EURO-MILS

Secure European Virtualisation for Trustworthy Applications in Critical Domains

Common Criteria Protection Profile

Multiple Independent Levels of Security: Operating System

V2.03 (2016)

Project number	318353
Project acronym	EURO-MILS
Project title	EURO-MILS: Secure European Virtualisation for Trustworthy Applications in Critical Domains
Start date of the project	1st October, 2012
Duration	42 months
Programme	FP7/2007-2013
Proiect website	www.euromils.eu

Common Criteria Protection Profile

Common Criteria

registration ID

Editors/Authors: Igor Furgel, Viola Saftig; T-Systems International GmbH (TSYS)

Further information on the EURO-MILS Project: http://www.euromils.eu

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 318353.

MILS Protection Profile

Whitepaper 2016

Executive Summary

This Protection Profile ‘Multiple Independent Levels of Security: Operating System (MILS PP: Operating System)’ is issued by the EURO-MILS Consortium.

This PP addresses only Operating System as part of a MILS final integrated system. This PP is intended to be part of a set of MILS PPs that should comprise, in the future, also other PPs regarding MILS architecture, like a PP addressing both underlying Hardware Platform and Operating System together and a PP for the entire integrated system.

The TOE, as addressed in the current PP, does not include any hardware. If it is desired to certify a TOE also comprising hardware components, the related ST will include these hardware components as part of the TOE.

If appropriate, the re-assignment operation may be applied:

”The ST may specify that certain objectives for the operational environment in the PP are security objectives for the TOE in the ST. […] If a security objective is re-assigned to the TOE the security objectives rationale has to make clear which assumption or part of the assumption may not be necessary any more“ ([1], chapter 9.3).

The document has been prepared as a Protection Profile (PP) following the rules and formats of Common Criteria version 3.1 [1], [2], [3], Revision 4.

Table of Content

1 PP Introduction 1
1.1 PP reference 1
1.2 TOE Overview 1
1.2.1 TOE definition and operational usage 1
1.2.2 TOE type 1
1.2.3 Non-TOE hardware/software/firmware 2
1.3 TOE Description 3
1.3.1 TOE Architecture 3
1.3.2 TOE 3
1.3.3 TOE Operational Environment 5
1.3.4 TOE Life Cycle 6
1.3.5 TOE Physical Boundary 9
1.3.6 TOE Logical Boundary 10
2 Conformance Claims 12
2.1 CC Conformance Claim 12
2.2 Protection Profile Claim 12
2.3 Package Claim 12
2.4 Conformance Rationale 12
2.5 Conformance statement 12
3 Security Problem Definition 13
3.1 Introduction 13
3.1.1 Assets and Objects 13
3.1.2 Subjects, Roles, and External Entities 18
3.2 Threats 19
3.3 Organizational Security Policies 20
3.4 Assumptions 22
4 Security Objectives 23
4.1 Security Objectives for the TOE 23
4.2 Security Objectives for the Operational Environment 24
4.3 Security Objectives Rationales 26
4.3.1 Security Objective Rationales: Threats 27
4.3.2 Security Objective Rationales: Security Policies 27
4.3.3 Security Objective Rationales: Assumptions 27
5 Extended Components Definition 28
6 Security Requirements 29

6.1 Security Functional Requirements for the TOE 29
6.1.1 Overview 29
6.1.2 Class FAU Security Audit 31
6.1.3 Class FDP User Data Protection 32
6.1.4 Class FIA Identification and Authentication 36
6.1.5 Class FMT Security Management 37
6.1.6 Class FPT Protection of the TSF 40
6.1.7 Class FRU Resource Utilization 41
6.2 Security Assurance Requirements for the TOE 44
6.3 Security Requirements Rationale 44
6.3.1 Security Functional Requirements Rationale 44
6.3.2 Security Functional Requirements Dependencies Analysis 48
6.3.3 Security Assurance Requirements Rationale 48
6.3.4 Security Assurance Requirements Dependencies Analysis 49
7 Acknowledgement 49
8 Glossary 50
9 Abbreviations 54
10 Bibliography 56

1 PP Introduction

1 This section provides document management and overview information required to register the protection profile and to enable a potential user of the PP to determine, whether the PP is of interest.

1.1 PP reference

2

Title:	Protection Profile 'Multiple Independent Levels of Security: Operating System (MILS PP: Operating System)'
Sponsor:	EURO-MILS Consortium
Editor(s):	Dr. Igor Furgel, Viola Saftig T-Systems GEI GmbH, SC Security Analysis & Testing
CC Version:	3.1 (Revision 4)
Assurance Level:	Minimum assurance level for this PP is EAL5 augmented.
General Status:	released
Version Number:	2.03 as of 31th March 2016
Registration:	registration ID
Keywords:	Operating System, Separation Kernel, MILS (Multiple Independent Levels of Security), Virtualization, Hypervisor

1.2 TOE Overview

1.2.1 TOE definition and operational usage

3 The Target of Evaluation (TOE) addressed by the current protection profile is a special kind of operating system, that allows to effectively separate different applications running on the same platform from each other.

4 The TOE can host user applications that can also be operating systems. User applications can even be malicious, and even in that case the TOE ensures that malicious user applications are neither harming the TOE nor other applications in other partitions. The TOE will be installed and run on a hardware platform (e.g. embedded systems).

5 The TOE is intended to be used as a component (the separation kernel) in MILS systems. MILS (Multiple Independent Levels of Security) systems are explained in [9], [10] and [11].

6 The TOE controls usage of memory, devices, processors, and communication channels to ensure *complete separation* of user applications and to prevent unexpected interference between user applications. The TOE enforces restrictions on the communication between the separated user applications as specified by the configuration data.

1.2.2 TOE type

7 The TOE is a special kind of operating system providing a separation kernel with real-time support.

8 The typical *life cycle* phases for this TOE type are development (source code development), manufacturing (compilation to binary), system integration (by the system integrator), installation (by the system operator), and finally, operational use (by the system operator). Operational use of the TOE is explicitly in the focus of this PP. A security evaluation/certification according to the assurance package chosen in this PP (see the statement "This PP does not claim conformance to any protection profile" in Section 2.1) involves all these life cycle phases.

1.2.3 Non-TOE hardware/software/firmware

9 The TOE may run on various hardware platforms. The TOE, as addressed in the current PP, does not include any hardware. If it is desired to certify a TOE also comprising hardware components, the related ST will include these hardware components as part of the TOE.

10 The minimum requirements and obligations on hardware usage like memory management and support for different CPU privilege modes are given in Section 3.3, organizational security policies P.SYSTEM_INTEGRATOR.

11 **Explanatory Note 1: Fehler! Verweisquelle konnte nicht gefunden werden.**

1.3 TOE Description

12 Though no TOE description statement within a PP is required by [1], the current PP includes the related statement as it is exceedingly important for the TOE type addressed here.

1.3.1 TOE Architecture

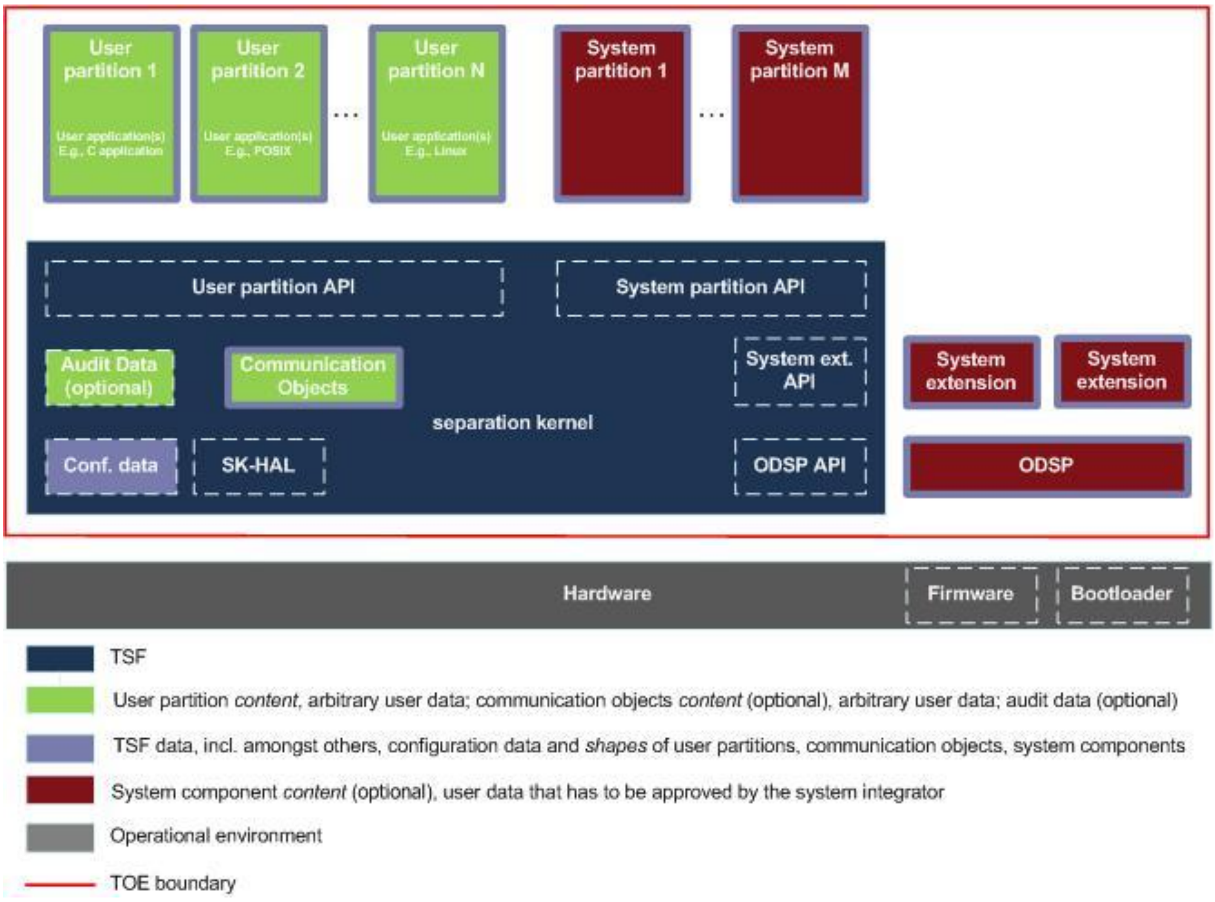
User partition 1
User application(s) E.g., C application
User partition 2
User application(s) E.g., POSIX
...
User partition N
User application(s) E.g., Linux
System partition 1
...
System partition M
User partition API
System partition API
Audit Data (optional)
Communication Objects
System ext. API
System extension
System extension
separation kernel
Conf. data
SK-HAL
ODSP API
ODSP
Hardware
Firmware
Bootloader
TSF
User partition *content*, arbitrary user data; communication objects *content* (optional), arbitrary user data; audit data (optional)
TSF data, incl. amongst others, configuration data and *shapes* of user partitions, communication objects, system components
System component *content* (optional), user data that has to be approved by the system integrator
Operational environment
TOE boundary

Figure 1: TOE and TOE Operational Environment During Operational Use

13 Figure 1, especially the difference between 'green' and 'red' components, will be explained in detail in the next sections (Section 1.3.2 and 1.3.3).

1.3.2 TOE

14 The TOE, delineated within the red line in Figure 1 consists of a separation kernel (TSF), TSF data and user data. The separation kernel and TSF data represent the TOE operating system.

1.3.2.1 TOE Operating System

15 The separation kernel provides the TSF and operates the TOE, by implementing mechanisms to assign resources to partitions, providing the execution environments for applications, and implementing communication between partitions as defined by the configuration data.

16 The separation kernel provides Application Programming Interfaces (APIs) to user partitions and system partitions as well as APIs to system extensions and on-board device support package (ODSP).

17 A Separation Kernel Hardware Abstraction Layer (SK-HAL) provides specific low-level functionality for each supported CPU architecture. In operational use, the TOE always contains only one SK-HAL.

