GlobalPlatform Card

Digital Letter of Approval

Version 1.0

Public Release

November 2015

Document Reference: GPC_SPE_095

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

The deployment of applications from different providers on a certified secure components (SE or TEE) implies a management process that checks the compatibility of the application with the platform at the functional level and in some cases at the security evaluation level.

GlobalPlatform has developed a Card Composition Model (described in GlobalPlatform Card Composition Model [GP Comp Mdl]): a common, cross-industry certification model for SEs supporting application loading in post-issuance. In essence, it outlines a methodology that streamlines the security evaluation of a mobile composite product by specifying how EMVCo and Common Criteria certificates can be re-used for chips and SE platforms that have previously been certified.

In the context of the deployment of secure applications, GlobalPlatform identifies and defines the roles and responsibilities of all stakeholders within the multiple applications ecosystem. As an example, GlobalPlatform Messaging Specification for Management of Mobile-NFC Services ([GP SM]) outlines the 'language' that should be used across all parties to ensure global consistency and clear communication within a NFC based eco systems. This 'language' is referred to as 'messages'.

[GP SM] supports back-office activities that facilitate the secure deployment of a mobile service to a mobile device. This specification is also essential for dealing with post deployment events. This could be voluntary – such as an end-user requesting an update to service privileges or personal detail changes – or involuntary – a device is lost or stolen and the services on the mobile (for example banking or loyalty) need to be suspended.

In this dynamic world, the provisioning flow defined in [GP SM] needs to be synchronized with the certification process and results defined in the Composition Model [GP Comp Mdl].

The aim of this document is to describe the additional flows needed to synchronize both processes, and a standardized data structure called Digital Letter of Approval that represents a proof of a certification or qualification process.

1.1 Audience

This document is intended primarily for the use of Product Issuers. It provides useful information to all stakeholders such as TSM vendors, Certification Bodies, Evaluation Bodies, Platform Developers, Application Developers, Platform Evaluators, Platform Issuers, platform manufacturers, Verification Authorities, and so on.

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://www.globalplatform.org/specificationsipdisclaimers.asp. 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
GlobalPlatform Card Specification	GlobalPlatform Card Specification v2.3	[GP CS]
GlobalPlatform Messaging Specification	GlobalPlatform System – Messaging Specification for Management of Mobile-NFC Services v1.2	[GP SM]
GlobalPlatform System Protocol Discovery Mechanism Specification	GlobalPlatform System – System Protocol Discovery Mechanism Specification v1.0	[GP SPDM]
GlobalPlatform Card Composition Model	GlobalPlatform Card Composition Model v1.1	[GP Comp Mdl]
GlobalPlatform Card Composition Model Security Guidelines for Basic Applications	GlobalPlatform Card Composition Model – Security Guidelines for Basic Applications v2.0	[GP Guidelines Basic Applications]
GlobalPlatform Card Overview of Complete Life Cycle for GlobalPlatform SE Products	GlobalPlatform Card Overview of Complete Life Cycle for GlobalPlatform SE Products v1.0	[GP Life Cycle]
GlobalPlatform Device TEE System Architecture	GlobalPlatform Device Technology – TEE System Architecture v1.0	[GP TEE Sys Arch]
GlobalPlatform Device TEE Management Framework	Global Platform Device Technology – TEE Management Framework v1.0	[GP TEE Mgmt]
ISO/IEC 8859-1	Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No.1	[8859-1]
IETF JSON Schema Draft 04	JSON Schema: core definitions and terminology Draft 04	[JSON Schema]
ITU-T X.680	Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation	[ASN.1]
RFC 1778	The String Representation of Standard Attribute Syntaxes	[RFC 1778]
RFC 2119	Key words for use in RFCs to Indicate Requirement Levels	[RFC 2119]
RFC 3986	Uniform Resource Identifier (URI): Generic Syntax	[RFC 3986]
RFC 4627	The application/json Media Type for JavaScript Object Notation (JSON)	[RFC 4627]
RFC 4646	Tags for Identifying Languages	[RFC 4646]
RFC 5246	The Transport Layer Security (TLS) Protocol Version 1.2	[RFC 5246]

Standard / Specification	Description	Ref
W3C.REC-xmlschema-2	Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004 http://www.w3.org/TR/2004/REC-xmlschema-2-20041028	[XML Data Types]
W3C, XML Signature Syntax and Processing Version 1.1	W3C, XML Signature Syntax and Processing, Version 1.1 http://www.w3.org/TR/2009/WD-xmldsig-core1-20090226/	[XML SIG]
W3C, Exclusive XML Canonicalization Version 1.0	W3C, Exclusive XML Canonicalization, Version 1.0 http://www.w3.org/TR/xml-exc-c14n	[XML C14N]

1.4 Terminology and Definitions

Tables may include an "MOC" column meaning "Mandatory/Optional/Conditional". This column specifies whether data must be present in the function or message. The following definitions apply to these terms:

- Mandatory (M): Means that an entry must be supplied.
- Optional (O): Means that an entry can be supplied, but is not required to be supplied.
- Conditional (C): Means that the need to supply an entry is dependent upon a particular condition.

The key words “MUST”, “MUST NOT”, “SHALL”, “SHALL NOT”, “REQUIRED”, “SHOULD, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document indicate normative statements and are to be interpreted as described in [RFC 2119].

Selected technical terms used in this document are included in Table 1-2. Additional technical terms are defined in [GP CS], in [GP Life Cycle], and in [GP SM].

Table 1-2: Terminology and Definitions

Term	Definition
Application Developer	As defined in [GP Comp Mdl], an entity responsible for development of an application.
Application Issuer	As defined in [GP Comp Mdl], an entity responsible for the security of an application.
Authority	In the context of this document, a certification, evaluation, approval, qualification, or validation scheme that may issue Digital Letters of Approval.
Device and Mobile Subscription Registrar (DMSR)	As defined in [GP SM], a role with responsibilities that include: • Performing the eligibility checking on the Mobile Subscription • Acting as a Device registrar to track the association between the Secure Element, the Device, and the Mobile Subscription. • Providing the capabilities of the Device.
Digital Letter of Approval (DLOA)	A digital representation of a Letter of Approval, signed by an Authority.
DLOA Registrar	A role that stores DLOAs and provides an interface to enable TSMs to retrieve them.
Letter of Approval (LOA)	A letter generated by an Authority, identifying a platform or application that has completed certification, evaluation, approval, qualification, or validation.
Management System	In the context of this document, the local or remote back-end systems that support a GlobalPlatform secure component. Management Systems perform functions such as authentication; administration; initialization; personalization; and post-issuance application and data loading.
Platform	One computing engine and executable code that provides a set of functionalities; may contain one Primary Root of Trust and unlimited Secondary Roots of Trust.

Term	Definition
Rich Execution Environment (REE)	An environment that is provided and governed by a Rich OS, potentially in conjunction with other supporting operating systems and hypervisors; it is outside of the TEE. This environment and applications running on it are considered un-trusted. Contrast *Trusted Execution Environment*.
Secure Element (SE)	A tamper resistant component which is used in a device to provide the security, confidentiality, and multiple application environment required to support various business models. May exist in any form factor such as UICC, embedded SE, smartSD, smart microSD, etc.
Secure Element Issuer (SEI)	A role that holds the ultimate responsibility for the GlobalPlatform card. Responsible for developing the card product profile, choosing the platform and application technologies, and designing the card layout. Usually holds a particular Security Domain in the SE: the Issuer Security Domain (ISD).
Secure Element Issuer Trusted Service Manager (SEI TSM)	An actor that acts on behalf of the Secure Element Provider. An SEI TSM controls access to the SE and provides Card Content Management operations to other actors.
Secure Element Provider	As defined in [GP SM], an actor that is the owner of a Secure Element.
Service Provider	As defined in [GP SM], an actor such as a bank, a transport company, a retailer, etc., that owns services provided to consumers. These services need to be deployed in a platform within the end-users' mobile equipment.
Service Provider Trusted Service Manager (SP TSM)	An actor that provides one or more technical roles and possibly business roles to the Service Provider. An SP TSM is trusted by the other actors to be in charge of service management and delivery. The SP TSM acts as an aggregator and may simultaneously support several Service Providers and SEI TSMs while maintaining confidentiality between the actors.
Trusted Application (TA)	An application running inside the Trusted Execution Environment that provides security related functionality to Client Applications outside of the TEE (i.e. running in the Rich Execution Environment). (For more information, see [GP TEE Sys Arch].)
Trusted Execution Environment (TEE)	An execution environment that runs alongside but isolated from an REE. A TEE has security capabilities and meets certain security-related requirements: It protects TEE assets from general software attacks, defines rigid safeguards as to data and functions that a program can access, and resists a set of defined threats. There are multiple technologies that can be used to implement a TEE, and the level of security achieved varies accordingly. (For more information, see [GP TEE Sys Arch].) Contrast *Rich Execution Environment*.
Waiver	An exemption to the rules defined by an Authority, provided to a given Service Provider for a very specific usage of the application in a specific context and for a limited duration.

1.5 Abbreviations and Notations

Selected abbreviations and notations used in this document are included in Table 1-3. Additional abbreviations and notations are defined in [GP CS].

Table 1-3: Abbreviations and Notations

Abbreviation / Notation	Meaning
DLOA	Digital Letter of Approval
DMSR	Device and Mobile Subscription Registrar
LOA	Letter of Approval
OID	Object Identifier as defined in “ISO/IEC 9834 series” and “ITU X660”
REE	Rich Execution Environment
SE	Secure Element
SEI	Secure Element Issuer
SEI TSM	Secure Element Issuer TSM
SP	Service Provider
SP TSM	Service Provider TSM
SPDM	System Protocol Discovery Mechanism
TEE	Trusted Execution Environment
TSM	Trusted Service Manager

1.6 Revision History

Table 1-4: Revision History

Date	Version	Description
Nov 2015	1.0	Public Release

2 Scope of the Specification

The objective of the specification is to provide a standardized infrastructure to retrieve the information related to certification of a device (or device components) or a secure component needed to deploy an application.

The specification considers:

- Letters of Approval obtained by a Platform, called Platform_DLOA
- Letters of Approval obtained by an Application, called Application_DLOA

2.1 Letters of Approval at the Platform Level

Authorities issue Letters of Approval for platforms. A platform can be an SE platform, a TEE platform, or an REE platform.

Some authorities are also issuing Letters of Approval for some other components of a device, like the certification of NFC communication interface component. In this document, Letters of Approval related to a component can be managed in the same way as Letters of Approval at the Platform level.

A Letter of Approval related to a platform shall include a reference to the specific platform. See section 4.4.1 for details.

2.2 Letters of Approval at the Application Level

Authorities, which can be either a certification, approval, or validation scheme, issue Letters of Approval for applications on top of a platform.

A Letter of Approval related to an application shall include a reference to the specific application and a reference to the targeted platform. See section 4.4.2 for details.

Prior to authorizing the deployment of a service on a given platform, the validity of obtained Letters of Approval corresponding to the applications on this platform may have to be verified. Verification could be done during the eligibility process performed by the TSM.

2.2.1 Letters of Approval Related to SE Applications

For SE applications, GlobalPlatform has developed a Card Composition Model (described in GlobalPlatform Card Composition Model [GP Comp Mdl]). It outlines a methodology that streamlines the security evaluation of a mobile composite product by specifying how EMVCo and Common Criteria certificates can be re-used for chips and SE platforms that have previously been certified.

[GP Comp Mdl] defines two types of SE applications:

Sensitive Application	• Requires formal security certification by a certification scheme such as Common Criteria or EMVCo.
Basic Application	• Does not have a negative impact on the certification status of the product and therefore may reside on a Secure Element along with one or more Sensitive Applications. • Shall be verified against a set of basic security rules defined in GlobalPlatform Card Composition Model Security Guidelines for Basic Applications [GP Guidelines Basic Applications]. • Shall also be verified against a set of rules that are specific to the certified platform onto which the application will be loaded. The specific set of rules might be empty on a particular certified platform.

The validity of obtained certificates must be verified prior to authorizing the deployment of a service using sensitive applications on a given SE; for example, for payment applications. Verification could be done during the eligibility process performed by the SP TSM. This process may also be used for basic applications, such as loyalty applications.

This specification defines the process used by a SP TSM to check the validity of the certificates of an application to be deployed on a specific product. For this process, the Service Provider needs a Letter of Approval generated by an Authority, which can be either a certification, evaluation, approval, qualification or validation scheme.

The Letter of Approval shall include a reference to the specific application and a reference to the targeted platform (i.e. Product); for details, please refer to the GlobalPlatform Card Overview of Complete Life Cycle for GlobalPlatform SE Products [GP Life Cycle].

