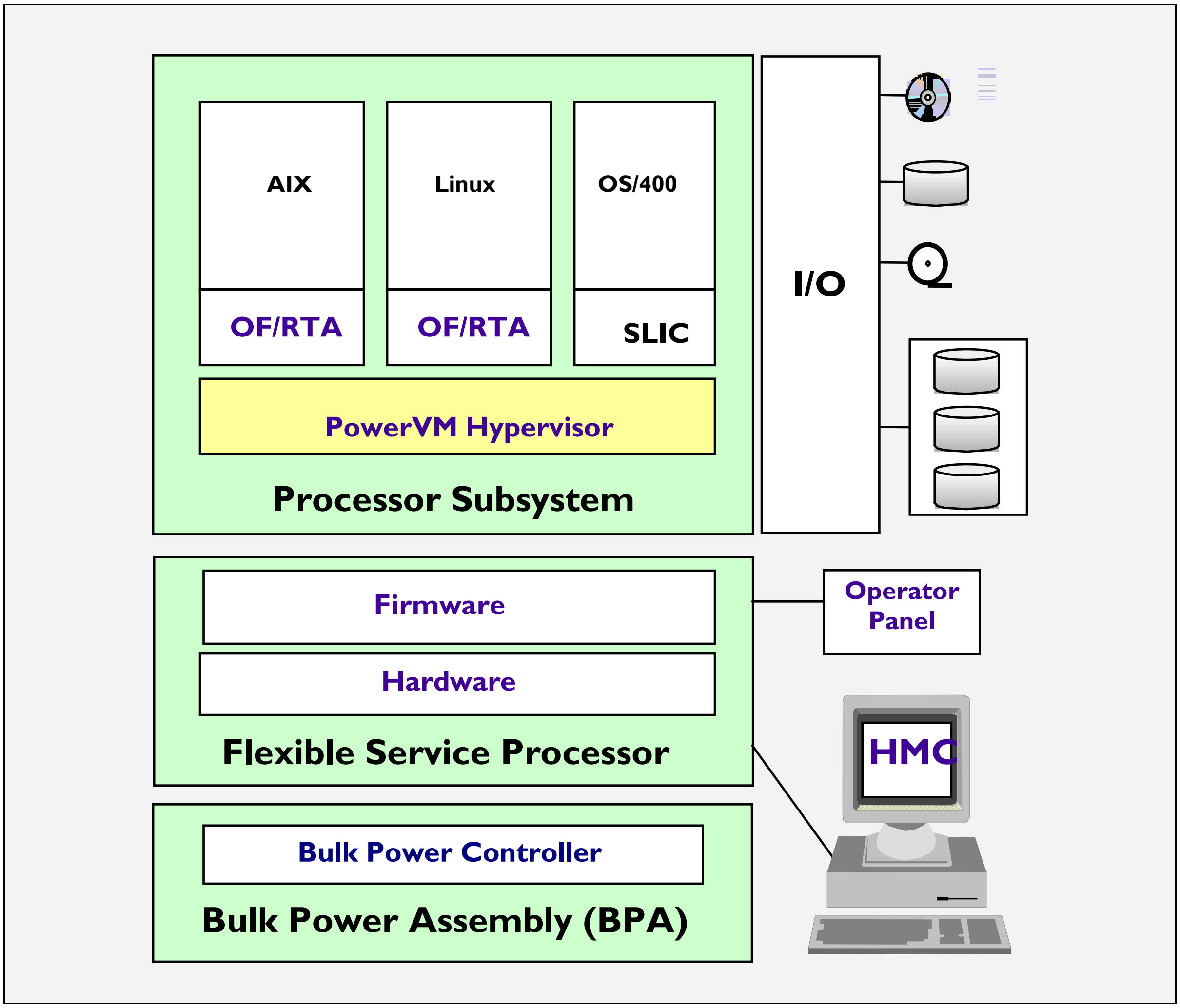

Figure 1: LPAR Architecture

Note that Figure 1 identifies the TOE components in the yellow-filled (PowerVM Hypervisor). The other components (HMC, FSP, BPA and operating systems) are outside scope of the TOE.

