



OIPF

Release 2 Specification

**Volume 7 – Authentication, Content Protection and
Service Protection**

[V2.3] – [2014-01-24]

Open IPTV Forum

Open IPTV Forum

Postal address

Open IPTV Forum support office address
650 Route des Lucioles – Sophia Antipolis
Valbonne – FRANCE
Tel.: +33 4 92 94 43 83
Fax: +33 4 92 38 52 90

Internet

<http://www.oipf.tv>

Disclaimer

The Open IPTV Forum accepts no liability whatsoever for any use of this document.

Copyright Notification

No part may be reproduced except as authorized by written permission.
Any form of reproduction and/or distribution of these works is prohibited.

Copyright © 2014 Open IPTV Forum e.V.

All rights reserved.

Contents

FOREWORD	7
INTRODUCTION	7
1 SCOPE	8
2 REFERENCES	9
2.1 Normative References.....	9
2.2 Open IPTV Forum References	10
2.3 Informative References.....	10
3 CONVENTIONS AND TERMINOLOGY	11
3.1 Conventions	11
3.2 Terminology	11
3.2.1 Definitions	11
3.2.2 Abbreviations	12
4 CONTENT AND SERVICE PROTECTION	15
4.1 Terminal-Centric Approach	15
4.1.1 Interfaces for CSP and CSP-T Server	15
4.1.2 Protected Content Usages	24
4.1.3 Content Encryption (informative)	27
4.1.4 Protected File Formats	28
4.1.5 Protection of MPEG-2 Transport Streams	28
4.1.6 Operation of Marlin Technologies	32
4.1.7 DRM Data	32
4.2 Gateway-Centric Approach	37
4.2.1 Capabilities	37
4.2.2 CSPG-DAE Interface	37
4.2.3 CI+ based Gateway	38
4.2.4 DTCP-IP based Gateway	51
5 USER IDENTIFICATION, AUTHENTICATION, AUTHORISATION AND SERVICE ACCESS PROTECTION	56
5.1 General Principals	56
5.2 Interfaces	57
5.2.1 HNI-INI	57
5.2.2 HNI-IGI	57
5.2.3 Common Requirements.....	57
5.3 Service Access Protection (informative).....	58
5.3.1 SAA Co-located with Service (informative).....	58
5.3.2 SAA Standalone (informative).....	58
5.4 OITF Authentication Mechanisms	59
5.4.1 HTTP Basic and Digest Authentication.....	59
5.4.2 Network Based Authentication (informative)	60
5.4.3 Web Based Authentication.....	61
5.4.4 HTTP Digest Authentication – Using IMS Gateway	62
5.4.5 GBA Authentication – Using IMS Gateway.....	66
5.5 IMS Registration – OITF	69
5.5.1 Relevant Functional Entities and Reference Points	69
5.5.2 Prerequisites.....	70
5.5.3 SIP Digest Message Flows.....	71
5.5.4 IMS AKA Message Flows	72
5.6 Session Management and Single Sign On	74
5.6.1 Cookie Session.....	74
5.6.2 URL Parameters (informative).....	75
5.6.3 HTTP Authentication Session.....	75
5.6.4 SAML Web-based SSO	76
6 FORCED PLAY OUT USING MEDIA ZONES.....	78
ANNEX A LINK OF USER AUTHENTICATION AND DRM DEVICE AUTHENTICATION (INFORMATIVE).....	79

ANNEX B	XML SCHEMAS	81
B.1	XML Schema for MarlinPrivateDataType Structure	81
B.2	XML Schema for MIPPVControlMessage Format	82
B.3	XML Schema for HexBinaryPrivateDataType Structure	82
ANNEX C	DRM MESSAGES USED IN DAE (INFORMATIVE)	83
ANNEX D	CSPG-CI+ USAGE EXAMPLES (INFORMATIVE)	84
D.1	CSPG-CI+ Initial Power-on (informative)	84
D.2	CSPG-CI+ Normal Power-on (informative)	84
D.3	Live Session Example (informative)	85
D.4	Parental Control Management Example (informative)	86
D.5	No Rights Event and Purchase Example (informative)	87
D.6	VOD Session Example (informative)	88
ANNEX E	CSPG-DTCP SESSION SETUP SEQUENCE EXAMPLES (INFORMATIVE)	89
E.1	Multicast streaming with SIP session management	90
E.2	Unicast Streaming with SIP session management	92
E.3	Unicast Streaming with RTSP session management	93
E.4	HTTP Streaming and Download	93
ANNEX F	EMBEDDED CSPG (INFORMATIVE)	94
F.1	Application to Simple and Secure Streaming	96