As all security certificates and approvals concern an application on a specific platform:

- For sensitive applications, a product certificate is issued for a specific combination of an application and a platform (for example, for the same sensitive application SensApp_v1 we have two certificates: one for the first product SensApp_v1 on Platform1 and a second on a second product SensApp_v1 on Platform2).
- For basic applications, unfortunately, currently the platform may have requirements in addition to those listed in [GP Guidelines Basic Applications] that should be checked before loading it on the targeted product.

2.2.2 Letters of Approval Related to Trusted Applications

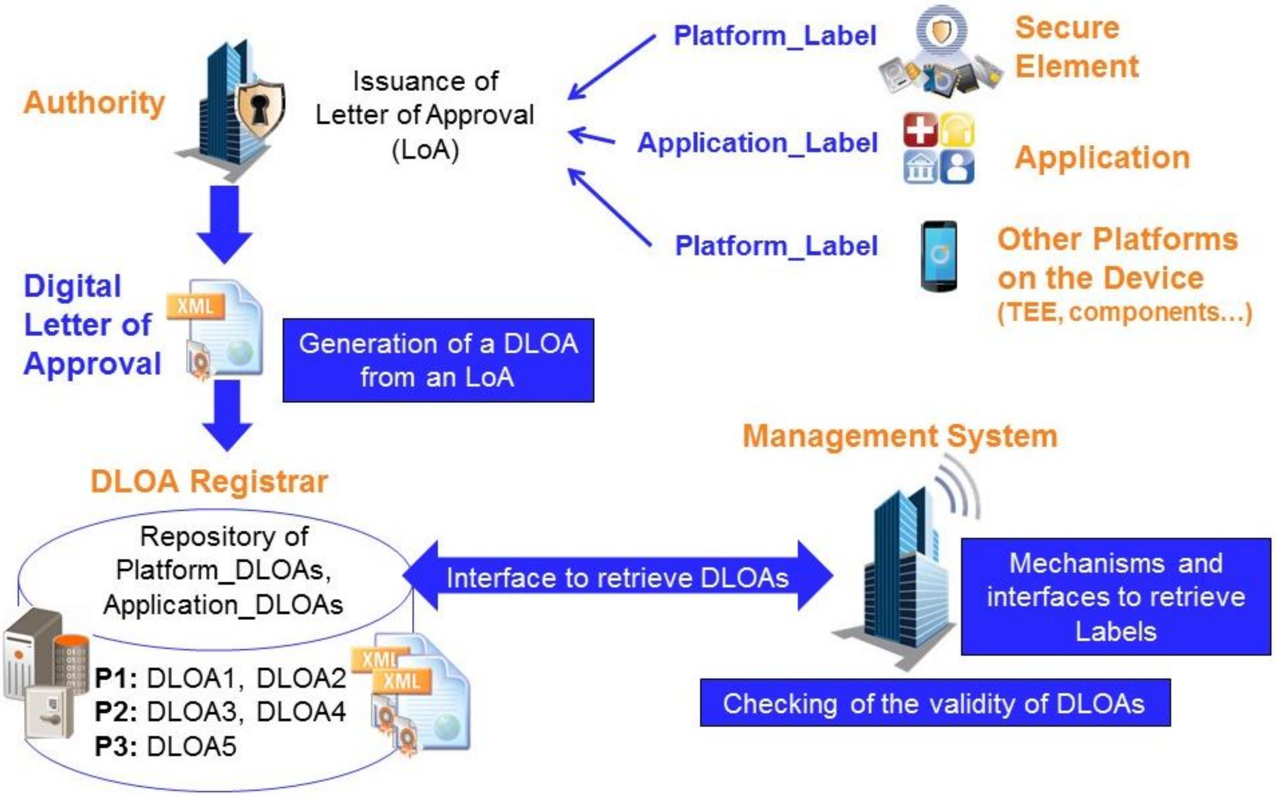
For Trusted Applications, it is not expected to require a specific process for verification before loading but some Authorities may generate Letters of Approval related to a specific platform.

2.3 List of All Items Covered by this Specification

The global solution proposed by GlobalPlatform is intended to address the following items:

1. Definition of a unique format for **Digital Letter of Approval (DLOA):**
    - A DLOA is a digital representation of a Letter of Approval issued by an Authority (i.e. a certification, evaluation, approval, qualification, or validation schemes). See Chapter 4.
2. Definition of common product naming and identification used in the DLOAs and during the whole product life with:
    - A Label identifying a Platform: the **Platform_Label**. See section 3.1.
    - A Label identifying an Application: the **Application_Label**. See section 3.2.
    - A Label identifying an Authority: the **Authority_Label**. See section 3.3.
3. Clarification of the process to issue DLOAs and manage the life cycle of DLOAs. See sections 4.2 and 4.3.
4. Definition of a standard interface to have access to DLOAs through a new entity called a **DLOA Registrar**:
    - A DLOA Registrar stores and provides an interface to enable a Management System to retrieve the Digital Letters of Approval (DLOAs) issued for a given Platform_Label, Application_Label. See section 5.2.
5. Example of usage of the DLOAs with the clarification of their usage by the SP TSMs during the global eligibility process when deploying an NFC service; that is, a given application on a given SE platform installed in a given device.
    - During the eligibility process, the SP TSM may check that the targeted SE platform is eligible to deploy the SE application requested by the Service Provider. See section 6.1.
    - During the eligibility process, the SP TSM may check that the device, into which the targeted SE is installed, is eligible to deploy the application requested by the Service Provider. See Chapter 6.
6. Definition of mechanisms provided by the SEI TSM or DMSR to get the Platform_Label of the different platforms available in a targeted device. See sections 6.2 and 6.3.
7. Definition of a command and a data structure to retrieve the Platform_Label from a Secure Element itself. See section 7.2.
8. Definition of a command and a data structure to retrieve the Platform_Label from a TEE itself. See Chapter 8.

Figure 2-1: Global Overview of the DLOA Solution


Authority
Issuance of Letter of Approval (LoA)
Platform_Label
Secure Element
Application_Label
Application
Platform_Label
Other Platforms on the Device (TEE, components…)
Digital Letter of Approval
XML
Generation of a DLOA from an LoA
DLOA Registrar
Repository of Platform_DLOAs, Application_DLOAs
P1: DLOA1, DLOA2
P2: DLOA3, DLOA4
P3: DLOA5
XML
XML
Interface to retrieve DLOAs
Management System
Mechanisms and interfaces to retrieve Labels
Checking of the validity of DLOAs

3 Labels Used for Naming and Identification

The following section describes the different labels that are used to identify the object that has been evaluated:

- The Platform_Label,
- The Application_Label,

An Authority_Label is also defined to identify the Authority that issues Letters of Approval.

3.1 The Platform_Label

The Platform_Label SHALL uniquely identify, without any ambiguity, the platform that has been evaluated.

The Platform_Label SHALL be assigned by the platform manufacturer that originated the evaluation request.

The value of the Platform_Label SHALL be a string assigned by the platform manufacturer using the following format:

"<OID of the platform manufacturer>/<Unique Identifier of the platform>"

The representation of the OID part SHALL follow the numerical dot notation of an OID, as specified in section 2.15 "Object identifier" of [RFC 1778], meaning integers are separated with dots (e.g.: "1.2", "3.4.5"). As an example, the GlobalPlatform OID will be represented here as `"1.2.840.114283"`.

The OID part and the Identifier part SHALL be separated by the character '/' without any space before or after.

The Identifier part SHALL include displayable characters (alphabetic, numerical, and special) or space corresponding to the range '0x20' to '0x7E' of the ASCII encoding defined in [8859-1], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

The length of the string of the Platform_Label, i.e. corresponding to "<OID>/<Unique Identifier>", SHALL be less than 120 characters.

The platform manufacturer SHALL guarantee the uniqueness of a Platform_Label, among the different platforms but also between two versions of the same platform.

The platform manufacturer SHALL provide the Platform_Label in the evaluation request.

The Platform_Label SHALL be mentioned in the formal Letter of Approval generated by the Authority after evaluation of the platform itself. The Platform_Label SHALL also be mentioned in the formal Letter of Approval generated by an Authority, after evaluation of an application on this platform.

The platform manufacturer SHALL guarantee that the Platform_Label stored in a given platform is compliant with the object that has been submitted to the evaluation process using this Platform_Label.

The Platform_Label stored in a given platform SHALL be updated if there is any change in the code of this platform.

A platform SHALL provide mechanisms to retrieve its Platform_Label. When the platform is an SE platform, such mechanisms are defined in Chapter 7. When the platform is a TEE platform such mechanisms are defined in Chapter 8.

A device SHALL provide mechanisms to retrieve the Platform_Label of all the platforms or device components available on the device.

3.2 The Application_Label

The Application_Label SHALL uniquely identify, without any ambiguity, the application.

The Application_Label SHALL be assigned by the Application Developer or the Application Issuer.

The value of the Application_Label SHALL be a string assigned by the Application Developer or the Application Issuer using the following format:

“<OID of the Application Developer>/<Unique Identifier of the application>”

Or, “<OID of the Application Issuer>/<Unique Identifier of the application>”

The representation of the OID part SHALL follow the numerical dot notation of an OID, as specified in section 2.15 "Object identifier" of [RFC 1778], meaning integers are separated with dots (e.g.: "1.2", "3.4.5"). As an example, GlobalPlatform OID will be represented here as `"1.2.840.114283"`.

The OID part and the Identifier part SHALL be separated by the character ‘/’ without any space before or after.

The Identifier part SHALL include displayable characters (alphabetic, numerical, and special) or space corresponding to the range '0x20' to '0x7E' of the ASCII encoding defined in [8859-1], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

The length of the string of the Application_Label, i.e. corresponding to "<OID>/<Identifier>", SHALL be less than 120 characters. The Application Developer or the Application Issuer SHALL guarantee the uniqueness of an Application_Label, among the different applications but also between two versions of the same application.

The Application Issuer SHALL provide the Application_Label in the evaluation request. It SHALL also provide the Platform_Label on top of which the application is evaluated.

The Application_Label and the Platform_Label SHALL be mentioned in the formal Letter of Approval generated by the Authority after evaluation of this application on top of the given platform.

For each application, it SHALL be possible to retrieve the corresponding Application_Label from the description of the service stored in the Management System.

The Application Issuer SHALL guarantee that the application provided with a given Application_Label is compliant with the object that has been submitted to the evaluation process using this Application_Label.

The Application_Label SHALL be updated if there is any change in the application.

3.3 The Authority_Label

The Authority_Label SHALL identify an Authority that certifies, evaluates, approves, qualifies, or validates platforms or applications, and that issues Letters of Approval.

The value of the Authority_Label SHALL be a string assigned by the Authority using one of the following formats:

“<OID of the Authority>”

Or, “<OID of the Authority>/<Authority_Subsection>”

The representation of the OID part SHALL follow the numerical dot notation of an OID, as specified in section 2.15 "Object identifier" of [RFC 1778], meaning integers are separated with dots (e.g.: "1.2", "3.4.5"). As an example, GlobalPlatform OID will be represented here as `"1.2.840.114283"`.

When the Authority contains an Authority_Subsection part, the OID part and the Authority_Subsection part SHALL be separated by the character ‘/’ without any space before or after.

When present, the Authority_Subsection part SHALL include displayable characters (alphabetic, numerical, and special) or space corresponding to the range '0x20' to '0x7E' of the ASCII encoding defined in [8859-1], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

The length of the string of the Authority_Label, i.e. corresponding to "<OID>" or "<OID>/<Authority_Subsection>", SHALL be less than 120 characters.

4 Digital Letter of Approval (DLOA)

The different certification, evaluation, approval, qualification, or validation schemes (called Authority in this document) currently issue Letters of Approval on paper or in different digital formats with a proprietary content. Such Letters of Approval cannot always be processed automatically by a Management System. This document serves the need to define a common digital format for a Letter of Approval, called a Digital Letter of Approval (DLOA), and to define its life cycle.

4.1 Format of a DLOA

The Digital Letter of Approval (DLOA) is an XML file containing the minimum fields required to:

- Identify the platform – the combination of the application and the platform – this DLOA corresponds to
- Identify the Authority that issued the corresponding Letter of Approval
- Provide the expiration date of the corresponding Letter of Approval
- Identify the Letter of Approval from which this DLOA has been generated (i.e. include the identifier of the Letter of Approval issued by the Authority)
- Ensure authenticity and integrity of the DLOA thanks to a digital signature computed by the Authority
- Provide additional information such as the date of issuance of the corresponding Letter of Approval or a URL where the original Letter of Approval can be retrieved

This generic DLOA format is then varied, creating different types of Letter of Approval:

- A DLOA corresponding to a Letter of Approval at the Platform level SHALL include the platform identifier, called the Platform_Label. Such a DLOA will be called a Platform_DLOA in the remainder of this document.
- A DLOA corresponding to a Letter of Approval at the Application level SHALL include the application identifier, called the Application_Label, but also the platform identifier, the Platform_Label, on which the application has been evaluated. Such a DLOA will be called an Application_DLOA in the remainder of this document.