2.2.1 Physical Boundaries

As indicated above, the TOE consists of a number of architectural components. The components expose a number of interfaces both externally and internally.

The external interfaces include the interfaces to the subject operating in a partition. These include the Hypervisor interfaces as well as the hardware instructions available to applications. Note that when operating in the evaluated configuration, the Hardware Management Console (HMC) used to configure the TOE is detached and, hence, does not represent an interface. There is also an operator panel where basic, non-security related operator functions can be performed by a user with direct physical access to the TOE.

The internal interfaces, specifically those not also available externally, include the FSP interface to the Hypervisor.

I/O represents the physical I/O slots either integrated into the hardware drawers or I/O drawers external to the server. The I/O adapters allow for the connection of disk, network, SAN, tape and other individual I/O devices.

Note that connections to a broad or public network are supported, but they would be treated as resources that can be granted to partitions for operating system use, but would not be used by TOE for its own purposes. Along these lines, while the TOE controls which devices a given partition can access, it does not control or otherwise constrain the nature of those devices. Any functions or connections of those devices are outside the scope of control of the TOE.

2.2.2 Logical Boundaries

The physical boundaries can then be broken down into individual logical components. For example, a physical drawer may contain 8 different I/O devices and these individual devices are assigned by the Hardware Management Console (HMC) to the configured virtual machines (partitions). When assigned to a partition, the logical I/O devices are available to be used by the partition (disk, network, tape and such).

This section summarizes the security functions provided by Logical Partition Architecture for Power7:

- User data protection
- Identification and authentication
- Security management
- Protection of the TSF

2.2.2.1 User data protection

The Hypervisor manages the association of CPUs, memory, and I/O devices, in a relatively static environment, with partitions containing operating system instances. Memory and I/O devices can be assigned to single partitions and when assigned are accessible only by the partition (including OF/RTAS and the OS running in the partition). CPUs can also be assigned a single partition, and only that partition (and occasionally the TOE) can use that CPU. CPUs can also be configured to be shared among a collection of partition (shared processor partition or also called micro-partitions) and the Hypervisor will save/restore the hardware register state when switching between partitions.

The Hypervisor also provides a mechanism where users can create LPAR groups (also referred to as eWLM groups) where a list of partitions are allowed to share the quantity of resources (memory and processors but not I/O) between the partitions. The resource is still owned at any point in time by one and only one partition but the operating system is given the ability to remove the resource from one partition and another partition can add the resource to their partition in the same group. The Hypervisor clears out the state of the resource before it is moved between partitions.

Partitions have no control over the resources they are assigned. The Hypervisor receives the partition management information from the HMC when it is being configured. Once configured, the HMC is disconnected and the TOE is placed in an operational state where those assignments would be continuously enforced.

2.2.2.2 Identification and authentication

Partitions are implicitly identified and authenticated by internal numerical identifiers associated with partitions (using internal data structures) as they are defined. Being implicitly identified by the TOE, partitions have no need, nor means, to identify themselves. Furthermore, the identification of a partition is guaranteed by the TOE and as such each partition is also continuously authenticated.

2.2.2.3 Security management

All of the TOE configuration occurs via the interface to the HMC. Since the HMC is disconnected while the TOE is operational the TOE effectively doesn’t offer any security management functions. However, the TOE serves to restrict the ability to change its own configuration nonetheless.

2.2.2.4 Protection of the TSF

The components of the TOE protect themselves using the domains provided by the Power7 processors. The TOE operates in the privileged domain and the partitions operate in the unprivileged domain. This allows the TOE to protect itself as well as the resources it makes selectively available to the applicable partitions.

Beyond protecting itself and its resources, the TOE is also designed such that when the hardware that supports a partition fails, the other partitions will continue uninterrupted.

2.3 TOE Documentation

IBM offers a series of documents that describe the installation process for LPAR as well as guidance for subsequent use and administration of the applicable security features.

3. Security Problem Definition

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

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

Furthermore, the TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL4 augmented with ALC_FLR.2 as defined in the CC.