Tables

Table 1: Recording Control access_criteria_descriptor	30
Table 2: Bit Assignments of recording_control_information_byte	30
Table 3: DNR and DNTS Combinations	30
Table 4: Parental_Control_URL Parameter Syntax	31
Table 5: DRMControlInformation Mapping for Marlin	32
Table 6: MarlinPrivateData Structure	36
Table 7: MIPPVControlMessage Format	37
Table 8: Open IPTV Forum private_host_application_ID.....	39
Table 9: SAS_async_msg() APDU syntax	39
Table 10: Generic message_byte() syntax	40
Table 11: OIPF specific messages and command_id values.....	40
Table 12: OIPF specific datatype_id values	41
Table 13: Mapping to DAE API or Events	41
Table 14: send_msg message data types	42
Table 15: reply_msg message data types	42
Table 16: parental_control_info message data types.....	43
Table 17: rights_info message data types	45
Table 18: system_info message data types	46
Table 19: can_play_content_req message data types.....	46
Table 20: can_play_content_reply message data types	47
Table 21: can_record_content_req message data types	47
Table 22: can_record_content_reply message data types.....	47
Table 19: Scrambling Modes	49
Table 20: DRMControlInformation Mapping for CSPG-CI+.....	50
Table 21: HexBinaryPrivateData Structure	51
Table 22: CA_descriptor.....	54
Table 23: DRM Messages used in DAE	83

Figures

Figure 1: CSP-T System Overview.....	15
Figure 2: Node Acquisition Sequence	17
Figure 3: Link Acquisition Sequence.....	19
Figure 4: Deregistration Sequence.....	21
Figure 5: License Acquisition Sequence.....	23
Figure 6: License Evaluation Sequence	25
Figure 7: Scramble Key Decryption Sequence	26
Figure 8: Content on Demand Encryption Sequence using Content Key (for (P)DCF [OMARLIN] or Marlin IPMP [MRL FF]).....	27
Figure 9: Content on Demand Encryption Sequence using Content Key (for MPEG-2 TS).....	27
Figure 10: Scheduled Content Encryption Sequence using Scramble Key (for MPEG-2 TS)	28
Figure 11: Conditional Access Descriptors Signalling ECM and EMM Messages	29
Figure 12: Outline of DRMControlInformationType with MarlinPrivateData.....	35
Figure 13: Outline of MIPPVControlMessage	37
Figure 14: CSPG-CI+ Overview.....	38
Figure 15: CSPG-CI+ Context.....	39
Figure 16: CSPG-DTCP Overview.....	51
Figure 17: Overview of Involved Reference Points.....	52
Figure 18: General Message Flow for Service Access Protection and User Authentication	56
Figure 19: SAA Co-located with Requested Service.....	58
Figure 20: Standalone SAA, Redirection Mode	59
Figure 21: HTTP Basic and Digest Authentication	60
Figure 22: Network Based Authentication.....	60
Figure 23: Web Based Authentication with Form	61
Figure 24: Initial procedure	63
Figure 25: Authentication between an OITF and an SAA based on HTTP credentials stored in IG	64
Figure 26: Authentication between an OITF and an SAA Based on GBA Credentials	65
Figure 27: Initial GBA Registration	67
Figure 28: Authentication between an OITF and an SAA Based on GBA Keys.....	68
Figure 29: OIPF Functional Entities and Reference Points Involved in IMS Registration.....	70
Figure 30: SIP Digest Message Flow Interlaced into IMS Registration	71
Figure 31: User Identification and Authentication based on the IMS AKA procedure	73
Figure 32: Session Management Using Cookie	74
Figure 33: Session Management Using URL Parameters.....	75
Figure 34: HTTP Authentication Session	76
Figure 35: SAML Web-based SSO.....	77
Figure 36: User Authentication for CSP, CSP-T Server communication	79
Figure 37: CSPG-CI+ First Power-on.....	84
Figure 38: CSPG-CI+ Normal Power-on.....	84
Figure 39: CSPG-CI+ Live Session Example.....	85
Figure 40: Parental Control Management Example.....	86
Figure 41: No Rights Event and Purchase Example	87
Figure 42: VOD Session Example	88
Figure 43: Session Setup Sequence for multicast streaming with SIP session management	90
Figure 44: CSPG-DTCP Initiated Teardown Sequence for multicast streaming with SIP session management.....	91
Figure 45: Session Setup Sequence for unicast streaming with SIP session management	92
Figure 46: Session Setup Sequence for unicast streaming with RTSP session management.....	93
Figure 47: Session Setup Sequence for HTTP Streaming and Download	93

Figure 48: Simple and Secure Streaming with CSPG.....96

Foreword

This Technical Specification (TS) has been produced by the Open IPTV Forum.