The detailed format of the Platform_DLOA, Application_DLOA are given in section 4.4.

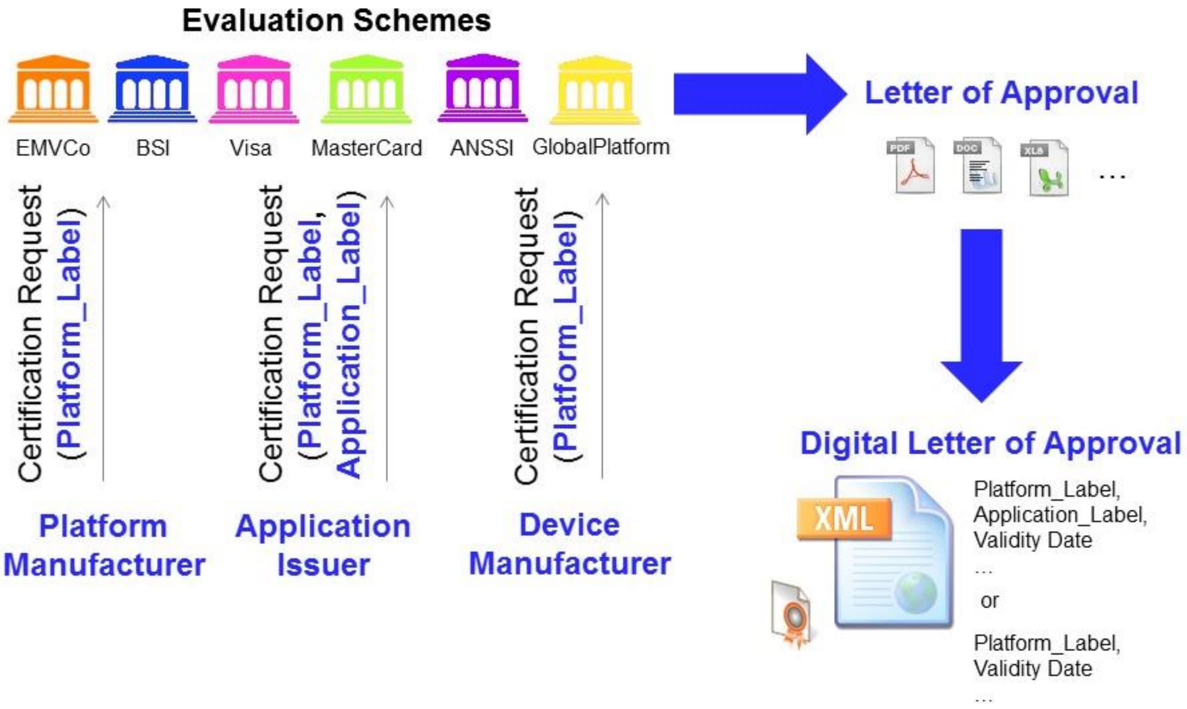
4.2 Generation of a DLOA

The Digital Letter of Approval (DLOA) SHALL be generated from a Letter of Approval issued by an Authority.

The DLOA and the signature on this DLOA SHALL be generated by the Authority that generated the corresponding Letter of Approval.

Note: If for some technical reasons, an Authority cannot generate Digital Letters of Approval, it MAY delegate the generation to another entity, but the Authority issuing the Letter of Approval SHALL at least generate the signature on the DLOA.

Figure 4-1: High Level Overview of DLOA Generation


Evaluation Schemes
EMVCo
BSI
Visa
MasterCard
ANSSI
GlobalPlatform
Certification Request (Platform_Label)
Certification Request (Platform_Label, Application_Label)
Certification Request (Platform_Label)
Platform Manufacturer
Application Issuer
Device Manufacturer
Letter of Approval
PDF
DOC
XLS
…
Digital Letter of Approval
XML
Platform_Label, Application_Label, Validity Date
…
or
Platform_Label, Validity Date
…

4.3 Life Cycle of a DLOA

This section describes the life cycle of a Digital Letter of Approval according to the different life cycle states of the corresponding Letter of Approval:

- Issuance of the Letter of Approval
- Expiration of the Letter of Approval
- Revocation of the Letter of Approval
- Renewal of the Letter of Approval
- Waiver management

Note: If an Authority implements a DLOA Registrar for all the LOAs it issues, the life cycle of the DLOAs can be significantly simplified.

If a DLOA Registrar is not managed by an Authority, the DLOA Registrar SHALL be responsible for maintaining an accurate list of DLOAs in connection with the Authority.

4.3.1 Issuance of the DLOA

When a platform successfully completes its certification, or when an application on a specific platform successfully completes its verification (for a basic application) or certification (for a sensitive application), the corresponding Letter of Approval SHALL be issued by the Authority.

Then, the corresponding Digital Letter of Approval (DLOA) SHALL be generated as defined in section 4.2.

If the Authority implements a DLOA Registrar, the Authority SHALL add the new DLOA to its DLOA Registrar.

If the DLOA Registrar is not implemented by the Authority, the entity that requested the Letter of Approval from the Authority SHALL ensure that the corresponding DLOA is available in the concerned DLOA Registrar(s).

4.3.2 Expiration of the DLOA

When a Letter of Approval (either at the Platform level or at the Application level) has expired, the corresponding entry in the DLOA Registrar SHOULD be made unavailable.

As the validity date inside the DLOA will have expired, there is no issue if such a DLOA is still available in a DLOA Registrar. The deletion of expired DLOAs from the DLOA Registrar SHOULD be managed periodically.

4.3.3 Revocation of the DLOA

When a Letter of Approval (either at the Platform level or at the Application level) is revoked, the corresponding entry in the DLOA Registrar(s) SHALL be deleted.

If the Authority implements a DLOA Registrar, the Authority SHALL remove the revoked DLOA from its DLOA Registrar.

If the DLOA Registrar is not implemented by the Authority, the entity that requested the Letter of Approval from the Authority or the responsibility of the Authority itself SHALL ensure that the corresponding revoked DLOA is made unavailable in the concerned DLOA Registrar(s).

4.3.4 Renewal of the DLOA

When a Letter of Approval (either at the Platform level, the Application level) is renewed, the Letter of Approval identifier is the same as for the initial Letter of Approval but the expiration date is updated. The Authority SHALL issue a new DLOA with an extended expiration date, as described in section 4.2.

If the Authority implements a DLOA Registrar, the Authority SHALL replace the former DLOA by this new DLOA in its DLOA Registrar.

If the DLOA Registrar is not implemented by the Authority, the entity that requested the renewal of the Letter of Approval from the Authority, SHALL ensure that the corresponding DLOA is replaced in the concerned DLOA Registrar(s).

4.3.5 Management of a Waiver

A waiver is an exemption to rules defined by an Authority, provided to a given Service Provider for a very specific usage of the application by the Service Provider in a specific context and for a limited duration.

A waiver is requested by the SP to the Authority. If the SP obtains a waiver, the SP will have to ask the SP TSM to consider this waiver and allow service deployment despite the corresponding DLOA(s) having expired or not being available.

The mechanism used to manage waivers is to be handled according to specific business agreements between the SP and the SP TSM.

4.4 Detailed Description of the Format of a DLOA

One XML format is defined for Digital Letters of Approval, with the following options:

- Digital Letter of Approval at the Platform level (called Platform_DLOA)
- Digital Letter of Approval at the Application level (called Application_DLOA)

4.4.1 Format of a Platform_DLOA

The following fields are included in the XML format of the DLOA at the Platform level:

Table 4-1: Format of a DLOA at Platform Level

Field	Description	MOC
`Authority_Label`	Label identifying the Authority issuing the Letter of Approval	M
`LOA_Identifier`	Identifier of the Letter of Approval assigned by the Authority	M
`LOA_Scope`	Scope of the Letter of Approval	M
`Platform_Label`	The Platform_Label of the platform that has been evaluated	M
`Issuance_Date`	Date of issuance of the corresponding LOA by the Authority	M
`Expiration_Date`	Date of expiration of the LOA according to the policy of the Authority	O
`LOA_URL`	URL that can be used to retrieve the original Letter of Approval	M
`Signature`	Electronic signature on the DLOA ensuring integrity and authenticity of the DLOA	M

Some additional data might be defined in the original Letter of Approval issued by the Authority. This data has not been considered as relevant for the use case defined in this document. However, the URL defined in the DLOA should allow the retrieval of the original Letter of Approval or of a more detailed and specific digital representation of the original Letter of Approval as issued by the Authority.

4.4.2 Format of an Application_DLOA

As mentioned in section 2.2, the Digital Letter of Approval at the Application level is always valid on a specific Platform. Thus, a DLOA at Application level SHALL include both an Application_Label and a Platform_Label.

The following items are included in the XML format of the DLOA at the Application level.

Table 4-2: Format of a DLOA at Application Level

Field	Description	MOC
`Authority_Label`	Label identifying the Authority issuing the Letter of Approval	M
`LOA_Identifier`	Identifier of the Letter of Approval assigned by the Authority	M
`LOA_Scope`	Scope of the Letter of Approval	M
`Application_Label`	The Application_Label of the application that has been evaluated	M
`Platform_Label`	The Platform_Label of the platform on which the application has been evaluated	M
`Issuance_Date`	Date of issuance of the corresponding LOA by the Authority	M
`Expiration_Date`	Date of expiration of the LOA according to the policy of the Authority	O
`Application_Limit_Date`	Date after which the Application cannot be used, according to the policy of the Authority	O
`LOA_URL`	URL that can be used to retrieve the original Letter of Approval	M
`Signature`	Electronic signature on the DLOA ensuring integrity and authenticity of the DLOA	M

Some additional data might be defined in the original Letter of Approval issued by the Authority. This data has not been considered as relevant for the use case defined in this document. However, the URL defined in the DLOA should allow the retrieval of the original Letter of Approval or of a more detailed and specific digital representation of the original Letter of Approval as issued by the Authority.

4.4.3 Format of the Various Fields of a DLOA

4.4.3.1 Authority Label

The field `Authority_Label` SHALL correspond to the Authority_Label of the Authority that has evaluated the Platform or the Application and that has issued the Letter of Approval.

The format of the `Authority_Label` is defined in section 3.3.

4.4.3.2 LOA Identifier

The field `LOA_Identifier` SHALL correspond to the Identifier of the Letter of Approval assigned by the Authority. The format of the `LOA_Identifier` depends on the Authority.

As an example, for the DLOA corresponding to the certification of a Platform by EMVCo, the `LOA_Identifier` contains the Platform Certificate Number (PCN) assigned by EMVCo.

The `LOA_Identifier` SHALL be a string, which includes displayable characters (alphabetic, numerical, and special) or space in the range from '0x20' to '0x7E' of the ASCII encoding as defined in ISO/IEC 8859-1 [8859-1], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

4.4.3.3 LOA Scope

The field `LOA_Scope` SHALL correspond to scope of the evaluation covered by the LOA.

The `LOA_Scope` SHALL be a string, which includes displayable characters (alphabetic, numerical, and special) or space in the range from '0x20' to '0x7E' of the ASCII encoding as defined in ISO/IEC 8859-1 [8859-1], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

4.4.3.4 Platform Label

The field `Platform_Label` SHALL contain a string corresponding to:

- The Platform_Label of the platform that has been evaluated, when defined in a Platform_DLOA
- The Platform_Label of the platform on which the application has been evaluated, when defined in an Application_DLOA

The format of the `Platform_Label` is defined in section 3.1.

4.4.3.5 Application Label

The field `Application_Label` SHALL contain a string corresponding to the Application_Label of the application that has been evaluated.

The field `Application_Label` is only applicable for an Application_DLOA.

The format of the `Application_Label` is defined in section 3.2.

4.4.3.6 Issuance Date

The field `Issuance_Date` SHALL correspond to the issuance date of the Letter of Approval by the Authority.

The format of the `Issuance_Date` SHALL be a string "YYYY-MM-DD" where:

- YYYY indicates the year
- MM indicates the month
- DD indicates the day

4.4.3.7 Expiration Date

The field `Expiration_Date` SHALL correspond to the expiration date of the Letter of Approval according to the policy of the Authority.

The format of the `Expiration_Date` SHALL be a string "YYYY-MM-DD" where:

- YYYY indicates the year
- MM indicates the month
- DD indicates the day

The presence of this field is optional. If the `Expiration_Date` field is not present or is present but with an empty value, the corresponding DLOA SHALL be considered as always valid.