18 TSF data consists of

- Configuration data: Data used by the TSF to enforce the *System Security Policy* (SSP, Section 1.3.4.2), depicted as a bright blue box in Figure 1.
- Shape data: A shape is TSF data that contains an entity's identity, the entity's resource usage data, a set of security attributes according to the SSP assigned to the entity, and links the content assigned to an entity to the resources assigned to the entity (Section 3.1.1.2). Shapes are depicted as bright blue frames in Figure 1.

1.3.2.2 Partition

19 A partition is a logical unit maintained by the separation kernel and configured by the configuration data. A partition contains user data. For each partition, the separation kernel provides resources. Resources of a partition comprise physical memory space and allocated CPU time for each CPU.

20 The TOE supports two different kinds of partitions: user and system partitions. User partitions, depicted as green content surrounded by bright blue shapes in Figure 1, are defined in Section 1.3.2.2.1. System partitions, depicted as red content surrounded by bright blue shapes in Figure 1, are defined in Section 1.3.2.2.2.

21 Partitions can communicate with each other under the supervision of the TOE's separation kernel. This communication occurs via communication objects. A communication object is an object exposed to one or multiple partitions with access rights as defined in the configuration data.

1.3.2.2.1User Partition

22 User partition: A *user partition* contains user applications and/or data being executed and/or stored in a user partition. User applications can be arbitrary and even malicious. User applications use the user partition API of the separation kernel. The content of a user partition does not have to be approved by the system integrator. The content of a user partition can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE, see Section 1.3.4.2.

1.3.2.2.2System Partition

23 System partition: A *system partition* contains applications and/or data supplied and approved by the system integrator. An application in a system partition is a *system application* and uses the system partition API of the separation kernel. The content of a system partition can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

24 **Explanatory Note 2**: The ability of the TOE to support system partitions is **optional** and a ST/PP compliant to this PP can choose to have system partitions or not to have system partitions. The author of the related ST/PP shall clearly state it.

1.3.2.3 System Component

25 A *system component* is a system partition (Section 1.3.2.2.2), system extension (Section 1.3.2.4), or an ODSP (Section 1.3.2.5). A system component contains user data supplied and approved by the system integrator.

1.3.2.4 System Extension

26 System extension: A *system extension* contains a software component (a system application) supplied and approved by the system integrator and coupled with the separation kernel via the system extension API. A system extension can provide specific functionality to applications within partitions only under supervision of the separation kernel. A system extension can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

27 **Explanatory Note 3**: The ability of the TOE to support system extensions is **optional** and a ST/PP compliant to this PP can choose to have system extensions or not to have system extensions. The author of the related ST/PP shall clearly state it.

1.3.2.5 On-board Device Support Package (ODSP)

28 On-board device support package: An *on-board device support package* is a special purpose HAL and may contain a set of drivers for specific hardware components (a system application). It is supplied and approved by the system integrator. An *on-board device support package* can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE. An *on-board device support package* uses the TSF's *on-board device support package* API. In operational use, the TOE always contains only one *on-board device support package*.

1.3.2.6 Audit Data

29 *Audit data* is user data consisting of electronic records reflecting events to be audited.

30 **Application Note 1**: The ability of the TOE to support the generation of audit data is **optional** and a ST/PP compliant to this PP can choose to have the generation of audit data or not. The author of the related ST/PP shall clearly state it.

1.3.2.7 Communication Object

31 A *communication object* contains user data. See Section 1.3.2.2

32 **Explanatory Note 4**: If a concrete TOE implementation cannot principally use any communication objects, the author of the related ST/PP shall clearly state it. Such a TOE implementation is considered to be compliant to this PP.

1.3.3 TOE Operational Environment

33 The TOE operational environment, outside the red line in Figure 1, consists of:

34 Hardware: *Hardware platform* is the physical part of the TOE operational environment on which the TOE is executed. Usually, hardware is a board with several components such as CPUs, serial interfaces, network adapters, I/O devices etc. There are Separation Kernel Hardware Abstraction Layer controlled components (e.g. CPUs, caches) and ODSP controlled components (e.g. serial interfaces, timer).

35 Hardware platform may also comprise the following hardware-specific software:

- Firmware: *Firmware* is software and data stored in non-volatile memory of the hardware platform that initializes the hardware after the power on.
- Bootloader: A *bootloader* is software that loads the TOE on the hardware and hands over the full control to the TOE. In particular, a TOE-external check of the TOE may be implemented in the bootloader (e.g. for “secure boot”).

1.3.4 TOE Life Cycle

The generic lifecycle of the TOE comprises of development/manufacturing, System Integration, Installation and Operational Use.

1.3.4.1 Development, Manufacturing

36 At the TOE manufacturer’s site the TSF is developed (source code development), and manufactured (compiled to binary). The TOE manufacturer also produces the TOE User Manuals.

1.3.4.2 System Integration

37 At the system integrator’s site, the TOE is integrated. Figure 2 presents the generic Lifecycle of the TOE. Components used to build the product based on the TOE are provided by different sources: user application developers, system integrators, and the TOE manufacturer.

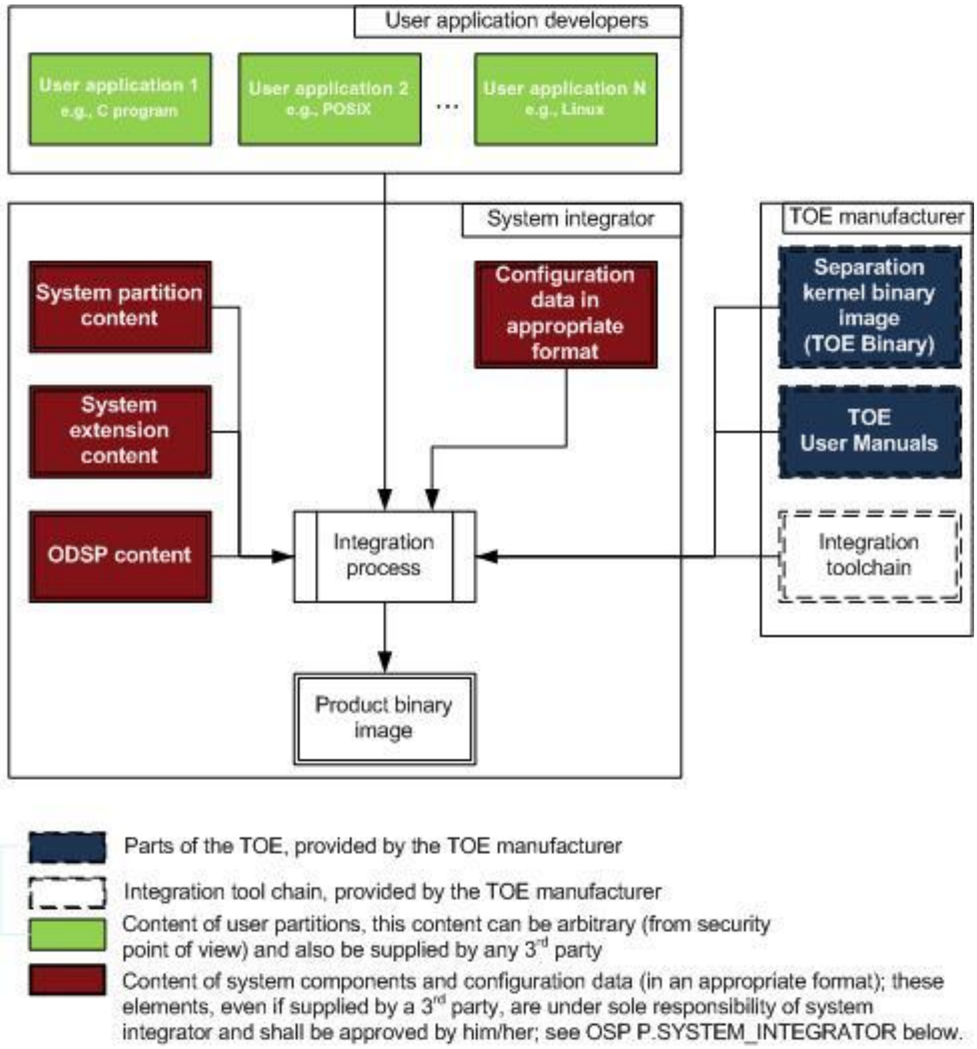
User application developers
User application 1
e.g., C program
User application 2
e.g., POSIX
...
User application N
e.g., Linux
System integrator
System partition content
System extension content
ODSP content
Configuration data in appropriate format
Integration process
Product binary image
TOE manufacturer
Separation kernel binary image (TOE Binary)
TOE User Manuals
Integration toolchain
Parts of the TOE, provided by the TOE manufacturer
Integration tool chain, provided by the TOE manufacturer
Content of user partitions, this content can be arbitrary (from security point of view) and also be supplied by any 3rd party
Content of system components and configuration data (in an appropriate format); these elements, even if supplied by a 3rd party, are under sole responsibility of system integrator and shall be approved by him/her; see OSP P.SYSTEM_INTEGRATOR below.

Figure 2: Generic Lifecycle of the TOE.

38 The system integration phase of the generic lifecycle can be split into the three steps: selection of the TOE operational environment and system applications and user applications (Step 1), configuration of the TOE (Step 2), and integration (Step 3).

39 The outcome of Step 2 is referred to as configuration data. The *configuration data* defines a set of rules on how the TOE behaves. For example, the configuration data comprises the assignment of resources and communication objects to partitions. The *System Security Policy* (SSP) consists of configuration choices made by a system integrator based on the subset of the configuration data rules evaluated in this PP (for details: see this section, below, in the description of Step 2). The SSP is enforced by the TSF and it cannot be circumvented by malicious user applications.

40 The *combined* outcome of Step 1 and Step 2 is referred to as the *System Integration Policy* (SIP). The SIP comprises user applications to be integrated (‘green’ components), user data that need to be approved by the system integrator (‘red’ components: the content of the ODSP, of system partitions, of system extensions) and system integration rules also covering hardware choices, see P.SYSTEM_INTEGRATOR for details.

41 Step 1 Selection

The system integrator selects hardware, and if applicable, firmware and bootloader the TOE runs on.

The system integrator selects the content of components: ODSP, optional system extension(s), optional system partition(s), and user partition(s) to be integrated in the TOE.

The content of any user partition is arbitrary and can be provided by arbitrary application developers.

The content of the ODSP, any system extension, any system partition shall be developed complying with the obligations given in Section 3.3, organizational security policy P.SYSTEM_INTEGRATOR and be approved by the system integrator.

42 Step 2 Configuration

The system integrator configures the product by, for example,

- defining user partitions, setting their content, shapes and resources, see Glossary,
- defining communication objects, setting their shapes and resources,
- defining system components, setting their content, shapes and resources,
- hardware selection parameters,
- setting TOE attributes, comprising
    - scheduling scheme,
    - policy for memory cache handling on a partition switch to the extent supported by the operational environment's hardware,
    - scheme for automatic handling of error conditions, defining the meaning of the secure state,
    - configuration of management functions; the audit function, if implemented by the TOE[1], is the only one.

The result of this activity is a representation, in appropriate format, of the configuration data.

The default configuration is that there is no information flow between any partitions. Any information flow between partitions has to be explicitly allowed by the system integrator in the configuration data.

The configuration data uniquely defines the System Security Policy (SSP). The SSP is defining user partitions, setting their content, shapes and resources, defining communication objects, setting their shapes and resources, defining system components, setting their content, shapes and resources, hardware selection

[1] see **Application Note 1**

parameters, setting TOE attributes, comprising scheduling scheme, policy for memory cache handling on a partition switch to the extent supported by the operational environment's hardware, scheme for automatic handling of error conditions, configuration of management functions; the audit function, if implemented by the TOE, is the only one. An example for a rule defined by the configuration data, but not in the SSP, is the content of user partitions.

The result of performing Step 2 is that the configuration data has been defined. The result of performing Step 1 and Step 2 is that a SIP has been defined.

43 Step 3 Integration

The system integrator uses the integration tool chain to create a product binary image according to the SIP from the selected components and the representation, in appropriate format, of the TOE configuration data. The tool chain

- imports, into the user partitions user applications and/or data,
- imports, into system partitions applications and/or data supplied by the system integrator,
- links the content of the on-board device support package and the content of system extensions with the TOE separation kernel binary image, creating the product binary image, including configuration data in a representation readable by the product binary image.