3.1 Threats

T.ACCESS An entity operating within a partition may be able to gain access to resources that belong to another partition as configured by an authorized user.

T.COMMUNICATE An entity operating within a partition may be able to establish a communication channel with another partition.

T.INTERFERE An entity operating within a partition may be able to disrupt the operation of another partition.

3.2 Assumptions

A.CONNECT The TOE is assumed to be appropriately installed, including connections to device resources as well as being disconnected from the management console when operational.

A.LOCATE The TOE and its connections are assumed to be physically protected from unauthorized access or modification.

A.MANAGE The TOE is assumed to be managed by users who are capable and trustworthy and will follow the applicable guidance correctly.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats and address applicable assumptions.

4.1 Security Objectives for the TOE

O.AUTHORIZATION The TOE must ensure that resources can be assigned to partitions only by an authorized user and that those resources will not be accessible to other partitions.

O.COMMUNICATION The TOE must not provide a direct means of communication between partitions.

O.NONINTERFERE The TOE must ensure that each partition cannot access resources or communicate with other partitions.

4.2 Security Objectives for the Environment

OE.ADMIN A suitable management console must be configured for use by a capable and trustworthy user assigned to follow the applicable guidance in order to install and operate the TOE in a secure manner.

OE.INSTALL The TOE must be installed and configured in accordance with its guidance documents, including connecting appropriate device resources and disconnecting the management console when the TOE is operational.

OE.PHYSICAL The TOE must be established in a physical environment suitable to protect itself and its external connections from inappropriate access and modification.

5. IT Security Requirements

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

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Logical Partition Architecture for Power7.

Requirement Class	Requirement Component
FDP: User data protection	FDP_ACC.2: Complete access control
	FDP_ACF.1: Security attribute based access control
	FDP_IFC.2: Complete information flow control
	FDP_IFF.1: Simple security attributes
	FDP_RIP.1: Subset residual information protection
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
FPT: Protection of the TSF	FPT_FLS.1: Failure with preservation of secure state

Table 1 TOE Security Functional Components

5.1.1 User data protection (FDP)

5.1.1.1 Complete access control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the [**Resource Access Control Policy**] on [**subjects: partitions and objects: logical and physical CPUs, logical and physical memory, and logical and physical I/O devices**] and all operations among subjects and objects covered by the SFP.

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

5.1.1.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Resource Access Control Policy**] to objects based on the following: [**partition, logical and physical CPU, logical and physical memory, and logical and physical I/O device 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: [**a given partition can access only logical and physical CPUs, logical and physical memory, and logical and physical I/O devices explicitly assigned to it**].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**no explicit authorization rules**].

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

5.1.1.3 Complete information flow control (FDP_IFC.2)

FDP_IFC.2.1 The TSF shall enforce the [**Partition Separation Policy**] on [**partitions and attached resource contents**] and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

5.1.1.4 Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the [**Partition Separation Policy**] based on the following types of subject and information security attributes: [**partition identities and no attached resource content attributes**].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [**I/O devices have been associated with partitions such that those devices enable some means of communication via their contents outside the scope of the TOE**].

FDP_IFF.1.3 The TSF shall enforce the [

1) **partitions cannot communicate with one another using physical CPU or physical memory resource contents;**
2) **partitions assigned to a group can release logical CPU and logical memory resources and those resources can be acquired by another partition within the same group; and**
3) **when a physical CPU is designated as shared, it can be assigned to partitions in successive time slots**

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [**no explicit authorization rules**].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [**no explicit denial rules**].

5.1.1.5 Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [**physical CPUs and physical memory**].

5.1.2 Identification and authentication (FIA)

5.1.2.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 Refinement: The TSF shall maintain the following list of security attributes belonging to individual **partitions** ~~users~~: [**unique partition id**].

5.1.2.2 User-subject binding (FIA_USB.1)

FIA_USB.1.1 Refinement: The TSF shall associate the following user security attributes with subjects acting on the behalf of that **partition** ~~user~~: [**unique partition id**].

FIA_USB.1.2 Refinement: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of **partitions** ~~user~~: [**partitions are identified internally when defined**].

