GLOBALPLATFORM®
THE STANDARD FOR SECURE DIGITAL SERVICES AND DEVICES

GlobalPlatform Technology

Virtual Primary Platform – OFL VNP Extension

Version 1.0.1.11

Public Review

March 2021

Document Reference: GPC_SPE_141 (formerly GPC_FST_141)

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Figures

Tables

1 Introduction

This document specifies a logical interface that enables operations on the VPP Firmware Loader of the Primary Platform of the integrated Tamper Resistant Element (TRE) within a System on Chip.

The VPP Firmware Loader interface is an extension of VNP Core Services defined in the GlobalPlatform Virtual Primary Platform – Network Protocol ([VNP]) and supports the operations defined in the Open Firmware Loader defined in the GlobalPlatform Open Firmware Loader for Tamper Resistant Secure Hardware ([OFL]).

1.1 Audience

This document is intended primarily for the use of TRE Makers, Image Makers, and Firmware Makers.

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://globalplatform.org/specifications/ip-disclaimers/. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GPC_SPE_134	GlobalPlatform Technology Open Firmware Loader for Tamper Resistant Secure Hardware v2.0	[OFL]
GPC_SPE_140	GlobalPlatform Technology Virtual Primary Platform – Network Protocol v2.0	[VNP]
GPC_SPE_142	GlobalPlatform Technology Virtual Primary Platform – Concepts and Interfaces v2.0	[VCI]
GPC_SPE_143	GlobalPlatform Technology Virtual Primary Platform – VPP Firmware Format v2.0	[VFF]
IETF RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]
RFC 4122	A Universally Unique IDentifier (UUID) URN Namespace	[RFC 4122]

Table 1-2: Informative References

Standard / Specification	Description	Ref
BSI-CC-PP-0084-2014	Security IC Platform BSI Protection Profile 2014 with Augmentation Packages	[PP-0084]

1.4 Terminology and Definitions

Terms starting with a Capital letter are defined in Table 1-3 or described in a section of this document or both.

Table 1-3: Terminology and Definitions

Term	Definition
Firmware	Defined in [VCI]. See also *VPP Firmware*.
Gate	Defined in [VNP].
Host	Defined in [VNP].
Image	Defined in [OFL].
Image Owner	Defined in [OFL].
OFL Agent	Defined in [OFL].
Part Number	Defined in [OFL].
Primary Platform	Defined in [VCI].
Public Firmware Identifier	UUID of the Firmware as defined in [OFL].
Tamper Resistant Element (TRE)	Defined in [VCI].
TRE Maker	Entity issuing the TRE and allowed to manage/update the Primary Platform Firmware.
VPP Firmware	Firmware compliant with the VPP Firmware Format as defined in [VFF].
VPP Firmware Loader	A VPP Application with special privileges that manages VPP Firmware, as specified in this document.

1.5 Abbreviations and Notations

Selected abbreviations and notations used in this document are included in Table 1-4.

Table 1-4: Abbreviations

Abbreviation	Meaning
ARP	Access Right Pattern (see [OFL])
FI	Firmware Index
NA	Not Applicable
OFL	Open Firmware Loader on the TRE
PN	Chip Part Number
REE	Regular Execution Environment as defined in [VNP]
RFU	Reserved for Future Use
RO	Read-Only

Abbreviation	Meaning
RW	Read / Write
SoC	System on Chip
TEE	Trusted Execution Environment as defined in [VNP]
TRE	Tamper Resistant Element
UUID	Universal Unique IDentifier version 5 or version 3 as defined in [RFC 4122]
VNP	VPP Network Protocol as defined in [VNP]
WO	Write-Only

1.6 Revision History

GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial changes are indicated, often with revision marks.

Table 1-5: Revision History

Date	Version	Description
March 2018	1.0	Public Release (with document reference GPC_FST_141)
October 2019	1.0.1.4	Committee Review
December 2019	1.0.1.5	Member Review
March 2021	1.0.1.11	Public Review
TBD	2.0	Public Release (with document reference GPC_SPE_141)

2 VPP Firmware Loader Platform

2.1 Requirements

Reference	Description
REQ1	The long-term credentials of the VPP Firmware Loader shall be compliant with the OFL specification defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Secure Hardware ([OFL]).
REQ2	The VPP Firmware Loader shall claim conformance with loader package 2 of the Protection Profile BSI-CC-PP-0084-2014 ([PP-0084]).
REQ3	The VPP Firmware Loader shall be compliant with the OFL specification in [OFL].
REQ4	The VPP Firmware Loader shall run in Privileged CPU Mode as defined in GlobalPlatform Virtual Primary Platform – Concepts and Interfaces ([VCI]).
REQ5	The OFL Service and Application Gates shall implement Credit-based Data Flow Control as defined in GlobalPlatform Virtual Primary Platform – Network Protocol ([VNP]).