This specification provides multiple options for some features. The Open IPTV Forum Profiles specification complements the Release 2 specifications by defining the Open IPTV Forum implementation and deployment profiles.

Introduction

The Open IPTV Forum Release 2 Specification consists of ten Volumes:

- Volume 1 – Overview,
- Volume 2 – Media Formats,
- Volume 2a – HTTP Adaptive Streaming,
- Volume 3 – Content Metadata,
- Volume 4 – Protocols,
- Volume 4a – Examples of Protocol Sequences,
- Volume 5 – Declarative Application Environment,
- Volume 5a – Web Standards TV Profile,
- Volume 6 – Procedural Application Environment
- Volume 7 – Authentication, Content Protection and Service Protection (the present document).

The present document, the Authentication, Content Protection and Service Protection Specification (Volume 7), specifies the Authentication, Content and Service Protection functionality of the OIPF Release 2 solution.

The requirements for this functionality are derived from the following sources:

- Open IPTV Forum Service and Platform Requirement for R2, see [OIPF_SERV2];
- Open IPTV Forum Functional Architecture for R2, see [OIPF_ARCH2].

1 Scope

For the system-wide scope, refer to [OIPF_OVIEW2], section 1.

The scope of the present Volume is content protection, service protection, service access protection, user identification, user authentication, and user authorisation.

The following sections contain features for which the criteria that determine under which circumstances these features are implemented are out of the scope of the present document or contain conditional normative statements referring to other volumes of the Open IPTV Forum specifications:

- 4.1 Terminal-Centric Approach
- 4.1.4 Protected File Formats
- 4.1.5 Protection of MPEG-2 Transport Streams
- 4.2.3 CI+ based Gateway
- 4.2.3.6 Protected Streaming and File Formats
- 4.2.3.7 Personal Video Recorder
- 4.2.3.8 Time Shifting
- 4.2.4 DTCP-IP based Gateway
- 4.2.4.5 Protected Streaming and File Formats
- 5.4.4 HTTP Digest Authentication using IMS Gateway
- 5.4.5 GBA Authentication using IMS Gateway

Note that GBA authentication can be achieved using either the mechanism in section 5.4.5 GBA Authentication using IMS Gateway or the, more general, mechanism in section 5.4.4 HTTP Digest Authentication using IMS Gateway. Section 5.4.4 allows the use of different authentication mechanism in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that section 5.4.5 GBA Authentication using IMS Gateway will be deprecated and removed in future versions of this specification.

2 References

2.1 Normative References

[3GPP24.109]	3GPP, TS 24.109, “Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details”.
[3GPP24.229]	3GPP, TS 24.229, “3GPP; Technical Specification Group Core Network and Terminals; Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 8)”.
[3GPP33.203]	3GPP, TS 33.203, “3GPP; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services (Release 8)”.
[3GPP33.220]	3GPP, TS 33.220, “Generic Authentication Architecture (GAA); Generic bootstrapping architecture”.
[ATIS-IDSa]	ATIS-0800006, IIF Default Scrambling Algorithm (IDSa)
[CEA-2014-A]	Consumer Electronics Association, CEA-2014-A, July 2007, “Web-based Protocol Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE)”, including the August 28, 2008 Errata.
[CENC]	ISO/IEC FDIS 23001-7:2011, “Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files”
[CI+]	CI Plus LLP, CI Plus Specification, “Content Security Extensions to the Common Interface”, V1.3 (2011-01).
[DTCP]	Hitachi, Intel, Matsushita, Sony, Toshiba, “Digital Transmission Content Protection Specification, Volume 1 (Informational Version)”, Revision 1.51.
[DTCP-AA]	DTLA, DTCP Adopter Agreement, “Digital Transmission Protection License Agreement”.
[DTCP-IP]	Hitachi, Intel, Matsushita, Sony, Toshiba, “DTCP Volume 1 Supplement E, Mapping DTCP to IP, (Informational Version)”, Revision 1.2.
[DVB-CA]	ETSI, ETR 289, “Digital Video Broadcasting (DVB); Support for the use of scrambling and Conditional Access (CA) within digital broadcasting systems”, October 1996.
[DVB-CI]	ETSI, EN 50221, “Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications”, February 1997, and ETSI, TS 101 699 V1.1.1, “Digital Video Broadcasting (DVB); Extensions to the Common Interface Specification”.
[DVB-SC]	ETSI, TS 103 197 V1.5.1, “Digital Video Broadcasting (DVB); Head-end implementation of DVB SimulCrypt”, March 2007.
[DVB-SI]	ETSI, EN 300 468 V1.13.1, “Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems”.
[DVB-SRM]	DVB BlueBook A135, “Digital Video Broadcasting (DVB); System Renewability Messages (SRM) in DVB systems”
[HDCP]	Digital Content Protection LLC, “High-bandwidth Digital Content Protection System”, Revision 1.3.
[IEC62455]	IEC, IEC 62455, “Internet protocol (IP) and transport stream (TS) based service access”.
[MPEG2TS]	ISO/IEC, ISO/IEC 13818-1, “Information technology – Generic coding of moving pictures and associated audio information: Systems”.
[MRL BBTS]	Marlin Developer Community, “Marlin Broadband Transport Stream Specification”, Version 1.0
[MRL BNSP]	Marlin Developer Community, “Marlin – Broadband Network Service Profile Specification”, Version 1.1
[MRL DMZ]	Marlin Developer Community, “Marlin Dynamic Media Zones”, Version 1.1
[MRL CORE]	Marlin Developer Community, “Marlin - Core System Specification”, Version 1.3
[MRL FF]	Marlin Developer Community, “Marlin – File Formats Specification”, Version 1.1
[MRL MS3]	Marlin Developer Community, “Marlin –Simple Secure Streaming Specification”, Version 1.1.1
[OMARLIN]	Marlin Developer Community, “OMArLin Specification”, Version 1.0
[RFC2109]	IETF, RFC 2109, “HTTP State Management Mechanism”.
[RFC2119]	IETF, RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels”.
[RFC2617]	IETF, RFC 2617, “HTTP Authentication: Basic and Digest Access Authentication”.