4.4.3.8 Limit Date of Application Usage

The field `Application_Limit_Date` SHALL correspond to the limit date of usage of the application that might be defined according to the policy of the Authority. If such limit date is defined by the Authority, after its expiration, the application SHOULD not be used anymore by the end user.

The field `Application_Limit_Date` SHALL be only applicable for an Application_DLOA.

The format of the `Application_Limit_Date` SHALL be a string "YYYY-MM-DD" where:

- YYYY indicates the year
- MM indicates the month
- DD indicates the day

The presence of this field in an Application_DLOA is optional. If the `Application_Limit_Date` field is not present or is present but with an empty value, the corresponding application shall be considered as always usable.

4.4.3.9 URL of LOA

The field `LOA_URL` SHALL contain the URL where the original Letter of Approval as issued by the Authority can be retrieved through a web browser.

The format of the `LOA_URL` SHALL be compliant with [RFC 3986], excluding the double quote (") '0x42' and the quote (') '0x47' characters.

The usage of such URL is out of scope of this specification.

As an example, for LOAs issued by GlobalPlatform, the field `LOA_URL` will be set to:

```
http://globalplatform.org/complianceproducts.asp
```

4.4.3.10 Signature on the DLOA

The field `Signature` SHALL contain an electronic signature on the DLOA ensuring integrity and authenticity of the DLOA.

This digital signature SHALL be generated by the Authority.

This digital signature MAY be verified (e.g. by the Management System) to check the integrity and the authenticity of the DLOA.

The format of the field `Signature` SHALL be compliant with W3C, XML Signature Syntax and Processing Version 1.1 ([XML SIG]).

The `<ds:SignatureType>` element SHALL include at least the following elements:

- `<ds:SignedInfo>` element, describing the signature algorithm used and which sections of the message are part of the signature. This specification mandates that the signature SHALL include the element `<dloa:SignedElements>`.
- `<ds:KeyInfo>` element, identifying the certificate of the Authority used for the signature.
- `<ds:SignatureValue>` element, providing the signature value of the `<ds:SignedInfo>` element.

As defined in [XML SIG], the following methods SHALL be used:

- Canonicalization method: Exclusive canonicalization as specified in W3C, Exclusive XML Canonicalization, Version 1.0 [XML C14N], for which identifier is "http://www.w3.org/2001/10/xml-exc-c14n#"
- Message Digest method: The following message digest methods defined in [XML SIG] are applicable:
    - SHA-256, which identifier is "http://www.w3.org/2001/04/xmlenc#sha256"
    - SHA-384, which identifier is "http://www.w3.org/2000/09/xmldsig#sha384"
    - SHA-512, which identifier is "http://www.w3.org/2001/04/xmlenc#sha512"
- Signature algorithm: The following signature algorithms defined in [XML SIG] are applicable:
    - RSA (PKCS#1 v1.5), which identifiers are "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256", "http://www.w3.org/2001/04/xmldsig-more#rsa-sha384" or "http://www.w3.org/2001/04/xmldsig-more#rsa-sha512" with an RSA key length of at least 3072 bits
    - ECDSA, which identifiers are "http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256", "http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384" or "http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512" with an ECC key length of at least 256 bits

The certificate of the Authority SHALL be provided using the XML structure rawX509Certificate defined in [XML SIG] to provide a binary (ASN.1 DER) X.509 Certificate, which identifier is "http://www.w3.org/2000/09/xmldsig#rawX509Certificate".

4.5 XML Schemas

4.5.1 Namespaces

In the context of this specification, a new namespace is defined:

- dloa: http://namespaces.globalplatform.org/systems-dloa/1.0

The XML schema defined in this specification refers to other namespaces:

- ds: W3C XML-Signature Syntax and Processing, W3C Recommendation
- gpm: GlobalPlatform Messaging Specification - XSD - WSDL, v 2.2.0.
- xsd: Extensible Markup Language (XML) 1.0, W3C Recommendation

4.6 Data Types and String Encoding

The following data types are applicable:

Table 4-3: Applicable Data Types

Data	Description
Integer	Integer values SHALL be in the range of a signed 64-bit Integer: from -(2^{63}) to 2^{63}-1.
Internationalizable strings	When an element is a language dependent string, it SHOULD have an attribute `xml:lang="xx"` where xx is the language identifier as specified in [RFC 4646]. If no `xml:lang` attribute is present, implementations MUST assume the language to be English as defined by setting the attribute value to "en" (i.e. `xml:lang="en"`).
Date	The date SHALL be expressed as a dateTime in "canonical representation" (refer to [XML Data Types]). Implementations SHOULD NOT rely on time resolution finer than milliseconds and MUST NOT generate time instants that specify leap seconds.

Figure 4-2: XML Format – High Level Description of the Elements

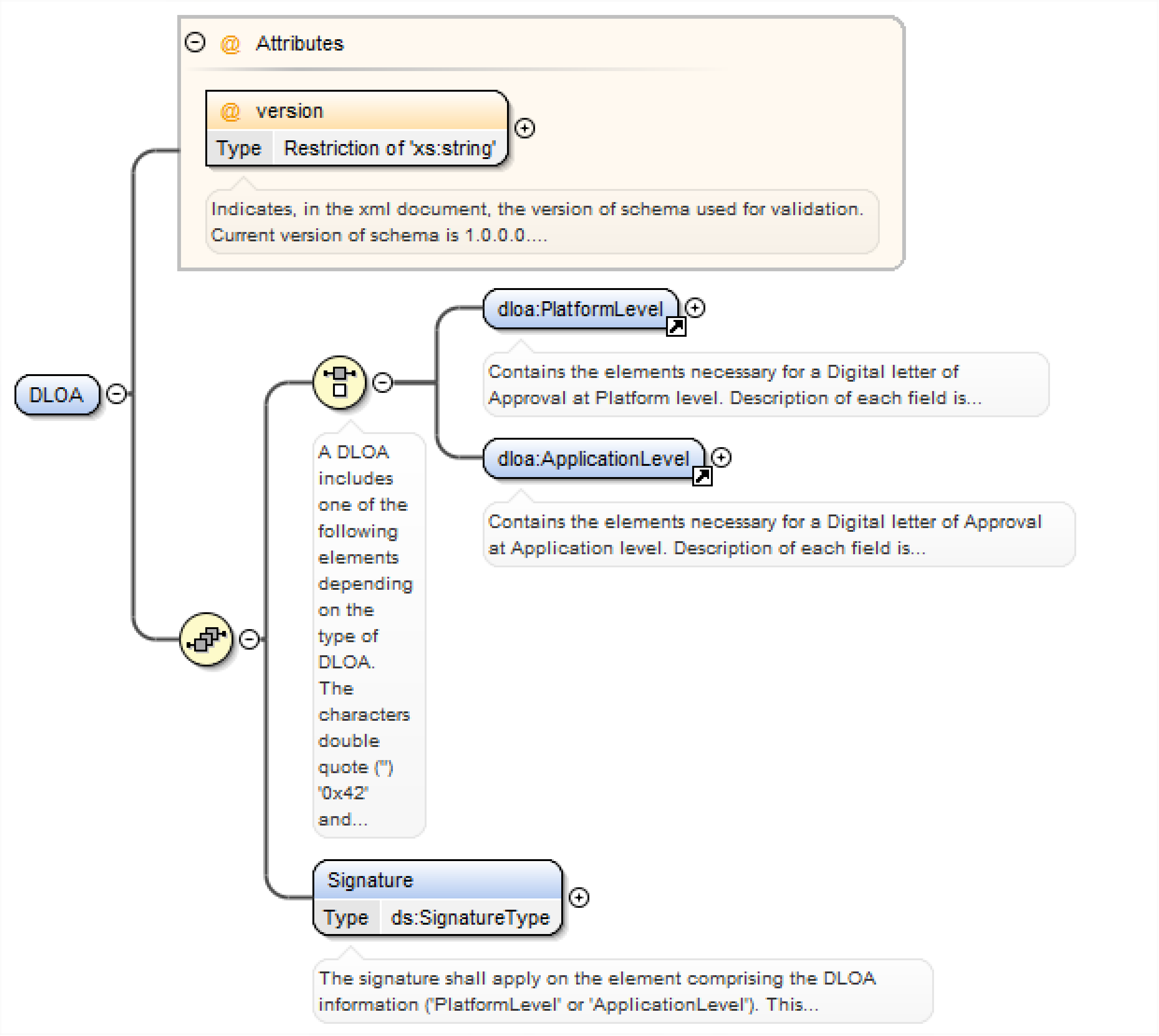

Attributes
version
Type Restriction of 'xs:string'
Indicates, in the xml document, the version of schema used for validation. Current version of schema is 1.0.0.0....
DLOA
dloa:PlatformLevel
Contains the elements necessary for a Digital letter of Approval at Platform level. Description of each field is...
dloa:ApplicationLevel
Contains the elements necessary for a Digital letter of Approval at Application level. Description of each field is...
A DLOA includes one of the following elements depending on the type of DLOA. The characters double quote (") '0x42' and...
Signature
Type ds:SignatureType
The signature shall apply on the element comprising the DLOA information ('PlatformLevel' or 'ApplicationLevel'). This...

Figure 4-3: XML Format – Details of `<dloa:PlatformLevel>` Element

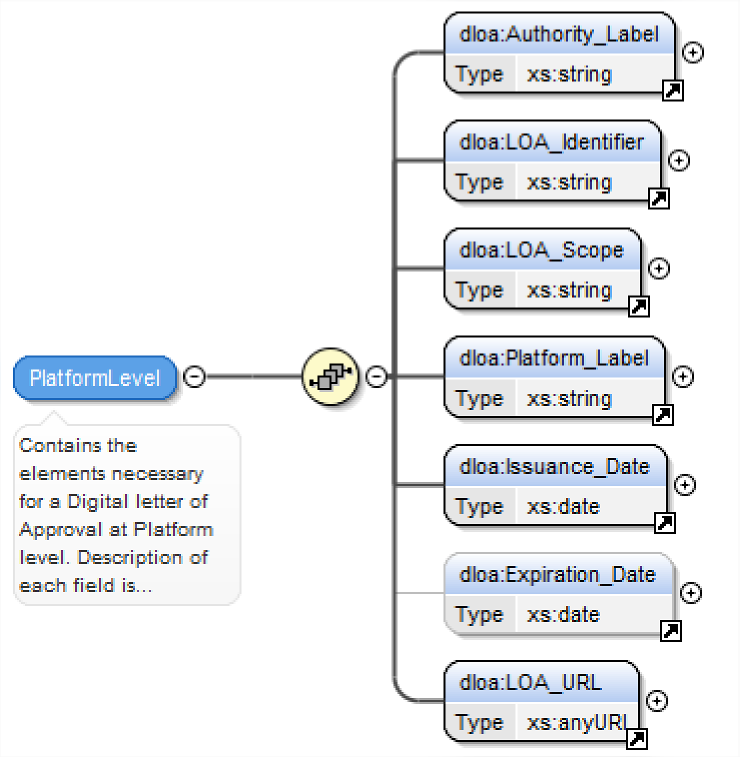
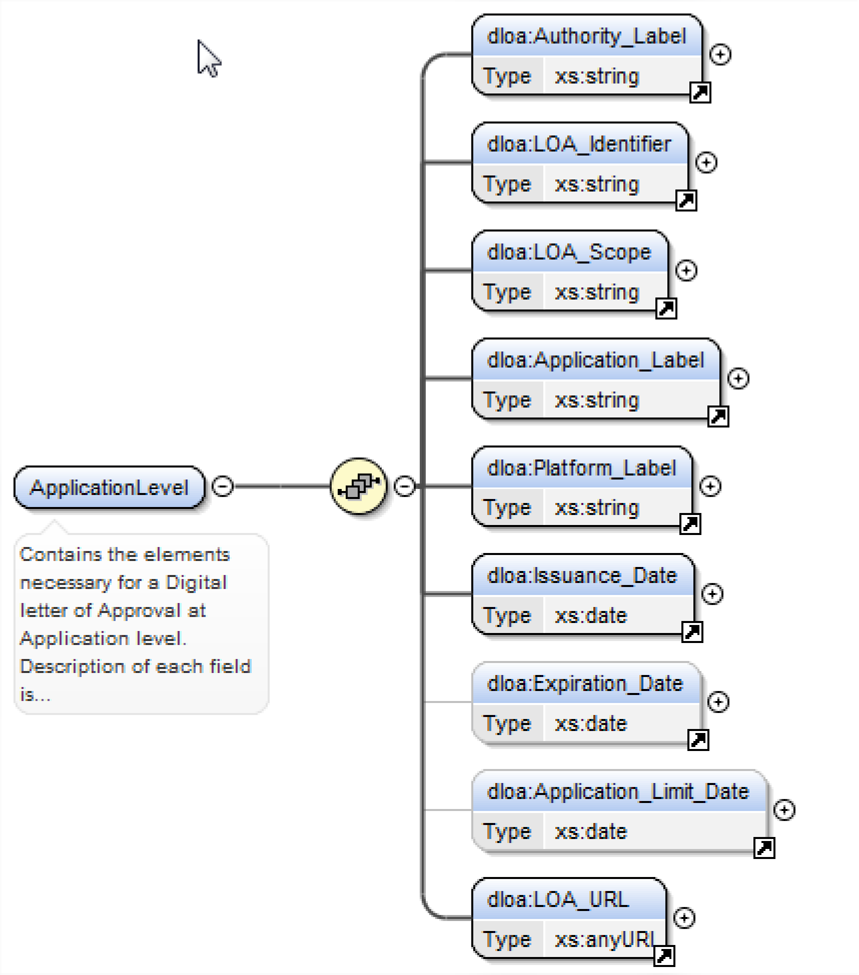


Figure 4-4: XML Format – Details of `<dloa:ApplicationLevel>` Element



5 The DLOA Registrar

5.1 Role of the DLOA Registrar

A DLOA Registrar is responsible for:

- Managing a list of Digital Letters of Approval (DLOAs). The DLOAs can be Platform_DLOA or Application_DLOA.

