LPAR for Power4
Security Target

Version: 1.6

Last Update: 05 November 2003

Document History

Version	Date	Changes	Summary	Author
0.1	23-02-03	Initial Version		Helmut Kurth, atsec
0.2	26-02-03	All chapters	First draft version with chapters 1 to 7 completed	Helmut Kurth, atsec
0.3	28-02-03	Mainly chapter 8	Making a few modifications to chapters 1 to 7 with respect to the comments from Dave and George	Helmut Kurth, atsec
1.0	01-03-03	Minor	Last minor changes with respect to comments from different persons	Helmut Kurth, atsec
1.1	11-04-03	Minor	Some minor changes in response to initial ETR for the ST	Helmut Kurth, atsec
1.2	28-05-03	Some	Addressing comments from BSI and the evaluation team	Helmut Kurth, atsec
1.3	01-07-03	Minor	Enhancing readability and understandability, addressing issues of the evaluation team	Helmut Kurth, atsec
1.4	10-07-03	Minor	Addressing further comments from the evaluation team	Helmut Kurth, atsec
1.5	22-09-03	Minor	Updating TOE version numbers	Helmut Kurth, atsec
1.6	05-11-03	Minor	Updating the TOE version numbers	Helmut Kurth, atsec

Table of Contents

References

[CC] Common Criteria for Information Technology Security Evaluation, CCIMB-99-031, Version 2.1, August 1999, Part 1 to 3

[CEM] Common Methodology for Information Technology Security Evaluation, CEM-99/045, Part 2 - Evaluation Methodology, Version 1.0, 1999

[GUIDE] ISO/IEC PDTR 15446 Title: Information technology – Security techniques – Guide for the production of protection profiles and security targets, ISO/IEC JTC 1/SC 27 N 2449, 2000-01-04

[ITSEC] Information Technology Security Evaluation Criteria, Version 1.2, CEC, June 1991

[TARGET] LPAR Security Target (this document)

[OPEN_FIRM] IEEE Std P1275: IEEE Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices, 1994

[OF_PCI_EXT] PCI Bus Binding to: IEEE Std 1275-1994 Standard for Boot (Initialization Configuration) Firmware Revision 2.1

[OF_ISA_EXT] IEEE Draft Std P1275.1/D14a Standard for Boot (Initialization Configuration) Firmware Supplement for IEEE 1754 ISA, 18 August, 1994

[OF_64BIT_EXT] IEEE Draft Std P1275.6/D5 Standard for Boot (Initialization Configuration) Firmware 64 Bit Extensions, March 20, 1995

[CHRP] PowerPC™ Microprocessor Common Hardware Reference Platform (CHRP™) System binding to: IEEE Std 1275-1994 Standard for Boot (Initialization, Configuration) Firmware, Revision: 1.5 (UNAPPROVED DRAFT), Date: May 8, 1996

[CPG] The Complete Partitioning Guide for IBM eServer pSeries Servers, IBM Redbook SG247039, January 2003

1 Introduction

This is version 1.6 of the Security Target document for the evaluation of the IBM LPAR architecture for pSeries “LPAR for Power4, functional release R3”.

1.1. ST Identification

Title: LPAR for Power 4 Security Target Version 1.6

Keywords: Logical Partitioning, POWER4, Open Firmware, virtual machine.

This document is the security target for the CC evaluation of the logical partitioning architecture of the IBM pSeries (LPAR for Power4) and is conformant to the Common Criteria for Information Technology Security Evaluation [CC].

1.2. ST Overview

This security target documents the security characteristics of the IBM logical partitioning architecture (LPAR) for the pSeries: LPAR for Power4, firmware releases 3R031021 (p630), 3K031021 (p650) and 3H031021 (p690) for the pSeries servers.

This architecture allows to define separate logical partitions on the logical partitioning capable pSeries eServers p630, p650 and p690. Each logical partition can operate like a separate machine with its own resources (processors, main memory, I/O devices). A Hypervisor controls the logical partitions and operating systems running in those partitions can communicate with the Hypervisor using defined Hypervisor calls.

1.3. CC Conformance

This ST is CC *Part 2 conformant* and *Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL4 augmented by ALC_FLR.1.

EAL 4 has been augmented by ALC_FLR.1 since this is also covered by the Mutual Recognition Arrangement.

1.4. Strength of Function

No security function of the TOE is based on a permutational or probabilistic algorithm. Therefore no security function is rated according to the criteria for strength of function. As a result the Security Target does not require a strength of function claim.

1.5. Structure

The structure of this document is as defined by [CC] Part 1 Annex C.

- Section 2 is the TOE Description.
- Section 3 provides the statement of TOE security environment.
- Section 4 provides the statement of security objectives.
- Section 5 provides the statement of IT security requirements.
- Section 6 provides the TOE summary specification, which includes the detailed specification of the IT Security Functions.
- Section 7 provides the Protection Profile claim
- Section 8 provides the rationale for the security objectives, security requirements, TOE summary specification and PP claims.

1.6. Terminology

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

Target of Evaluation (TOE): The TOE is defined as the LPAR logical partitioning system for the IBM pSeries, running and tested on the hardware and firmware specified in this Security Target.

User: The TOE does not have a notion of a "user" as a human. Instead the "users" of the TOE are software functions (usually operating systems) operating in one of the partitions defined by the TOE.

Partition: A partition in this Security Target is a collection of memory regions, processors and I/O slots. Those partitions define the operational environment for an operating system to run.

2 TOE Description

2.1 Summary

The target of evaluation (TOE) is the logical partitioning architecture (LPAR) for Power4 for the IBM pSeries systems p630, p650 and p690 (firmware releases 3R031021 (p630), 3K031021 (p650) and 3H031021 (p690)), consisting of the Open Firmware, Run Time Abstraction Layer (RTAS) and the Hypervisor executing in hypervisor mode on the above mentioned hardware platforms.

The logical partitioning capable pSeries eServers p630, p650 and p690 support a logical partitioned environment that enables the pSeries systems to run multiple logical partitions concurrently. The maximum number of partitions that can concurrently run depends on the specific partitioning-capable pSeries server model. For example, the pSeries 690 support up to 16 partitions running concurrently while the pSeries 650 supports up to 8 partitions and the pSeries 630 supports up to 4 partitions..

In a logical partition, an operating system instance runs with dedicated resources: processors, memory, and I/O slots. These resources are statically assigned to the logical partition. The total amount of assignable resources is limited by the physically installed resources in the system.

Because the implementation of logical partitioning is static, one has to shut down every operating system instance in all logical partitions to change the resource assignment of running logical partitions.

From a functional point of view, applications on top of an operating system are running inside partitions in the same way they run on a stand-alone pSeries machine. There are no issues when moving an application from a stand-alone server to a partition. Operating system software needs to be modified in some areas to call Hypervisor functions instead of native code. The design of partitioning-capable pSeries servers is such that one partition is isolated from software running in the other partitions, including protection against natural software defects and even deliberate software attempts to break the partition barriers. It has the following security features:

- Protection against inter-partition data access

The design of partitioning-capable pSeries servers prevents any data access between partitions, other than intended connections via I/O devices (networks, serial links etc.). This protects the partitions against unauthorized from other partitions.

- Unexpected partition crash

A software crash within a partition shall not cause any disruption to the other partitions. Neither an application failure nor an operating system failure inside a partition interfere with the operation of other partitions.

- Denial of service across shared resources

The design of partitioning-capable pSeries servers prevents partitions from making extensive use of a shared resource so that other partitions using that resource become starved. This means that partitions sharing the same PCI bridge chips, for example, cannot occupy the bus indefinitely.

In this way, applications can be safely consolidated in partitions in a partitioning-capable pSeries server without compromising overall system security.

The logical resources that can be assigned to a partition are:

- Processors
- Main memory regions
- I/O slots

The assignment of those resources to the individual logical partitions is stored in non-volatile RAM. This part of the NVRAM is maintained by the service processor and can not be read or modified directly by software running in a logical partition. The assignment itself is performed by a System Administrator, who uses a “Hardware Management Console”

(HMC) to define those assignments. The HMC communicates with a “Service Processor” that accepts the commands from the HMC and sets the values to define the logical partitions in the non-volatile RAM (NVRAM) accordingly.

The NVRAM also contains the NVRAM parts that the operating system would expect when operating in non-partitioned mode. For each partition such a portion of NVRAM is reserved in the overall NVRAM. Access to this NVRAM is established via a modified “Real Time Abstraction Service” (RTAS) layer, which has been adapted to use calls to the Hypervisor to emulate those and other RTAS services.

The LPAR for Power4 in the version that is evaluated allows only dedicated assignment of processors, main memory and I/O slots to individual partitions. Any sharing of those resources between different partitions is not supported by the LPAR architecture. This separation is accomplished by the fact that for each resource there is just one partition number that can be entered into the table in NVRAM. In future releases of this architecture a controlled sharing is planned to be possible.

The separation between the different partitions in accordance with the assignments defined in the NVRAM is then enforced by the hardware with the support of the “Hypervisor”. The Hypervisor is software with special privileges running in a separate piece of memory. Partitions may communicate with the Hypervisor using “Hypervisor Calls” or by a machine check or reset interrupt.

The partitions are managed via a “Hardware Management Console” (HMC), which is connected via a serial link to the “Service Processor”. An administrator uses the HMC to define the partitions with their resources and then transfers this information to the service processor. The service processor stores this information in designated areas of the NVRAM, which is then used by the TOE to define, manage and control the logical partitions.

The TOE itself consists of the “Boot Firmware (Open Firmware) / RTAS / Hypervisor” part shown in figure 1.