[RFC3261]	IETF, RFC 3261, "SIP: Session Initiation Protocol".
[RFC5746]	IETF, RFC 5746, "Transport Layer Security (TLS) Renegotiation Indication Extension".
[SAMLCORE]	OASIS, "Assertions and Protocols for the OASIS Security Markup Language (SAML) V2.0".
[SAMLPROF]	OASIS, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0".

2.2 Open IPTV Forum References

[OIPF_SERV2]	Open IPTV Forum, "Service and Platform Requirements", V2.0, December 2008.
[OIPF_ARCH2]	Open IPTV Forum, "Functional Architecture - V2.3", January 2014.
[OIPF_MEDIA2]	Open IPTV Forum, "Release 2 Solution Specification, Volume 2 - Media Formats", V2.3, January 2014.
[OIPF_META2]	Open IPTV Forum, "Release 2 Solution Specification, Volume 3 - Content Metadata", V2.3, January 2014.
[OIPF_PROT2]	Open IPTV Forum, "Release 2 Solution Specification, Volume 4 - Protocols", V2.3, January 2014.
[OIPF_DAE2]	Open IPTV Forum, "Release 2 Solution Specification, Volume 5 - Declarative Application Environment", V2.3, January 2014.

2.3 Informative References

[OIPF_OVIEW2]	Open IPTV Forum, "Release 2 Solution Specification, Volume 1 - Overview", V2.3, January 2014.
---------------	---

3 Conventions and Terminology

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Terminology

3.2.1 Definitions

Term	Definition
Business Token	collection of information defined in [MRL BNSP] that contains the service specific information for a given business model
Content and Service Protection Gateway	optional gateway function that provides a conversion from a (proprietary) content and service protection solution in the network to one that is supported by an OITF, as defined in this specification
Client Function	function that interacts with the Marlin Client Function in a Content and Service Protection
Content and Service Key Management Function	entity responsible for storing and providing Service, Program, Content Keys and ECM attached information This function MAY be physically co-located with other functions (e.g. the Content Delivery Network Controller for Content on Demand services), see [OIPF_ARCH2]. This entity has been identified in release 1 just to illustrate informatively the separation between content encryption which is part of content preparation and content delivery.
Content on Demand Encryption Management Function	back office Content on Demand function in charge of launching Content on Demand encryption. This entity has been identified in release 1 just to illustrate informatively the separation between content encryption which is part of content preparation and content delivery.
Content and Service Protection	function that handles service protection and content protection for the client in the OITF
CSP-G Server	functional entity in the network that handles content protection and service protection for the Content and Service Protection Gateway (CSPG) in the residential network
CSP-T Server	functional entity in the network that handles service protection and content protection for the CSP-T client in the OITF
Marlin Action Token	token defined either in [MRL BNSP] or in [MRL MS3] that is used to trigger the Marlin Protocols from the Marlin Client Function in CSP, and from which some information (e.g., business token) is used in the Marlin protocols. The mimeType attribute is used to qualify which Marlin token type is returned.
Marlin Client Function	compliant implementation of the Marlin Client that is defined in [MRL BNSP] and that enables secure communications (Marlin Protocols) with the Marlin Server Function in a CSP-T Server
Marlin Configuration Token	token defined in [MRL BNSP] that includes the location information of the Marlin Server Function in CSP-T Server with which the CSP communicates
Marlin Server Function	compliant implementation of the Marlin Server that is defined in [MRL BNSP] and that enables secure communications (Marlin Protocols) with the Marlin Client Function in a CSP