  Note: If a DLOA Registrar is not managed by an Authority, the DLOA Registrar is also responsible to maintain an accurate list of DLOAs in connection with the Authority.

- Implementing an interface for Management Systems (e.g. TSMs) to retrieve all the DLOA entries of a particular type known by the DLOA Registrar (for example for a full refresh of the SP TSM cache), either Platform_DLOAs or Application_DLOAs as defined in section 5.2.1.
- Implementing an interface for Management Systems (e.g. TSMs) to retrieve the DLOA entries for a given platform or application (for example for an individual DLOA retrieval by the SP TSM) as defined in section 5.2.2.

In addition the DLOA Registrar MAY act as an SPDM Provider Actor as defined in [GP SPDM]; see section 6.1.3.1 for details.

A DLOA Registrar MAY also implement mechanisms to define access rights to some DLOAs (as some DLOAs may not be public). To have access to these specific DLOAs, a Management System SHALL authenticate with the DLOA Registrar. Mechanisms for a Management System to authenticate are out of scope of this specification.

In any case, only active DLOAs SHALL be provided by the DLOA Registrar. Expired or revoked DLOAs SHALL not be returned.

5.2 Interfaces Provided by the DLOA Registrar

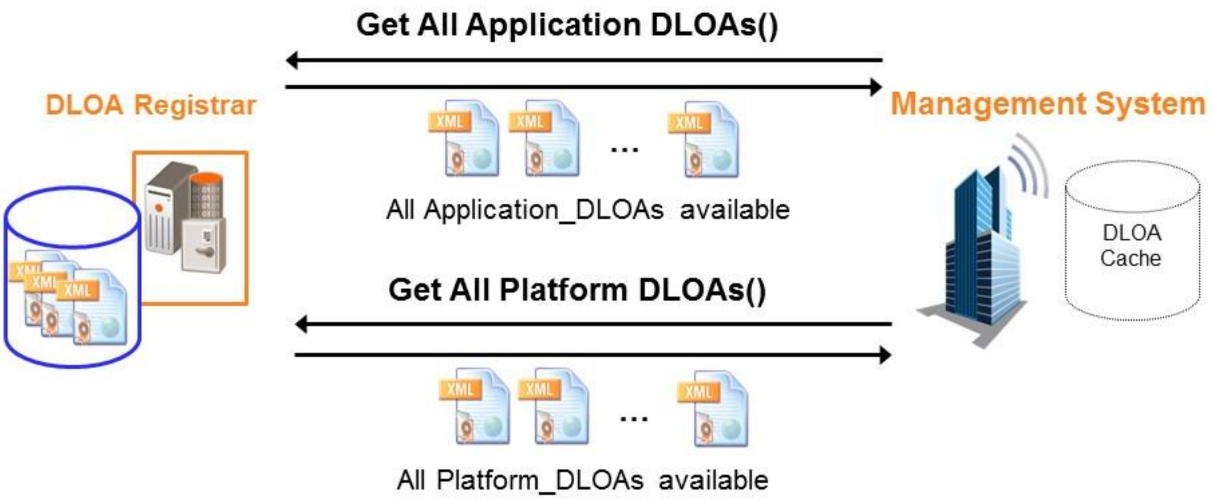
The DLOA Registrar needs to implement:

- An interface to retrieve all DLOA entries corresponding to a given DLOA type. This interface is called "Get All DLOAs" and is defined in section 5.2.1.
- An interface to retrieve the DLOA entries corresponding to a given platform or a given application. This interface is called "Get DLOA" and is defined in section 5.2.2.

5.2.1 Get All DLOAs

The DLOA Registrar SHALL implement an interface to retrieve all Platform_DLOA or Application_DLOA entries known by the DLOA Registrar and matching the access rights of the caller, if any.

Figure 5-1: Get All DLOAs Interface Provided by the DLOA Registrar


Get All Application DLOAs()
DLOA Registrar
Management System
XML
...
All Application_DLOAs available
DLOA Cache
Get All Platform DLOAs()
All Platform_DLOAs available

The DLOA Registrar SHALL provide the following REST endpoints for that purpose:

HTTP Command:

GET

Resource URL:

`<DLOA_Registrar_URL>/v1/dloas/platform`, for retrieving all the platform DLOAs

`<DLOA_Registrar_URL>/v1/dloas/application`, for retrieving all the application DLOAs

Where:

- The `<DLOA_Registrar_URL>` is either
  - Directly known by the SP TSM as part of (as detailed in section 6.1.3):
    - the description of the service,
    - the SE capability profile, or
    - the device capability profile.
  - Or might have been obtained previously using the System Protocol Discovery Mechanism described in the GlobalPlatform System Protocol Discovery Mechanism Specification [GP SPDM], as detailed in section 6.1.3 and Chapter 7. In that case, the DLOA Registrar SHALL act as an SPDM Provider Actor as defined in [GP SPDM].
- "v1" indicates version 1 of the DLOA Registrar interface (the version described in this specification).

Parameters:

None.

Output Format:

The HTTP response body SHALL contain a JSON flow that is compliant with the JSON format [RFC 4627] below:

- For Platform_DLOAs:

```
[
  {
    "Platform_Label": "<Platform label #A>",
    "DLOA": [
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #1>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #2>"
      }
    ]
  },
  {
    "Platform_Label": "<Platform label #B>",
    "DLOA": [
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #1>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #2>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #3>"
      }
    ]
  }
]
```

- For Application_DLOAs:

```
[
  {
    "Platform_Label": "<Platform label #A>",
    "Application_Label": "<Application label #A>",
    "DLOA": [
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #1>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #2>"
      }
    ]
  },
  {
    "Platform_Label": "<Platform label #B>",
    "Application_Label": "<Application label #B>",
    "DLOA": [
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #1>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #2>"
      },
      {
        "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #3>"
      }
    ]
  }
]
```

The global lists represent the list of platforms, applications that are known by the DLOA Registrar. If no platform or application is known by the DLOA Registrar, then this list SHALL be empty.

Each item of the list represents a single platform or application for which the DLOA is provided. Those items contain the following properties:

- For Platform_DLOA:
  - `Platform_Label` (string, mandatory): the label of the platform, in the format defined in section 3.1.
- For Application_DLOA:
  - `Platform_Label` (string, mandatory): the label of the platform, in the format defined in section 3.1.
  - `Application_Label` (string, mandatory): the label of the application, for the platform identified by the `Platform_Label` property, in the format defined in section 3.2.
- `DLOA` (list, mandatory): the list of DLOAs for the given platform or application. Only active DLOAs SHALL be returned: expired or revoked DLOAs shall not be returned.

  Each item of the list represents a single DLOA. Each item contains the following properties:
  - `DLOA_XML` (string, mandatory): the base64 encoded string representation of the DLOA XML file corresponding to the given platform or application. The XML format of the DLOA is defined in section 4.4.

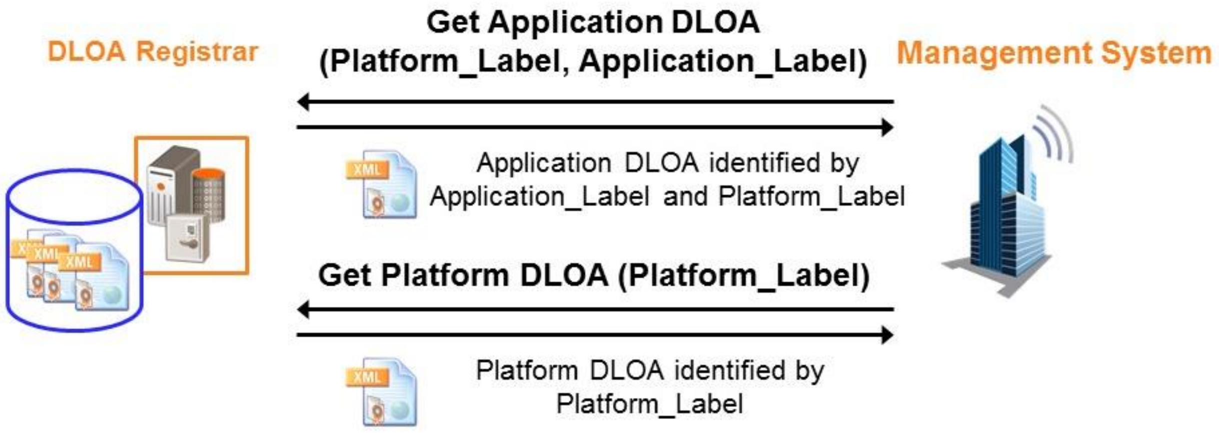
The corresponding JSON Schema representation of this format (valid for all types of DLOA) is described in section B.1.1 and some examples of Get All DLOAs requests and responses are provided in section B.2.1.

5.2.2 Get DLOA

The DLOA Registrar SHALL implement an interface to retrieve the DLOA entries known by the DLOA Registrar, matching the access rights of the caller, if any, and corresponding to:

- A given platform (identified by its Platform_Label), or
- A given application (identified by both its Application_Label and the Platform_Label of the platform onto which the application is to be deployed).

Figure 5-2: Get DLOA Interface Provided by the DLOA Registrar


DLOA Registrar
Get Application DLOA
(Platform_Label, Application_Label)
Management System
XML
Application DLOA identified by
Application_Label and Platform_Label
Get Platform DLOA (Platform_Label)
Platform DLOA identified by
Platform_Label

The DLOA Registrar SHALL provide the following REST endpoints for that purpose:

HTTP Command:

GET

Resource URL:

`<DLOA_Registrar_URL>/v1/dloa/platform`, for retrieving a platform DLOA

`<DLOA_Registrar_URL>/v1/dloa/application`, for retrieving an application DLOA

Where:

- The `<DLOA_Registrar_URL>` is either
  - Directly known by the SP TSM as part of (as detailed in section 6.1.3):
    - the description of the service,
    - the SE capability profile, or
    - the device capability profile.
  - Or might have been obtained previously using the System Protocol Discovery Mechanism described in [GP SPDM], as detailed in section 6.1.3 and Chapter 7. In that case, the DLOA Registrar SHALL act as an SPDM Provider Actor as defined in [GP SPDM].
- "v1" indicates version 1 of the DLOA Registrar interface (the version described in this specification).

Parameters:

- For Platform_DLOA:
    - `Platform_Label` (string, mandatory): The label of the platform, as a URL-encoded string representation of the platform label defined in section 3.1.
- For Application_DLOA:
    - `Platform_Label` (string, mandatory): The label of the platform, as a URL-encoded string representation of the platform label defined in section 3.1.
    - `Application_Label` (string, mandatory): The label of the application, for the platform identified by the `Platform_Label` parameter, as a URL encoded string representation of the application label defined in section 3.2.

Those parameters SHALL be provided as query parameters to the Get DLOA URL.

Output Format:

The HTTP response body SHALL contain a JSON flow that is compliant with the JSON format [RFC 4627] below:

```
[
  {
    "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #1>"
  },
  {
    "DLOA_XML": "<base64-encoded string representation of the DLOA XML file #2>"
  }
]
```

The global list represents the list of DLOAs for the given platform or application. Only active DLOAs SHALL be returned: expired or revoked DLOAs SHALL not be returned. If no active DLOA for the given platform or application is known by the DLOA Registrar, then this list SHALL be empty.