The HMC and the service processor are just used for the configuration of the partitions but will not be used in the evaluated configuration to perform any modification to this configuration while the TOE is in operation. Therefore those parts of the system are regarded as part of the TOE environment.

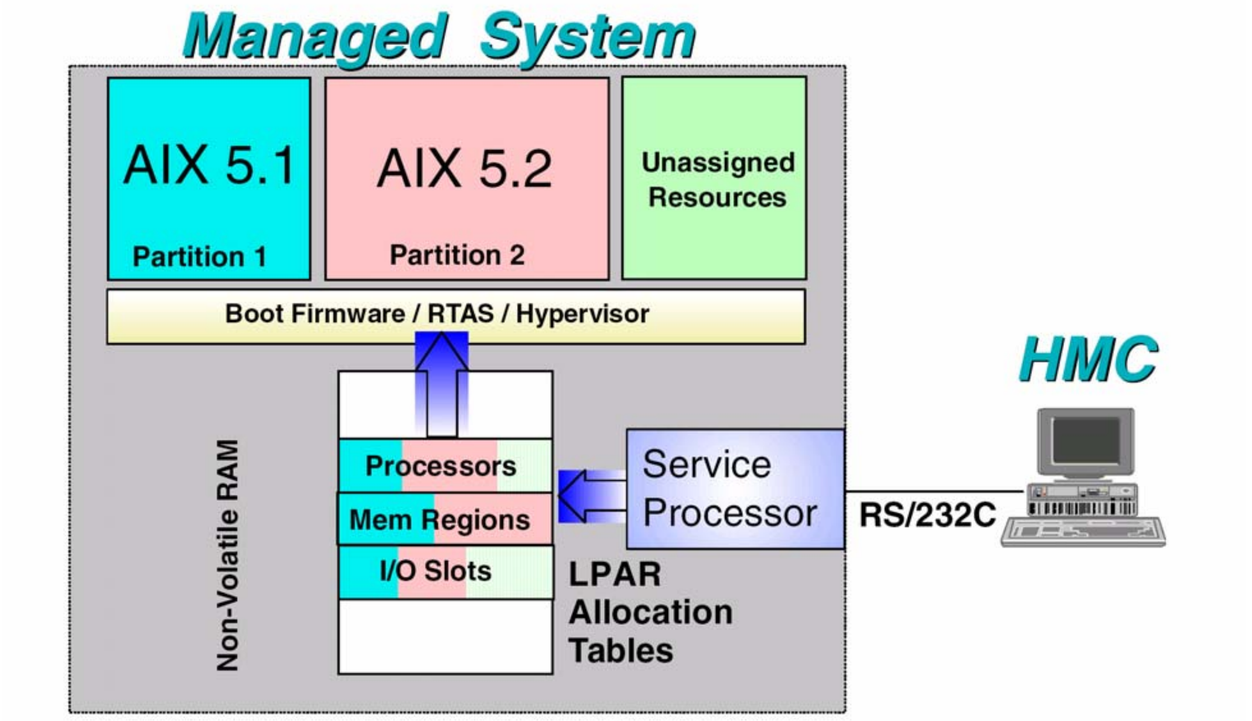

Figure 1: LPAR Architecture Overview

2.2 TOE Components

2.2.1 Hypervisor

The Hypervisor firmware provides major additions to firmware functionality. It implements the following three major categories of service calls:

- Virtual memory management

The Hypervisor becomes the only function that can update the address translation page tables in memory or the TCEs of the PHBs. In this way, Hypervisor controls the physical memory locations that can be accessed from within a partition.

- Debug register and memory access

For the debug and dump environments, Hypervisor provides controlled access to protected facilities and memory locations.

- Virtual TTY support

The Hypervisor provides input/output streams for a virtual TTY device that can be used on the HMC.

The Power4 processor support a "Hypervisor mode" in addition to the existing user and supervisor mode. Some resources can only be accessed and some instructions only be executed when the processor is in Hypervsior mode. This mode is indicated by a specific bit in the Machine State Register (MSR) of the processor.

2.2.2 Open Firmware (Boot Firmware)

A partitioning-capable pSeries server has one instance of Open Firmware both in the partitioned environment and the Full System Partition. (Full System Partition is the configuration where the LPAR is not active and an OS is directly running on the hardware. Full System Partition is **not** an evaluated configuration and mentioned here just for completeness since this mode is mentioned in other documentation. Full System Partition is often named „SMP mode" in the design documentation).

Open Firmware has access to all devices and data in the system. Open Firmware is started when the system goes through a power-on reset. Open Firmware, which runs in addition to the Hypervisor in a partitioned environment, runs in two modes: global and partition. Global and partition Open Firmware share the same firmware binary stored in the flash memory. Only when running in global mode (i. e. in Hypervisor state) the Open Firmware is considered to be part of the TOE. When in partition mode Open Firmware performs the boot loading within the partition and is not executing in hypervisor mode.

In both modes Open Firmware uses hypervisor calls to communicate when it requires hypervisor functions. Although using this interface would not be required for Open Firmware executing in global mode, it allows to use the same binary in both modes. The reader should be aware that only when Open Firmware is executed in global mode it has the capability to access TSF data (which it anyhow only does via hypervisor calls).

The partition Open Firmware is started when a partition is activated. Each partition has its own instance of the partition Open Firmware, which is copied starting at address 0 in the partition's RAM. It has access to all the devices assigned to that partition, but has no access to devices outside the partition in which it runs. As explained, the partition Open Firmware resides within the partition memory, but is replaced when the operating system takes control; it is just needed for the time necessary to load the operating system into the partition system memory.

The global Open Firmware resides with the Hypervisor firmware in the first 256 MB of the physical memory which is a memory area never assigned to any partition.

The global Open Firmware includes the partition manager component. The partition manager is an application in the global Open Firmware that establishes partitions and their corresponding resources, such as CPU, memory, and I/O slots, which are defined in partition profiles. The partition manager manages the operational partitioning transactions. The partition manager component will not work when in partition mode, since the required hardware privileges to perform those functions are not available in any partition.

The partition profiles are stored in and retrieved from nonvolatile random access memory (NVRAM) by system firmware. After the profiles are set up, the system will automatically return to this configured state on a power-on, even if the HMC is unavailable. The NVRAM also provides separate address spaces, called slots, to store partition specific information for each partition. These slots, or partition IDs, are numbered from 1 to the supported maximum partition number.

2.2.3 Run-Time Abstraction Services (RTAS)

RTAS presents the same platform service calls (with a few exceptions) that are presented in a non-partitioned environment, but have some underlying implementation changes to properly handle multiple operating system images, including:

- RTAS calls are only serialized within a partition.

  In general, RTAS operations are restricted to only those resources dedicated to that partition, with an error code return for invalid requests.

- Multiple virtual operator panels for all partitions.

  The information provided by operator panels in a traditional pSeries server are represented on the HMC on a per-partition basis.

- Per-partition time-of-day clock values.

  Time-of-day (TOD) is virtualized for each partition (including in Full System Partition mode. The reader is reminded that this mode is not allowed in the evaluated configuration). Each partition can set its own time and date.

- Restricted access to the per-partition NVRAM areas.

  Each partition has its own segment of NVRAM for the storage of its configuration variables, including a unique boot list for every partition. There is also a unique segment of NVRAM for when the system is in Full System Partition, with its own boot list. For example, there are up to 16 partition boot lists and a seventeenth Full System Partition boot list on the pSeries 690 equipped with 7040-61D I/O drawers.

- Partition reset capabilities.

  Previous pSeries servers had a service processor-based serial port snoop function that allowed remote reset of an unresponsive operating system image. The service processor would snoop the serial port data stream (which is the OS console), but when the operating system is not longer reachable through the keystrokes, and when a certain special command sequence was seen, the service processor would reset the system.

  On partitioning-capable pSeries servers, with the HMC, each partition has its own, very powerful, reset capabilities: a soft reset that causes the partition operating system to get a PowerPC reset interrupt, and a hard reset that is the equivalent of a virtual power off of a partition. For the hard reset, no matter how disabled the partition operating system is, the hard reset will bring all the processors out of the partition and back to the global firmware partition manager so that the partition is ready to be reactivated.

2.3 Resources Managed by the TOE

The TOE manages three types of resources:

- Processors
- Memory regions
- I/O slots

Those resources are explained in the next sections.

2.3.1 Processors

Each installed and configured processor in the partitioning-capable pSeries server can be assigned to a partition. The System Administrator at the HMC defines just the numbers of the processors allocated to a partition. Based on this specification the actual processors are then assigned. The System Administrator does define which processor is assigned to which partition. This is selected by the system.

At least one processor must be assigned to each partition. Sharing processors between multiple active partitions is not possible.

2.3.2 Memory

In a partitioned environment, some of the physical memory areas are reserved by several system functions to enable partitioning in the partitioning-capable pSeries server. Unused physical memory can be assigned to a partition. It is not possible and necessary to specify the precise address of the assigned physical memory in the partition profile, because the system selects the resources automatically. The partition manager part of the TOE selects memory regions and assigns them to partitions.

The minimum amount of physical memory for each partition is 256 MB. One can assign further physical memory to partitions in increments of 256 MB. Sharing memory between multiple active partitions is not possible.

The operating system's Virtual Memory Manager (VMM) manages the logical memory within a partition as it does the real memory in a stand-alone pSeries server. The hypervisor and the POWER4 processor manage access to the physical memory.

2.3.3 I/O slots

I/O devices are assignable to partitions on a PCI slot (physical PCI connector) basis. This means that it is not the PCI adapters in the PCI slots that are assigned as partition resources, but the PCI slots in which the PCI adapters are plugged.