FIA_USB.1.3 Refinement: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of **partitions** ~~user~~: [**partition security attributes do not change once assigned**].

5.1.3 Security management (FMT)

5.1.3.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Resource Access Control Policy and Partition Separation Policy**] to restrict the ability to [***modify***] the security attributes [**partition and resource identities (and association of resources to partitions)**] to [**no user**[1]].

5.1.3.2 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Resource Access Control Policy and Partition Separation Policy**] to provide [***restrictive***[2]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**no user**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4 Protection of the TSF (FPT)

5.1.4.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [**memory and I/O device failures**].

5.2 TOE Security Assurance Requirements

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

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: basic design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused vulnerability analysis

Table 2 EAL 4 augmented with ALC_FLR.2 Assurance Components

[1] The intention here is to indicate that the TOE does not allow any modifications to security attributes while it is operational. Note that this applies to potential changes associated with FMT_MSA.3 as well.

[2] The policy is restrictive in that resources can be accessed only after being explicitly assigned to a partition and that a given resource can be assigned only to a single partition.

5.2.1 Development (ADV)

5.2.1.1 Security architecture description (ADV_ARC.1)

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

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

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

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

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

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

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

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

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

5.2.1.2 Complete functional specification (ADV_FSP.4)

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

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

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

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

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

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

ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

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

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

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

5.2.1.3 Implementation representation of the TSF (ADV_IMP.1)

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

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

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

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

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

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

5.2.1.4 Basic modular design (ADV_TDS.3)

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

ADV_TDS.3.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

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

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

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

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

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

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

ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.

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

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

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

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

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

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

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

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

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

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

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

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

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

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

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

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

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

AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

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

AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

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

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

5.2.3.2 Problem tracking CM coverage (ALC_CMS.4)

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

5.2.3.3 Delivery procedures (ALC_DEL.1)

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

5.2.3.4 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.
ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.2.3.5 Flaw reporting procedures (ALC_FLR.2)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.2.3.7 Well-defined development tools (ALC_TAT.1)

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

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

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

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

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

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

5.2.4 Tests (ATE)

5.2.4.1 Analysis of coverage (ATE_COV.2)

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

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

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

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

5.2.4.2 Testing: basic design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

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

5.2.4.3 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.

ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, expected test results and actual test results.

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

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

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

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

5.2.4.4 Independent testing - sample (ATE_IND.2)

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

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

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

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

ATE_IND.2.2e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

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

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Focused vulnerability analysis (AVA_VAN.3)

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

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

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

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

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

AVA_VAN.3.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 User data protection

The TOE is designed to instantiate partitions for the purpose of supporting multiple simultaneous operating systems. As such, it implements a policy where by partitions can access only those resources explicitly assigned to it.

In terms of access control, CPU, memory, and I/O devices can be assigned to a given partition and a partition can access those resources only when they are assigned to it. This is accomplished using hardware features supporting the mapping of these resources to established partitions. Hence, even when using hardware instructions directly, a partition cannot directly perceive that other resources may exist. During operation of the TOE, CPU, memory, and I/O device resources can be assigned to only a single partition at any given point in time and cannot be simultaneously shared among partitions.

Normally, CPU, memory, and I/O resources are permanently assigned to a partition at configuration time. Alternately, partitions can be placed in groups (one per partition) and partitions within those groups can release CPU and memory resources and alternately acquire available CPU and memory resources, though they can be accessed by only a single partition at any given time. Also, a given CPU can be configured to be shared among partitions and subsequently partitions can utilize that CPU, one at a time, based on available time slots.

In terms of communication, a user can optionally choose to configure a virtual communication path between partitions. Also, partitions can be assigned to devices (NICs for example) and those devices might be capable of enabling some means of communication outside the scope of control of the TOE.