Each item of the list represents a single DLOA. Each item contains the following properties:

- `DLOA_XML` (string, mandatory): The base64-encoded string representation of the DLOA XML file corresponding to the given platform or application, or device. The XML format of the DLOA is defined in section 4.4.

The corresponding JSON Schema representation of this format is described in section B.1.2 and some examples of Get DLOA requests and responses are provided in section B.2.2.

5.2.3 Security Mechanisms

TLS security with server authentication SHALL be set up between the Management System and the DLOA Registrar to ensure the Management System that it is talking to the right DLOA Registrar.

The DLOA Registrar MAY require authentication of the Management System to restrict access to some DLOAs. In that case, TLS with mutual authentication SHALL be used.

Minimum TLS version 1.2 [RFC 5246] SHALL be used.

6 Example of Management of the DLOAs in the NFC SE Ecosystem

A Service Provider MAY request that, prior to deploying a NFC service on a given Secure Element, the SP TSM SHOULD check that the targeted SE implements a platform that has been recognized by the concerned Authority as a valid SE platform to host the SE application(s) of this service.

A Service Provider MAY also request that, prior to deploying a service on a Secure Element installed in a given device, the SP TSM SHOULD check that this device implementation (considered as a Platform in this specification and composed of device components also considered as Platforms in this specification) has been recognized by the concerned Authority as a valid device to interact with the Secure Element application(s) of this service.

These checks MAY be performed by the SP TSM during the global eligibility process of the service as defined in [GP SM].

DLOAs are provided by DLOA Registrars. The interface between the SP TSM and the DLOA Registrar to retrieve DLOA(s) is defined in section 5.2. In order to improve efficiency of the global eligibility process as defined in [GP SM] and to avoid sending frequent requests to the DLOA Registrar, the SP TSM MAY temporary store DLOA entries.

DLOAs required by a given SP TSM MAY be handled by different DLOA Registrars. The mechanisms to retrieve the URL of the DLOA Registrar(s) in charge of a given DLOA are defined in section 6.1.3.

6.1 Overall Process of the Eligibility Check by the SP TSM

This chapter refers to mechanisms defined in [GP SM].

During the `CheckGlobalEligibility` process of a given Service:

- The SP TSM MAY be required to check that the Application_DLOA corresponding to the card applications composing the service are still valid on the platform of the targeted Secure Element
- The SP TSM MAY be required to check that the Platform_DLOAs of the different platforms available on the device onto which the service is to be deployed is still valid

In order to achieve these checks, the SP TSM need to:

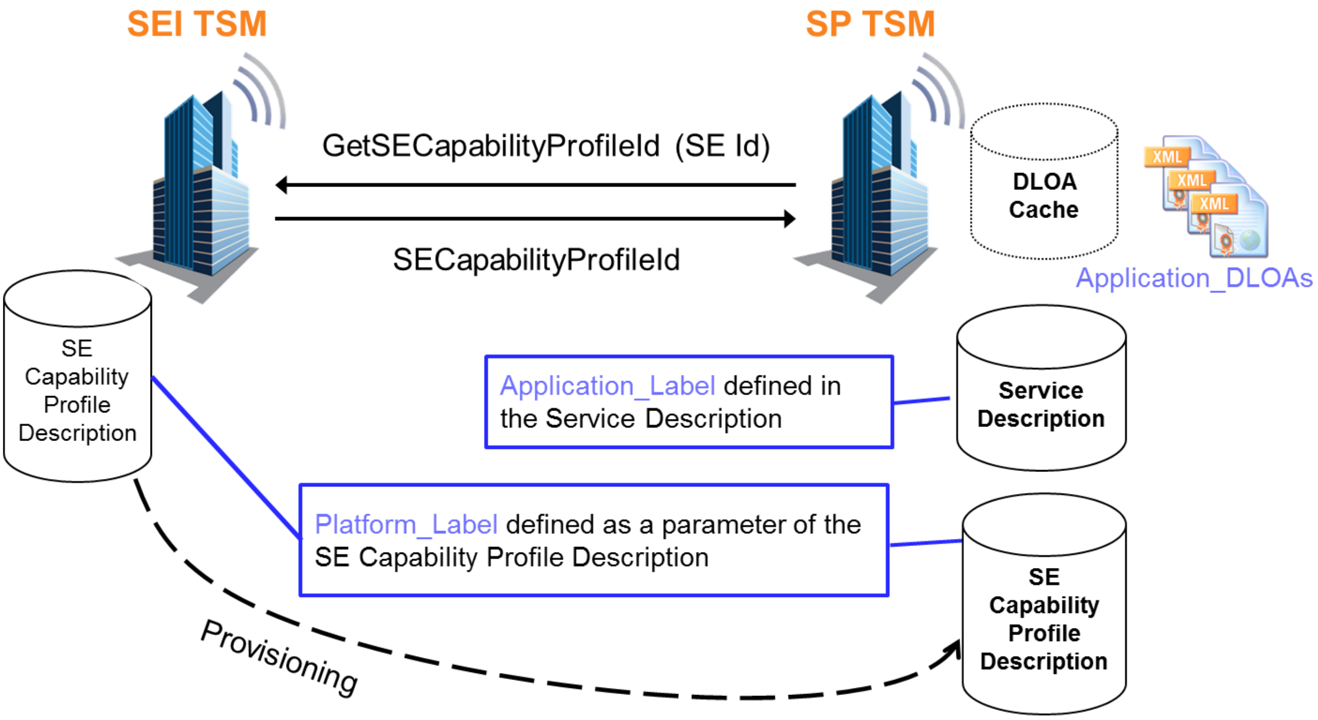
- Retrieve the different labels needed to get an SE_Application_DLOA as defined in section 6.1.1,
- Retrieve the different labels needed to get the Platform_DLOA as defined in section 6.1.2,
- Retrieve the URL of the DLOA Registrar(s) as defined in section 6.1.3,
- Retrieve the corresponding DLOAs either from its DLOA cache or by interfacing with the DLOA Registrars as defined in section 6.1.4,
- And then, check the validity of the applicable DLOAs as defined in section 6.1.5.

6.1.1 Retrieval of Labels to Get SE_Application_DLOAs

To retrieve an Application_DLOA, the SP TSM needs the Application_Label of the SE application to be deployed and the Platform_Label of the SE platform onto which the application is to be deployed:

- The SP TSM SHALL retrieve the `SECapabilityProfileID` corresponding to the targeted SE using the function `GetSECapabilityProfileId` implemented by the SEI TSM as defined in [GP SM], and then retrieve the Platform_Label that shall be part of the Secure Element capabilities associated with this `SECapabilityProfileID` as defined in section 6.2.
- The SP TSM SHALL retrieve, from the description of the service the SP TSM has on its system, the Application_Label associated with the requested Service and for the targeted SE.

Figure 6-1: SP TSM Process to Retrieve SE_Platform_Label and SE_Application_Label

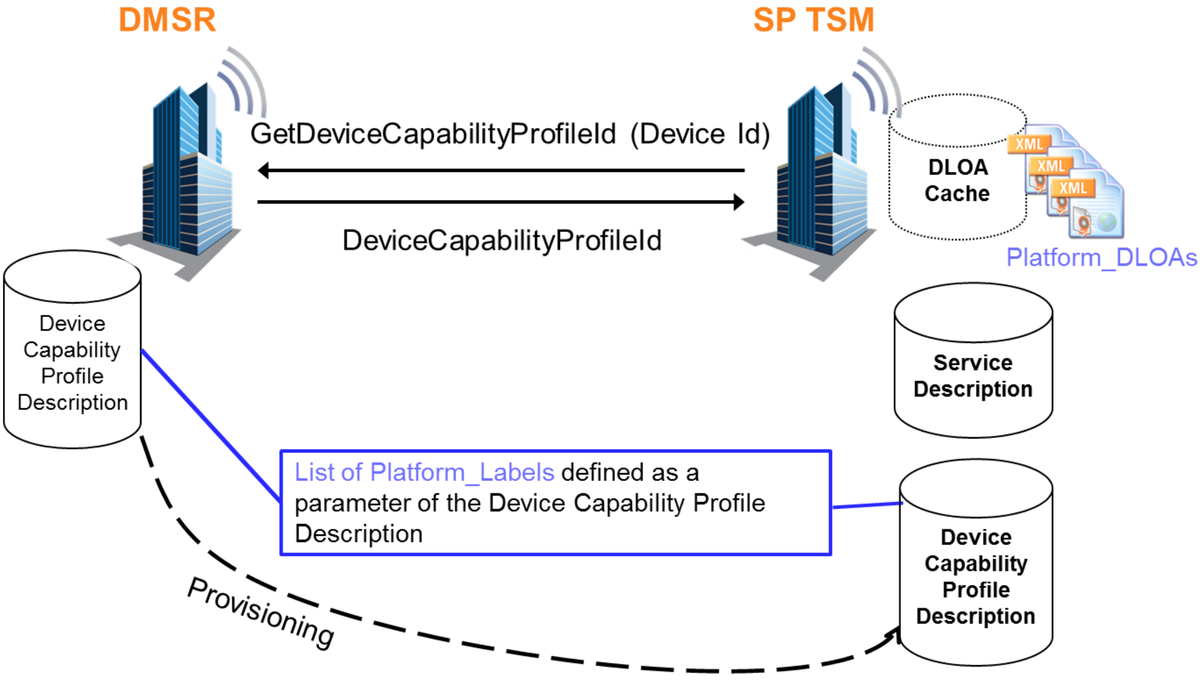

SEI TSM
SP TSM
GetSECapabilityProfileId (SE Id)
SECapabilityProfileId
DLOA Cache
XML
XML
XML
Application_DLOAs
SE Capability Profile Description
Application_Label defined in the Service Description
Service Description
Platform_Label defined as a parameter of the SE Capability Profile Description
SE Capability Profile Description
Provisioning

6.1.2 Retrieval of Labels to Get Platform_DLOAs

To retrieve the Platform_DLOA of the different platforms available in a device, the SP TSM needs the Platform_Labels available in the device onto which the service is to be deployed:

- The SP TSM SHALL retrieve the `DeviceCapabilityProfileID` corresponding to the targeted device using the function `GetDeviceCapabilityProfileId` implemented by the DMSR as defined in [GP SM], and then retrieve the list of Platform_Labels that shall be part of the Device capabilities associated with this `DeviceCapabilityProfileID` as defined in section 6.3.

Figure 6-2: SP TSM Process to Retrieve Other Platform_Labels Available in the Device


DMSR
SP TSM
GetDeviceCapabilityProfileId (Device Id)
DeviceCapabilityProfileId
DLOA Cache
XML
XML
XML
Platform_DLOAs
Device Capability Profile Description
Service Description
List of Platform_Labels defined as a parameter of the Device Capability Profile Description
Device Capability Profile Description
Provisioning

6.1.3 Retrieval of the URL of the DLOA Registrar(s)

DLOAs required by a given SP TSM MAY be handled by different DLOA Registrars.

- The SP TSM SHOULD retrieve the URL of the DLOA Registrar in charge of the SE_Application_DLOA from the service description for the targeted SE, if available. Otherwise the SP TSM SHOULD retrieve a default URL of the DLOA Registrar associated with the SE platform via the corresponding Secure Element capabilities defined for the `SECapabilityProfileID`; see details in section 6.2.

  The SP TSM MAY also retrieve the URL of the DLOA Registrar associated with an SE platform by sending a request to the Secure Element and using the System Protocol Discovery Mechanism defined in [GP SPDM]. In that case, the process to follow is defined in section 6.1.3.1.

- The SP TSM SHOULD retrieve the URL of the DLOA Registrar in charge of the other Platform_DLOAs from the description of the service the SP TSM has on its system, if available. Otherwise the SP TSM SHOULD retrieve a default URL of the DLOA Registrar associated with the Device via the corresponding Device capabilities defined for the `DeviceCapabilityProfileID`; see details in section 6.3.