To install an operating system, one has to assign at least one device adapter, typically a SCSI adapter, that is able to boot the operating system, and an adapter to access the install media.

Once installed, one needs at least one device adapter connected to the boot disk or disks. For application use and system management purposes, one also may have to assign at least one network adapter.

One can allocate slots in any I/O drawer on the system. It is recommended to assign more PCI slots than required for the number of adapters in the partition, even if these PCI slots are not populated with PCI adapters. This provides the flexibility to add PCI adapters into the empty slots of an active partition, using the PCI Hot Plug insertion/removal capability.

Sharing I/O slots between multiple active partitions is not possible.

2.4 Resource Assignment

In a partition profile, the administrator has to specify three kinds of values for each resource. For CPU and memory, the administrator has to specify minimum, desired, and maximum values. For I/O slots, the administrator has to specify the required and desired values. If any of the three types of resources cannot satisfy the specified minimum and required values, the activation of a partition will fail. If the available resources satisfy all the minimum and required values, but do not satisfy desired values, the activated partition will get as many of the resources as are available.

The maximum value is used to limit the maximum CPU and memory resources when dynamic logical partitioning operations are performed on the partition. Those values have no impact in the evaluated configuration, because dynamic partitioning is not available there.

2.5 Reserved memory regions in a partitioned environment

In a partitioned environment, some of the physical memory regions are reserved by several system functions to enable partitioning on partitioning-capable pSeries servers. Before understanding the mapping between the logical memory address of a partition and the physical memory address, you have to consider the following memory regions:

- Hypervisor
- Partition page tables
- Translation control entry (TCE) tables

These three memory regions are not usable for the physical memory allocation of the partition.

2.5.1 Hypervisor

The Hypervisor is a passive object loaded into the first physical memory block (PMB) in a partitioned environment. It is loaded only when the system is running in a partitioned environment and does not reserve a processor resource for itself. The Hypervisor only runs when a partition needs a service executed on its behalf, such as creating a page table entry. The Hypervisor can be thought of as a call-back library used as any partition requires. Care has been taken to minimize the number of instructions required to implement the call-backs (i. e. the instructions emulated by the hypervisor), so in most cases, operating system performance is identical for the operating system in a non-partitioned environment where call-backs are not made, versus operating systems in a partitioned environment where call-backs are required.

The Hypervisor resides outside of the partition system memory in the first PMB at the physical address zero. This first PMB is not usable by any of the partition operating systems in a partitioned environment.

2.5.2 Partition page tables

An operating systems Virtual Memory Manager (VMM) uses Hypervisor services to manage the partition page table. The operating systems VMM communicates the desired virtual to logical mapping, and Hypervisor translates that into the virtual to physical mapping within the page table.

The partition table resides outside of the physical address range mapped to the logical address of the partition. Partition page tables are additional memory that is required for a partition to operate, in addition to the total logical memory size of a partition. The partition page table size is determined to be four page table entries per 4096 bytes real page. Each page table entry has a size of 16 bytes.

Therefore, the partition page table is an amount of contiguous physical memory blocks equal to 1/64 of the logical memory address range of the partition, rounded up to the nearest power of two, and it must be on an address alignment equal to its size.

2.5.3 Translation control entry (TCE) tables

In a Full System Partition, TCE tables are controlled by the operating system, as in conventional pSeries systems, but this mode is not an evaluated configuration. In a partitioned environment, the operating system uses Hypervisor services to mange the TCE tables. The operating system communicates the desired I/O bus address to logical mapping, and the Hypervisor translates that into the I/O bus address to physical mapping within the specific TCE table. The Hypervisor needs a dedicated memory region for the TCE tables in order for the I/O address to partition memory address translation to perform direct memory access (DMA) transfers to PCI adapters. Each PCI slot that can be assigned to a partition is isolated underneath a PCI-to-PCI bridge.

This PCI-to-PCI bridge is programmed with the window of allowable DMA addresses from this slot. This window corresponds to a window of TCEs allocated from the parent PCI host bridge (PHB) TCE table. Therefore, TCE tables can be shared across partitions when slots under the same PHB are assigned to different partitions.

An individual TCE table cannot be larger than 8 MB, which contains 2^{20} 8-byte entries capable of mapping 4 K entries each, which results in a 4 GB physical address range that can be mapped by a single TCE table. An individual PHB is programmed to index a single TCE table.

In a partitioned environment, TCE tables are allocated at the top of the physical memory and extend downward. The total size of TCE tables is based on the number of PHBs; therefore, it depends on the number of configured I/O drawers on the pSeries 690. If you configured up to four I/O drawers (80 PCI slots), then a PMB is reserved for the TCE tables. If you configured more than four I/O drawers, then two PMBs, 512 MB in total, are reserved for the TCE tables.

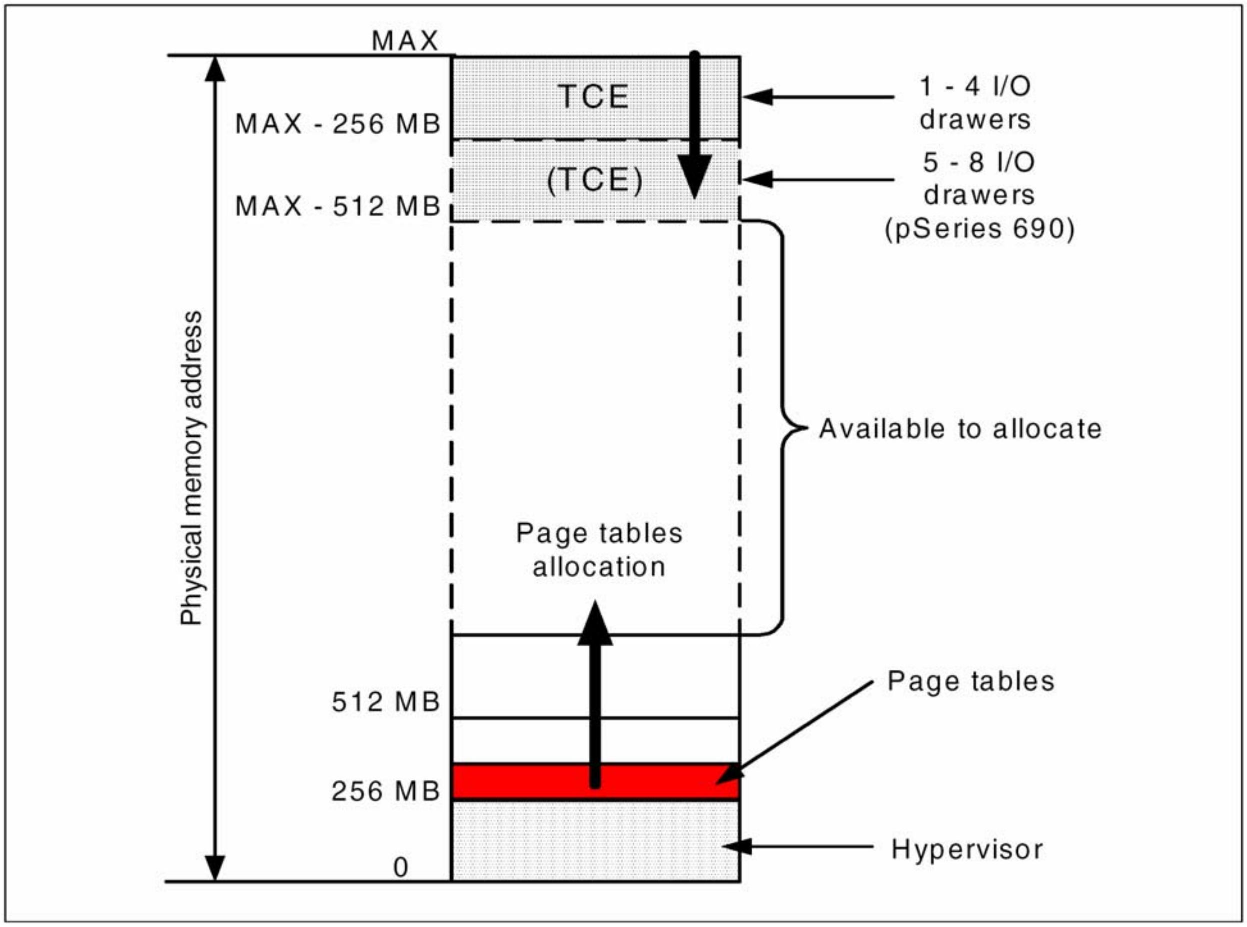

Figure 2: Reserved Memory in a Partitioned Environment

2.6 Hypervisor calls

The POWER4 processor supports a special form of instructions. These instructions are exclusively used by a new controlling firmware named Hypervisor. If an operating system instance in a partition requires access to hardware, it first invokes the Hypervisor using Hypervisor calls. The Hypervisor allows privileged access to an operating system instance for dedicated hardware facilities and includes protection for those facilities in the processor.

The Hypervisor is entered by way of three interrupts: the System Reset Interrupt, the Machine Check Interrupt and System (Hypervisor) Call Interrupt. These use absolute interrupt vectors 0x0100, 0x0200, and 0x0C00 respectively.