2.2 Overview

Two Hosts are involved in the support of VPP Firmware Loader operations:

- OFL Agent Host loads the Images or manages the Firmwares defined in [OFL] and runs in a TEE Host Domain or REE Host Domain as defined in [VNP].
- OFL Host is in the TRE Host Domain as defined in [VNP] and runs in Privileged CPU Mode as defined in [VCI].

Both Hosts exchange commands, responses, and events using the VNP Core Services as specified in [VNP]. The Gates define the entry point to a service that operates inside the OFL Host to perform the loading or update of a Firmware, and to manage the Firmware.

Figure 2-1 illustrates Hosts and Gates in a valid network supporting access to the above functions.

Figure 2-1: Hosts and Gates

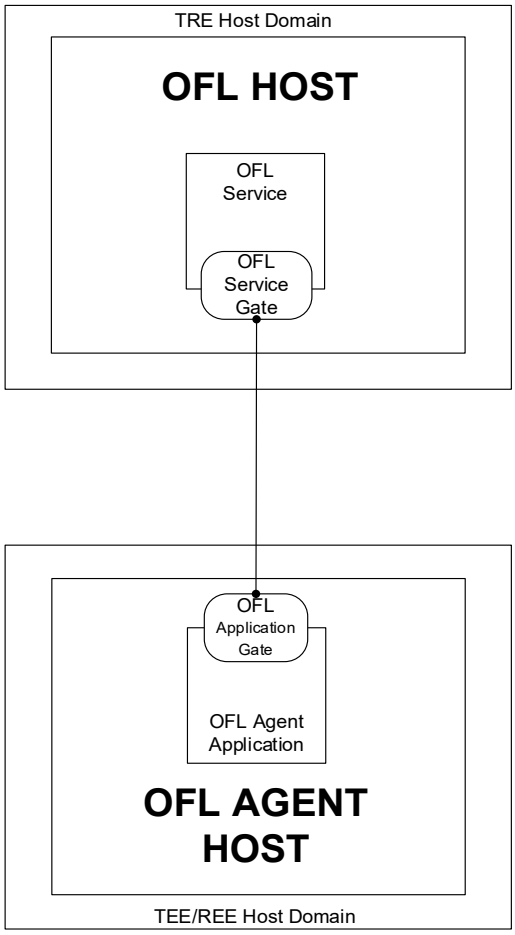


Table 2-1 lists the Gate URN as defined in [VNP] for the OFL Host providing the OFL Service Gate. The syntax of the Gate URN is defined in [VNP].

Table 2-1: Gate URN

Gate	NID	NSS	Pre-generated Identifier
OFL Service Gate	urn:globalplatform.org	TRE:HCI:OFL	BB780E30-419A-5B71-9B98-18A042E75899

2.3 OFL Service Gate

2.3.1 Overview

An OFL Service Gate allows the implementation of operations defined in [OFL] for the OFL Service within the OFL Host.

2.3.2 Commands

The OFL Service Gate shall support the following commands:

Table 2-2: OFL Commands

Value	Command
'10'	OFL_DO_OPERATE
'11'	OFL_LOAD_SEGMENT
'12'	OFL_CHANGE_SEGMENT
'13'	OFL_DELETE_SESSION
'14'	OFL_ENABLE_FIRMWARE
'15'	OFL_DISABLE_FIRMWARE
'16'	ANY_SET_PARAMETER
'17'	ANY_GET_PARAMETER
'80' to '8F'	Range of Identifiers reserved for ETSI

The commands are described in the following sections.

2.3.2.1 OFL_DO_OPERATE

With the command OFL_DO_OPERATE, an OFL Agent Application requests the OFL Service to load an Image.

The command parameter is defined in [OFL] as follows:

Table 2-3: OFL_DO_OPERATE Command Parameter

Description	Length
OFL_DO_OPERATE parameter in [OFL]	N_1 in [OFL]

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameter.

In case of failure, the appropriate response shall be provided. The response code may be selected from the responses defined in Table 2-12.

2.3.2.2 OFL_LOAD_SEGMENT

With the command OFL_LOAD_SEGMENT, an OFL Agent Application requests the OFL Service to load a segment of the Firmware into the TRE.

The command parameter is defined in [OFL] as follows:

Table 2-4: OFL_LOAD_SEGMENT Command Parameter

Description	Length
OFL_LOAD_SEGMENT parameter in [OFL]	N_2 in [OFL]

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameter. The command may be rejected as defined in [OFL].

In case of failure, the appropriate response shall be provided. The response code may be selected from the responses defined in Table 2-12.