1.3.4.3 Installation

44 The system integrator provides this product binary image to the system operator who, at the system operator's site, installs it on the hardware.

1.3.4.4 Operational Use

45 At the system operator's site, the TOE is operated. At power on the hardware is initialized, then the product binary image is loaded. Immediately after the product binary has been loaded, the on-board device support package, being part of the product binary image, gets invoked. The on-board device support package then starts the TOE separation kernel (TSF), also being part of the product binary image, which initializes itself and starts enforcing the SSP. During operational use, user applications cannot change the product binary image, e.g. no new user or system partitions can be created, no new communication objects can be created, no new user or system applications can be loaded.

1.3.5 TOE Physical Boundary

46 The TOE is a software product; additionally, TOE User Manuals also belong to the TOE. In Figure 1, each component within the red line is within the TOE physical boundary. Each component outside of the red line is outside of the TOE physical boundary. Thus, no hardware belongs to the TOE. The TOE also includes the TOE User Manuals.

47 **Explanatory Note 5: Fehler! Verweisquelle konnte nicht gefunden werden.**

1.3.6 TOE Logical Boundary

48 The TOE provides at least the following TOE security services, abbreviated as TSS_XXX, cf. also Security Functional Groups defined in sec. 6.1.1:

- TSS_SSA: Separation in space of applications hosted in different partitions from each other and from the TOE operating system according to the SSP by using the underlying hardware.

  Applications can be hosted in different partitions. Partitions get assigned resources (i.e. space) according to the SSP, which comprise memory ranges and a set of CPUs. The TSF enforces the corresponding part of the SSP by the enforcement of access control on partition content, per-partition provision of physical memory space and allocated CPU time for each CPU.

  By confining applications into user partitions, the TSF enforces that these applications can affect neither applications in other partitions (user or system applications) nor the TOE operating system itself.

- TSS_STA: Separation in time of applications hosted in different partitions from each other and from the TOE operating system according to the SSP.

  Applications can be hosted in different partitions. Partitions get assigned CPU time (i.e. time windows) according to the SSP. The TSF enforces the corresponding part of the SSP by per-partition allocation of a predefined amount of CPU time for each CPU. Several user and/or system partitions can share the same time window. On a partition switch CPUs will be reused. The TSF enforces that no residual information is in CPU registers or memory caches according to the SSP. The TSF assigns a priority to every subject to allow priority based scheduling within one time window.

- TSS_COM: Provision and management of communication objects.

  Applications hosted in different partitions can get assigned a set of communication objects. A communication object is an object exposed to one or multiple partitions with access rights as defined in the configuration data, thus allowing communication between partitions.

- TSS_MAN: Management of and access to the TSF and TSF data.

  The TSF restricts access to TSF data. Resource usage data is data accounting for the usage of resources. For example, the partition resource usage data accounts for how much memory a partition has already used and how much there is still available. Resource usage data is stored in shapes. The TSF protects the confidentiality, integrity and availability of resources and shapes (see Table 2 for more details). The TSF restricts the executability of the system application API to system applications. Management functions are used for the management of the security behavior of the TSF. The management functions as configured in the SSP can only be invoked by system applications, but can never be invoked by user applications.

- TSS_SPT: TSF self-protection and accuracy of security functionality.

TSF self-protection and accuracy of functionality supports preserving a secure state of the TOE. The TSF statically assigns automatic invocations of error handling functions to recover from or respond to error conditions.

Application Note 2: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall extend the logical TOE boundary by the following TOE security service:
TSS_AUD: Generation and treatment of audit data according to the SSP.

The TSF provides a function for the start-up and shutdown of the audit functions. When the audit function is active, the system collects events written by user applications to audit data, including events to be audited as defined by the SSP. Audit data can be treated by subjects according to the SSP.

2 Conformance Claims

2.1 CC Conformance Claim

49 This protection profile claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, September 2012, CCMB-2012-09-001 [1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, September 2012, CCMB-2012-09-002 [2]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 4, September 2012, CCMB-2012-09-003 [3]

as follows

- Part 2 conformant,
- Part 3 conformant.

The

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; Version 3.1, Revision 4, September 2012, CCMB-2012-09-004, [4]

has to be taken into account.

2.2 Protection Profile Claim

50 This PP does not claim conformance to any protection profile.

2.3 Package Claim

51 The current PP is conformant to the following security assurance package:
Assurance package EAL5 augmented with AVA_VAN.5 as defined in the CC, part 3 [3].

2.4 Conformance Rationale

52 Since this PP does not claim conformance to any protection profile, this section is not applicable.

2.5 Conformance statement

53 This PP requires *strict* conformance of any ST or PP claiming conformance to this PP.

3 Security Problem Definition

3.1 Introduction

54 **Explanatory Note 6:** Some of the entities listed below, depending on context, can act both as an object to be protected (Section 3.1.1) as well as a subject (Section 3.1.2). Example: The SSP specifies that a user application may, for example, query itself. Thus, in FDP_ACC.2/AS.COMMUN_OBJ_CONT (Section 6.1.3.1) the SSP is applied on the user application acting as object number 1 in Table 1 (Section 3.1.1.1) and on the (same) user application acting as subject number 1 in Table 3 (Section 3.1.2).

55 **Explanatory Note 7**: For a subject, the following operations are possible:

- Treat. The verb “*treat*” is used as a synonym for “read”, “execute” and “write”. The verb "treat" is limited to TSF-mediated operations on objects.
- Write. To “*write*” means to write an object by invocation of the TSFI. To “write” an object also may mean partial writing of an object or changing an object’s state (“modification”).
- Read. To “*read*” means to read an object by invocation of the TSFI. To “read” an object also may mean partial reading of an object or obtaining information about an object’s state (“querying”).
- Execute. To “*execute*” application content means to run the application content by invocation of the TSFI. To “*execute*” an API of the TSFI means to “invoke” the TSFI.
- Consume. To “*consume*” means to use and deplete a quantifiable resource like memory (by invocation of the TSFI) or CPU time (with or without invocation of the TSFI).
- Address. To “*address*” means to address memory directly without invocation of the TSFI and reading (by CPU “load” instructions) or writing it (by CPU “store” instructions).

3.1.1 Assets and Objects

56 Each partition, each communication object, and each system component consists of a triple: *content*, *resources* used by the content, and a *shape*, which contains a set of security attributes according to the SSP assigned to an entity linking content and resources (see Glossary for more details).

3.1.1.1 Primary Assets

57 Primary assets represent user data.

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
1	User partition content (AS.USER_PART_CONT)	*User partition content* is user applications and/or data being executed and/or stored in a user partition. This asset can be *addressed* and *treated* by user applications within their user partition. This asset can be *treated* by subjects. This asset can be *addressed* by system applications.	confidentiality, integrity
2	Communication object content (AS.COMMUN_OBJ_CONT)	*Communication object content* is the content of a communication object and exchanged (received/read and sent/written) between partitions. This asset can be *treated* by subjects. This asset can be *addressed* by system applications.	confidentiality, integrity
3	System component content (AS.SYS_COMP_CONT)	*System component content* are system applications and/or data being executed and/or stored in a system component (a system partition, a system extension or the on-board device support package). This asset can be *addressed* and *treated* by system applications.	confidentiality, integrity

Table 1: Primary Assets Representing User Data

Application Note 3: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following primary asset in Table 1:

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
4	Audit data (AS.AUD)	*Audit data* – audit data is electronic records reflecting events to be audited. This asset is generated by the TSF. This asset can be *treated* by subjects. This asset can be *addressed* by system applications. Each audit data object has a unique object identity.	confidentiality, integrity

3.1.1.2 Secondary Assets

58 Secondary assets represent the TSF and TSF data.

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
5	User partition resources (AS.USER_PART_RES)	*User partition resources* comprise physical memory space and allocated CPU time for each CPU. Resources are assigned according to the SSP. This asset can be *consumed* by subjects. Please note, that this asset is managed by the TSF to enforce the SSP.	availability
6	User partition shape (AS.USER_PART_SHAPE)	A *user partition shape* contains a set of security attributes according to the SSP assigned to a user partition that links its *user partition resources* and its *user partition content*. A user partition shape contains the following security attributes: a unique partition identity, a flag indicating that the partition is a user partition (i.e. the role for all applications	confidentiality, integrity

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
		in the partition), and the *resource usage data* (i.e. here partition resource usage data), SSP enforcement data. This asset can be *treated* and *addressed* by system applications. On behalf of user applications, this asset is *read* and *written* by the TSF. **Explanatory Note 8:** SSP enforcement data is data used by the TSF to enforce the SSP. For example, SSP enforcement data may contain page tables. User partition shapes can contain also other, security irrelevant data, e.g. information on optimising virtualised guests that is not security relevant. For each instantiation of this object, the TSF assigns a unique object identity (partition identity). Please note, that this asset is used by the TSF to enforce the SSP.	
7	Communication object resources (AS.COMMUN_OBJ_RES)	*Communication object resources* are memory space. Resources are assigned according to the SSP. This asset can be *consumed* by subjects. Please note, that this asset is managed by the TSF to enforce the SSP.	availability
8	Communication object shape (AS.COMMUN_OBJ_SHAPE)	A *communication object shape* contains a set of security attributes according to the SSP assigned to a communication object, which links its *communication object resources* and its *communication object content*. A communication object shape contains, amongst other, a unique communication object identity and the *resource usage data* (i.e. here communication object resource usage	confidentiality, integrity

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
		data). This asset can be *addressed* by system applications. For each instantiation of this object, the TSF assigns a unique object identity (communication object identity). Please note, that this asset is used by the TSF to enforce the SSP.	
9	System component resources (AS.SYS_COMP_RES)	*Resources of a system component* comprise physical memory space and allocated CPU time for each CPU. Resources are assigned according to the SSP. This asset can be *consumed* by system applications. Please note, that this asset is managed by the TSF to enforce the SSP.	availability, confidentiality, integrity
10	System component shape (AS.SYS_COMP_SHAPE)	A *system component shape* contains a set of security attributes according to the SSP assigned to a system component that links its *system component resources* and its *system component content*. A system component shape of a system partition also contains, amongst other a flag indicating that the partition is a system partition, and the *resource usage data* (i.e. here partition resource usage data). This asset can be *treated* and *addressed* by system applications. For each instantiation of this object the TSF assigns a unique object identity (system component identity). Please note, that this asset is used by the TSF to enforce the SSP.	confidentiality, integrity

Object Number	Asset Name	Description, Operations	Generic Security Properties to be Maintained by the TOE, as long as the TOE is operational
11	Configuration data (AS.CONF_DATA)	*Configuration data* are data used by the TOE to enforce the SSP. This asset can be *addressed* by system applications. Please note, that this asset is stored and used by the TSF to enforce the SSP.	confidentiality, integrity
12	System application API (AS.SYS_APP_API)	The *system application API* is an interface to functions of the TSF available for system applications. This asset can be *executed* by system applications.	availability (in the sense of 'executability') only for system applications

Table 2: Secondary Assets Representing the TSF and TSF Data

59 **Explanatory Note 9**: If a concrete TOE implementation cannot principally use any communication objects, the author of the related ST/PP shall clearly state it.
In such a case the assets AS.COMMUN_OBJ_CONT, AS.COMMUN_OBJ_RES and AS.COMMUN_OBJ_SHAPE do not exist any more and, hence, should be ommited in all the related items like security objectives and security requirements.
The ability of the TOE to support system components is optional. If the TOE does not support system components, the assets AS.SYS_COMP_CONT, AS.SYS_COMP_RES and AS.SYS_COMP_SHAPE do not exist any more and, hence, should be ommited in all the related items like security objectives and security requirements.