The return from the Hypervisor to the OS is via the rfid (Return from Interrupt Doubleword) instruction. The target of the rfid (instruction at the address contained in SRR0) is either a firmware glue routine (in the case of System Reset or Machine Check) or the instruction immediately following the invoking Hypervisor Call. The reason for the firmware glue routines is that the OS must do its own processing because of the asynchronous nature of System Reset or Machine Check interruptions. The firmware glue routine calls an OS registered recovery routine for the System Reset or Machine Check condition. The glue routines are registered by the partition's operating system through RTAS. Until the glue routines are registered, the OS will not receive direct reports of either System Reset or Machine Check interrupts but will simply be re-IPLed by the Hypervisor. The glue routines contain a register buffer area that the Hypervisor fills with register values that the glue routine must pass to the OS when calling the interrupt handler. The last element in this buffer is a lock word. The lock word is set with the value of the using processor, and reset by the glue routine just before calling the OS interrupt handler. This way only one buffer is needed per partition rather than one per processor.

2.7 PowerPC Processor Extensions to support Logical Partitioning

2.7.1 Hypervisor State Bit in the Machine State Register

The Hypervisor State Bit in the Machine State Register indicates if the processor is in Hypervisor mode. Hypervisor resources can only be accessed and instructions reserved for the Hypervisor can only be executed when the processor is in Hypervisor mode.

2.7.2 Logical Partition Identity Register (LPIDR)

This register contains a value that identifies to which partition the processor is assigned.

2.7.3 Real mode offset (RMO) register

The POWER4 processor supports a real mode offset register (RMO). By utilizing the RMO, the processor enables a mapping between the logical memory of a partition and the physical memory. A logical memory address of zero in a partition is registered at a fixed offset in the physical memory address space. The RMO is the physical address that corresponds to the beginning of a partitions memory. This is the partition's logical address 0. This address offset is set in the RMO register when this partition is activated. The processor adds the value stored in the RMO to each logical address fetch and store made by code running in a partition.

The RMO address space only applies to Supervisor Real Mode Execution (MSR.DR/MSR.IR = 0). The RMO defines the low logical memory (starting at 0) that the operating system (Supervisor) can directly address in translation-off mode. Other logical address ranges (beyond the RMO) have no interaction or dependency on the real mode offset register, because they cannot be addressed in the translate-off mode.

2.7.4 Real mode limit register (RML)

The POWER4 processor supports a real mode limit register (RML). By utilizing the RML, the processor limits the range of real mode addressing. The RML is the size of a partition's memory region that is accessed in real mode.

2.8 Service authority

There is the possibility to give one of the partitions in a partitioning-capable pSeries server the service authority attribute. Service authority would enable this partition, to perform system firmware updates or to set system policy parameters.

In the evaluated configuration no partition must have this authority.

Firmware updates also can be done from the service processor menus. Firmware updates are done at the system level, not on a per-partition basis.

A partition with service authority can perform firmware updates without having to power off the managed system. All other partitions must be shut down before the firmware update is initiated. The partition that has service authority must also have access to the firmware update image. If the firmware update image is provided on diskette, the diskette drive must belong to the partition that has service authority. If you are downloading the firmware update from the network, download it to the partition with service authority.

In the Full System Partition, you do not have to take additional steps to prepare for firmware updates.

2.9 TOE Boundary

The Target of Evaluation as defined in this Security Target consists of the Boot Firmware / RTAS / Hypervisor Layer as shown in figure 1. The Hardware Management Console is only used for the system initialization and start-up phase and is regarded as part of the TOE environment. Also the Service Processor is seen as part of the TOE environment, since with no HMC attached during operation it does not affect the operation of the TOE security functions.

2.10 Intended Method of Use

The intended method of use of the LPAR for Power4 architecture in the evaluated configuration is a static definition of partitions when the system is configured and before any partition is started. To change this configuration, all partitions have to be stopped, the partition reconfiguration has to be performed and all partitions have to be fully re-initialized and restarted. Dynamic reconfiguration while partitions are still running is not part of the evaluated configuration.

In this scenario the Hardware Management Console becomes a tool for the initialization and startup of the system and has no operational functionality when the system is in operation. Therefore the Hardware Management Console may be completely removed from the system during operation (to ensure that no unauthorized modification to the LPAR configuration can be made). The only aspect that might need support from the HMC during operation is the situation when a partition hangs completely and needs to be reset.

The functions of the LPAR architecture need to be used by different parts of an operating systems. The following figure shows the parts of AIX that interact with the functions of the TOE.

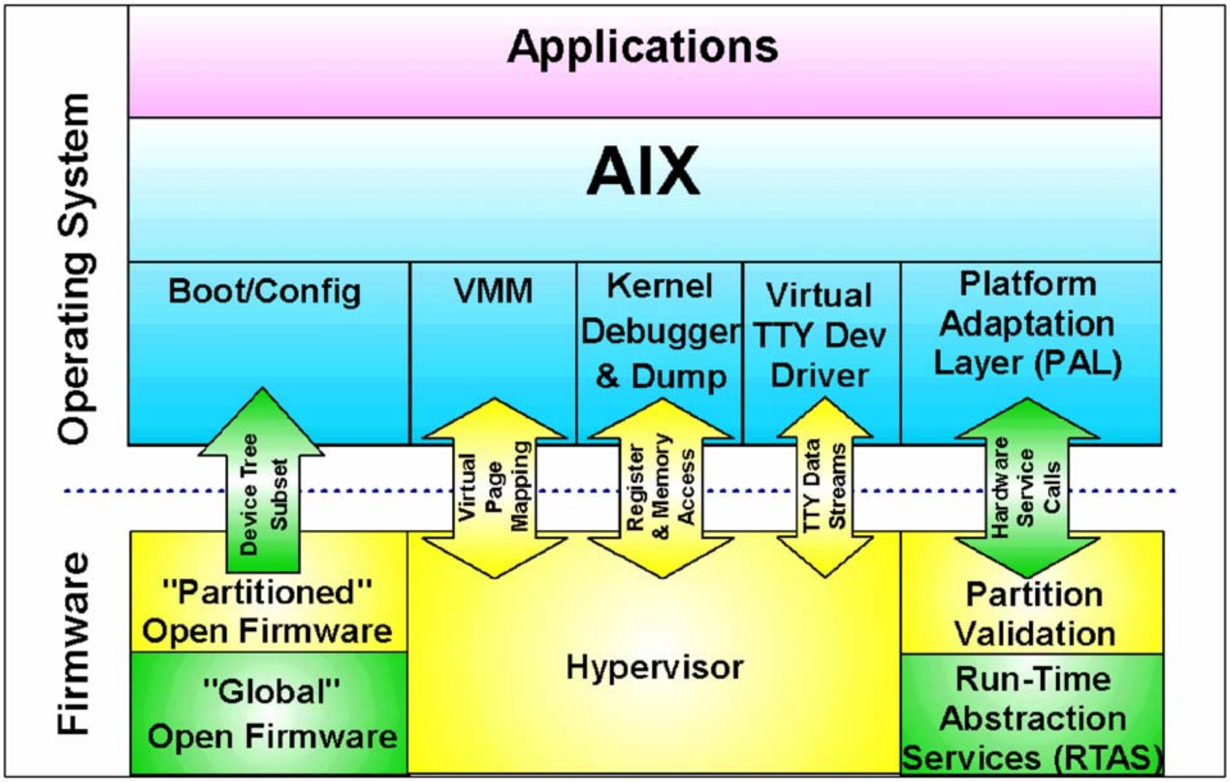

Figure 3: Operating System Use of the TOE with the example of AIX

2.11 Summary of Security Features

The primary security features of the product are:

- Identification of Partitions
- Discretionary Access Control (very static) to TOE resources
- Full resource separation prohibiting direct information flow between partitions
- Object reuse functionality
- Security Management

- TSF Protection.

2.12 Technical Environment for Use

The following assumptions about the technical environment the TOE is intended to be used in are made:

a) The TOE is running on the following hardware platforms:

- IBM pSeries Symmetric Multiprocessor (SMP) Systems, using Power4 CPUs (p 630, p650 and p690)

b) The following peripherals can be run with the TOE preserving the security functionality:

  - All devices that are connected via the PCI bus
  - The service processor (no external device other than the HMC connected)
  - The hardware management console (not to be used when the TOE is running except for starting a partition and restarting a hanging partition)
  - Floppy disk drive and CD-ROM attached to the ISA bus via a SCSI adapter.

3 TOE Security Environment

3.1 Introduction

The statement of TOE security environment describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

To this end, the statement of TOE security environment identifies the list of assumptions made on the operational environment (including physical and procedural measures) and the intended method of use of the for the product, defines the threats that the product is designed to counter, and the organisational security policies with which the product is designed to comply.

3.2 Threats

The assumed security threats are listed below.

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 a server, including data in transit between workstations.

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:

- individuals who have been granted the right to access the system up to the right to run their own software in a logical partition.

The threat agents are assumed to originate from a well managed community of individuals allowed to install and run software in a logical partition. The TOE is assumed to operate in a non-hostile working environment, and hence the product protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well funded attackers to breach system security.

The threats listed below are grouped according to whether or not they are countered by the TOE. Those that are not countered by the TOE are countered by environmental or external mechanisms.

3.2.1 Threats countered by the TOE

T.UAPACCESS A user of the TOE (which may have complete control over his partition) may get access to information in resources belonging to other partitions.

Note: If system administration has established a dedicated communication channel between two partitions e. g. if both partitions have a network adapter as part of one of their I/O slots and those adapters are connected to the same network, communication between the two partitions is of course possible. The same applies if removable storage media are transferred between two partitions. Such a kind of sharing via I/O devices is regarded as the only accepted way for two partitions to share information.

T.UATACCESS A user of the TOE (which may have complete control over his partition) may access information resources belonging to the TOE via other than the defined TOE user functions (Hypervisor calls) or may misuse a Hypervsior call to obtain access to resources that belong to another partition.