6.1.3.1 Retrieval of Default URL of SE DLOA Registrar from Discovery URL Stored in SE

To retrieve the URL of the DLOA Registrar associated with an SE by sending a request to the Secure Element itself, the SP TSM needs to follow this process:

- Retrieve the Discovery Base URL of the SE DLOA Registrar associated with an SE by sending a request to the SE as defined in section 7.2.
- Use the retrieved Discovery Base URL to build the SPDM URL as defined in [GP SPDM]. The built SPDM URL SHALL have the following format:

  ```
  {http|https}://<Actor Domain and
  Path>/spdm/protocols?version=<version>&requesterId=<Discovering actor Id>&
  secureComponentType={SE|TEE}
  &secureComponentId=<SC Id>[&<additional query parameters>]
  ```

  Where

  - The `<Actor_Domain and Path>` is the location domain and path obtained from the Discovery Base URL retrieved from the Secure Element. Refer to section 7.1.2.
  - The `version` query parameter is the version of the System Protocol Discovery Mechanism that is used by the SP TSM, as defined in [GP SPDM].
  - The `requesterId` query parameter represents the identifier of the Discovering actor, as defined in the section “Usage for GlobalPlatform System” of [GP SPDM].
  - The `secureComponentType` query parameter is the type of Secure Component, as defined in [GP SPDM]. In the context of a Secure Element, it SHALL be "`SE`".
  - The `secureComponentId` query parameter is the identifier of the Secure Component, as defined in [GP SPDM]. In the context of a Secure Element, it SHALL be the Card Unique Data value.

  If the Discovery Base URL contains additional query parameters, then the SP TSM SHALL also manage the location of the '?' character so that when building the SPDM URL, the domain and path part of the Discovery Base URL are separated from the additional query parameters part of this Discovery Base URL.

- Send HTTP GET Request using this built SPDM URL as defined in [GP SPDM]

  The HTTP GET Response will be as defined in [GP SPDM], with specific parameters values defined below:

  - The `providerId` parameter SHALL represent the identifier of the DLOA Registrar actor,
  - The `supportedProtocols` parameter SHALL contain an item corresponding to the DLOA Registrar interface. This item SHALL provide the following parameters:
    - The `protocolId` parameter, representing the identifier of the DLOA protocol interface, SHALL be set to "`gps-dloa`".
    - The `protocolVersion` parameter, representing the version of the DLOA protocol interface, SHALL be set to "`1.0.0`".
    - As for the GlobalPlatform System messaging for mobile-NFC services management (defined in the section ‘Usage for "GlobalPlatform System Messaging for Mobile-NFC Services Management" for Secure Elements’ of [GP SPDM]), the `protocolInfo` parameter SHALL contain a `functionGroups` parameter (Mandatory) with at least an item as follows:
      - The `functionGroup` parameter (Mandatory) set to "`DLOARetrieval`".

    - The endPoint parameter (Mandatory), representing the DLOA_Registrar_URL that will be used by the SP TSM to retrieve the DLOAs as defined in section 5.2.

Here is an example of HTTP GET RESPONSE

```
{
  "version": "1.0.0",
  "providerId": "5.6.7.8",

  "supportedProtocols": [
    {
      "protocolId": "gps-dloa",
      "protocolVersion": "1.0.0",
      "protocolInfo": {
        "functionGroups": [
          {
            "functionGroup": "DLOARetrieval",
            "endPoint": "https://myCompany.com/DLOARegistrarEndPoint"
          }
        ]
      }
    }
  ]
}
```

6.1.4 Retrieval of Application_DLOAs and Platform_DLOAs

This specification defines two options to enable the SP TSM to retrieve Application_DLOAs and Platform_DLOAs during the CheckGlobalEligibility process:

- Interfacing directly with the DLOA Registrar using the interface defined in section 5.2, or
- Implementing a DLOA caching mechanism.

If SP TSM implements a DLOA caching mechanism, the SP TSM SHALL use the interface defined in section 5.2 to refresh the DLOA cache. How often the cache is refreshed SHALL be part of the business agreement between the SP and the SP TSM.

Note: The cache MUST be flushed and refreshed to consider new, updated or revoked DLOAs. In this context, refreshing the cache daily sounds a good compromise between the objectives of removing revoked DLOAs and of minimizing the number of exchanges between the SP TSM and the DLOA Registrar.

6.1.5 Checking the Validity of Application_DLOAs and Platform_DLOAs

During the CheckGlobalEligibility function execution, the SP TSM SHALL consider an Application_DLOA or a Platform_DLOA as valid in the following cases:

- The DLOA exists in the DLOA Registrar or in the SP TSM up-to-date DLOA cache, and
- The Expiration_Date has not yet occurred.

Otherwise, the SP TSM SHALL consider that Application_DLOA or Platform_DLOA is not valid and return '11' (SE capability) or '21' (Device capability) as Non Eligibility Reason in the CheckGlobalEligibility function output data.

6.2 Extension of the Secure Element Capabilities

The Secure Element capabilities defined in [GP SM] are extended to retrieve the Platform_Label and the URL of the DLOA Registrar associated with a given SE.

Refer to [GP SM] for the exact definition of the ‘`Platform_Label`’ and '`SE_DLOA_Registrar_URL`’ capabilities.

6.3 Extension of the Device Capabilities

The Device capabilities defined in [GP SM] are extended to retrieve the Platform_Label of the different platforms available in the device and the URL of the DLOA Registrar associated with a given device.

Refer to [GP SM] for the exact definition of the ‘`Platform_Label`’ and '`Device_DLOA_Registrar_URL`’ capabilities.

7 Interface Provided by the Secure Element

The Platform_Label corresponding to the platform implemented on a given SE and the Discovery Base URL of the default DLOA Registrar associated with this SE SHALL be stored in the SE in a Certification Data object as defined in section 7.1.

The whole Certification Data object can be retrieved from the SE using the mechanism defined in section 7.2.

The Discovery Base URL of the SE default DLOA Registrar can be updated as defined in section 7.3.

As the Platform_Label stored in a given SE shall identify the object that was submitted to the evaluation process – in other words, the Platform_Label shall change if there is any change in the code of the platform – the mechanisms that may lead to an update of the Platform_Label are very specific and out of scope of this version of the specification.

7.1 Certification Data Object

The Platform_Label corresponding to the platform implemented by the SE and the Discovery Base URL of the default DLOA Registrar associated to the SE SHALL be stored by OPEN in a Certification Data object as defined below.

Table 7-1: Certification Data Object

<table>
<tr><th>Tag</th><th>Len</th><th colspan="3">Description</th><th>Presence</th></tr>
<tr><td rowspan="4">'7F22'</td><td rowspan="4">Var</td><td colspan="3">Certification Data</td><td></td></tr>
<tr><th>Tag</th><th>Len</th><th>Description</th><th></th></tr>
<tr><td>'5F45'</td><td>Var</td><td>Platform_Label</td><td>Mandatory</td></tr>
<tr><td>'5F50'</td><td>Var</td><td>Discovery Base URL of the SE default DLOA Registrar</td><td>Optional</td></tr>
</table>

7.1.1 Description of the Platform_Label Stored in the SE

The Platform_Label available in the Certification Data object corresponds to the ASCII encoding, coded on one byte and left-justified (see [8859-1]), of the `Platform_Label` defined as a string in section 3.1.

As an example, the Platform_Label `"1.2.840.114283/My_Platform_Label_1a"` will be encoded as follows:

```
'312E322E3834302E3131343238332F4D795F506C6174666F726D5F4C6162656C5F3161'
```

7.1.2 Description of the Discovery Base URL of the SE Default DLOA Registrar

The Discovery Base URL of the SE default DLOA Registrar available in the Certification Data object defines the Base URL that SHALL be used as defined in section 6.1.3.1 to build the SPDM URL required to discover the DLOA Registrar URL (`DLOA_Registrar_URL`).

This Discovery Base URL SHALL respect the format of Base URLs defined in [GP SPDM]:

```
{http|https}://<Actor Domain and Path>[?<additional query parameters>]
```

Where, as defined in [GP SPDM]:

- The URL scheme is either `http` or `https`.
- The `<Actor Domain and Path>` is the location domain and path of the DLOA Registrar, to be used as the domain and path for the System Protocol Discovery Mechanism URL (refer to section 6.1.3.1).
- Additional query parameters may (Optional) be provided in the `<additional query parameters>` part of the URL, each individual query parameter respecting the following format `"<tag>=<value>"`, and being separated by a `'&'` character.

The Discovery Base URL of the SE default DLOA Registrar SHALL not exceed 200 characters.

An example of Discovery Base URL of the DLOA Registrar is:

```
https://mycompany.com/myDLOARegistrar
```

The Discovery Base URL of the SE default DLOA Registrar stored in the Certification Data SHALL correspond to the ASCII encoding, coded on one byte and left-justified (see [8859-1]) of the Discovery Base URL of the SE default DLOA Registrar defined above.

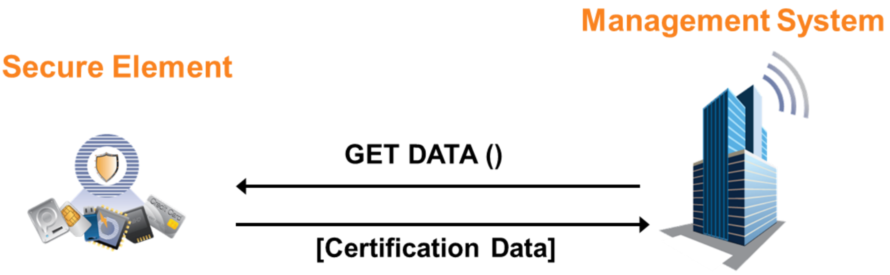
As an example, the Discovery Base URL `"https://mycompany.com/myDLOARegistrar"` will be encoded as follows:

```
'68747470733a2f2f6d79636f6d70616e792e636f6d2f6d79444c4f41526567697374726172'
```

7.2 Retrieval of Certification Data Object from Secure Element

As defined in section 7.1, the Certification Data object includes the Platform_Label and the Discovery Base URL of the SE default DLOA Registrar. This Certification Data object can be retrieved from the Secure Element by sending a GET DATA command to any Security Domain as defined in section 7.2.1.

Figure 7-1: Overview of the Interface Provided by the Secure Element



7.2.1 GET DATA Command Used to Retrieve the Certification Data Object

A mechanism is defined to enable retrieval of data forwarded from the OPEN. This feature allows an off-card entity to select (or target through an OTA Administration Session) its own Security Domain to retrieve data actually owned by the OPEN.

The off-card entity shall send to its own Security Domain a GET DATA command complying with the following requirements:

- The P1-P2 parameters shall be set to 'BF31'
- The command shall have a command data field encoding a request for one (and only one) of the following data:
  - Certification Data object: '5C 02 7F22'

If the command data field is coded differently, a status word of '6A80' shall be returned.

If the OPEN is not able to provide such data, a status word of '6A88' shall be returned. Otherwise, the response shall contain the requested data object retrieved from the OPEN, encapsulated in a data object with tag 'BF31'.

For example,

- The following GlobalPlatform GET DATA command:

  '80 CA BF 31 04 5C 02 7F 22'

  would receive an answer of the following form:

  'BF 31' (length) '7F 22' (length) (value)

- The following ISO GET DATA command:

  '00 CA BF 31 04 5C 02 7F 22'

  would receive an answer of the following form:

  '7F 22' (length) (value)

The current Security Level as defined in [GP CS] section 10.6 SHALL include at least C-MAC and R-MAC, otherwise a status word '6982' shall be returned.

7.3 Updating the Discovery Base URL of the SE DLOA Registrar

The Discovery Base URL of the SE default DLOA Registrar that is part of the Certification Data object SHALL be updated by sending a STORE DATA command as defined in [GP CS] section 6.3 to the CASD. If and only if no CASD is present on the SE, the STORE DATA command SHALL be sent to the ISD.

The data field of the STORE DATA command SHALL be as follows:

Table 7-2: STORE DATA Command Data field

Tag	Len	Description			Presence
'7F22'	Var	Certification Data			
		Tag	**Len**	**Description**	
		'5F50'	Var	Discovery Base URL of the SE default DLOA Registrar	Mandatory

8 Interface Provided by the Trusted Execution Environment

The TEE properties defined in [GP TEE Mgmt] are extended to include the Platform_Label corresponding to the platform implemented on a given TEE.

Refer to [GP TEE Mgmt] for the exact definition of the `'gpd.tee.platformLabel'` TEE property and the extension of the TEE type.

9 Additional Features

9.1 Interaction between the SP TSM and the Service Provider

In order to make some business decisions regarding services already deployed, a Service Provider (SP) MAY want to be informed that a given Application_DLOA is no longer valid (expiration or revocation).

In that case, the business agreement between the SP and the SP TSM will define mechanisms for the SP TSM to inform the SP that an Application_DLOA or a Platform_DLOA is no longer valid. These mechanisms are out of scope of this specification.

On the other side, if the SP has obtained a waiver for a given Application_DLOA that is no longer valid, the SP needs to inform the SP TSM to consider the waiver during the eligibility process of services using this application.

In that case, the business agreement between the SP and the SP TSM will define mechanisms for the SP to inform the SP TSM about the obtained waivers and their context of use. These mechanisms are out of scope of this specification.