Term	Definition
MS3 Manifest File	A text document carrying a Stream Access Statement URL and a Content URI Template, as specified in [MRL MS3]
MS3 Stream Access Statement	Stream Access Statement as specified in [MRL MS3]
Output Control Information	Output Control Information as defined in [MRL BNSP] and [MRL BBTS]
Program Key	symmetric key defined in [IEC62455] that encrypts an ECM
Scramble Key	symmetric key that scrambles the content
Server Function	function that interacts with the Marlin Server Function in a CSP-T Server
serviceBaseCID	the part of the Content ID that is the same for all content in a service
Service Key	symmetric key defined in [IEC62455] that encrypts an ECM or a Program Key
Single Sign On	method of service access control that enables the user to authenticate once and gain access to the resources of multiple services

3.2.2 Abbreviations

In addition to the Abbreviations provided in Volume 1, the following abbreviations are used in this Volume.

Acronym	Definition
3GPP	Third Generation Partnership Project
AES	Advanced Encryption Standard
AKE	Authentication and Key Exchange
APDU	Application Protocol Data Unit
ATIS	Alliance for Telecommunications Industry Solutions
BBTS	Broadband Transport Stream – MPEG-2 transport stream as defined by [MRL BBTS]
BNS	Broadband Network Service
BSF	Bootstrapping Server Function
bslbf	bit string, left bit first
B-TID	Bootstrapping Transaction Identifier
CA	Conditional Access
CAD	Content Access Descriptor
CAM	Conditional Access Module
CAT	Conditional Access Table
CBC	Cipher-Block Chaining
CE-HTML	Consumer Electronics – HTML
CI	Common Interface
CSKMF	Content and Service Key Management Function
CSPG-CI+	CSPG – CI+ based
CSPG-DTCP	CSPG – DTCP-IP based
CSP-T	Content and Service Protection – Terminal-Centric Approach
DCF	DRM Content Format
DMZ	Dynamic Media Zones
DNR	Do Not Record
DNTS	Do Not Time Shift
DTCP	Digital Transmission Content Protection
DTLA	Digital Transmission Licensing Administrator

Acronym	Definition
DVB	Digital Video Broadcasting
ECM	Entitlement Control Message
EMM	Entitlement Management Message
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
HDCP	High-bandwidth Digital Content Protection
HDD	Hard Disk Drive
HNI-AMNI	Home Network Interface – Additional Managed Network Interface
HNI-CSP	Home Network Interface – Content and Service Protection
HNI-IGI	Home Network Interface – IMS Gateway Interface
HNI-INI	Home Network Interface – ITF (IPTV Terminal Function) Network Interface
ID	Identity
IDSA	IIF Default Scrambling Algorithm
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IIF	IPTV Interoperability Forum
IPMC	IP Multicast
IPMP	Intellectual Property Management Protocol
ISO	International Organization for Standardization
IV	Initialization Vector
KDF	Key Derivation Function
KSM	Key Stream Message
M-CID	Marlin Content ID
MIME	Multipurpose Internet Mail Extensions
MP4	MPEG-4
MPEG	Moving Pictures Experts Group
MS3	Marlin Simple Secure Streaming
NAF	Network Application Function
NPI	Network Provider Interface
OASIS	Organization for the Advancement of Structured Information Standards
PCMCIA	Personal Computer Memory Card International Association
PCP	Protected Content Packet
PDCF	Packetized DRM Content Format
PES	Packetized Elementary Stream
PID	Packet Identifier
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PMT	Program Map Table
SAML	Security Assertion Markup Language
SAS	Specific Application Support
SRM	System Renewability Message
TEK	Traffic Encryption Key
TISPAN	Telecoms & Internet converged Services & Protocols for Advanced Networks
TLS	Transport Layer Security
TLV	Type Length Value
TS	Transport Stream
TV	Television
uimsbf	unsigned integer most significant bit first
UNIS-CSP-G	User Network Interface Specific – Content and Service Protection Gateway

Acronym	Definition
UPnP	Universal Plug and Play
URI	Usage Rule Information

4 Content and Service Protection

This section specifies the Content and Service protection functionality.

It consists of a specification of

- the Terminal-Centric Approach, see section 4.1, and
- the Gateway-Centric Approach, see section 4.2.

4.1 Terminal-Centric Approach

This section specifies the functionality for the OIPF Terminal-Centric Approach to Content & Service Protection. In order to do this, this section provides a mapping from all relevant functions and interfaces from [OIPF_ARCH2] to specific sections of Marlin specifications [MRL BNSP] and [MRL MS3].