T.CRASHPAR Software running in one partition may either crash another partition or dominate resources in a way that another partition is no longer able to execute their software at an acceptable performance.

3.2.2 Threats to be countered by measures within the TOE environment

The following threats apply in environments where specific threats to distributed systems need to be countered.

TE.HWMF A partition or the TOE are losing stored data due to hardware malfunction.

TE.MODNVRAM The content of the NVRAM is modified by unauthorized access to the Hardware Management Console or the Service Processor.

TE.HW_SEP The underlying hardware functions of the hardware the TOE is running on does not provide sufficient capabilities to support the self-protection of the TSF from partitions.

3.3 Organizational Security Policies

The TOE complies with the following organizational security policies:

P.PARTDEF The organization operating the TOE has a defined policy for the amount of processors and memory and the actual I/O slots to be allocated to the individual partitions. This policy is defined in accordance with the operational needs of the software running in the individual partitions. This policy is implemented by an administrator of the organization using the HMC.

3.4 Assumptions

This section indicates the minimum physical and procedural measures required to maintain security of the TOE.

3.4.1 Physical Aspects

A.LOCATE The processing resources of the TOE as well as the Hardware Management Console and the Service Processor will be located within controlled access facilities which will prevent unauthorized physical access.

A.PROTECT The TOE hardware and software critical to security policy enforcement as well as the Hardware Management Console and the Service Processor will be protected from unauthorized physical modification.

A.SP-HMC The Service Processor works properly and will not modify the NVRAM storing the assignment of resources to individual partitions or otherwise interfere with the operation of the TOE except for starting / restarting an existing partition. The Hardware Management Console works properly and will not send a command to the Service Processor unless this command was entered by a system administrator.

Note: The System Administrator is supposed not to submit any commands from the HMC when the TOE is executing except for performing for starting a partition and restarting a hanging partition without modifying the resources assigned to the partitions or for performing a full shutdown. See the assumption A.NOPMOD in the following section.

3.4.2 Personnel Aspects

A.MANAGE It is assumed that there are one or more competent individuals who are assigned to manage the TOE and assigns the resources to the logical partitions such that the software in those partitions is able to operate properly.

A.NO_EVIL_ADMIN The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.NOPMOD The system administrator will not modify the assignment of TOE resources to partitions without performing a shutdown on all partitions before performing such a modification and the performing a full restart of the system.

3.4.3 Other Aspects

A.CONNECT All direct connections to I/O devices, the Service Processor and the Hardware Management Console reside within the controlled access facilities. No network connection is established to the Hardware Management Console and the Service Processor.

A.NOSERVICE No partition has been given Service Authority by the administrator.

A.PDEF Installation and configuration of the TOE as well as the Hardware Management Console and the Service Processor is done correctly. Assignment of resources to partitions is performed according to a defined policy in accordance with the operational needs of the software intended to run in the different logical partitions.

A.HW_FUNC The hardware underlying the TOE operates correctly in accordance with its specification. Periodical checks are performed to verify that this is the case.

4 Security Objectives

4.1 Security Objectives for the TOE

O.AUTHORIZATION The TOE must ensure that logical partitions can only access and use the resources assigned to them

O.RESIDUAL_INFO The TOE must ensure that any information contained in a protected resource is not released when the resource is reused by another partition after re-configuration of the partition information.

O.NONINTERFERE The TOE must ensure that software running in one partition can not interfere with software running in another partition in a way that causes an interrupt or exception in this partition or that causes a major performance loss in this partition.

O.NOFLOW The TOE must ensure that no information can be transferred between different partitions.

4.2 Security Objectives for the TOE IT-Environment

There is one objective for the IT-Environment of the TOE:

OE.HW-SEP The underlying hardware must provide separation mechanism that can be used by the TOE to protect the TSF and TSF data from unauthorized access and modification.

4.3 Security Objectives for the TOE Non-Environment

All security requirements listed in this section are targeted at the non-IT environment of the TOE.

OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

OE.INSTALL Those responsible for the TOE must establish and implement procedures to ensure that the firmware components that comprise the TOE as well as the underlying hardware, the Hardware Management Console and the Service Processor are distributed, installed and configured in a secure manner.

OE.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy as well as the Service Processor and the Hardware Management Console are protected from physical attack which might compromise IT security objectives.

OE.HW-CHECK Periodical checks are performed using a diagnostics tool to ensure that the underlying hardware works correctly.

OE.NONET The service processor and the hardware management console are not connected to any network or remote service connection.

5 Security Requirements

5.1 TOE Security Functional Requirements

FDP_ACC.1 Subset access control

FDP_ACC.1.1 The TSF shall enforce the ***LPAR Resource Access Control Policy*** on ***processors, memory regions and I/O slots as objects and partitions as subjects and all access to those resources by a partition.***

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the ***LPAR Resource Access Control Policy*** to objects based on ***the partition number***

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 partition shall have access to a processor, a memory region or an I/O slot only if the resource is allocated to the partition by the table in the NVRAM***.

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

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the ***no other rules***

FDP_IFC.2 Complete information flow control

FDP_IFC.2.1 The TSF shall enforce the ***Separation Information Flow Control Policy*** on ***all information*** 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 TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the ***Separation Information Flow Control Policy*** based on the following types of subject and information security attributes: ***assignment of objects to a partition***.

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: ***in no case.***

FDP_IFF.1.3 The TSF shall enforce the ***no other rules***

FDP_IFF.1.4 The TSF shall provide the following ***no other capabilities***.

FDP_IFF.1.5 The TSF shall explicitly authorize an information flow based on the following rules: ***none.***

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: ***none***.

FDP_RIP.1 Subset residual information protection

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: ***memory regions, processors.***

Application Note: I/O slots are not mentioned explicitly here, since of course data stored on I/O devices should be available to other regions when the I/O slot is allocated to another partition. Nevertheless the TOE needs to ensure that internal state register of the device or the PCI slot are reset, also just to ensure proper operation of the device when allocated to another partition.

FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the ***LPAR Resource Access Control Policy*** to provide ***TOE defined (within the boundary of the profile specified by the external administrator)*** default values for security attributes that are used to enforce the SFP.

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

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: ***software failure in a partition, hardware failure of a hardware component belonging to a partition.***

FPT_RVM.1 Non-bypassability of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 TSF domain separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FRU_FLT.1 Degraded fault tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of ***programs executing in the other partitions*** when the following failures occur: ***software failure in a partition, hardware failure of a hardware component belonging to a partition.***

5.1.1 Strength of Function

Note: There is no security function in the TOE that is based on statistical or probabilistic algorithms. Therefore no SOF claim is contained in this Security Target.

5.2 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL4 [CC] augmented by ALC_FLR.1.

5.3 Security Requirements for the IT Environment

The only IT environment where requirements are stated is the underlying processor, that has to provide the mechanism to protect the TSF and TSF data from unauthorized access and tampering. This is expressed with the following security functional requirement for the processor used to execute TOE software:

FDP_ACC.1 Subset access control

FDP_ACC.1.1 The TSF shall enforce the **memory access control policy** on **instructions as subjects and memory locations and processor register as objects.**

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **memory access control policy** to objects based on **the processor state (user, supervisor or hypervisor).**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **access to memory locations and special registers is based on the processor state and the state of the memory management unit. Access to dedicated processor registers is allowed only if the processor is in hypervisor state when the instruction accessing the register is executed**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules**: some dedicated processor registers may be read but not modified when the instruction accessing the register is in user or supervisor mode.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **following rule: none**.

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **memory access control policy** to provide **permissive** default values for security attributes that are used to enforce the SFP.

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

Application Note: The „default" values in this case are seen as the values the processor has after start-up. They have to be „permissive", since the initialization routine needs to set up the memory management unit and the device register etc.. With respect to the hardware there is no „role" model implemented but the access control policy is purely based on a single attribute („user", „supervisor" or "hypervisor" state) that can not be managed or assigned to a „user". The attribute changes under well defines conditions (when the processor encounters an exception, an interrupt or when a hypervisor call is executed (which effectively causes an interrupt to occur). The security requirement FMT_MSA.1 was therefore not applicable because the security attribute can not be „managed". For this reason there is also no security requirement FMT_SMR.1 included, because there are no „roles" that need to be managed or assigned to „users". The dependency of FMT_MSA.3 to FMT_MSA.1 and FMT_SMR.1 is therefore unresolved.

5.4 Security Requirements for the Non-IT Environment

There are no security requirements for the non-IT environment that are not already covered by the assumptions on the TOE environment.

5.5 TOE Assurance Requirements

The assurance requirements for the TOE are those defined for the assurance evaluation level EAL4 augmented by ALC_FLR.1. No refinements are made to those assurance requirements. All those requirements can be found in part 3 of the Common Criteria. Therefore those assurance requirements are not reproduced in this Security Target.

6 TOE Summary Specification

The TOE provides the separation between the logical partitions within the LPAR architecture for the IBM pSeries systems.

The TOE implements the following security functions:

- Identification of a partition using its partition number (IA)
- Static assignment of memory regions, processors and I/O slots to logical partitions according to the information stored in the NVRAM and control of access to those resources (AC)
- Protection from interference between software running in different partitions (IP)
- Object reuse functionality for resources assigned to a partition (OR)
- Self protection of the Hypervsior from unauthorized access and modification by software running in a partition (TP)

6.1 Identification (ID)

The TOE needs to identify the partition that calls a TOE function via a Hypervisor Call or via an interrupt. This identification is achieved with the partition number that is stored in the Logical Partition ID (LPID) register. This partition number can only be assigned and modified when the processor is in Hypervisor mode. Partitions are numbered starting with one and then sequentially numbered up to the maximum number of partitions defined.