2.3.2.3 OFL_CHANGE_SEGMENT

With the command OFL_CHANGE_SEGMENT, an OFL Agent Application requests the OFL Service to change the segment parameters used to decipher the parameters of the corresponding segment loaded using the OFL_LOAD_SEGMENT command.

The command parameter is defined in [OFL] as follows:

Table 2-5: OFL_CHANGE_SEGMENT Command Parameter

Description	Length
OFL_CHANGE_SEGMENT parameter in [OFL]	N_3 in [OFL]

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameters.

In case of failure, the appropriate response shall be provided. The response code may be selected from the responses as defined in Table 2-12.

2.3.2.4 OFL_DELETE_SESSION

With the command OFL_DELETE_SESSION, an OFL Agent Application requests the OFL Service to delete a Firmware and its Firmware Session (see [OFL]). The parameter of the OFL_DELETE_SESSION command shall be set to one of the Public Firmware Identifiers listed in the UUID_FIRMWARE_LIST registry entry.

The OFL_DELETE_SESSION command has a single parameter. No policy may prevent the deletion of a session from the Host.

The command parameter is defined in [OFL] as follows:

Table 2-6: OFL_DELETE_SESSION Command Parameter

Description	Length
Public Firmware Identifier in [OFL]	16

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameters.

In case of failure, the appropriate response shall be provided. The response code may be selected from the responses defined in Table 2-12.

OFL shall deregister the Host linked to the deleted Firmware, as defined in [VNP] Host deregistration.

2.3.2.5 OFL_ENABLE_FIRMWARE

With the command OFL_ENABLE_FIRMWARE, an OFL Agent Application requests the OFL Service to set a Firmware in ENABLED state (see [OFL]). The parameter of the OFL_ENABLE_FIRMWARE command shall be set to one of the Public Firmware Identifiers listed in the UUID_FIRMWARE_LIST registry entry.

The OFL_ENABLE_FIRMWARE command has a single parameter. This command shall not be performed when the OFL state is TERMINATED. This command shall fail if the targeted Firmware has been previously disabled by using the OFL_DO_OPERATE command.

The command parameter is defined in [OFL] as follows:

Table 2-7: OFL_ENABLE_FIRMWARE Command Parameter

Description	Length
Public Firmware Identifier in [OFL]	16

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameter.

In case of failure, the appropriate response shall be provided. The response code may be selected from the responses defined in Table 2-12.

OFL shall register the Host linked to the enabled Firmware, as defined in [VNP] Host registration.

2.3.2.6 OFL_DISABLE_FIRMWARE

With the command OFL_DISABLE_FIRMWARE, an OFL Agent Application requests the OFL Service to set a Firmware in the DISABLED state (see [OFL]). The parameter of the OFL_DISABLE_FIRMWARE command shall be set to one of the Public Firmware Identifiers listed in the UUID_FIRMWARE_LIST registry entry.

The OFL_DISABLE_FIRMWARE command has a single parameter. This command shall not be performed when the OFL state is TERMINATED.

The command parameter is defined in [OFL] as follows:

Table 2-8: OFL_DISABLE_FIRMWARE Command Parameter

Description	Length
Public Firmware Identifier in [OFL]	16

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameter.

In case of failure, the appropriate response shall be provided. The response code may be selected from responses defined in Table 2-12.

OFL shall deregister the Host linked to the disabled Firmware, as defined in [VNP] Host deregistration.

2.3.2.7 ANY_SET_PARAMETER

With the command ANY_SET_PARAMETER, an OFL Agent Application requests the OFL Service to set the content of the registry entry to the value corresponding to the requested parameter identifier, if the corresponding access right allows it.

The command parameter is defined in [OFL] as follows:

Table 2-9: ANY_SET_PARAMETER Command Parameters

Description	Length
Parameter Identifier in OFL Service Gate Registry	1
Value of the parameter	N_4

When the command is successfully executed, the OFL Host shall send the response ANY_OK without parameter.

In case of failure, the appropriate response shall be provided. The response code may be selected from responses defined in Table 1-1Table 2-12.

2.3.2.8 ANY_GET_PARAMETER

With the command ANY_GET_PARAMETER, an OFL Agent Application requests the OFL Service to get the content of the registry entry corresponding to the requested parameter identifier, if the corresponding access right allows it.

The command parameter is defined in [OFL] as follows:

Table 2-10: ANY_GET_PARAMETER Command Parameter

Description	Length
Parameter Identifier in OFL Service Gate Registry	1

When the command is successfully executed, the OFL Host shall send the response ANY_OK with the following parameter:

Table 2-11: ANY_GET_PARAMETER Response Parameter

Description	Length
Value of the parameter	N_5

In case of failure, the appropriate response shall be provided. The response code may be selected from responses defined in Table 2-12.