All normative statements in this section and its sub-sections apply only in case the Terminal-Centric Approach is supported by the OITF.

OITFs that support the OIPF Terminal-Centric Approach to Content & Service Protection **SHALL** be compliant with [MRL BNSP] and **MAY** be compliant with [MRL MS3].

NOTE: The criteria that determine under which circumstances the Terminal-Centric Approach is implemented are out of the scope of the present document.

NOTE: The criteria that determine under which circumstances the support for Marlin metering for content or rights owner settlement is implemented in the OITF are out of the scope of the present document.

4.1.1 Interfaces for CSP and CSP-T Server

This section describes the interfaces related to a CSP and CSP-T Server in the Functional Architecture described in [OIPF_ARCH2].

4.1.1.1 Scope

The main scope of this section is to describe CSP interfaces (CSP-1, UNIS-CSP-T) and CSP-T Server interfaces (NPI-CSPT_x, $x = 1, 2, 3$). CSP-1 is the interface between CSP and OITF Functions. NPI-CSPT_x, $x = 1, 2, 3$, are the interfaces between the CSP-T Server and Providers Network Functions. This section informatively touches upon the Marlin License Evaluation and Content Encryption.

Only the UNIS-CSP-T interface and the interface to DAE in CSP-1 are normative. The other interfaces are informatively described for comprehension.

Figure 1 shows the message flow overview.

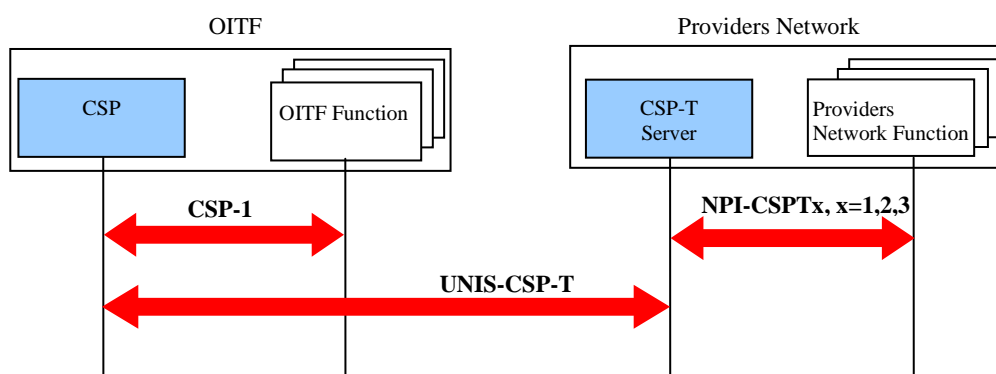


Figure 1: CSP-T System Overview

The four functional entities in Figure 1 are described below:

- CSP in this document consists of Marlin Client Function and a part of the Client Function which deals with Marlin elements.

- CSP-T Server in this document consists of Marlin Server Function and a part of the Server Function which deals with Marlin elements.
- OITF Function is the function in the OITF that interacts with the CSP. The OITF Function also interacts with a Providers Network Function to acquire the necessary information for the CSP. How the Providers Network Function is called in this document depends on the process to be performed.
- Providers Network Function is the function in the Providers Network that interacts with the CSP-T Server. How the Providers Network Function is called in this document depends on the process to be performed.

Note that the OITF Function with which the CSP communicates is not limited as described in this document and may vary depending on the implementation of the OITF.

4.1.1.2 Interface CSP - CSP-T Server (UNIS-CSP-T)

When requested from a native application or from the DAE application to handle a Marlin Action Token or a MIPPVControlMessage, see section 4.1.7.3, the CSP SHALL act as a Marlin DRM Client and SHALL perform Marlin Protocols as specified in [MRL BNSP] or [MRL MS3] as applicable. Furthermore, in the context of [MRL BNSP], if there are no available rights when trying to use content, the CSP SHALL comply with [MRL BBTS] and [OMARLIN] and SHALL try to use the URL specified in the content to acquire new rights.

In both cases, the CSP-T Server SHALL comply with Marlin Protocols as specified in [MRL BNSP].

These protocols are:

- Marlin registration: Node acquisition and Link acquisition.
- Marlin de-registration.
- Marlin License acquisition.

If Marlin Simple Secure Streaming feature is supported, the CSP-T Server MAY comply with MS3 Protocols as specified in [MRL MS3]

- MS3 Protocol.

Marlin Protocols are described in section 4.1.1.4.