During startup the TOE assigns the processors to the logical partitions. Some processors may be unassigned (i. e. have a partition number of 0) and will then not be used within any partition.

This addresses requirement FIA_UID.2

6.2 Access Control (AC)

The TOE will assign processors, memory regions and I/O slots to at most one partition. This assignment is managed by a table in NVRAM, which indicates for each resource to which partition the resource is allocated. This is done by storing the number of this partition in the table. A partition number of zero indicates that the resource is not allocated to any partition.

The TOE will ensure that resources or part of resources are not shared between partitions. Resources can only be allocated to a partition by the use of the hardware management console. When the software within a partition performs a shutdown the TOE will return the resources used by the partition to the hypervisor.

This addresses requirements FDP_ACC.1, FDP_ACF.1 and FMT_MSA.3

6.3 Interference Protection (IP)

The TOE does not allow direct communication between software in different partitions. No function of the TOE can be used for such communication. The TOE also ensures that none of the resources it manages is shared between partitions. As a result software in one partition can not affect software in another partition such that the other partition crashes or can not access its resources. Also hardware failures caused by resources allocated to one partition do not affect the operation of the software in other partitions. In case of a hardware failure at the worst a machine check interrupt may be generated, which is intercepted by the TOE. The TOE ensures that such a machine check interrupt from one partition does not influence the other partitions.

Any other software problem is encapsulated within the logical partition. Events that in normal operation systems would result in system crashes, memory exhaustion, I/O locking problems or infinite loops with interrupts disabled are confined within a partition, since no resources are shared between different partitions. Since also the TOE does not share memory or I/O with a partition, also the TOE is not affected. Even a partition that tries to dominate the Hypervsisor e. g. by issuing Hypervsisor calls in an infinite loop may only have some limited timing effects on other partitions that need TOE services, since the Hypervisor function called by a partition will be executed by the processor that executed the call. Some performance degradation may occur due to the Hypervisor internal locking mechanisms for synchronization of TOE internal resources.

Of course if a machine check occurs for Hypervisor resources the whole system will terminate.

This addresses requirements FDP_IFC.2, FDP_IFF.1, FPT_FLS.1 and FRU_FLT.1

6.4 Object Reuse (OR)

When the TOE allocates resources to a partition the residual information in those resources is cleared. This applies to processors that are fully reset, memory objects that are cleared and I/O slot, which are fully reset when they are allocated to a partition. Since no dynamic re-allocation of resources during the operation of the TOE is performed, object reuse activities are only performed before a partition is started.

This addresses requirement FDP_RIP.1

6.5 TSF Protection (TP)

The underlying hardware of the Power4 processors allow the TOE to reserve areas in main memory and in NVRAM for its own operation. The TOE will not allocate those memory areas to any partition thereby protecting its own data structures and code from any type of access by software running in any partition.

This addresses requirements FPT_RVM.1 and FPT_SEP.1

6.6 TOE Assurance Summary Specification

The following table provides an overview, how the assurance measures of EAL4 augmented by ALC_FLR.1 are met by the TOE..

Part of the documentation (especially for the development environment and the configuration management) will be basically identical to the documentation provided previously for the EAL4+ evaluation of AIX 5.2 since LPAR for Power4 uses basically the same development environment

Table 6-1: Mapping Assurance Requirements to Documentation

Assurance Component	Documentation describing how the requirements are met
ACM_AUT.1	CMVC is the tool used for configuration management of LPAR source code, documentation test plans and test cases. The configuration management procedures and tools are identical to the ones used for AIX 5.2 that has been evaluated at a level of EAL4. The procedures are described in separate documents.
ACM_CAP.4	See above
ACM_SCP.2	IBM has a problem tracking procedure in place that covers all aspects of ACM_SCP.2. This description is included in the documentation of the configuration management and the software development procedures..
ADO_DEL.2	A separate document describes the delivery process.
ADO_IGS.1	Installation, generation and start-up procedures are described in the installation manual.
ADV_FSP.2	The functional specification is provided in the documents describing the hypervisor call interface
ADV_HLD.2	The high level design is described in an overview document and subsystem specific documents.
ADV_IMP.1	The full source code of the parts of LPAR for Power4 that build the TSF is provided for the evaluation.
ADV_LLD.1	Low level design documentation is provided for all subsystems that implement TSF.
ADV_RCR.1	The correspondence information will be provided as part of the functional specification and the high level design. An additional document providing the correspondence to the TOE Summary Specification will be provided to the evaluation facility.
ADV_SPM.1	A separate document describing the Security Policy Model is provided to the evaluation facility.
AGD_ADM.1	Administrator guidance is provided in the documents describing the

Assurance Component	Documentation describing how the requirements are met
	functions for the HMC. Security aspects are also addressed in those documents
AGD_USR.1	User guidance is not required, since there are no human users.
ALC_DVS.1	The security procedures on the development site have not changed from the AIX 5.2 evaluation. Those procedures are described in general documents that apply for IBM as a whole as well as site specific documents and specific documents for eServer development.
ALC_FLR.1	The defect handling procedure IBM has in place for the development of LPAR for Power4 requires to describe the defect with its effects, security implications, fixes and required verification steps. Defects are managed within the configuration management system and can therefore be easily tracked for each version of the TOE. The configuration management system also allows to check at any time the status of the defect and the corrective actions taken.
ALC_LCD.1	The life cycle definition is described in separate documents. This addresses the software development process as well as the description of the tools used for development and how they are used.
ALC_TAT.1	See above
ATE_COV.2	Detailed test plans are produced to test the functions of LPAR for Power4. Those test plan include an analysis of the test coverage, an analysis of the functional interfaces tested and an analysis of the testing against the high level and low level design.
ATE_DPT.1	See above
ATE_FUN.1	Testing will be performed on different platforms that are defined in the Security Target. Test results are documented such that the tests can be repeated.
ATE_IND.2	All the required resources to perform their own tests are provided to the evaluation facility to perform their test. The evaluation facility will perform and document the tests.
AVA_MSU.2	The misuse analysis will be provided.
AVA_SOF.1	A Strength of Function analysis is not required, because no security function of the TOE is based on a probabilistic or permutational mechanism.
AVA_VLA.2	A vulnerability analysis will be provided that describes IBM's approach to identify vulnerabilities of LPAR for Power4 as well as the results of the findings.

7 Protection Profile Claims

No claim of conformance to an existing Protection Profile is made.

8 Rationale

8.1 Security Objectives Rationale

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat, assumption or policy and that each threat, assumption or policy is covered by at least one security objective.

8.1.1 Security Objectives Coverage

Table 8-1: Mapping Security Objectives for the TOE to Threats and Policies

O.AUTHORIZATION	T.UAPACCESS, T.UATACCESS, P.PARTDEF
O.RESIDUAL_INFO	T.UAPACCESS
O.NONINTERFERE	T.CRASHPAR, T.UAPACCESS
O.NOFLOW	T.UAPACCESS

Table 8-2: Mapping Security Objectives for the TOE-Environment to Threats and Assumptions

OE.ADMIN	A.MANAGE, A.NO_EVIL_ADMIN, A.NOPMOD, A.NOSERVICE
OE.INSTALL	A.PDEF, A.MANAGE
OE.PHYSICAL	A.CONNECT, A.LOCATE, A.PROTECT, TE.MODNVRAM
OE.HW-SEP	A.HW_FUNC, A.PROTECT, TE.HW-SEP
OE.HW-CHECK	A.HW_FUNC, A.SP-HMC, TE.HWMF
OE.NONET	A.LOCATE, A.CONNECT, TE.MODNVRAM

Table 8-3: Mapping Threats to Security Objectives for the TOE

T.UAPACCESS	O.AUTHORIZATION, O.RESIDUAL_INFO, O.NONINTERFERE, O.NOFLOW
T.UATACCESS	O.AUTHORIZATION
T.CRASHPAR	O.NONINTERFERE

Table 8-4: Mapping Assumptions to Security Objectives for the TOE Environment

A.LOCATE	OE.PHYSICAL, OE.NONET
A.PROTECT	OE.PHYSICAL, OE.HW-SEP
A.SP-HMC	OE.HW-CHECK
A.MANAGE	OE.ADMIN, OE.INSTALL
A.NO_EVIL_ADMIN	OE.ADMIN
A.NOPMOD	OE.ADMIN
A.CONNECT	OE.NONET, OE.PHYSICAL
A.NOSERVICE	OE.ADMIN
A.PDEF	OE.INSTALL

A.HW_FUNC	OE.HW-SEP, OE.HW-CHECK

Table 8-5: Mapping Organizational Security Policies to Objectives

P.PARTDEF	O.AUTHORIZATION

8.1.2 Security Objectives Sufficiency

T.UAPACCESS:

The threat of a user getting access to information in resources belonging to other partitions is addressed by O.AUTHORIZATION requiring that a subject in a logical partition can only access and use resources assigned to this partition, O.RESIDUAL_INFO which prohibits information flow via residuals in resources assigned to the partition and O.NOFLOW which prohibits any other information flow between subjects in different logical partitions and O.NONINTERFERE prohibiting communication channels based on interference between the activities in different partitions..

T.UATACCESS:

The threat of getting access to TOE resources other than via the defined TOE interfaces is addressed by O.AUTHORIZATION requiring that a subject in a logical partition can only access and use resources assigned to this partition by the TOE.