3.1.2 Subjects, Roles, and External Entities

External Entity Number	Subject Number	Role	Definition
1	1	User application	A *user application* is any application within a user partition. A user application is allowed to use only the TOE user partition API. For each instantiation of this subject the TOE assigns a unique subject identity.
2	2	System application	A *system application* is any application within a system partition, a system extension, or the on-board device support package (ODSP). Only a system application in a system partition is allowed

External Entity Number	Subject Number	Role	Definition
			to use the TOE system partition API. Only a system application in a system extension is allowed to use the TOE system extension API. Only a system application in the ODSP is allowed to use the TOE ODSP API. For each instantiation of this subject the TOE assigns a unique subject identity.
3	-	System integrator	A *system integrator* is a person trusted to (re-)configure and integrate the TOE. This includes identifying system partitions and user partitions and assigning applications into partitions. *System integrator* may (and usually do) act on behalf of an organisation.
4	-	System operator	A *system operator* is a person trusted to (re-)install, stop, start, restart, or access (also physically) the TOE in the field. *System operator* may (and usually do) act on behalf of an organisation.
5	-	Attacker	An *attacker* is a threat agent (a person or a process acting on his/her behalf) trying to undermine the TOE security policy defined by the current PP and, hence, the SSP. The attacker especially tries to change properties of the assets having to be maintained according to the TOE security policy defined by the current PP (see Table 1 and Table 2 in Section 3.1.1). The attacker is assumed to possess an at most *high* attack potential. Note that the TOE security policy defined by the current PP only addresses attacks carried out by *user applications* and does not address any physical attacks, see **P.SYSTEM_INTEGRATOR** and P.SYSTEM_OPERATOR. All attacks from other sources than *user applications* shall be averted by the TOE operational environment.

Table 3: Subjects, Roles and External Entities

60 In Table 3, if there is a number in the “subject” column, it means that, during operational use, the TSF recognizes the external entity as subject, and assigns a role to it. If there is no such number (“-”), then, during operational use, the TSF does not recognize that external entity as subject.

61 **Explanatory Note 10**: The ability of the TOE to support system components is **optional**. If the TOE does not support system components, the the role “System application” does not exist any more and, hence, should be ommited in all the related items like security objectives and security requirements.

3.2 Threats

62 Assets are defined in Table 1 in Section 3.1.1.1 (user data) and Table 2 in Section 3.1.1.2 (TSF data). An attacker is an external entity defined in Table 3 in Section 3.1.2.

T.DISCLOSURE

63 An attacker discloses user data and/or TSF data of which the confidentiality shall be maintained according to Table 1 in Section 3.1.1.1 (user data) and Table 2 in Section 3.1.1.2 (TSF data).

T.MODIFICATION

64 An attacker writes user data and/or TSF data of which the integrity shall be maintained according to Table 1 in Section 3.1.1.1 (user data) and Table 2 in Section 3.1.1.2 (TSF data).

T.DEPLETION

65 By requesting resources for communication objects and/or partitions and/or system extensions and/or ODSP, an attacker makes these resources unavailable to the TOE itself and/or to user applications and/or to system applications.

T.EXECUTION

66 An attacker invokes a system application API without being authorized to do so.

67 **Explanatory Note 11**: For example, attacks can be initiated in the following ways:

- An arbitrary user application developer who, e.g. by subcontracting, is authorized to develop a user application for the TOE, tries to attack the TOE, e.g. to implant malicious code in the user application.
- An arbitrary external human entity or IT entity that has authorized access to a user application, e.g. from the Internet, compromises this user application to attack the TOE.

3.3 Organizational Security Policies

68 The TOE and/or its environment shall comply with the following organizational security policies (OSP) as security rules, procedures, practices or guidelines imposed by an organization upon its operation.

P.SECURE_STATE

69 The TOE shall preserve a secure state in which the TOE enforces the SSP.

P.SYSTEM_INTEGRATOR

70 Obligations for a system integrator comprise, as follows:

(1) The system integrator shall select hardware such that:

(1.1) The hardware shall have CPU(s) with at least two privilege modes (“user” and “supervisor” mode).

Explanatory Note 12: Only the TOE separation kernel itself and system components may run in the “supervisor” mode. User applications always run in "user mode". In "user mode" only a limited set of instructions is available, in the "supervisor mode" all instructions are available.

(1.2) The hardware shall have memory management, which restricts accesses of user applications to memory regions according to the SSP.

Explanatory Note 13: Memory management can, for example, be provided by an MMU or a MPU. The MMU or MPU may be be configurable through the TOE by policies specifying these restrictions. These MMU / MPU configuration policies are part of the SSP.

(1.3) The hardware (CPU or CPUs) shall provide instructions to switch between privilege modes and to use the memory management to set up different segments of memory.

(1.4) The hardware (CPU or CPUs) shall allow the TOE to reuse CPU(s) for different user applications, in a way that there is no residual information flow through CPU registers.

(1.5) The hardware shall provide default values for security-relevant settings at power-on (e.g. program counter, a full list shall be included in the TOE User Manuals).

Explanatory Note 14: This supports the TOE reaching the initial secure state.

(1.6) If the hardware possesses any other active components beside CPUs, then either the hardware shall provide support to either turn these components completely off or the TOE separation kernel and/or system components control them as described in TOE User Manuals.

Explanatory Note 15: For example, if devices can execute DMA, then all DMA shall be switched off or, in order to control DMA, the hardware shall provide an I/O MMU, with the I/O MMU controlled by the TOE separation kernel and/or system components.

Application Note 4: The writer of a ST shall state all the CPU architectures which should be subject of consideration during the security evaluation. These architectures shall fulfill requirements (1.1) to (1.3). Depending on the system integrator's requirements for residual information flow on the hardware, special attention may have to be paid to (1.4) to (1.6).

(2) The system integrator shall ensure that the TOE separation kernel gets exclusively executed, so that the TSF starts operating exclusively controlling the CPU(s) and other hardware resources it has to control.
For this reason, the system integrator shall ensure an appropriate implementation (see item #(3) below) and configuration (see item #(4) below) of firmware and bootloader and ODSP.

(3) The system integrator shall ensure that any system component content has been developed following the guidance in the TOE User Manuals. The system integrator shall validate that system component content complies with the SSP and approve this system component content for integration.

(4) The system integrator shall correctly perform the integration process according to the guidance in the TOE User Manuals.
The system integrator is fully responsible for the definition of an appropriate – for the purpose of the system integrator – *System Security Policy* (SSP). The TSF will enforce any SSP as defined by the system integrator.

(5) The system integrator shall define an operational policy for the product in the field which at least enables enforcing the SSP during operational use. The system integrator shall oblige the system operator to follow this policy. The operational policy shall at least require that:

(5.1) The system operator shall ensure that the operational environment provides the TOE with appropriate physical security measures commensurate with the value and properties of the assets protected by the TOE.

(5.2) The system operator shall ensure that the hardware selected for the TOE operates correctly according to the operational policy (and, if necessary, according to the hardware manuals)

(6) The system integrator shall be aware that the TSF has no knowledge of whether a specific SSP is appropriate for a specific product based on the TSF. The TSF will enforce any SSP as defined by the system integrator.

P.SYSTEM_OPERATOR

71 The system operator shall follow the operational policy for the product in the field defined by the system integrator.

72 **Application Note 5**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following OSP:
P.AUDIT
The TOE shall be able to record all events to be audited as defined by the SSP.
Thereby, the TOE enforces each possible SSP, i.e. a set of SSPs, concrete configuration parameters with their allowed values shall be exactly described in the TOE User Manuals[2].
For providing reliable timestamps for the audit security functionality, the system integrator shall select timer facilities in the TOE operational environment according to the SIP.

3.4 Assumptions

73 This section describes the assumptions about the operational environment of the TOE.

74 **A.TRUSTWORTHY_PERSONNEL**

[2] please note that a concrete treatment of audit data AS.AUD is covered by SSP, see also P.SYSTEM_INTEGRATOR, item #(4).

The personnel configuring and integrating the TOE (system integrator) are trustworthy, act according to Section 3.3, organizational security policy P.SYSTEM_INTEGRATOR and are sufficiently qualified for this task.

The personnel installing and operating the TOE (system operator) are trustworthy, act according to Section 3.3, organizational security policy P.SYSTEM_OPERATOR and are sufficiently qualified for this task.

4 Security Objectives

4.1 Security Objectives for the TOE

75 **OT.CONFIDENTIALITY**

For each asset, the TOE shall preserve its confidentiality as defined by the SSP according to Table 1 in Section 3.1.1.1 (user data) and Table 2 in Section 3.1.1.2 (TSF data).

76 **OT.INTEGRITY**

For each asset, the TOE shall preserve its integrity as defined by the SSP according to Table 1 in Section 3.1.1.1 (user data) and Table 2 in Section 3.1.1.2 (TSF data).

77 **OT.RESOURCE_AVAILABILITY**

For user partition resources, communication object resources and system component resources (see Table 2), the TOE shall preserve their availability as defined by the SSP.

78 **OT.SECURE_STATE**

The TOE shall preserve a secure state. A secure state is a TOE state in which the TOE enforces the SSP.

79 **OT.SYSTEM_APPLICATION_API_PROTECTION**

The TOE shall prevent any execution of the system application API by a user application. Thus, the API availability is restricted to only system applications.

80 **Application Note 6**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following objective for the TOE:
OT.AUDIT
The TOE shall be able to record all events to be audited as defined by the SSP.
Thereby, the TOE enforces each possible SSP, i.e. a set of SSPs, concrete

configuration parameters with their allowed values shall be exactly described in the TOE User Manuals[3].

4.2 Security Objectives for the Operational Environment

81 OE.SYSTEM_INTEGRATOR

Obligations for a system integrator comprise, as follows:

(1) The system integrator shall select hardware such that:

(1.1) The hardware shall have CPU(s) with at least two privilege modes (“user” and “supervisor” mode).

(1.2) The hardware shall have memory management, which restricts accesses of user applications to memory regions according to the SSP.

(1.3) The hardware (CPU or CPUs) shall provide instructions to switch between privilege modes and to use the memory management to set up different segments of memory.

(1.4) The hardware (CPU or CPUs) shall allow the TOE to reuse CPU(s) for different user applications, in a way that there is no residual information flow through CPU registers.

(1.5) The hardware shall provide default values for security-relevant settings at power-on (e.g. program counter, a full list shall be included in the TOE User Manuals).

(1.6) If the hardware possesses any other active components beside CPUs, then either the hardware shall provide support to either turn these components completely off or the TOE separation kernel and/or system components control them as described in TOE User Manuals.

(2) The system integrator shall ensure that the TOE separation kernel gets exclusively executed, so that the TSF starts operating exclusively controlling the CPU(s) and other hardware resources it has to control.
For this reason, the system integrator shall ensure an appropriate implementation (see item #(3) below) and configuration (see item #(4) below) of firmware and bootloader and ODSP.

(3) The system integrator shall ensure that any system component content has been developed following the guidance in the TOE User Manuals. The system integrator shall validate that system component content complies with the SSP and approve this system component content for integration.

[3] please note that a concrete treatment of audit data AS.AUD is covered by SSP, see also OE.SYSTEM_INTEGRATOR, item #(4).

(4) The system integrator shall correctly perform the integration process according to the guidance in the TOE User Manuals.
The system integrator is fully responsible for the definition of an appropriate – for the purpose of the system integrator – System Security Policy (SSP). The TSF will enforce any SSP as defined by the system integrator.

(5) The system integrator shall define an operational policy for the product in the field which at least enables enforcing the SSP during operational use. The system integrator shall oblige the system operator to follow this policy. The operational policy *shall at* least require that:

(5.1) The system operator shall ensure that the operational environment provides the TOE with appropriate physical security measures commensurate with the value and properties of the assets protected by the TOE.

(5.2) The system operator shall ensure that the hardware selected for the TOE operates correctly according to the operational policy (and, if necessary, according to the hardware manuals)

(6) The system integrator shall be aware that the TSF has no knowledge of whether a specific SSP is appropriate for a specific product based on the TSF. The TSF will enforce any SSP as defined by the system integrator.

82 **OE.SYSTEM_OPERATOR**

The system operator shall follow the operational policy for the product in the field defined by the system integrator.