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

- FDP_ACC.2: The TOE controls all operations that a partition may perform on CPU, memory, and I/O device resources by allowing partitions to access (in any manner) only the resources explicitly assigned to it.
- FDP_ACF.1: As indicated above, partitions can access only those resources that have been assigned to it.
- FDP_IFC.2: The TOE offers no means of direct communication among partitions, so all means of inter-partition communication within the scope of the TOE are controlled.
- FDP_IFF.1: CPU, memory, and I/O device resources can be assigned to only one partition at a time. CPUs, memory, and I/O devices cannot be dynamically re-allocated, though they could be reallocated when the TOE is reconfigured while not in an operational state.
- FDP_RIP.1: When a partition initially starts and when it is assigned a new CPU, the corresponding CPU context is initialized to a known state appropriate to the partition (either a new starting state when initially assigned or restoration of the previous partition state when reassigned). In the case of memory, the volatile nature of the memory ensures it is clear when the TOE starts operation. When memory is acquired by a partition after start-up, it is cleared of any residual data before it can be accessed. *Note that I/O devices cannot be addressed with this claim since essentially any I/O device could be used and the TOE does not have the ability to clear the contents of all applicable I/O devices. Hence, it is left to the partitions themselves to address any associated issues related to reuse of information in devices when the TOE is reconfigured such that a device may be reassigned to a different partition.*

6.1.2 Identification and authentication

The TOE is aware of one type of active entity (users): partitions which it instantiates. *Note that the HMC is assumed to be disconnected while the TOE is operational and there is also a directly connected operator panel, it allows only basic functional operations.*

When partitions are defined they are assigned unique numbers in TOE-internal data structures which are subsequently used to identify the partition and to associate resources with the partition. Once a partition is created, its number will not change except when it is deleted and recreated. Given that each partition is uniquely identified by the TOE using TOE-internal data structures, the TOE effectively ensures that each partition is authentic on a continuous basis.

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

- FIA_ATD.1: Each partition is identified by a unique partition number by the TOE and there is only one HMC identified by virtue of its dedicated physical connection to the TOE.
- FIA_USB.1: Unique identifying partition numbers are assigned when partitions are created and cannot change except by deleting and recreating a partition.

6.1.3 Security management

All functions to configure the TOE are available only through the dedicated physical HMC interface. However, the HMC is expected to be disconnected while the TOE is operational and as a result the HMC is outside the scope of evaluation. Regardless, the HMC allows a user of the HMC to create partitions and to assign CPU, memory, and I/O device resources to those partitions. Furthermore, each given resource can be assigned only to a single partition. The resulting configuration data is pushed to the TOE prior to it being placed in an operational, evaluated configuration.

When operational, the TOE restricts the security management functions by offering no interfaces to manipulate them to its subjects (i.e., partitions). The available interfaces (i.e., PowerPC Hypervisor) offer no ability to perform any security management related function and as summarized below, the architecture of the TOE prevents bypass and tampering of its mechanisms to ensure that inappropriate users cannot perform any security management functions.

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

- FMT_MSA.1: The only interfaces available to manipulate the assignment of resources to partitions are offered through the dedicated HMC connection.
- FMT_MSA.3: Partitions cannot access resources until they are defined and explicitly assigned resources via the HMC. The only interfaces available to create partitions and manipulate the assignment of resources to partitions are offered through the dedicated HMC connection.

6.1.4 Protection of the TSF

The FSP firmware depend on the FSP hardware (i.e., IBM Power7) to provide a separate domain for its execution.

The Power7 hardware provides a privileged mode of execution specifically for the Hypervisor firmware. Only the Hypervisor firmware executes in that mode and it is only from this privileged execution mode that full, unconstrained access to the available resources (CPUs, memory, and I/O devices) is available. Even though the Hypervisor shares the available CPUs with its instantiated partitions, the contexts of the CPUs are saved and restored appropriately during every context switch to ensure uninterrupted operation of the Hypervisor and the partitions.

The Hypervisor firmware instantiates partitions that execute in other execution modes offered by the Power7. Additionally, those partitions can access only those resources that have been specifically allocated for use by the associated partitions. While a partition can freely access the resources it has been assigned, there are no interfaces that might allow access to (or even the perception of) other unassigned or otherwise assigned resources.