4.1.1.3 Interface CSP - OITF Function (CSP-1)

The DAE DRM Agent API, as defined in [OIPF_DAE2] section 7.6.1, triggers handling of DRM Message, e.g. Marlin Action Token, MIPPVControlMessage or Marlin License. When the sendDRMMessage API is called for a DRMSystemID set to the value defined for Marlin, OITF SHALL forward the DRM Message to the CSP function. The result of calling sendDRMMessage is notified through the onDRMMessageResult event handler.

Typical DRM events SHALL be triggered by CSP to DAE via A/V or video/broadcast object when content cannot be played, recorded or time shifted, due to a lack of rights (no license, invalid license) or parental control locking. These events are defined in [OIPF_DAE2] sections 7.13.6 and 7.14.7. The DRMSystemID of these events SHALL be set to the value defined for Marlin.

A DAE application or native application MAY use DRMControlInformation, defined as an extension to PurchaseItem in [OIPF_META2], present in the BCG and SD&S retrieved by the metadata client. SilentRightsURL, PreviewRightsURL and RightsIssuerURL in DRMControlInformation MAY be used to get updated rights. If the DRMSystemID in DRMControlInformation is set to the value defined for Marlin, the application SHALL forward the DRMPrivateData, if present, to the CSP. A DAE application SHALL use sendDRMMessage, defined in [OIPF_DAE2] section 7.6.1.2, to forward the DRMPrivateData.

All objects defined in [OIPF_DAE2] that are requested to handle a content-access descriptor, defined in [OIPF_DAE2] SHALL check if the content-access descriptor includes DRMControlInformation. These objects or the underlying functions SHALL forward the available DRMPrivateData in the DRMControlInformation to the CSP if the DRMSystemID is set to the value defined for Marlin.

NOTE: The DRMSystemID is defined in section 4.1.7.1 for Marlin.

4.1.1.4 Marlin Protocol Sequences (informative)

4.1.1.4.1 Marlin Registration (informative)

Marlin Registration provides functions which enable a Marlin Client Function in CSP to register to a Marlin domain. Marlin Registration consists of Node Acquisition and Link Acquisition.

4.1.1.4.1.1 Node Acquisition (informative)

Node Acquisition provides an Octopus Node Object from a Marlin Server Function in CSP-T Server to a Marlin Client Function in CSP.

Note that Node Acquisition is performed prior to the respective Link Acquisition to provide the Octopus Node Objects necessary for the Link Acquisition.

Marlin Node Acquisition Protocol is triggered by a Marlin Action Token for Node Acquisition (hereafter Node Acquisition Action Token) from CSP. The OITF Function acquires the Node Acquisition Action Token and then the OITF Function feeds it to CSP. After CSP acquires corresponding Marlin Configuration Token from CSP-T Server, CSP executes Marlin Node Acquisition Protocol with CSP-T Server. Note that Marlin Node Acquisition Protocol is to provide one Octopus Node Object per its request and response.

The message flow in case of Node Acquisition is shown in Figure 2.