83 **OE.TRUSTWORTHY_PERSONNEL**

The personnel configuring and integrating the TOE (system integrator) are trustworthy, act according to Section 3.3, organizational security policy P.SYSTEM_INTEGRATOR and are sufficiently qualified for this task.

The personnel installing and operating the TOE (system operator) are trustworthy, act according to Section 3.3, organizational security policy P.SYSTEM_OPERATOR and are sufficiently qualified for this task.

84 **Application Note 7**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following objective for the TOE:
OE.AUDIT
For providing reliable timestamps for the audit security functionality, the system integrator shall select timer facilities in the TOE operational environment according to the SIP.

4.3 Security Objectives Rationales

85 The following table provides an overview for security objectives coverage (TOE and its environment) and also gives an evidence for sufficiency and necessity of the defined objectives. It shows that all threats and OSPs are addressed by the security objectives and it also shows that all assumptions are addressed by the security objectives for the TOE operational environment.

	OT.CONFIDENTIALITY	OT.INTEGRITY	OT.RESOURCE_AVAILABILITY	OT.SYSTEM_APPLICATION_API_PROTECTION	OT.AUDIT (optional)	OT.SECURE_STATE	OE.SYSTEM_INTEGRATOR	OE.SYSTEM_OPERATOR	OE.AUDIT (optional)	OE.TRUSTWORTHY_PERSONNEL
T.DISLOSURE	X									
T.MODIFICATION		X								
T.DEPLETION			X							
T.EXECUTION				X						
P.AUDIT (optional)					X				X	
P.SECURE_STATE						X				
P.SYSTEM_INTEGRATOR							X			
P.SYSTEM_OPERATOR								X		
A.TRUSTWORTHY_PERSONNEL										X

Table 4: Security Objectives Rationale

86 **Application Note 8**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following affinities between the Security Problem Definition and Security Objective statements:

P.AUDIT is covered by **OT.AUDIT** and **OE.AUDIT**. Thereby OT.AUDIT directly enforces the TOE-depended share of P.AUDIT and OE.AUDIT ensures a reliable TOE-external real time source.

87 A justification required for *suitability* of the security objectives to cope with the security problem definition is given below:

4.3.1 Security Objective Rationales: Threats

4.3.1.1 Threat: T.DISCLOSURE

88 If the security objective OT.CONFIDENTIALITY has been reached, the threat T.DISCLOSURE is completely eliminated.

4.3.1.2 Threat: T.MODIFICATION

89 If the security objective OT.INTEGRITY has been reached, the threat T.MODIFICATION is completely eliminated.

4.3.1.3 Threat: T.DEPLETION

90 If the security objective OT.RESOURCE_AVAILABILITY has been reached, the threat T.DEPLETION is completely eliminated.

4.3.1.4 Threat: T.EXECUTION

91 If the security objective OT.SYSTEM_APPLICATION_API_PROTECTION has been reached, the threat T.EXECUTION is completely eliminated.

4.3.2 Security Objective Rationales: Security Policies

92 Each identified security policy in this Protection Profile is addressed by at least one security objective for the TOE or security objective for the operational environment. This section provides a mapping from each security policy to the security objectives and provides a rationale how the security policy is fulfilled.

4.3.2.1 Policy: P.SECURE_STATE

93 OT.SECURE_STATE directly enforces P.SECURE_STATE.

4.3.2.2 Policy P.SYSTEM_INTEGRATOR

94 OE.SYSTEM_INTEGRATOR directly enforces P.SYSTEM_INTEGRATOR.

4.3.2.3 Policy: P.SYSTEM_OPERATOR

95 OE.SYSTEM_OPERATOR directly enforces P.SYSTEM_OPERATOR.

4.3.3 Security Objective Rationales: Assumptions

96 Each security assumption in this Protection Profile is addressed by at least one security objective for the operational environment. This section maps assumptions to environmental security objectives and provides a rationale how the assumption is fulfilled.

4.3.3.1 Assumption: A.TRUSTWORTHY_PERSONNEL

97 OE.TRUSTWORTHY_PERSONNEL directly upholds A.TRUSTWORTHY_PERSONNEL.

5 Extended Components Definition

98 This PP does not include any extended components.

6 Security Requirements

99 This part of the PP defines the detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the *functional* and *assurance* security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE.

100 The CC allows several operations to be performed on security requirements (on the component level); *refinement*, *selection*, *assignment* and *iteration* are defined in sec. 8.1 of Part 1 [1] of the CC. Each of these operations is used in this PP.

101 The **refinement** operation is used to add detail to a requirement, and, thus, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in underlined and removed words are ~~crossed out~~.

102 The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the PP author are *italicised*. Selections to be filled in by the ST author appear in square brackets with an indication that a selection has to be made, [selection:], and are *italicised*.

103 The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made by the PP author are denoted by showing as **bold** text. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment has to be made [assignment:], and are *italicised*. In some cases the assignment made by the PP authors defines a selection to be performed by the ST author. Thus this text is underlined and italicised like *this*.

104 The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier. For example, FDP_ACF.1/AS.USER_PART_CONT indicates an iteration of FDP_ACF.1 on the asset 'user partition content'. Iterations applied to assets follow the order of Table 1 in Section 3.1.1.1 (primary assets) and Table 2 in Section 3.1.1.2 (secondary assets).
For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate their relation to other SFRs with the same iteration indicator. In such a case, the iteration operation is applied to only one single component.

6.1 Security Functional Requirements for the TOE

6.1.1 Overview

105 In order to give an overview of the SFRs in the context of the security services offered by the TOE, in the following table the authors of this PP defined security functional groups and allocated the functional requirements described in the following sections to them.

Security Functional Group	Security Functional Requirements (SFRs) (SFRs always used together are grouped by "{}")
SFG_SSA: Separation in space of applications hosted in different partitions from each other and from the TOE operating system	{FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FRU_RSA.2/AS.USER_PART_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_STA: Separation in time of applications hosted in different partitions from each other and from the TOE operating system	{FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FDP_RIP.2, FRU_PRS.1, FRU_RSA.2/AS.USER_PART_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_COM: Provision and management of communication objects	{FDP_ACC.2/AS.COMMUN_OBJ_CONT, FDP_ACF.1/AS.COMMUN_OBJ_CONT}, {FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FRU_RSA.2/AS.COMMUN_OBJ_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_MAN: Management of and access to the TSF and TSF data	FIA_UID.2, all selected components of the class FMT
SFG_SPT: TSF self-protection and accuracy of security functionality	FPT_FLS.1, FPT_RCV.2 Supported by: FIA_UID.2, all selected components of the class FMT

Table 5: Security Functional Groups and their SFRs

106 **Application Note 9**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following Security Functional Group in Table 5:

Security Functional Group	Security Functional Requirements (SFRs) (SFRs always used together are grouped by "{}")
SFG_AUD: Generation and treatment of audit data according to the SSP.	FAU_GEN.1, {FDP_ACC.2/AS.AUD, FDP_ACF.1/AS.AUD} Supported by: FIA_UID.2, FMT_MOF.1, FMT_SMF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMR.1, all selected components of the class FPT

Explanatory Note 16: The SFP (Security Functional Policy) is a set of rules that are parameterised by the SSP. These rules are fix-coded in the implementation of the TSF. Thus, the behavior of the product binary image depends on the SFP and SSP.

In the following, the SFP is split up into sub-SFPs as follows:

- SFP-COMMUN-OBJ is the SFP for access control on communication object content;
- SFP-INF-FLOW is the SFP for information flow control;
- SFP-SEC-ATTR is the SFP to enforce management of security attributes.

107 **Application Note 10**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define a set of SFRs (see related suggestions for {FDP_ACC.2/AS.AUD, FDP_ACF.1/AS.AUD} below) modelling the additional functional security policy SFP-AUD for access control on audit data.

6.1.2 Class FAU Security Audit

108 **Application Note 11**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following SFR:

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1: not fulfilled, but justified: reliable timestamps shall be provided to the TOE by the TOE operational environment as required by **P.SYSTEM_INTEGRATOR**.

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [selection: *minimum, basic, detailed, not specified*] level of audit; and

c) **All events to be audited as defined by the SSP**[4].

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

[4] [assignment: *other specifically defined auditable events*]

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

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

6.1.3 Class FDP User Data Protection

109 Objects (user data assets) are defined in Table 1 in Section 3.1.1.1. Subjects are defined in Table 3 in Section 3.1.2. For the security attributes “asset” see column “Asset Name” in Table 1, for “object identity” see Table 2, for “role” and “subject identity” see Table 3. The set of all operations among subjects and objects is defined in Table 1 in Section 3.1.1.1, column “Description, Operations”.

6.1.3.1 FDP_ACC.2 Complete Access Control

110 **FDP_ACC.2/AS.COMMUN_OBJ_CONT for Asset: ‘Communication Object Content’ as Object**

Hierarchical to: FDP_ACC.1

Dependencies: FDP_ACF.1: fulfilled by FDP_ACF.1/AS.COMMUN_OBJ_CONT.

FDP_ACC.2.1 The TSF shall enforce **the SFP-COMMUN-OBJ**[5] on **all subjects with role ‘user application’ and ‘communication object content’ as object**[6] and all operations among subjects and objects.

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.

111 **Application Note 12**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following SFR:

FDP_ACC.2/AS.AUD for Asset: ‘Audit Data’ as Object

Hierarchical to: FDP_ACC.1

[5] [assignment: *access control SFP*]

[6] [assignment: *list of subjects and object*]

Dependencies: FDP_ACF.1: fulfilled by FDP_ACF.1/AS.AUD

FDP_ACC.2.1 The TSF shall enforce **the SFP-AUD**[7] on **all subjects with roler 'user application' and 'audit data' as object**[8] and all operations among subjects and objects.

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.

6.1.3.2 FDP_ACF.1 Access Control Functions

112 **FDP_ACF.1/AS.COMMUN_OBJ_CONT for Asset: 'Communication Object Content' as Object**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1: hierarchically fulfilled by FDP_ACC.2/AS.COMMUN_OBJ_CONT; FMT_MSA.3: fulfilled by FMT_MSA.3.

FDP_ACF.1.1 The TSF shall enforce the **SFP-COMMUN-OBJ**[9] to objects based on the following: **subject security attributes 'role', 'subject identity' and object security attribute 'object identity'**[10].

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 subject with the attribute 'role' set to 'user application' is allowed to treat the object of asset AS.COMMUN_OBJ_CONT, if and only if the attributes 'subject identity' and 'object identity' have values for which the SSP allows treating this object by this subject**[11].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

[7] [assignment: *access control SFP*]

[8] [assignment: *list of subjects and object*]

[9] [assignment: *access control SFP*]

[10] [assignment: *list of subjects and object*]

[11] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]**.**

113 **Application Note 13**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following SFR:

FDP_ACF.1/AS.AUD for Asset: ‘Audit Data’ as Object

Hierarchical to: No other components

Dependencies: FDP_ACC.1: hierarchically fulfilled by FDP_ACC.2/AS.AUD; FMT_MSA.3: fulfilled by FMT_MSA.3.

FDP_ACF.1.1 The TSF shall enforce the **SFP-AUD**[12] to objects based on the following: **subject security attributes ‘role’, ‘subject identity’ and object security attribute ‘object identity’.**[13].

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 subject with the attribute ‘role’ set to ‘user application’ is allowed to treat the object of asset AS.AUD, if and only if the attributes ‘subject identity’ and ‘object identity’ have values for which the SSP allows treating this object by this subject.**[14].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

[12] [assignment: *access control SFP*]

[13] [assignment: *list of subjects and object*]

[14] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

6.1.3.3 FDP_IFC.2 Complete Information Flow Control

Hierarchical to: FDP_IFC.1

Dependencies: FDP_IFF.1: fulfilled by FDP_IFF.1.

FDP_IFC.2.1 The TSF shall enforce the **SFP-INF-FLOW**[15] on

- **all subjects**[16]

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.

6.1.3.4 FDP_IFF.1 Simple Security Attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1: hierarchically fulfilled by FDP_IFC.2; FMT_MSA.3 fulfilled by FMT_MSA.3.