2.3.3 Events

The Gate has no additional events.

2.3.4 Responses

Table 2-12 defines the Error/Command matrix for the Gate.

Table 2-12: Error/Command Matrix

Error Code / Command	ANY_OK	ANY_E_CMD_PAR_UNKNOWN	ANY_E_NOK	ANY_E_REG_PAR_UNKNOWN	ANY_E_CMD_NOT_SUPPORTED	ANY_E_REG_ACCESS_DENIED
OFL_DO_OPERATE	•	•	•			
OFL_LOAD_SEGMENT	•		•	•		•
OFL_CHANGE_SEGMENT	•	•	•			
OFL_DELETE_SESSION	•	•	•			
OFL_ENABLE_FIRMWARE	•	•	•			
OFL_DISABLE_FIRMWARE	•	•	•			
ANY_SET_PARAMETER	•	•	•	•		•
ANY_GET_PARAMETER	•	•	•	•		•
UNKNOWN COMMANDS			•		•	

2.3.5 Registry

The registry refers to a loaded Firmware.

The parameter names are the same as the names used in [OFL], to ease the link between the current specification and the OFL specification.

Table 2-13 defines the registry for the OFL Gate as defined in [OFL].

OFL Agent selects a Firmware using a Firmware UUID contained in the UUID_FIRMWARE_LIST registry entry ’0D’ and writes it in the registry entry ’7F’. Registry entries flagged with a dot in the FI column of Table 2-13 are filled by OFL with information related to the selected Firmware. If the UUID_FIRMWARE registry entry is not set to a Public Firmware Identifier listed in the registry entry UUID_FIRMWARE_LIST, then the corresponding registry entries shall be empty.

Table 2-13: OFL Gate Registry

Identifier	Parameter	Access Right	FI	Description	Length	Default
'7F'	UUID_FIRMWARE	RW	-	Registry entry set by OFL Agent in order for OFL to select the Firmware that is targeted and fill the registry entries flagged in column FI.	16	-
'01'	OFL_VERSION	RO		As defined in [OFL]	2	'0200'
'02'	OFL_TYPE_UUID	RO			16	-
'03'	TRSH_CREDENTIALS _PARAMETER	RO			-	-
'04'	IDS_CREDENTIALS _PARAMETER	WO			-	-
'05'	OFL_STATE	RO			1	-
'06'	ARP_STATE	RO			1	
'07'	LIST_ISID	RO			N_6*16	-
'08'	UUID_CI _GENERATION_LIST	RO			N_7*16	-
'09'	UUID_CI _VERIFICATION_LIST	RO			N_8*16	-
'0A'	OPERATION_TOKEN	RO			32	-
'0B'	SERIAL_NUMBER	RO			16	-
'0C'	PART_NUMBER	RO			16	-
'0D'	BATCH_NUMBER	RO			16	-
'0E'	UUID_FIRMWARE _LIST	RO			N_9*16	-

Identifier	Parameter	Access Right	FI	Description	Length	Default
'0F'	GROUP_ID_LIST	RO			N_{10}*16	-
'10'	GROUP_ID	RO	●		16 or 0	-
'11'	CURR_VERSION	RO	●		2 or 0	-
'12'	FIRMWARE_STATE	RO	●		1 or 0	-
'13'	URN_PI	WO			N_{11}	-
'14'	DET_PI	RO			16	-
'15'	EPH_PI	RO			16	-
'16'	IMU_PI	RO			16	-
'17'	IDS2_EXT	RO			N_{12}	-
'18'	BIST	RO			1	-
'80' to '8F'	Registry entries reserved for ETSI	-		Reserved for ETSI	-	-

2.3.5.1 OFL State

Table 2-14 defines the possible values of the registry entry ‘OFL_STATE’ in the OFL Service Gate as defined in [OFL].

Table 2-14: OFL States

Description	Value
ENABLED: as defined in [OFL].	'01'
DISABLED: as defined in [OFL].	'02'
TERMINATED: as defined in [OFL].	'03'

2.3.5.2 ARP State

Table 2-15 defines the possible values of the registry entry ‘ARP_STATE’ in the OFL Service Gate as defined in [OFL].

Table 2-15: ARP States

Description	Value
BLANK: as defined in [OFL].	'00'
LOCKED: as defined in [OFL].	'01'
UNLOCKED: as defined in [OFL].	'02'

2.3.5.3 Firmware State

Table 2-16 defines the possible values of the registry entry ‘FIRMWARE_STATE’ in the OFL Service Gate as defined in [OFL].