The TOE ensures that its security mechanisms cannot be bypassed by encapsulating partitions with their assigned resources and offering only limited interfaces that are designed to ensure that partitions cannot interfere with other partitions or the TOE’s own operation.

When the TOE detects a memory or I/O device failure, the TOE will shut itself down. Given that the TOE is configured and stored in firmware, it will be restored to its previous state when it is restarted. While the contents of a given partition could potentially be corrupted, the TOE itself cannot be corrupted by transient failures (such as memory errors).

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

- FPT_FLS.1: When memory or I/O device errors are detected by the TOE, it shuts down and when restarted would revert to its previously secure configuration as defined in firmware.

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

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

8.1.1 Security Objectives Rationale for the TOE and Environment

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

	T.ACCESS	**T.COMMUNICATE**	**T.INTERFERE**	**A.CONNECT**	**A.LOCATE**	**A.MANAGE**
O.AUTHORIZATION	X					
O.COMMUNICATION		X				
O.NONINTERFERE			X			
OE.ADMIN						X
OE.INSTALL				X		
OE.PHYSICAL					X	

Table 3 Environment to Objective Correspondence

8.1.1.1 T.ACCESS

> *An entity operating within a partition may be able to gain access to resources that belong to another partition as configured by an authorized user.*

This Threat is satisfied by ensuring that:

- O.AUTHORIZATION: By ensuring that resources can be accessed only by the partition assigned by an authorized user, the TOE mitigates the threat of partitions gaining access to resources of other partitions.

8.1.1.2 T.COMMUNICATE

An entity operating within a partition may be able to establish a communication channel with another partition.

This Threat is satisfied by ensuring that:

- O.COMMUNICATION: By ensuring that partitions cannot communicate with one another using any direct means provided by the TOE, the TOE limits the potential for inter-partition communication.

8.1.1.3 T.INTERFERE

An entity operating within a partition may be able to disrupt the operation of another partition.

This Threat is satisfied by ensuring that:

- O.NONINTERFERE: By ensuring that partitions are limited to access their assigned resources ,the TOE mitigates the threat of interference among partitions.

8.1.1.4 A.CONNECT

The TOE is assumed to be appropriately installed, including connections to device resources as well as being disconnected from the management console when operational.

This Assumption is satisfied by ensuring that:

- OE.INSTALL: This objective is intended to directly address the need to ensure that the TOE is appropriately installed and connected to other devices.

8.1.1.5 A.LOCATE

The TOE and its connections are assumed to be physically protected from unauthorized access or modification.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: This objective is intended to directly address the need of physical protection for the TOE and its physical connections.

8.1.1.6 A.MANAGE

The TOE is assumed to be managed by users who are capable and trustworthy and will follow the applicable guidance correctly.

This Assumption is satisfied by ensuring that:

- OE.ADMIN: This objective is intended to directly address the need to assign capable and trustworthy administrators who will adhere to the applicable guidance.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 4** indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

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

	O.AUTHORIZATION	**O.COMMUNICATION**	**O.NONINTERFERE**
FDP_ACC.2	X		X
FDP_ACF.1	X		X
FDP_IFC.2		X	X
FDP_IFF.1		X	X
FDP_RIP.1	X		
FIA_ATD.1	X		
FIA_USB.1	X		
FMT_MSA.1	X		X
FMT_MSA.3	X		X
FPT_FLS.1	X		

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.AUTHORIZATION

The TOE must ensure that resources can be assigned to partitions only by an authorized user and that those resources will not be accessible to other partitions.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2: In order to ensure that resources are restricted to partitions appropriately, an access control policy is defined which covers all resources as well as all operations.
- FDP_ACF.1: In order to ensure that resources are restricted to partitions appropriately, the access control rules ensure that partitions gain access to resources only when they are appropriately configured for that purpose.
- FDP_RIP.1: In order to ensure that resources (including information they contain) are restricted to partitions appropriately, the TOE must ensure that memory and processor resources are cleared when allocated to partitions.
- FIA_ATD.1: In order to limit resource access to specific partitions, the TOE must define identities associated with partitions.
- FIA_USB.1: In order to limit resource access to specific partitions, the TOE must ensure that partitions are continuously identified and that identification cannot change.
- FMT_MSA.1: In order to ensure that resources are managed properly, the TOE must ensure that assignment of resources to partitions cannot be accomplished by unauthorized users.
- FMT_MSA.3: In order to ensure that resources are managed properly, the TOE must ensure that they are not accessible by partitions until they are explicitly assigned.
- FPT_FLS.1: In order to protect against inappropriate resource access, the TOE must protect itself against memory and disk failures.