FDP_IFF.1.1 The TSF shall enforce the **SFP-INF-FLOW** [17] based on the following types of subject and information security attributes:

- **subject security attributes 'subject identity';**
- **information security attributes: none**[18]

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: **The operation is allowed by the SSP**[19].

FDP_IFF.1.3 The TSF shall enforce the additional information flow rules: [assignment: *additional information flow control SFP rules*].

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].

[15] [assignment: *information flow control SFP*]

[16] [assignment: *list of subjects and information*]

[17] [assignment: *information flow control SFP*]

[18] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

[19] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes]*

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

6.1.3.5 FDP_IFF.5 No Illicit Information Flows

Hierarchical to: FDP_IFF.4

Dependencies: FDP_IFC.1, hierarchically fulfilled by FDP_IFC.2.

FDP_IFF.5.1 The TSF shall ensure that no illicit information flows exist to circumvent **the SFP-INF-FLOW**.[20]

6.1.3.6 FDP_RIP.2 Full Residual Information Protection

Hierarchical to: FDP_RIP.1

Dependencies: No dependencies.

FDP_RIP.2.1 The TSF shall ensure that any previous information content of ~~a resource~~ all CPU registers being relevant to a partition switch, [assignment: *list of other resources*] is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects.

114 **Explanatory Note 17**: Partition switches are defined by SSP as part of the scheduling scheme.

6.1.4 Class FIA Identification and Authentication

6.1.4.1 FIA_UID.2 User Identification

Hierarchical to: FIA_UID.1

Dependencies: No dependencies.

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

[20] [assignment: *name of information flow control SFP*].

Explanatory Note 18: A "user" of the TOE is a user application or a system application. Please note that in the context of the security policy defined by the PP, user identification is sufficient for supporting this security policy; no user authentication is necessary. The reason for this is OE.SYSTEM_INTEGRATOR with particular obligations #(3) and #(4). It means that user authentication (users can be 'system application' and 'user application', see FMT_SMR.1) is performed through organisational measures by Systems Integrator. Indeed, the Systems Integrator has to decide – during the integration process – which application shall be put into a system partition and which application – into a user partition. The result of the TOE integration process cannot be changed during the TOE operation, i.e. an initially assigned role 'system application' or 'user application' can never be changed in the TOE operational phase. Hence, user authentication does not have to be performed technically by the TOE itself.

6.1.5 Class FMT Security Management

6.1.5.1 FMT_MOF.1 Management of Security Functions Behavior

Hierarchical to: No other components.

Dependencies: FMT_SMF.1, fulfilled by FMT_SMF.1;
FMT_SMR.1, fulfilled by FMT_SMR.1.

FMT_MOF.1.1 The TSF shall restrict the ability to *execute*[21] the functions **identified in FMT_SMF.1**[22] to **as specified by the SSP**[23].

6.1.5.2 FMT_MSA.1 Management of Security Attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1, hierarchically fulfilled by
FDP_ACC.2/AS.COMMUN_OBJ_CONT,
(and optionally by FDP_ACC.2/AS.AUD, see **Application Note 12**);
FDP_IFC.1: hierarchically fulfilled by FDP_IFC.2];
FMT_SMF.1: fulfilled by FMT_SMF.1;
FMT_SMR.1: fulfilled by FMT_SMR.1.

FMT_MSA.1.1 The TSF shall enforce the **SFP-SEC-ATTR**[24] to restrict the ability to *read and write*[25], [selection: *change_default, query, delete,*

[21] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] refinement

[22] [assignment: *list of functions*]

[23] [assignment: *the authorised identified roles*]

[24] [assignment: *access control SFP(s), information flow control SFP(s)*]

[assignment: other operations]] the security attributes **role, subject identity, object identity, and SSP enforcement data** [26] to **the TSF acting on behalf of system applications**[27].

6.1.5.3 FMT_MSA.2 Secure Security Attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] hierarchically fulfilled by FDP_IFC.2 FMT_MSA.1 fulfilled by FMT_MSA.1; FMT_SMR.1: fulfilled by FMT_SMR.1.
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for **security attributes: SSP enforcement data**[28].

Explanatory Note 19: The SSP enforcement data are represented, for example, by user partition page tables stored in user partition shapes. These page tables define which memory is accessible to user partitions.

6.1.5.4 FMT_MSA.3 Static Policy Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1: fulfilled by FMT_MSA.1, FMT_SMR.1: fulfilled by FMT_SMR.1.
FMT_MSA.3.1	The TSF shall enforce the **SFP-SEC-ATTR**[29] to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

115 **Explanatory Note** 20: Default and alternative initial values for security attributes used to enforce the SSP as well as the related authorised identified roles should be appropriate for this purpose.

[25] [selection: *change_default, query, modify, delete, [assignment: other operations]*]

[26] [assignment: *list of security attributes*]

[27] [assignment: *the authorised identified roles*]

[28] [assignment: *list of security attributes*]

[29] [assignment: *access control SFP, information flow control SFP*]

6.1.5.5 FMT_MTD.1 Management of TSF Data

116 **FMT_MTD.1/AS.SYS_APP_API for Asset: 'System Application API'**

Hierarchical to: No other components.

Dependencies: Dependencies: FMT_SMF.1:fulfilled by FMT_SMF.1; FMT_SMR.1: fulfilled by FMT_SMR.1.

FMT_MTD.1.1 The TSF shall restrict the ability to ***execute***[30] the **'System Application API'**[31] to **system applications**[32].

6.1.5.6 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].

117 **Explanatory Note** 21: For example, en- / disabling the audit function, if the author of the related ST/PP decided to include **an optional generation of audit data** (FAU_GEN.1) in the logical TOE boundary (in the scope of TSF).

6.1.5.1 FMT_SMR.1 Security Roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1, hierarchically fulfilled by FIA_UID.2.

FMT_SMR.1.1 The TSF shall maintain the roles:

- **'system application' and**
- **'user application'**[33].
- [assignment: *list of further authorised identified roles compliant with* Table 3].

[30] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[31] [assignment: *list of TSF data*]

[32] [assignment: *the authorised identified roles*]

[33] [assignment: *the authorised identified roles*]

FMT_SMR.1.2 The TSF shall be able to associate ~~users with roles~~ each application with a role.

6.1.6 Class FPT Protection of the TSF

6.1.6.1 FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state according to the SSP when the following types of failures occur:

- [assignment: *list of types of failures in the TSF*].

118 **Explanatory Note** 22: An example for an instantiation of the list of types of failures may be "TOE initialization error", "TOE run-time error", "partition initialization error", "partition run-time error".

6.1.6.2 FPT_RCV.2 Automated Recovery

Hierarchical to: FPT_RCV.1.

Dependencies: AGD_OPE.1: fulfilled by the assurance package chosen.

FPT_RCV.2.1 When automated recovery from

- **TSF initialization error**[34]
- [assignment: *list of further failures/service discontinuities*].

is not possible, the TSF shall enter a halt state ~~a maintenance mode where the ability to return to a secure state is provided~~.

119 **Explanatory Note** 23: This element describes an early phase during initialization, where automated recovery as defined in FPT_RCV.2.2 is not yet possible, because the TSF's mechanism to handle errors is not yet active.

FPT_RCV.2.2 For

- [assignment: *list of failures/service discontinuities*]

the TSF shall ensure the return of the TOE to a secure state according to the SSP using automated procedures.

[34] [assignment: *list of failures/service discontinuities*]

120 **Explanatory Note** 24: An example for an instantiation of the list of failures may be "TSF initialization error", "TSF run-time error", "partition initialization error", "partition run-time error".

121 **Explanatory Note** 25: The SSP may be configured to a secure state for each kind of failure, for example, to halt the entire TOE, restart a partition or to ignore an error. Handling of TOE initialization errors according to the SSP is only possible after the TSF's mechanism to handle errors is active.

6.1.7 Class FRU Resource Utilization

6.1.7.1 FRU_PRS.1 Limited Priority of Service

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FRU_PRS.1.1	The TSF shall assign a priority to each subject in the TSF.
FRU_PRS.1.2	The TSF shall ensure that each access to **CPU resources**[35], [assignment: *further controlled resources*] shall be mediated on the basis of the subject's assigned priority.

6.1.7.2 FRU_RSA.2 Minimum and Maximum Quotas

122 **FRU_RSA.2/AS.USER_PART_RES for Asset: 'User Partition Resources'**

Hierarchical to:	FRU_RSA.1.
Dependencies:	No dependencies.
FRU_RSA.2.1	For each 'user partition', the TSF shall enforce maximum quotas of the following resources:

- **System memory: the maximum amount of physical memory that is available to the user applications within their partition;**
- **Processing time: each user application is confined to the time window(s) as specified by the SSP**[36]
- [assignment: *further controlled resources*]

[35] [assignment: *controlled resources*]
[36] [assignment: *controlled resources*]

that <u>*user applications executed in the corresponding partition*</u>[37] can use *simultaneously*[38].

FRU_RSA.2.2 <u>For each 'user partition',</u> the TSF shall ensure the provision of minimum quantity of each

- **System memory: the minimum amount of physical memory that is available to the user applications within their partition;**
- **Processing time: each user application gets access to its time window(s) within the corresponding partition schedule as specified by the SSP**[39]
- <u>[assignment: *further controlled resources*]</u>

that is available for <u>*user applications executed in the corresponding partition*[40]</u> to use *simultaneously*[41].

123 **Explanatory Note** 26: The refinement 'for each user partition' has been performed to indicate that resources shall be assigned per user partition.

124 FRU_RSA.2/AS.COMMUN_OBJ_RES for Asset: 'Communication Object Resources'

Hierarchical to: FRU_RSA.1

Dependencies: No dependencies.

FRU_RSA.2.1 <u>For each 'communication object',</u> the TSF shall enforce maximum quotas of the following resources:

- **System memory: the maximum amount of physical memory that can be allocated to the communication object;**[42]
- <u>[assignment: *further controlled resources*]</u>

that <u>*user applications*[43]</u> can use *simultaneously*[44].

[37] [selection: *individual user, defined group of users, subjects*], refinement

[38] [selection: *simultaneously, over a specified period of time*]

[39] [assignment: *controlled resources*]

[40] [selection: *individual user, defined group of users, subjects*], refinement

[41] [selection: *simultaneously, over a specified period of time*]

[42] [assignment: *controlled resources*]

[43] [selection: *individual user, defined group of users, subjects*], refinement

[44] [selection: *simultaneously, over a specified period of time*]

FRU_RSA.2.2 For each ‘communication object’, the TSF shall ensure the provision of minimum quantity of each

- **System memory: the minimum amount of physical memory that can be allocated to a communication object;[45]**
- [assignment: *further controlled resources*]

that is available for *user applications and system applications*[46] to use *simultaneously*[47].

125 **Explanatory Note** 27: The refinement ‘for each communication object’ has been performed to indicate that resources shall be assigned per communication object.

[45] [assignment: *controlled resources*]

[46] [selection: *individual user, defined group of users, subjects*], refinement

[47] [selection: *simultaneously, over a specified period of time*]

6.2 Security Assurance Requirements for the TOE

126 This PP claims conformance to the assurance package EAL5 augmented by AVA_VAN.5.

6.3 Security Requirements Rationale

6.3.1 Security Functional Requirements Rationale

127 The following table provides an overview for security functional requirements coverage also giving an evidence for sufficiency and necessity of the SFRs chosen.