Table 2-16: Firmware States

Description	Value
ENABLED: as defined in [OFL].	'00'
DISABLED: as defined in [OFL].	'01'

2.4 OFL Application Gate

2.4.1 Overview

An OFL Application Gate allows implementing operations defined in [OFL] for the OFL Agent Application within the OFL Agent Host.

2.4.2 Commands

This Gate has no additional commands.

2.4.3 Events

This Gate has no additional events.

2.4.4 Registry

This Gate has no registry entries defined.

2.5 Procedures

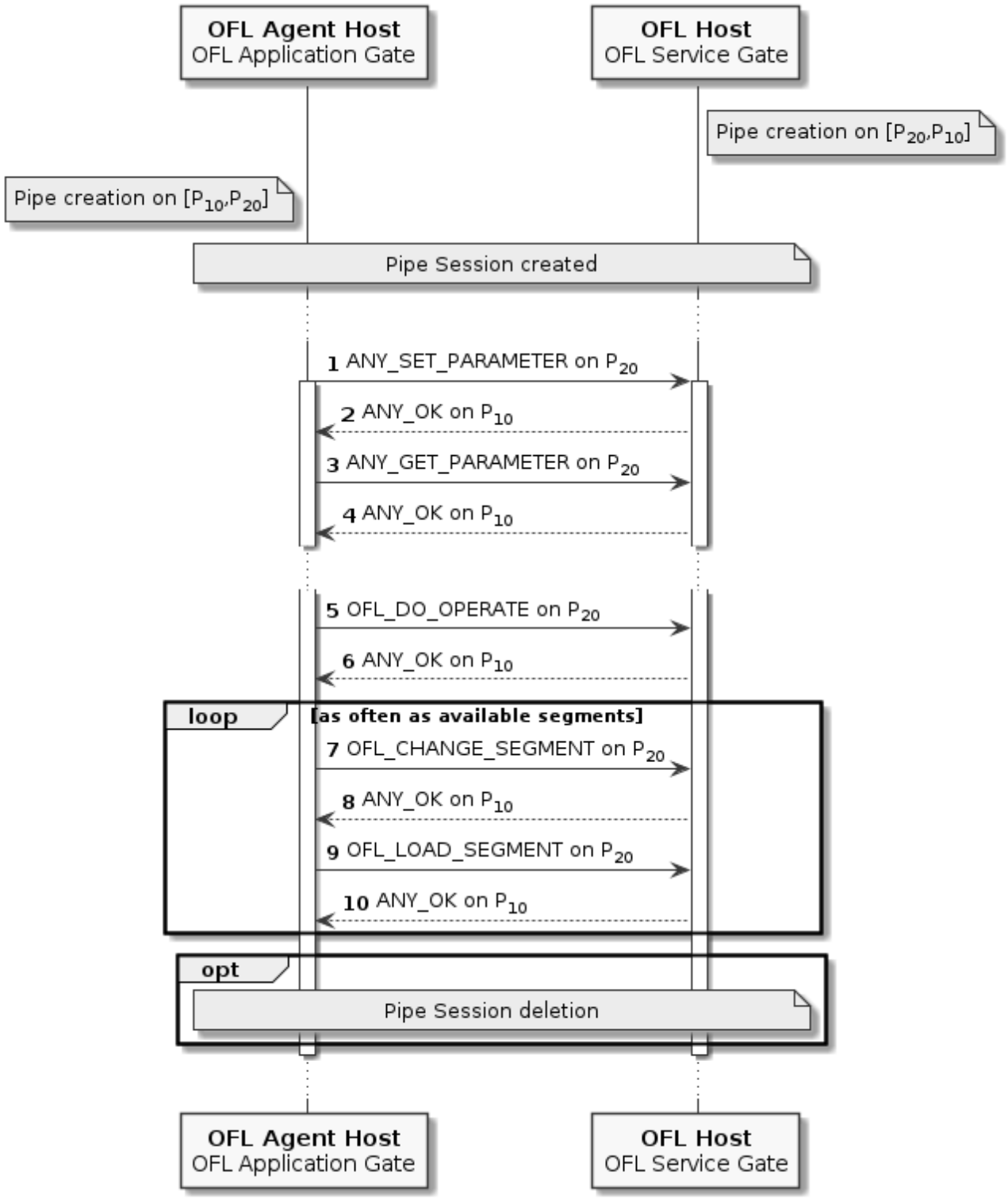
2.5.1 Overview

Note: We assume, for the following procedures, a Pipe Session opened as defined in [VNP].

2.5.2 OFL Image Loading Procedure

Figure 2-2 illustrates the basic exchange between the OFL Application Gate and the OFL Service Gate.