8.2.1.2 O.COMMUNICATION

The TOE must not provide a direct means of communication between partitions.

This TOE Security Objective is satisfied by ensuring that:

- FDP_IFC.2: In order to limit potential means of communication between partitions, an information flow policy is defined which covers any means of communication between partitions.
- FDP_IFF.1: In order to limit potential means of communication between partitions, the information flow policy rules ensure that inter-process communication is not allowed using any mean provided by the TOE.
-

8.2.1.3 O.NONINTERFERE

The TOE must ensure that each partition cannot access resources or communicate with other partitions.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2: In order to ensure that resources cannot be used for interference among partitions, an access control policy is defined which covers all resources as well as all operations.
- FDP_ACF.1: In order to ensure that resources cannot be used for interference among partitions, the access control rules ensure that partitions gain access to resources only when they are appropriately configured for that purpose.
- FDP_IFC.2: In order to ensure that communication mechanisms cannot be used for interference among partitions, an information flow policy is defined which covers any means of communication between partitions.
- FDP_IFF.1: In order to ensure that communication mechanisms cannot be used for interference among partitions, the information flow policy rules ensure that inter-process communication is allowed only using devices which may be subject to object reuse or other means of communication not controllable by the TOE.
- FMT_MSA.1: In order to protect against configuration-related interference attempts, the TOE must ensure that resource assignments cannot be established by unauthorized users.
- FMT_MSA.3: In order to protect against configuration-related interference attempts, the TOE must ensure that resource access is not allowed until it is explicitly configured.

8.3 Security Assurance Requirements Rationale

The TOE is intended for an environment requiring a moderate to high level of assurance in the security functionality of conventional commodity TOEs, as presented in the statement of security environment (Section 3). The target assurance level of EAL4 augmented with ALC_FLR.2 is appropriate for such an environment.

8.4 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST. As indicated in the table, all of the dependencies are satisfied with the exceptions of FMT_SMR.1 and FMT_SMF.1.

The CC indicates that the depending requirements need a security management role (FMT_SMR.1) and to provide the associated security management functions (FMT_SMF.1). However, the applicable functions are available only when the TOE is offline. While online, the applicable security attributes cannot be changed and the applicable default information flow settings are restrictive (FMT_MSA.1 and FMT_MSA.3). Given that the TOE offers no ability to change the applicable attributes while online, there is no real dependency on FMT_SMF.1 or FMT_SMR.1.

ST Requirement	CC Dependencies	ST Dependencies
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2 and FMT_MSA.3
FDP_IFC.2	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.2 and FMT_MSA.3
FDP_RIP.1	none	none
FIA_ATD.1	none	none

FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	**[FMT_SMR.1]** and [**FMT_SMF.1]** and FDP_ACC.2 and FDP_IFC.2
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and [**FMT_SMR.1]**
FPT_FLS.1	none	none

8.5 Explicitly Stated Requirements Rationale

This Security Target includes no requirements that are not defined in the CC.

8.6 TOE Summary Specification Rationale

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

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 5 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	User data protection	**Identification and authentication**	**Security management**	**Protection of the TSF**
FDP_ACC.2	X			
FDP_ACF.1	X			
FDP_IFC.2	X			
FDP_IFF.1	X			
FDP_RIP.1	X			
FIA_ATD.1		X		
FIA_USB.1		X		
FMT_MSA.1			X	
FMT_MSA.3			X	
FPT_FLS.1				X

Table 5 Security Functions vs. Requirements Mapping

8.7 PP Claims Rationale

See Section 7, Protection Profile Claims.