	OT.AUDIT (optional)	OT.CONFIDENTIALITY	OT.INTEGRITY	OT.RESOURCE_AVAILABILITY	OT.SECURE_STATE	OT.SYSTEM_APPLICATION_API_PROPTECTION
FAU_GEN.1 (optional, see **Application Note 14**)	X					
FDP_ACC.2/AS.COMMUN_OBJ_CONT		X	X			
FDP_ACC.2/AS.AUD (optional, see **Application Note 14**)	X	X	X			
FDP_ACF.1/AS.COMMUN_OBJ_CONT		X	X			
FDP_ACF.1/AS.AUD (optional, see **Application Note 14**)	X	X	X			
FDP_IFC.2		X				
FDP_IFF.1		X				
FDP_IFF.5		X				
FDP_RIP.2		X				

	OT.AUDIT (optional)	OT.CONFIDENTIALITY	OT.INTEGRITY	OT.RESOURCE_AVAILABILITY	OT.SECURE_STATE	OT.SYSTEM_APPLICATION_API_PROPTECTION
FIA_UID.2	X	X	X			
FMT_MOF.1	X					
FMT_MSA.1	X	X	X			
FMT_MSA.2		X	X			
FMT_MSA.3	X	X	X			
FMT_MTD.1/AS.SYS_APP_API		X	X			X
FMT_SMF.1	X	X	X			
FMT_SMR.1	X	X	X			
FPT_FLS.1					X	
FPT_RCV.2					X	
FRU_PRS.1				X		
FRU_RSA.2/AS.USER_PART_RES		X		X		
FRU_RSA.2/AS.COMMUN_OBJ_RES		X		X		

Table 6: Coverage of Security Objectives for the TOE by SFR. “X” is for where a dependency to an objective exists.

128 **Application Note 14**: If the author of the related ST/PP decided to include **an optional generation of audit data** in the logical TOE boundary (in the scope of TSF), a ST/PP compliant to this PP shall **additionally** define the following affinities between the Security Objectives and Security Requirements statements:
OT.AUDIT is covered by SFRs as showed in the Table above.
Thereby FMT_SMF.1 specifies a security management function on audit generation. FMT_MOF.1 controls usage of the security management function on audit generation.

FAU_GEN.1 ensures that when the audit function is active, the system collects events written by user applications to audit data, including events to be audited as defined by the SSP. FDP_ACC.2/AS.AUD, FDP_ACF.1/AS.AUD control that audit data can be *treated* by subjects according to the SSP. FIA_UID.2 ensures that applications are identified; FMT_SMR.1 provides security roles to applications; FMT_MSA.1 restricts the ability to read and write the security attributes role, subject identity, object identity, and SSP enforcement data to the TSF acting on behalf of user applications. FMT_MSA.3 provides well-defined default values for security attributes.

129 **Security Objective: OT.CONFIDENTIALITY**

For all assets, the operations of user applications are controlled by the TSF:

For the asset AS.COMMUN_OBJ_CONT, the SFRs {FDP_ACC.2/AS.COMMUN_OBJ_CONT, FDP_ACF.1/AS.COMMUN_OBJ_CONT}, ensure that user applications can only treat user data in the form of communication objects according to the SSP.

For the asset AS.AUD (if optionally defined), the SFRs {FDP_ACC.2/AS.AUD, FDP_ACF.1/AS.AUD} (optional) ensure that user applications can only treat audit data according to the SSP.

The TSF allows user applications to treat asset AS.USER_PART_SHAPE only according to FMT_MSA.1, FMT_MSA.2, and FMT_MTD.1/AS.SYS_APP_API.

The AS.SYS_COMP_SHAPE only can be treated by system applications via the system application API. The SFR FMT_MTD.1/AS.SYS_APP_API specifies that executing the system application API is limited to system applications.

The TSF configures the MMU of the underlying hardware to restrict each user application’s addressing to AS.USER_PART_CONT when in its own user partition to memory within its own partition according to FMT_MSA.1, FMT_MSA.2, and FMT_MSA.3. This also configures the MMU to disallow user applications to address any of these other assets (i.e., AS.COMMUN_OBJ_CONT, AS.SYS_COMP_CONT, AS.AUD (if optionally defined), AS.USER_PART_SHAPE, AS.COMMUN_OBJ_SHAPE, AS.SYS_COMP_SHAPE, AS.CONF_DATA).

FIA_UID.2 ensures that applications are identified; FMT_SMR.1 provides security roles to applications; FMT_SMF.1 specifies management functions. FMT_MSA.1 restricts the ability to read and write the security attributes role, subject identity, object identity, and SSP enforcement data to the TSF acting on behalf of user applications. FMT_MSA.2 ensures that the TSF accepts only secure values for SSP enforcement data. The TOE ensures that the security attributes role, subject identity, and object identity are only initialized once by the TSF and not written during run-time. FMT_MSA.3 provides well-defined default values for security attributes.

FDP_IFC.2 and FDP_IFF.1 ensure that (1) each user application is protected from other user applications, (2) each system application is protected from user applications, (3) the TSF is protected from user applications.

Note: this PP does not claim protection of user applications from the TSF or from system partitions because they belong to the trusted base and approved base correspondingly. Thus, FDP_IFF.1, FDP_IFC.2, FDP_IFF.5 ensure that information flows originating from user applications to other applications are restricted to information flows allowed according to the SSP, ensuring separation as defined in SSP of user partitions in space and time. FRU_RSA.2/AS.USER_PART_RES and FRU_RSA.2/AS.COMMUN_OBJ_RES ensure that no information flow against the SSP can be initiated by illicit resource depletion. FDP_RIP.2 ensures that no residual information is in CPU registers or memory caches according to the SSP, when CPU(s) are reused on a partition switch.

130 **Security Objective: OT.INTEGRITY**

For all assets, the operations of user applications are controlled by the TSF:

For the asset AS.COMMUN_OBJ_CONT, the SFRs FDP_ACC.2/AS.COMMUN_OBJ_CONT, FDP_ACF.1/AS.COMMUN_OBJ_CONT ensure that user applications can only treat user data in the form of communication objects according to the SSP.

For the asset AS.AUD (if optionally defined), the SFRs FDP_ACC.2/AS.AUD, FDP_ACF.1/AS.AUD (optional) ensure that user applications can only treat audit data according to the SSP.

The TSF allows user applications to treat asset AS.USER_PART_SHAPE only according to FMT_MSA.1, FMT_MSA.2, and FMT_MTD.1/AS.SYS_APP_API.

The AS.SYS_COMP_SHAPE only can be treated by system applications via the system application API. The SFR FMT_MTD.1/AS.SYS_APP_API specifies that executing the system application API is limited to system applications.

The TSF configures the MMU to restrict each user application’s addressing to AS.USER_PART_CONT when in its own user partition to memory within its own partition according to FMT_MSA.1, FMT_MSA.2, and FMT_MSA.3.

This also configures the MMU to disallow user applications to address any of these other assets (i.e., AS.COMMUN_OBJ_CONT, AS.SYS_COMP_CONT, AS.AUD (if optionally defined), AS.USER_PART_SHAPE, AS.COMMUN_OBJ_SHAPE, AS.SYS_COMP_SHAPE, AS.CONF_DATA).

FIA_UID.2 ensures that applications are identified; FMT_SMR.1 provides security roles to applications; FMT_SMF.1 specifies management functions. FMT_MSA.1 restricts the ability to read and write the security attributes role, subject identity, object identity, and SSP enforcement data to the TSF acting on behalf of user applications. FMT_MSA.2 ensures that the TSF accepts only secure values for SSP enforcement data. The TOE ensures that the security attributes role, subject identity, and object identity are only initialized once by the TSF and not written during run-time. FMT_MSA.3 provides well-defined default values for security attributes.

131 **Security Objective: OT.RESOURCE_AVAILABILITY**

FRU_RSA.2/AS.USER_PART_RES ensures that allocation limits are enforced on the minimum and maximum amount of memory and processing time available to a user applications within their partition.

Maximum amounts of memory and processing time available to user applications within their user partitions established by FRU_RSA.2/AS.USER_PART_RES ensure that AS.COMMUN_OBJ_RES and AS.SYS_COMP_RES are not depleted through operations of user applications.

FRU_RSA.2/AS.COMMUN_OBJ_RES ensures that allocation limits are enforced on the minimum and maximum amount of memory available to a communication object.

If the SSP defines that subjects from different user partitions share the same time window, FRU_PRS.1 ensures priority-based CPU access.

132 **Security Objective: OT.SECURE_STATE**

The TOE initialisation brings the TOE to a secure state unless any errors happen during initialisation. If errors happen, the TOE preserves the secure state through FPT_FLS.1/FPT_RCV.2.

After successful initialisation, the TOE is operating in secure state and enforces the SSP. If during operation any errors happen, the TOE preserves secure state via FPT_FLS.1/FPT_RCV.2.

133 **Security Objective: OT.SYSTEM_APPLICATION_API_PROTECTION**

FMT_MTD.1/AS.SYS_APP_API ensures that the TOE prevents any execution of the system application API by a user application. Thus, the availability of the API is restricted to only system applications.

6.3.2 Security Functional Requirements Dependencies Analysis

134 The dependency analysis for the security functional requirements shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analysed, and non-dissolved dependencies are appropriately explained.

135 The dependency analysis has directly been made within the description of each SFR in sec. 6.1 above. All dependencies being expected by CC part 2 are either fulfilled or their non-fulfilment is justified.

6.3.3 Security Assurance Requirements Rationale

136 EAL5+ has been considered appropriate to ensure the robust and reliable separation of partitions.

137 An operating system providing a generic MILS separation kernel needs to be at least as trustworthy as its guest applications, which also is an argument for a high degree of assurance.

138 A MILS separation kernel needs to be designed to be NEAT (non-bypassable, evaluable, always-invoked and tamperproof [14]). Demonstrating NEAT properties is an important argument for performing vulnerability requirements along a high level of AVA_VAN.5. The selection of the component AVA_VAN.5 provides a higher assurance than the pre-defined EAL5 package, namely requiring a vulnerability analysis to assess the resistance to penetration attacks performed by an attacker possessing a high attack potential.

139 The whole architecture of the separation kernel shall be implemented in a modular way as required by EAL5 to allow easy and thorough inspection for the NEAT properties.

Explanatory Note 28: In particular, EAL 5 has also been identified as good match for high-criticality avionics products [12].

6.3.4 Security Assurance Requirements Dependencies Analysis

140 In this section, we provide a dependency analysis for the security assurance requirements as defined by the CC. There are no unfulfilled dependencies.

141 This PP claims conformance to the standard EAL5 package augmented with AVA_VAN.5. For the EAL5 standard package, all dependencies in CC v3.1 part 3 provided packages are fulfilled. In addition, this PP also provides a dependency analysis for the security assurance requirement AVA_VAN.5.

142 AVA_VAN.5 depends on: ADV_ARC.1: fulfilled by ADV_ARC.1; ADV_FSP.4 hierarchically fulfilled by ADV_FSP.5; ADV_IMP.1: fulfilled by ADV_IMP.1; ADV_TDS.3: hierarchically fulfilled by ADV_TDS.4; AGD_OPE.1: fulfilled by AGD_OPE.1; AGD_PRE.1: fulfilled by AGD_PRE.1; ATE_DPT.1: hierarchically fulfilled by ATE_DPT.3.

7 Acknowledgement

143 Part of this PP is based on SKPP [6] [7], OSPP [8], HASK-PP [5], the security functional group approach is from [13].

8 Glossary

Application: An *application* is executable data. It is either a system application or a user application.

Attacker: An *attacker* is a threat agent (a person or a process acting on his/her behalf) trying to undermine the TOE security policy defined by the current PP and, hence, the SSP. The attacker especially tries to change properties of the assets having to be maintained according to the TOE security policy defined by the current PP (see Table 1 and Table 2 in Section 3.1.1). The attacker is assumed to possess an at most *high* attack potential.

Note that the TOE security policy defined by the current PP only addresses attacks carried out by user applications and does not address any physical attacks.

Audit Data: *Audit data* is electronic records reflecting events to be audited.

Bootloader: A *bootloader* is software that loads the TOE on the hardware and hands over the full control to the TOE. In particular, a TOE-external check of the TOE may be implemented in the bootloader (e.g. for “secure boot”).

Communication Object: Partitions can communicate with each other under the supervision of the TOE’s separation kernel. A *communication object* is an object exposed to one or multiple partitions with access rights as defined in the configuration data. The content of a communication object is the content of a communication object and exchanged (received/read and sent/written) between partitions. The resources of a communication object are physical memory space.

Configuration Data: *Configuration data* is data used by the TOE to enforce the SSP.

The configuration data defines a set of rules on how the TOE behaves. For example, the configuration data comprises the assignment of resources and communication objects to partitions. The configuration data is defined during Step 2 of the generic Lifecycle (Section 1.3.4.2).