Figure 2-2: OFL Image Loading



The procedure has ten steps:

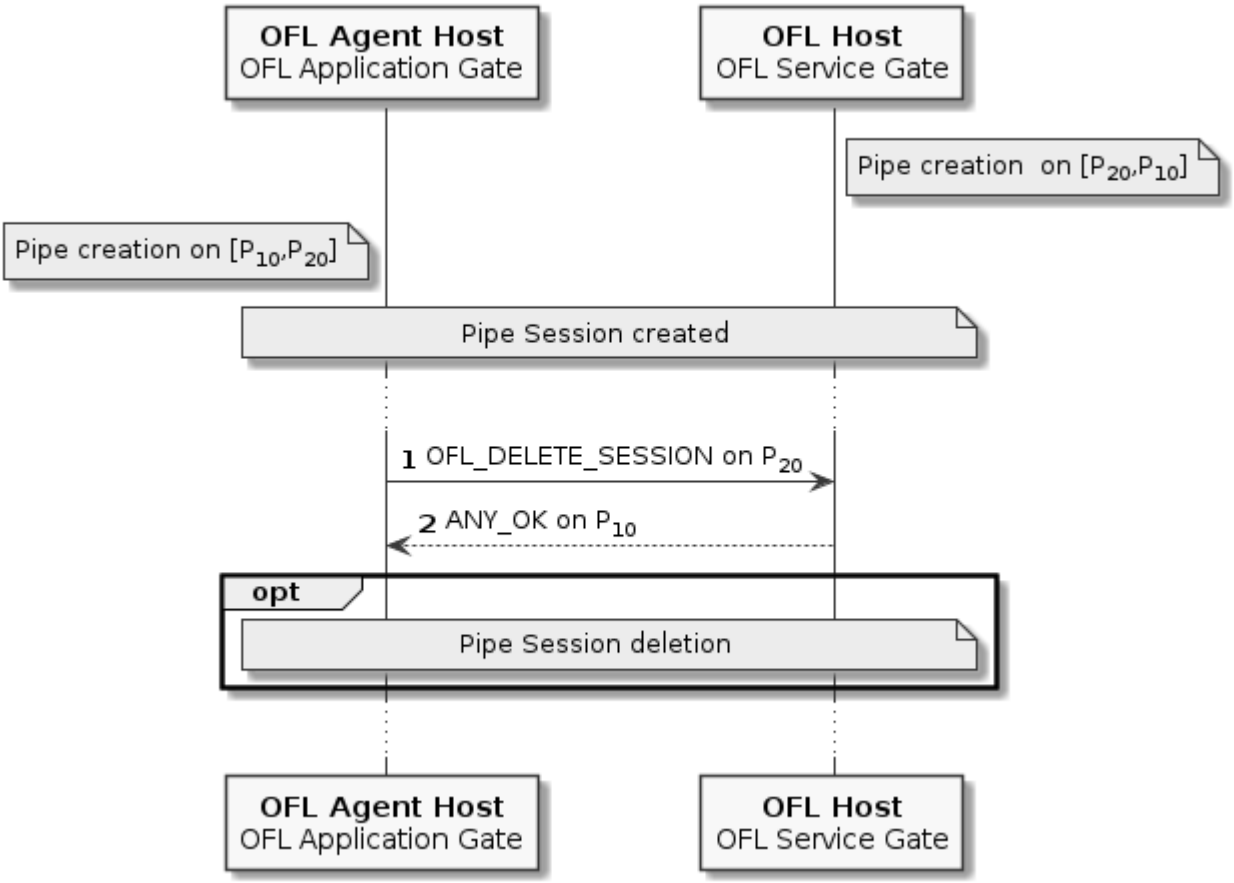
1. The OFL Application Gate sends the ANY_SET_PARAMETER command for storing the IDS credentials parameters (see [OFL]) from the Image Owner in the OFL Service Gate.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.
3. The OFL Application Gate sends the ANY_GET_PARAMETER command for getting the initial TRE credentials parameter (see [OFL]) computed from the Image Owner parameter.
4. The OFL Service Gate returns ANY_OK if the command is successfully executed.
5. The OFL Application Gate sends the OFL_DO_OPERATE command for executing the operation defined in the OFL_DO_OPERATE parameter to the OFL Host.
6. The OFL Service Gate returns ANY_OK if the command is successfully executed.
7. The OFL Application Gate may send the OFL_CHANGE_SEGMENT command to position the next set of data with the right deciphering parameters.
8. The OFL Service Gate returns ANY_OK if the command is successfully executed.
9. The OFL Application Gate sends the OFL_LOAD_SEGMENT command to load a memory segment into the TRE.
10. The Open Firmware Loader Service Gate returns ANY_OK if the command is successfully executed. We may return to step 7. Steps 7 to 10 may be repeated as needed.

If any error occurs between step 5 and step 10, then the procedure shall either be restarted at step 5 or shall be terminated, depending on the implementation.