Note that the subject in the partition may get information about TOE internal values and status as long as those can not be used to signal information between different partitions. For example it is not seen as a problem if a subject within a partition can see details of the allocation of the resources of its own partition in the overall set of resources. Since the allocation is static, this information does not leak anything about information from other partitions.

T.CRASHPAR:

The threat of a subjects in a partition trying to influence the behavior of subjects in other partitions is addressed by O.NONINTERFERE requiring that software running in one partition can not interfere with software running in other partitions.

TE.HWMF:

The threat of a partition or the TOE loosing data due to hardware malfunction is addressed by OE.HW-CHECK requiring to run hardware diagnostics.

Note that there is no requirement for a backup of TSF data, since due to the limited number of partitions this data can be easily entered again. Backup of application specific data of software running in a logical partition is not subject of this Security Target.

TE.MODNVRAM:

The threat of modifying the content of the NVRAM by unauthorized access to the Service Processor or the Hardware Management Console is addressed by OE.PHYSICAL which requires that the TOE hardware, the Service Processor and the Hardware Management Console are protected from physical access by unauthorized persons as well as by OE.NONET requiring that neither the Service Processor nor the Hardware Management Console are connected to any type of network or other external connection.

TE.HW_SEP:

The threat of the underlying hardware functions not providing sufficient capabilities to support the self-protection of the TOE is addressed by OE.HW-SEP requiring that the underlying must provide such protection mechanisms.

P.PARTDEF

The organizational security policy requiring a suitable definition of the resources allocated to the individual partitions is addressed by O.AUTHORIZATION requiring that logical partitions can only access the resources assigned to them.

A.LOCATE

The assumption that the processing resources of the TOE as well as the HMC and the Service Processor are located in a controlled area preventing unauthorized physical access is addressed by OE.PHYSICAL requiring a physical protected area for the TOE components, the HMC and the Service Processor and by OE.NONET which forbids any type of external connection to the Service Processor or the HMC which might be used for a logical attack.

A.PROTECT

The assumption that the TOE hardware and software as well as the HMC and the Service Processor are protected from unauthorized physical or logical modification is addressed by OE.PHYSICAL protecting the TOE hardware, the HMC and the Service Processor from physical access and attack by unauthorized persons. The TOE software uses the protection functions required by OE.HW-SEP to protect itself from unauthorized logical modifications.

A.SP-HMC

The assumption that the Service Processor and the HMC work properly and will not modify the NVRAM or interfere otherwise with the operation of the TOE is addressed by OE.HW-CHECK requiring that the hardware functions are checked regularly for correct operation.

A.MANAGE

The assumption of the TOE being managed by competent individuals is addressed OE.ADMIN requiring that the persons responsible for the TOE are capable of managing the TOE and OE.INSTALL requiring that the installation and configuration is done properly.

A.NO_EVIL_ADMIN

The assumption that system administration personnel is not careless, willfully negligent, or hostile is addressed by OE.ADMIN requiring that the persons responsible for the TOE are trustworthy and capable of managing the TOE.

A.NOPMOD

The assumption that the system administrator will not modify the assignment of TOE resources until all partitions are shut down is addressed by OE.ADMIN requiring competent and trustworthy administrative personnel.

A.CONNECT

The assumption that the TOE, all I/O devices, the HMC and the Service Processor reside within a controlled facility is addressed by OE.PHYSICAL. The second part of the assumption on the absence of external connections to the Service Processor and the HMC is addressed by OE.NONET.

A.NOSERVICE

The assumption that no partition is given SERVICE authority by the administrator is addressed by OE.ADMIN requiring the administrator to be trustworthy and capable to manage the TOE securely.

A.PDEF

The assumption that the TOE is installed and configured correctly is addressed by OE.INSTALL requiring this.

A.HW_FUNC

The assumption that the hardware works correct and is periodically checked for correct operation is addressed by OE.HW-SEP requiring the correctness of the hardware separation mechanism and OE.HW-CHECK requiring periodical checks of the hardware functions.

8.2 Security Requirements Rationale

8.2.1 Internal Consistency of Requirements

This section describes the mutual support and internal consistency of the security functions defined in this Security Target. This is done by justifying why and how the security functions satisfy the security functional requirements and why the security functions mutually support each other.

Identification (ID)

To enforce the access control policy the TOE needs to be able to identify the partition that requests a function from the TOE. This identification is provided by the partition number stored in the LPID register of the processor. Since this register can not be modified by any program executing in supervisor or user state, the TOE can rely on the correctness of this information without performing an explicit authentication process. Therefore the requirement for identification is sufficient as the basis to perform access control and only FIA_UID.2 is included as a requirement. Note that the "user" as stated in the requirement is not a human user but is the logical partition. The TOE identifies the partition that uses an interface to the TOE by the partition number.

Access Control (AC)

The TOE assigns the resources to individual partitions in accordance with the profile defined by an external administrator. The TOE decides himself, which processors and memory regions are assigned to a partition. The profile defined by the external administrator just defines the number of processors and the amount of memory regions leaving the decision on the assignment of individual processors and memory regions to the TOE. The TOE needs to perform the assignment of resources in accordance with the profile defined by the external administrator and needs to ensure that the resources are not shared between different partitions. Since modifying the assignment of resources to partitions requires the stop and re-start of all partitions the access control model used is a very static one not requiring the management of the access control policy. Therefore no security management requirements are included in the security requirements for the TOE. The security functional requirements FDP_ACC.1 and FDP_ACF.1 defines this discretionary access control policy.

There is just a requirement for this access control mechanism to adhere to the profile defined by the external administrator. But as mentioned this profile just defines some boundary conditions for the access control policy. The actual default values are defined by the TOE itself. Therefore the assignment in FMT_MSA.3.2 is not an "authorized identified role" as specified in part 2 of the CC, but the TOE itself. Just the boundary conditions are defined by an administrator, but this administrator is not known to the TOE but is a role in the TOE environment. As a result of this, the TOE does not need to have a functional requirement for different roles, i. e. the dependency of FMT_MSA.3 on FMT_SMR.1 does not apply in this case and is therefore not resolved. Also the dependency of FMT_MSA.3 on FMT_MSA.1 and FMT_SMF.1 is not resolved, since the management of the access control policy is left completely to the TOE itself within the boundary defined by the profile.

The security function access control addresses just the "discretionary" access control policy on the resources managed by the TOE. The additional "information flow" policy is supported by the LPAR Resource Access Control Policy such that sharing of those explicit named resources is prohibited.

To address the LPAR Resource Access Control Policy the functional requirements FDP_ACC.1 and FDP_ACF.1 as well as FMT_MSA.3 are included.

Interference Protection (IP)

The functionality of the TOE prohibits direct flow of information between different partitions (unless a communication channel is established in the TOE environment e. g. by connecting the network adapters allocated to different partitions to the same network) This information flow policy (which actually is a separation policy) is expressed in FDP_IFF.1. Due to the fact that no processors, memory regions and I/O slots are shared between different partitions, the operation of software in one partition should not be affected by the operation of software in another partition. This is expressed by the security functional requirement FDP_IFC.2. In addition even the failure of a resource (=hardware component) belonging to one partition will not affect the operation of software in other partitions, which is expressed in the requirements FPT_FLS.1 and FRU_FLT.1. According to the target evaluation assurance level and the threat model, a sophisticated analysis for non-obvious covert channels is not part of this evaluation. Potential covert channels may well be identified in the vulnerability analysis, but will not be addressed further if their exploitation is beyond the capabilities of an attacker as defined in the threat model.

Object Reuse (OR)

The TOE manages three types of resources:

- Memory regions
- Processors
- I/O slots

With respect to object reuse memory regions and processors are critical. But one has to keep in mind that none of those resources is dynamically released and re-assigned during the operation of the TOE. Instead any re-assignment of resources requires all partitions to shut-down and being restarted after the re-assignment has been made. In this light only the clearing of memory regions when they are assigned to a partition is highly critical. Of course also information stored in processor registers can transfer information and therefore it also needs to be ensured that they do not contain information left from their previous use within another partition. This is addressed by FDP_RIP.1.

With respect to I/O devices the situation is different. If the I/O device connected to the I/O slot is a storage medium like a disk, it is of course not required to erase the information there before the device is allocated to another partition. But also I/O devices that are not designed to be storage media may contain information transferred to them during the previous use within one partition. It may even be possible to read this information when the I/O slot to which the I/O device is connected is assigned to another partition. Here one has to keep in mind that the I/O devices themselves are not managed by the TOE, only the slots they are connected to are. The TOE does not know, if the I/O device connected to a slot has an internal memory or not and is therefore not responsible to perform any kind of object reuse activity on the devices themselves except for resetting the I/O slot.

TSF Protection (TP)

The TSF protection relies of course on the protection mechanism of the underlying hardware in the sense that the Power4 processor protects access to TOE resources from software in partitions. This support mechanism is the additional "Hypervisor Mode" and the protection of special processor registers such that they can only be modified when the processor is in Hypervisor Mode. Those aspects are addressed by the requirements for the IT-environment.

The TOE itself now has to use those functions properly to protect itself from interference and tampering by software running in a partition. This is expressed by the requirement FPT_SEP.1.

Software running in a partition can communicate with the TOE to request services. This is done by "Hypervisor Calls" or by generating interrupts that are intercepted by the TOE. The TOE now needs to ensure that all access to TOE resources that require the service of the TOE is done using the interface to the TOE where the TOE ensures that the security policy is enforced. This is expressed by FPT_RVM.1.