The default configuration is that there is no information flow between any partitions. Any information flow between partitions has to be explicitly allowed by the system integrator in the configuration data.

Content: *Content* can be either the content of a user partition or a system partition or a communication object. The content of a user partition is user applications and/or data being executed and/or stored in a user partition. The content of a system component is system applications and/or data being executed and/or stored in the system component, supplied by the system integrator. The content of a communication object is the content of a communication object and exchanged (received/read and sent/written) between partitions.

Events to be Audited: The system integrator selects the *events to be audited*, that is the internal TOE events to be detected and recorded by the TOE.

Firmware: *Firmware* is software and data stored in non-volatile memory of the hardware platform that initializes the hardware after the power on.

Hardware: *Hardware* platform is the physical part of the TOE operational environment on which the TOE is executed. Usually, hardware is a board with several components such as CPUs, serial interfaces, network adapters, I/O devices etc. There are Separation Kernel Hardware Abstraction Layer controlled components (e.g. CPUs, caches) and ODSP controlled components (e.g. serial interfaces, timer). This PP considers the following parts as part of the hardware: bootloader, firmware.

Separation Kernel Hardware Abstraction Layer: A *Separation Kernel Hardware Abstraction* Layer (SK-HAL) provides specific low-level functionality for each supported CPU architecture Since the CPU instruction set is also CPU dependent, the generic components are CPU specific at the object code level.

The usual responsibility of an SK-HAL may comprise: (1) abstraction of data type representation, (2) processor exception handling, and (3) low level address space and memory management.

In operational use, the TOE always contains only one SK-HAL.

Instruction Set Architecture: The *instruction set architecture* is the set of instructions available to operate on a CPU provided by a CPU manufacturer.

Life Cycle: The typical *life cycle* phases for this kind of TOE are development (source code development), manufacturing (compilation to binary), system integration (by the system integrator), installation (by the system operator), and finally, operational use (by the system operator). Operational use of the TOE is explicitly in the focus of this PP.

Object: An *object* is a passive entity in the TOE manipulated by subjects with operations. In policies, subjects are related to objects by authorizations. This defines the way objects may be accessed by subjects. Objects are listed in Section 3.1.1.

On-board Device Support Package: An *on-board device support package* is a special purpose HAL and may contain a set of drivers for specific hardware components (a system application). It is supplied *and approved by the* system integrator. An on-board device support package can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of *a system component of* the TOE. An *on-board device support* package uses the TSF's on-board device support package API. In operational *use, the TOE* always contains only one *on-board device support package*. The main tasks of a ODSP are (1) platform initialization, (2) interrupt management, (3) hardware timer management, (4) memory region management.

Operational Policy for the Product in the Field: The *operational policy for the product in the field* covers the life cycle phase "operational use". It is a set of rules issued by the system integrator how the product in the field is to be operated. The system integrator obliges the system operator to follow this policy.

Partition: A *partition* is a logical unit maintained by the separation kernel and configured by the SSP. A *partition* contains user data. For each partition, the separation kernel provides resources. Resources of a partition comprise physical memory space and allocated CPU time for each CPU.

Partition Isolation: In the context of this PP, *partition isolation* is achieved if the generic security objectives listed for the primary and secondary assets in Section 3.1.1 are met.

Partition Switch: A partition switch occurs when a CPU(s) is/are assigned to another partition. Partition switches are defined by SSP as part of the scheduling scheme. The TSF enforces that no residual information is in CPU registers or memory caches according to the SSP.

Product Binary Image: The *product binary image* is the output of the generic Lifecycle (Section 1.3.4.2). The product binary image contains the TOE separation kernel binary image, the configuration data in a representation readable by the product binary image, the content of the on-board device support package, the content of system extensions and the content of partitions. The system integrator provides this product binary image to the system operator who, at the system operator's site, installs it on the hardware. During operational use, user applications cannot change the product binary image, e.g. no new user or system partitions can be created, no new communication objects can be created, no new user or system applications can be loaded.

Resource: In this PP we consider *resources* of partitions, communication objects and system components. The resources of a partition comprise physical memory space and allocated CPU time for each CPU*.* The resources of a communication object are physical memory space. The resources of a system component comprise physical memory space and allocated CPU time for each CPU.

Resource Usage Data: *Resource usage data* is data accounting for the usage of resources. For example, the partition resource usage data accounts for how much memory a partition has already used and how much there is still available. Resource usage data is stored in shapes. The TSF protects the confidentiality, integrity and availability of resources and shapes (see Table 2 for more details).

Secure State: A *secure state* is a state in which the TOE enforces the SSP. The secure state is maintained by a scheme for automatic handling of error conditions (configured in Step 2 of Section 1.3.4.2).

Shape: A *shape* is TSF data that contains an entity's identity, the entity's resource usage data, a set of security attributes according to the SSP assigned to the entity, and links the content assigned to an entity to the resources assigned to the entity.

SSP Enforcemenet Data: SSP enforcement data is data used by the TSF to enforce the SSP. For example, SSP enforcement data may contain page tables.

Subject: A *subject* is an active entity that can perform operations on objects. A subject requires resources provided by the TOE to become operational. A subject is an abstraction created by the TSF. Subjects are listed in Section 3.1.2.

System Application: A *system application* is any application within a system partition, a system extension, or the on-board device support package (ODSP). Only a system application in a system partition is allowed to use the TOE system partition API. Only a system application in a system extension is allowed to use the TOE system extension API. Only a system application in the ODSP is allowed to use the TOE ODSP API.

System Application API: The *system application API* is an interface to functions of the TSF available for system applications. The system application API is the combined functionality of the system partition API, the system extension API, and the ODSP API. Only a system application in a system partition is allowed to use the TOE system partition API. Only a system application in a system extension is allowed to use the TOE system extension API. Only a system application in the ODSP is allowed to use the TOE ODSP API.

System Component: A *system component* is a system partition (Section 1.3.2.2.2), system extension (Section 1.3.2.4), or an ODSP (Section 1.3.2.5). A system component contains user data supplied and approved by the system integrator.

System Extension: A *system extension* contains a software component (a system application) supplied and approved by the system integrator and coupled with the separation kernel via the system extension API. A system extension can provide specific functionality to applications within partitions only under supervision of the separation kernel. A system extension can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

System Integration Policy (SIP): The *system integration policy* (SIP) is a set of rules issued by the system integrator for using and protecting assets. The SIP also defines events to be audited.

The SIP is defined during the generic Lifecycle (Section 1.3.4.2), which can be split into the three steps: selection of the TOE operational environment and system applications and user applications (Step 1), configuration of the TOE (Step 2), and integration (Step 3). The result of performing Step 1 and Step 2 is that a SIP has been defined.

System Integrator: A *system integrator* is a person trusted to (re-)configure and integrate the TOE. This includes identifying system partitions and user partitions and assigning applications into partitions. *System integrator* may (and usually do) act on behalf of an organisation.

System Operator: A *system operator* is a person trusted to (re-)install, stop, start, restart, or access (also physically) the TOE in the field. *System operator* may (and usually do) act on behalf of an organisation.

System Partition: A *system partition* contains applications and/or data supplied and approved by the system integrator. An application in a system partition is a *system application* and uses the system partition API of the separation kernel. The content of a system partition can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

System Security Policy (SSP): The *System Security Policy* (SSP) consists of configuration choices made by a system integrator based on the subset of the configuration data rules evaluated in this PP. The SSP is enforced by the TSF and it cannot be circumvented by malicious user applications.

Time Window: A *time window* is assigned CPU time a to user application. User applications hosted in different user partitions can be assigned to different time windows according to the SIP.

TOE Operating System: The TOE *operating system* consists of the separation kernel and TSF data.

TOE Security Service: A *TOE Security Service* is a logical part of the TOE that has to be relied upon for enforcing a related subset of the rules regulating how the SSP is maintained by the TOE.

TOE Separation Kernel: The separation kernel provides the TSF and operates the TOE, by implementing mechanisms to assign resources to partitions, providing the execution environments for applications, and implementing communication between partitions as defined by the configuration data.

TOE User Manuals: The *TOE User Manuals* are documentation provided with the TOE on how to use the TOE in general environments and in security critical environments.

Treat: The verb "*treat"* is used as a synonym for "read", "execute" and "write". It describes all possible operations by a subject on an asset.

User: A *user* is an external entity. External entities are listed in Section 3.1.2.

User Application: A *user application* is any application within a user partition. A user application is allowed to use only the TOE user partition API. User applications can even be malicious, and even in that case the TOE ensures that malicious user applications are neither harming the TOE nor other applications in other partitions.

User Application Developer: A *user application developer* is a developer of an application that has been placed into a user partition by the system integrator.

User Partition: A *user partition* is defined as such by system integrator by an appropriate definition of the SSP. The content of a user partition is user applications and/or data being executed and/or stored in a user partition. User data can be executable and/or non-executable. The organizational security policy **P.SYSTEM_INTEGRATOR** requires that into any user partition, the system integrator only loads user applications.

9 Abbreviations

API: Application Programming Interface

CC: Common Criteria for Information Technology Security Evaluation

CPU: Central Processing Unit

DMA: Direct Memory Access

EAL: Evaluation Assurance Level

HASK: High-Assurance Security Kernel

ISA: Instruction Set Architecture

I/O: Input / Output

IT: Information Technology

MILS: Multiple Independent Levels of Security

MMU: Memory Management Unit

NEAT: non-bypassable, evaluable, always-invoked and tamperproof

ODSP: On-board Device Support Package

OSP: Organizational Security Policy

OSPP: Operating Systems Protection Profile

SAR: Security Assurance Requirement

SFG: Security Functional Group

SFP: Security Function Policy

SFR: Security Functional Requirement

SIP: System Integration Policy

SK-HAL: Separation Kernel Hardware Abstraction Layer

SKPP: Separation Kernel Protection Profile

SSP: System Security Policy

ST: Security Target

TOE: Target of Evaluation

TSF: Target of Evaluation Security Functionality

TSFI: TSF Interface

TSS: TOE Summary Specification

TSS_XXX: TOE Security Service XXX

10 Bibliography

Common Criteria

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, September 2012, CCMB-2012-09-001

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, September 2012, CCMB-2012-09-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 4, September 2012, CCMB-2012-09-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; Version 3.1, Revision 4, September 2012, CCMB-2012-09-004

Protection Profiles

[5] Protection Profile for High-Assurance Security Kernel, Bundesamt für Sicherheit in der Informationstechnik (BSI) and Sirrix AG security technologies, Version 1.14, June 2008

[6] U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, Information Assurance Directorate,. Version 1.03, June 2007

[7] Separation Kernel Protection Profile revisited: Choices and rationale, Timothy E. Levin, Thuy D. Nguyen, Cynthia E. Irvine, Michael McEvilley, 4th Annual Layered Assurance Workshop (LAW), 2010

[8] Operating System Protection Profile, Stephan Mueller, Gerald Krummeck, Helmut Kurth, 2010

Other Sources

[9] Design and verification of secure systems, 8th ACM Symposium on Operating System Principles, John Rushby, 1981

[10] The MILS architecture for a secure global information grid, W. Scott Harrison, Nadine Hanebutte, Paul Oman, Jim Alves-Foss, CrossTalk 18 (10), p. 20–24, 2005

[11] The MILS architecture for high assurance embedded systems, Jim Alves-Foss, W. Scott Harrison, Paul Oman, Carol Taylor, International Journal of Embedded Systems 2 (3/4), p. 239-247, 2006

[12] Towards Common Criteria certification for DO-178B compliant airborne software systems, Jim Alves-Foss, Bob Rinker, Carol Taylor, , 2002, http://www.csds.uidaho.edu/papers/Alves-Foss02b.pdf

[13] How to Create a slim and comprehensive PP: The Frame Approach, International Common Criteria Conferrence (ICCC), Igor Furgel, 2013, https://www.fbcinc.com/e/iccc/presentations/T3_D2_12pm_Furgel_How_to_create_a_slim_and_comprehensive_PP.pdf

[14] MILS virtualization for Integrated Modular Avionics, David Kleidermacher, Mike Wolf, 27th Digital Avionics Systems Conference, 2008