2.5.3 OFL Delete Firmware Session Procedure

Figure 2-3 illustrates the exchange for deleting a Firmware Session as defined in [OFL] within the persistent memory dedicated to OFL Service Gate.

Figure 2-3: OFL Delete Session



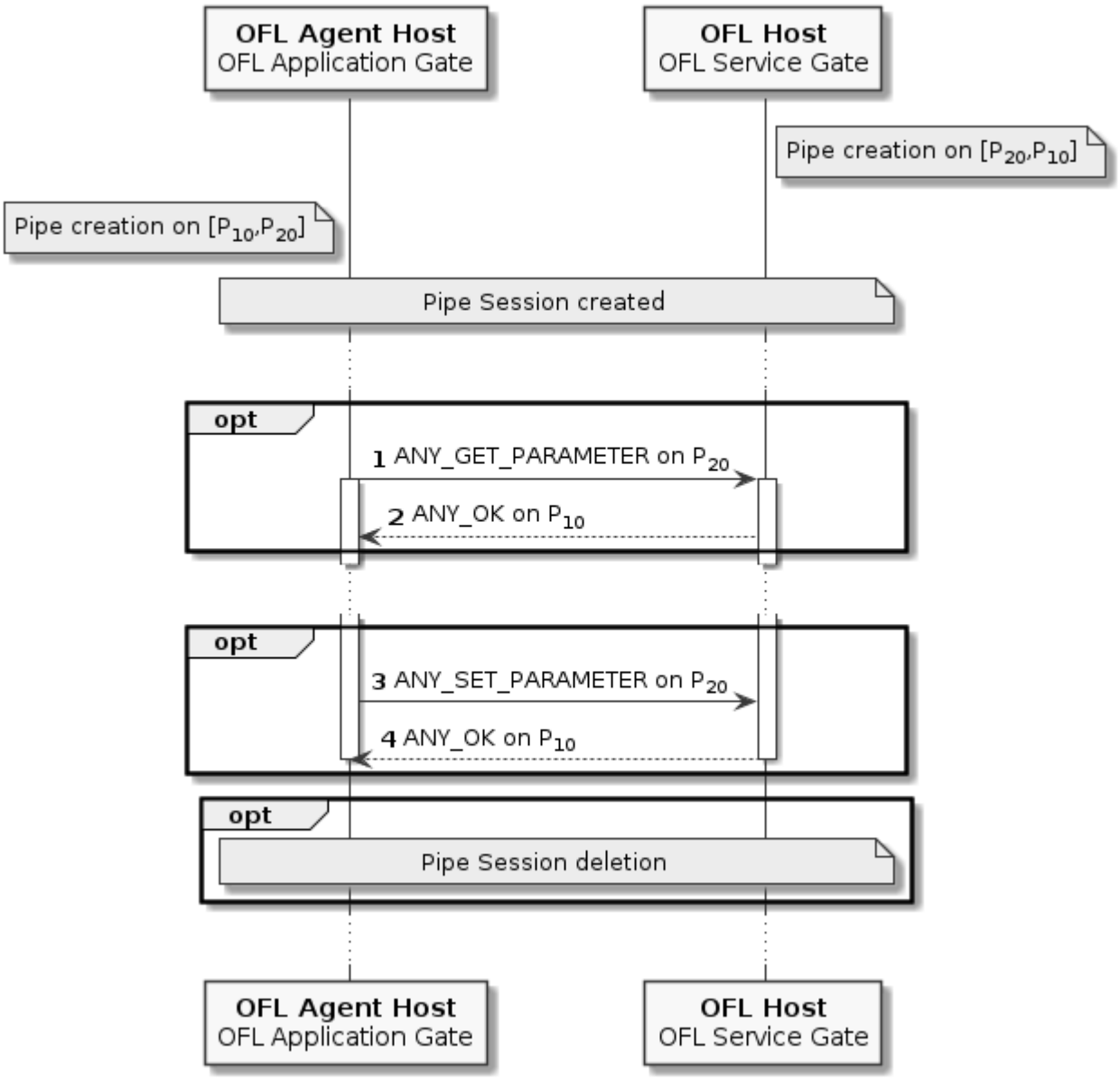
The procedure has two steps:

1. The Open Firmware Loader Application Gate sends the OFL_DELETE_SESSION command to the OFL Service Gate. The Firmware Session related to a given already loaded Firmware is deleted.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.

2.5.4 OFL Access Registry Procedure

Figure 2-3 illustrates the exchange for accessing the registry of the OFL Service Gate.

Figure 2-4: OFL Access Registry



The procedure for reading an entry of the registry has two steps:

1. The OFL Application Gate sends the ANY_GET_PARAMETER command to the OFL Service Gate.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.