Note that the TOE is not involved in every access by a partition to the resources it manages. On the memory regions the TOE needs to set the values in the RML and RMO registers (and some other resources controlled by the TOE) correctly. Then the processor architecture ensures that software running in a partition can not access other memory regions. The processors allocated to a partition are defined by the partition number in the LPID register of the processors. This register can not be changed by software running in a partition. Concerning I/O slots it is also the underlying hardware architecture that enforces the access control policy on those resources.

Due to this hardware support mechanisms there are only a limited number events that require software in a partition to request services from the Hypervisor. This has the effect that the performance of software running in a partition is almost identical to software running on a machine without partitioning (provided in both cases the same number and type of resources are used).

As a summary of this here is a table mapping the TSF to the functional requirements:

Table 8-6: Mapping TOE Security functions to Security Functional Requirements

TSF	Security Functional Requirements
Identification (ID)	FIA_UID.2
Access Control (AC)	FDP_ACC.1, FDP_ACF.1, FMT_MSA.3
Interference Protection (IP)	FDP_IFF.1, FDP_IFC.2, FPT_FLS.1, FRU_FLT.1
Object Reuse (OR)	FDP_RIP.1
TSF Protection (TP)	FPT_SEP.1, FPT_RVM.1

8.2.2 Security Requirements Coverage (TOE)

The following table shows the mapping of Security Objectives for the TOE to security functional requirements:

Table 8-7: Mapping Assumptions to Security Objectives for the TOE Environment

O.AUTHORIZATION	FIA_UID.2, FDP_ACC.1, FDP_ACF.1, FMT_MSA.3, FPT_RVM.1, FPT_SEP.1
O.RESIDUAL_INFO	FDP_RIP.1
O.NONINTERFERE	FPT_FLS.1, FRU_FLT.1, FPT_RVM.1, FPT_SEP.1
O.NOFLOW	FDP_IFC.2, FDP_IFF.1, FPT_RVM.1, FPT_SEP.1

O.AUTHORIZATION requires that logical partitions can only access resources assigned to them. This requires an access control policy which is addressed by FDP_ACC.1 and FDP_ACF.1, the identification of the partition accessing a resource which is addressed by FIA_UID.2, the specification of default values which is addressed by FMT_MSA.3 as well as the reference mediation (FDP_RVM.1) and the separation of the TOE from the partitions it manages (FPT_SEP.1).

O.RESIDUAL_INFO requires to prohibit information flow when a resource is assigned to another partition. This is addressed by FDP_RIP.1.

O.NONINTERFERE requires that a software in a running partition is not affected by the actions performed by software in other partitions (including hard- and software failure) which is addressed by FPT_FLS.1 which requires the preservation of a secure state in those situations and by FRU_FLT.1 which requires that partitions continue to work when such failures occur in other partitions. To be effective the separation of the TOE from the partitions (FDP_SEP.1) as well as the control that all critical actions are mediated by the TOE (FPT_RVM.1) is required.

O.NOFLOW requires that no information can be transferred directly between different partitions. This is addressed by the information flow policy requirements FDP_IFC.2 and FDP_IFF.1, which also require the separation of the TOE from the partitions (FDP_SEP.1) as well as the control that all critical actions are mediated by the TOE (FPT_RVM.1).

8.2.3 Security Requirements Coverage (TOE IT-Environment)

The security functional requirements FDP_ACC.1, FDP_ACF.1 and FMT_MSA.3 for the IT environment define an access control policy on memory regions and processor register such that defined memory regions and processor register can only be accessed or modified when the processor is in “Hypervisor Mode”. This allows the TOE to define a domain for its own execution that is not accessible to software running in a partition. The models of the Power4 processor used in the hardware underlying the TOE provides this support as documented in the processor manuals. This all contributes to satisfy the objective OE.HW-SEP for the IT environment.

8.2.4 Security Requirements Dependency Analysis

Table 8-8: Security Requirements Dependency Analysis

Security Functional Requirement	Dependency defined in the CC	Resolved
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Yes
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	Yes
FDP_IFC.2	FDP_IFF.1 Simple security attributes	Yes
FDP_IFF.1	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialization	Yes[1]
FDP_RIP.1	No dependencies	Yes
FIA_UID.2	No dependencies	Yes

Security Functional Requirement	Dependency defined in the CC	Resolved
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	No[2]
FPT_FLS.1	ADV_SPM.1 Informal TOE security policy model	Yes[3]
FPT_RVM.1	No dependencies	Yes
FPT_SEP.1	No dependencies	Yes
FRU_FLT.1	FPT_FLS.1 Failure with preservation of secure state	Yes

[1] The dependency on FDP_IFC.1 is satisfied by the inclusion of FDP_IFC.2 which is hierarchical to FDP_IFC.1.

[2] see next section for justification of unresolved dependencies.

[3] The dependency is resolved by the evaluation assurance level chosen. ADV_SPM.1 is included in EAL4.

8.2.5 Justification of unresolved dependencies

The dependencies of FMT_MSA.3 to FMT_MSA.1 and FMT_SMR.1 are both not resolved. The reason is the static access control policy which is completely defined and enforced by the TOE in accordance with the profile specified by an external administrator. The TOE itself therefore does not require management of the security attributes and therefore also roles are not required. This is supported by the assumptions A.LOCATE stating that the TOE is operated a physically secured environment, A.PROTECT stating that the TOE is protected against physical modifications, A.PDEF stating that the definition of the partitions is static between the start-up of the TOE and the shutdown of the TOE and A.NOSERVICE stating that no partition is given Service Authority (which would allow software in the partition to modify the allocation of resources to partitions) together enforce that allocation of resources to partitions is static for the time the TOE is operating and therefore no management aspects need to be considered.

For the security functional requirements for the IT-environment there is also a dependency of FMT_MSA.3 to FMT_MSA.1 and FMT_SMR.1. Both dependencies are also not resolved, because the access control policy of the underlying processor is also a static one, which can not be managed by any role. Therefore neither a management of the access control policy is required (which FMT_MSA.1 would define) nor a role model for management is required (which FMT_SMR.1 would define). The dependencies defined in part 2 of the CC for this security functional requirement therefore do not apply here.

8.2.6 Strength of function

The TOE does not use any function or mechanism based on permutational or probabilistic properties. Therefore no strength of function analysis is required. This Security Target therefore does not contain a strength of function claim.

8.2.7 Evaluation Assurance Level

The evaluation assurance level EAL4 has been chosen to comply with the strong security claims made with respect to information flow and non-interference. Having such claims at assurance levels below EAL4 does not seem to be appropriate.

8.3 TOE Summary Specification Rationale

8.3.1 Security Functions Justification

This section provides some more justification for the security functions defined in the TOE summary specification. The reader should note that some aspects of this justification is already provided in section 8.2.1 of this Security Target. Therefore the information in this section also forms part of the TOE summary specification rationale.

Identification

To enforce the access control policy the TOE needs to be able to identify the partition that requests a function from the TOE. Therefore a security function "Identification" has been introduced to address this. As already explained authentication is not required because the TOE completely manages and controls the identify of the partitions.

Access Control

The basic security function of the TOE is to control access to the three type of resources. Therefore a security function "Access Control" has been introduced. This function covers the discretionary access control aspects of the TOE to the memory regions, processors and I/O slots. The access control policy is actually very static and also defined by the TOE based on the requirements specified by an externally supplied profile for the resources to be allocated to partitions.

Interference Protection

Due to the strict separation of resources the TOE is also able to protect partitions from malicious software or from hardware failures of resources belonging to other partitions. Except for some potential performance downgrade a running partition will not be affected by those kind of problems. In addition the strict separation also enforces a simple information flow control policy where no information flow between different partitions is allowed. Therefore a security function "Interference Protection" has been introduced to describe those properties.

Object Reuse

To support the access control and interference protection security functions it is required that no information from a partition is passed within objects that are re-assigned to other partitions. Although this happens only when the system is started, such a transfer would still be possible since the system may be restarted without being powered down. Therefore a security function "Object Reuse" has been introduced to address this aspect.

TSF Protection

The strict separation policy defined by the security functions listed above requires the TOE to maintain a domain for its own execution that is protected from unauthorized access and tampering by software in a partition. In addition the TOE needs to control all access to TSF data which is done by defined calls to a defined interface between software in a partition and the TOE. Therefore a security function "TSF Protection" has been introduced to address those aspects.

8.3.2 Assurance Measures Justification

The assurance measures taken to satisfy the requirements of EAL4 and ALC_FLR.1 have been outlined in section 5.5 of this Security Target. It is subject to the evaluation to assess that those assurance measures are in place and sufficient to satisfy the requirements defined in the Common Criteria. But the table in section 5.5 demonstrates that there is a measure in place for all the assurance requirements of EAL4 as well as for the additional assurance requirement ALC_FLR.1.

8.4 PP Claims Rationale

No conformance with any existing protection profile is claimed

9 Abbreviations

AIX	Advanced Interactive Executive
CC	Common Criteria
CPU	Central Processing Unit
DMA	Direct Memory Access
FPR	Floating Point Register
GPR	General Purpose Register
HMC	Hardware Management Console
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
LPAR	Logical Partition Architecture
LPID	Logical Partition Identifier
MSR	Machine State Register
NVRAM	Non-Volatile Random Access Memory
PDF	Portable Data Format
PHB	PCI Host Bridge
PMB	Physical Memory Block
PP	Protection Profile
RML	Real Mode Limit
RMO	Real Mode Offset
RTAS	Run-Time Abstraction Services
SCSI	Small Computer System Interface
ST	Security Target
TCE	Translation Control Entry
TOE	Target of Evaluation
TSF	TOE Security Functions
VMM	Virtual Memory Manager