In addition, the SP MAY require the SP TSM to manage the limit date of usage of the Application (as defined in the `Application_Limit_Date` field of the DLOA). In that case, mechanisms already defined in [GP SM] for service suspension or service un-deployment can be re-used.

Annex A DLOA XML

A.1 DLOA XSD

Below is the xsd schema describing the DLOAs (Platform_DLOA, Application_DLOA).

Note: This xsd is provided here for the reader’s convenience but the xsd file “dloa.xsd” published with this specification shall be used as a reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:dloa="http://namespaces.globalplatform.org/systems-dloa/1.0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  targetNamespace="http://namespaces.globalplatform.org/systems-dloa/1.0"
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0.0.0">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="xmldsig-core-schema.xsd"/>
  <xs:element name="DLOA">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:annotation>
            <xs:documentation>A DLOA includes one of the following elements depending on the
type of DLOA. The characters double quote (") '0x42' and quote (') '0x47' shall not be used
in any attribute.</xs:documentation>
          </xs:annotation>
          <xs:element ref="dloa:PlatformLevel"/>
          <xs:element ref="dloa:ApplicationLevel"/>
        </xs:choice>
        <xs:element name="Signature" type="ds:SignatureType">
          <xs:annotation>
            <xs:documentation>The signature shall apply on the element comprising the DLOA
information ('PlatformLevel' or 'ApplicationLevel').
              This is well known by the entity in charge of building, and the entitity in
charge of verifying, the signature</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="version" use="required">
        <xs:annotation>
          <xs:documentation>Indicates, in the xml document, the version of schema used for
validation. Current version of schema is 1.0.0.0.
            Version type is indicated with a major, a minor, a revision number and optionally
a last patch number.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="\d+[.]\d+[.]\d+([.]\d+)?"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  <xs:element name="PlatformLevel">
    <xs:annotation>
      <xs:documentation>Contains the elements necessary for a Digital letter of Approval at
Platform level. Description of each field is provided in 'GlobalPlatform Card Digital Letter
of Approval' specification</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="dloa:Authority_Label"/>
        <xs:element ref="dloa:LOA_Identifier"/>
        <xs:element ref="dloa:LOA_Scope"/>
        <xs:element ref="dloa:Platform_Label"/>
        <xs:element ref="dloa:Issuance_Date"/>
        <xs:element ref="dloa:Expiration_Date" minOccurs="0"/>
        <xs:element ref="dloa:LOA_URL"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ApplicationLevel">
    <xs:annotation>
      <xs:documentation>Contains the elements necessary for a Digital letter of Approval at
Application level. Description of each field is provided in 'GlobalPlatform Card Digital Letter
of Approval' specification</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="dloa:Authority_Label"/>
        <xs:element ref="dloa:LOA_Identifier"/>
        <xs:element ref="dloa:LOA_Scope"/>
        <xs:element ref="dloa:Application_Label"/>
        <xs:element ref="dloa:Platform_Label"/>
        <xs:element ref="dloa:Issuance_Date"/>
        <xs:element ref="dloa:Expiration_Date" minOccurs="0"/>
        <xs:element ref="dloa:Application_Limit_Date" minOccurs="0"/>
        <xs:element ref="dloa:LOA_URL"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Authority_Label" type="xs:string"/>
  <xs:element name="LOA_Identifier" type="xs:string"/>
  <xs:element name="LOA_Scope" type="xs:string"/>
  <xs:element name="Platform_Label" type="xs:string"/>
  <xs:element name="Application_Label" type="xs:string"/>
  <xs:element name="Issuance_Date" type="xs:date"/>
  <xs:element name="Expiration_Date" type="xs:date"/>
  <xs:element name="Application_Limit_Date" type="xs:date"/>
  <xs:element name="LOA_URL" type="xs:anyURI"/>
</xs:schema>
```

A.2 Examples of DLOA XMLS

A.2.1 Example of Platform DLOA

Here is an example of Platform_DLOA in xml format:

Note: The values for the XML elements `<dloa:SignatureValue>` and `<dloa:X509Certificate>` are arbitrarily chosen.

```
<?xml version="1.0" encoding="UTF-8"?>
<dloa:DLOA xmlns:dloa="http://namespaces.globalplatform.org/systems-dloa/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://namespaces.globalplatform.org/systems-dloa/1.0 dloa.xsd"
version="1.0.0.0">
  <dloa:PlatformLevel>
    <dloa:Authority_Label>ACertificationScheme</dloa:Authority_Label>
    <dloa:LOA_Identifier>ACertificateIdentifier</dloa:LOA_Identifier>
    <dloa:LOA_Scope>ALoa scope1, scope2</dloa:LOA_Scope>
    <dloa:Platform_Label>MyPlatformLabel</dloa:Platform_Label>
    <dloa:Issuance_Date>2014-05-20</dloa:Issuance_Date>
    <dloa:LOA_URL>http://www.certificationschemeurl.com</dloa:LOA_URL>
  </dloa:PlatformLevel>
  <dloa:Signature>
    <SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#”"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"></SignatureMethod>
      <Reference>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"></DigestMethod>
        <DigestValue></DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">FFFFFFFFFFFF</SignatureValue>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>TW9uQ2VydGlmaWNhdA==</X509Certificate>
      </X509Data>
    </KeyInfo>
  </dloa:Signature>
</dloa:DLOA>
```

A.2.2 Example of Application DLOA

Here is an example of Application_DLOA in xml format:

Note: The values for the XML elements `<dloa:SignatureValue>` and `<dloa:X509Certificate>` are arbitrarily chosen.

```
<?xml version="1.0" encoding="UTF-8"?>
<dloa:DLOA xmlns:dloa="http://namespaces.globalplatform.org/systems-dloa/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://namespaces.globalplatform.org/systems-dloa/1.0 dloa.xsd"
version="1.0.0.0">
  <dloa:ApplicationLevel>
    <dloa:Authority_Label>ACertificationScheme</dloa:Authority_Label>
    <dloa:LOA_Identifier>ACertificateIdentifier</dloa:LOA_Identifier>
    <dloa:LOA_Scope>ALoa scope1, scope2</dloa:LOA_Scope>
    <dloa:Application_Label>MyApplicationLabel</dloa:Application_Label>
    <dloa:Platform_Label>MyPlatformLabel</dloa:Platform_Label>
    <dloa:Issuance_Date>2014-05-20</dloa:Issuance_Date>
    <dloa:Expiration_Date>2019-05-20</dloa:Expiration_Date>
    <dloa:Application_Limit_Date>2020-05-20</dloa:Application_Limit_Date>
    <dloa:LOA_URL>http://www.mycertificationschemeurl.com</dloa:LOA_URL>
  </dloa:ApplicationLevel>
  <dloa:Signature>
    <SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#”"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"></SignatureMethod>
      <Reference>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"></DigestMethod>
        <DigestValue></DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">FFFFFFFFFFFF</SignatureValue>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>TW9uQ2VydGlmaWNhdA==</X509Certificate>
      </X509Data>
    </KeyInfo>
  </dloa:Signature>
</dloa:DLOA>
```

Annex B DLOA Registrar REST Interface

B.1 JSON Schema of REST Interface

B.1.1 Get All DLOAs

This section represents the JSON Schema of the response of the Get All DLOAs interface (valid for all types of DLOA), according to the JSON Schema: core definitions and terminology Draft 04 [JSON Schema].

Note: This JSON Schema is provided here for the reader's convenience but the JSON file "DLOARegistrar#Get All DLOAs.schema.json" published with this specification shall be used as a reference.

```
{
  "title": "DLOA Registrar - Get All DLOAs: JSON Schema of the response Body",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "description": "The list of platforms or applications DLOAs that are known by the DLOA
Registrar. Only active DLOA shall be returned: expired or revoked DLOA shall not be returned.
If no platform or application DLOA is known by the DLOA Registrar, then this list shall be
empty.",
  "anyOf": [
    {
      "items": {
        "type": "object",
        "description": "The DLOAs of a Platform.",
        "properties": {
          "Platform_Label": {
            "$ref": "#/definitions/Platform_Label"
          },
          "DLOA": {
            "$ref": "#/definitions/DLOA"
          }
        },
        "required": [
          "Platform_Label",
          "DLOA"
        ],
        "additionalProperties": false
      }
    },
    {
      "items": {
        "type": "object",
        "description": "The DLOAs of an Application.",
        "properties": {
          "Platform_Label": {
            "$ref": "#/definitions/Platform_Label"
          },
          "Application_Label": {
            "$ref": "#/definitions/Application_Label"
          },
          "DLOA": {
            "$ref": "#/definitions/DLOA"
          }
        },
        "required": [
          "Platform_Label",
          "Application_Label",
          "DLOA"
        ],
        "additionalProperties": false
      }
    }
  ],
  "definitions": {
    "DLOA": {
      "type": "array",
      "description": "The list of DLOAs.",
      "minItems": 1,
      "items": {
        "type": "object",
        "properties": {
          "DLOA_XML": {
            "type": "string",
            "description": "The DLOA, in XML format, corresponding to the given platform or
application.",
            "media": {
              "binaryEncoding": "base64"
            }
          }
        },
        "additionalProperties": false,
        "required": [
          "DLOA_XML"
        ]
      }
    },
    "Platform_Label": {
      "type": "string",
      "maxLength": 120,
      "description": "The label of the platform."
    },
    "Application_Label": {
      "type": "string",
      "maxLength": 120,
      "description": "The label of the application."
    }
  }
}
```

B.1.2 Get DLOA

This section represents the JSON Schema representation of the response of the Get DLOA interface, according to [JSON Schema].

Note: This JSON Schema is provided here for the reader’s convenience but the JSON file "DLOARegistrar#Get DLOA.schema" published with this specification shall be used as a reference.

```
{
  "title": "DLOA Registrar - Get DLOA: JSON Schema of the response Body",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "description": "The list of DLOAs for the given platform or application. Only active DLOAs
shall be returned: expired or revoked DLOAs shall not be returned. If no active DLOA for the
given platform or application is known by the DLOA Registrar, then this list shall be empty.",
  "items": {
    "type": "object",
    "properties": {
      "DLOA_XML": {
        "type": "string",
        "description": "The DLOA, in XML format, corresponding to the given platform or
application.",
        "media": {
          "binaryEncoding": "base64"
        }
      }
    },
    "required": [
      "DLOA_XML"
    ],
    "additionalProperties": false
  }
}
```

B.2 Examples of REST Calls

B.2.1 Get All DLOAs

Example of request:

- For Platform_DLOAs:

```
https://mycompany.com/DLOARegistrarEndPoint/v1/dloas/platform
```

- For Application_DLOAs:

```
https://mycompany.com/DLOARegistrarEndPoint/v1/dloas/application
```

Examples of response:

- Response for Platform_DLOAs:

```
[
  {
    "Platform_Label": "1.2.840.114283/My_Platform_Label_1a",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  },
  {
    "Platform_Label": "1.2.840.114283/My_Platform_Label_2",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  },
  {
    "Platform_Label": "1.2.840.1234567/LBL12-04-14",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  }
]
```

- Response for Application_DLOAs:

```
[
  {
    "Platform_Label": "1.2.840.114283/My_Platform_Label_1a",
    "Application_Label": "1.2.840.1234567%2FMyApplication_1.2",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  },
  {
    "Platform_Label": "1.2.840.114283/My_Platform_Label_1a",
    "Application_Label": "1.2.840.1234567%2FMyApplication_2.0",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  },
  {
    "Platform_Label": "1.2.840.1234567/LBL12-04-14",
    "Application_Label": "1.2.840.1234567%2FMyApplication_1.2",
    "DLOA": [
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      },
      {
        "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
      }
    ]
  }
]
```

- Response for no DLOA matching the request:

```
[]
```

B.2.2 Get DLOA

Examples of request:

- For Platform_DLOA:

```
https://mycompany.com/DLOARegistrarEndPoint/v1/dloa/platform?Platform_Label=1.2.840.114283%2
    FMy_Platform_Label_1a
```

Note that the '/' character of the platform label has been URL-encoded into the '%2F' string

- For Application_DLOA:

```
https://mycompany.com/DLOARegistrarEndPoint/v1/dloa/application?Platform_Label=1.2.840.11428
    3%2FMy_Platform_Label_1a&Application_Label=1.2.840.1234567%2FMyApplication_1.2
```

Note that the '/' character of the platform label and the application label has been URL-encoded into the '%2F' string

Examples of response:

- Response for one single DLOA matching the request:

```
[
  {
    "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
  }
]
```

- Response for several DLOAs matching the request:

```
[
  {
    "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
  },
  {
    "DLOA_XML": "PD94bWwgdmVyc2lvbj0iMS4w…"
  }
]
```

- Response for no DLOA matching the request:

```
[]
```