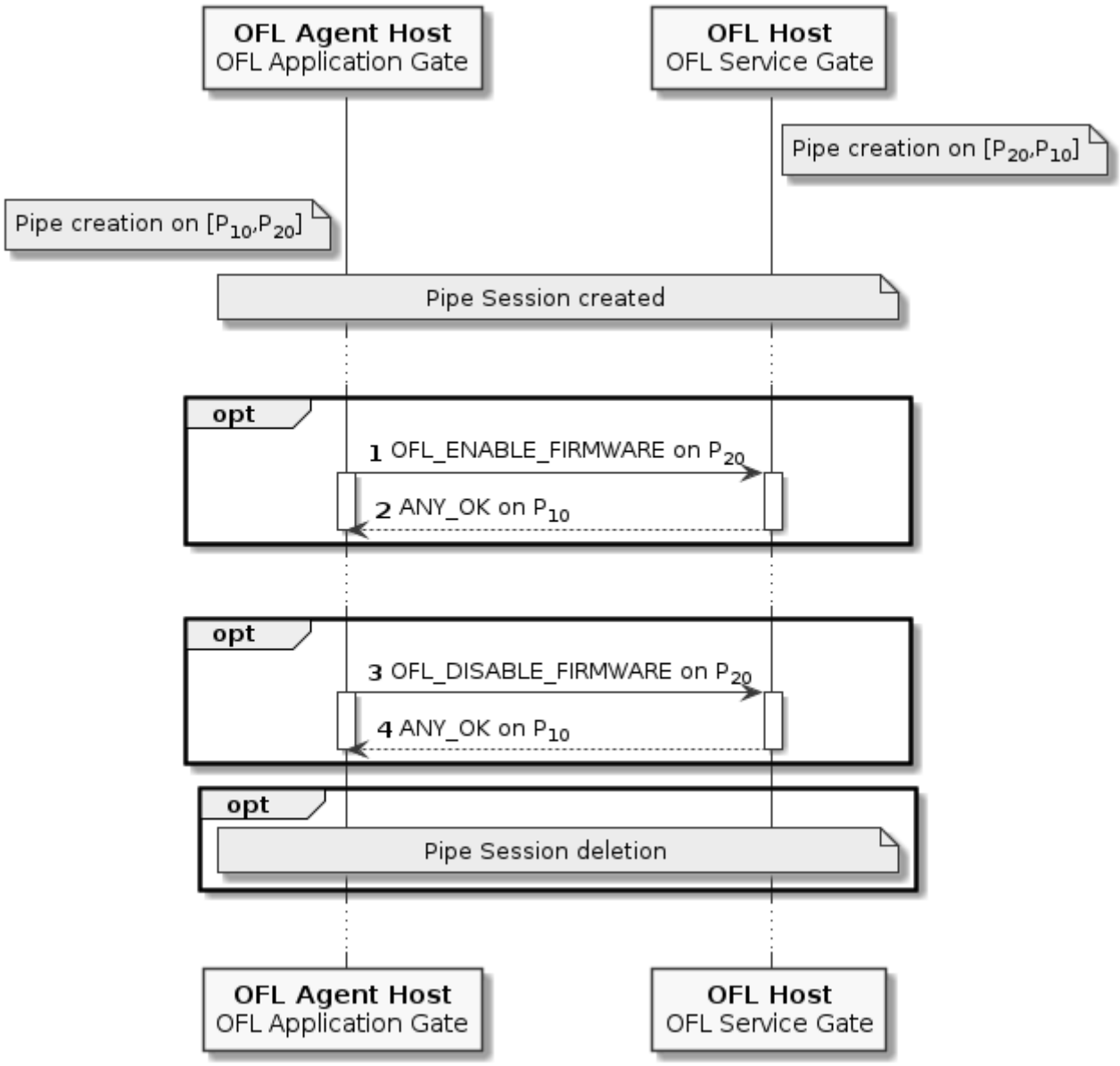
The procedure for writing an entry of the registry has two steps:

1. The OFL Application Gate sends the ANY_SET_PARAMETER command to the OFL Service Gate.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.

2.5.5 OFL Enable or Disable Firmware Procedure

Figure 2-5 illustrates the exchange for enabling or disabling a Firmware from OFL Application Gate.

Figure 2-5: OFL Enable/Disable Firmware



The OFL enable Firmware procedure has two steps:

1. The OFL Application Gate sends the OFL_ENABLE_FIRMWARE command to the OFL Service Gate.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.

The OFL disable Firmware procedure has two steps:

1. The OFL Application Gate sends the OFL_DISABLE_FIRMWARE command to the OFL Service Gate.
2. The OFL Service Gate returns ANY_OK if the command is successfully executed.