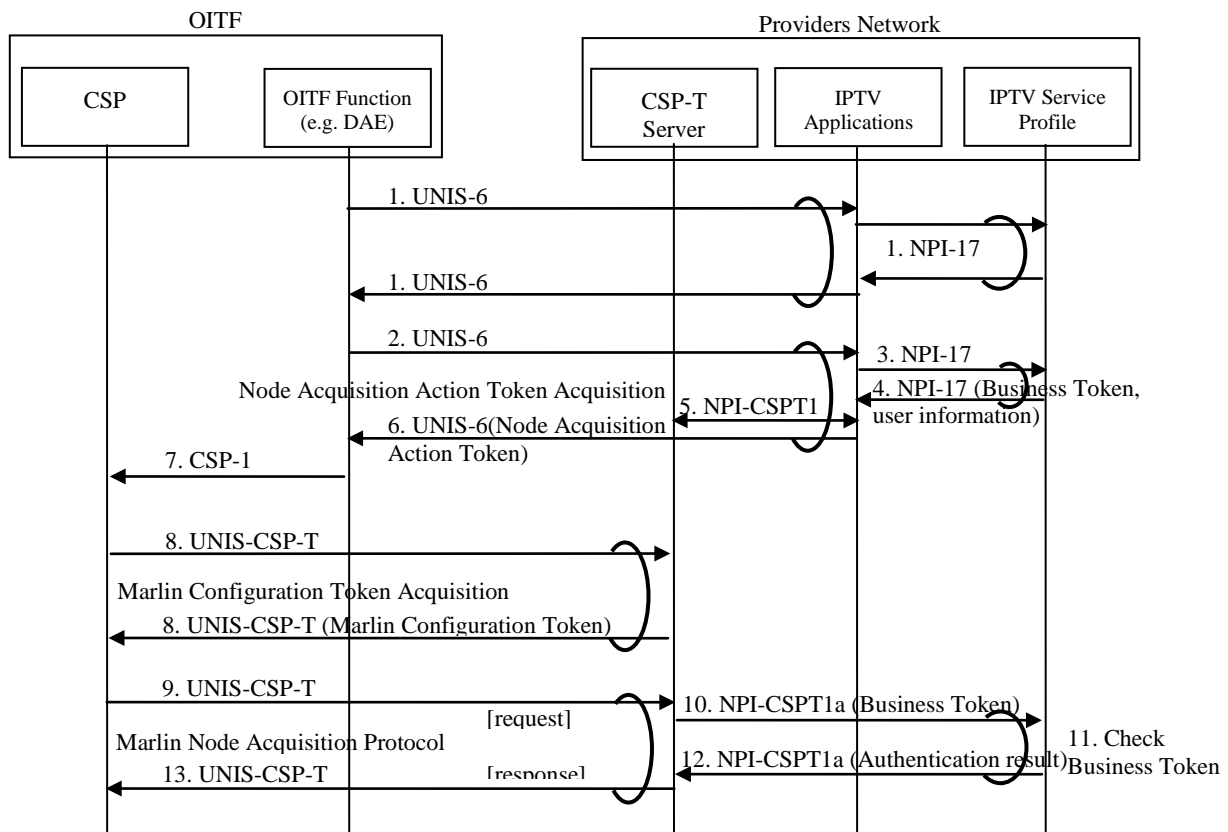


Figure 2: Node Acquisition Sequence

In Node Acquisition Sequence, the following steps are performed:

1. The OITF Function (e.g. DAE) communicates with IPTV Applications and IPTV Service Profile function via UNIS-6 and NPI-17 for Node Acquisition¹.

¹ Note that, although NPI-17 is assumed as the interface for communication between IPTV Applications and IPTV Service Profile, in the case of the managed network model, NPI-2 and NPI-6 may be used instead.

2. Given the Marlin Action Token URL (e.g. embedded into the webpage obtained in step 1), the OITF Function (e.g. DAE application) sends the request for the Node Acquisition Action Token to the IPTV Applications by UNIS-6.
3. When receiving the request from the OITF Function, the IPTV Applications sends a request to the IPTV Service Profile function via NPI-17 to get the necessary information to generate the Node Acquisition Action Token.
4. Receiving the request from IPTV Applications, the IPTV Service Profile function sends Business Token and user information to IPTV Applications.
5. Given the information from the IPTV Applications, when there is no Octopus Node for the given user information, the CSP-T Server generates Octopus Node and correlates user information with the Octopus Node, so that CSP-T Server can check for the existence of the Octopus Node next time from the user information. Then the CSP-T Server correlates the Business Token with Octopus Node so that the CSP-T Server can provide the corresponding Octopus Node from the Business Token included in the (Marlin Node Acquisition Protocol) request.
6. IPTV Applications sends the Node Acquisition Action Token to the OITF Function by UNIS-6.
7. The OITF Function sends the Node Acquisition Action Token to the CSP by CSP-1.
8. When the CSP does not have a corresponding Marlin Configuration Token, the CSP gets the Marlin Configuration Token from the CSP-T Server by referring to the URL specified in the Node Acquisition Action Token.
9. Given the Node Acquisition Action Token, the CSP sends a (Marlin Node Acquisition Protocol) request to CSP-T Server by UNIS-CSP-T.
10. To check the request from the CSP, the CSP-T Server sends the Business Token (and possibly other client data such as client version, model, etc... extracted from the request) to the IPTV Service Profile function.
11. The IPTV Service Profile function validates the data received from the CSP-T Server.
12. If validation succeeds, the IPTV Service returns to CSP-T Server the data necessary to fulfil the CSP request. If validation fails, an error is returned to the CSP-T Server.
13. The CSP-T Server sends a Marlin (Node Acquisition) response message to the CSP. This response includes either the Octopus Node correlated to the Business Token sent in the original CSP request, or an error message as defined in [MRL BNSP].

4.1.1.4.1.2 Link Acquisition (informative)

Link Acquisition provides an Octopus Link from Marlin Server Function in CSP-T Server to Marlin Client Function in CSP.

Note that this sequence assumes that the corresponding Node Acquisition has already been performed between the CSP and CSP-T Server.

Marlin Link Acquisition Protocol is triggered by a Marlin Action Token for Link Acquisition (hereafter Link Acquisition Action Token) from CSP. The OITF Function acquires the Link Acquisition Action Token, and then the OITF Function feeds it to CSP. After CSP acquires corresponding Marlin Configuration Token from CSP-T Server, CSP executes Marlin Link Acquisition Protocol with CSP-T Server.

The message flow in case of Link Acquisition is shown in Figure 3.

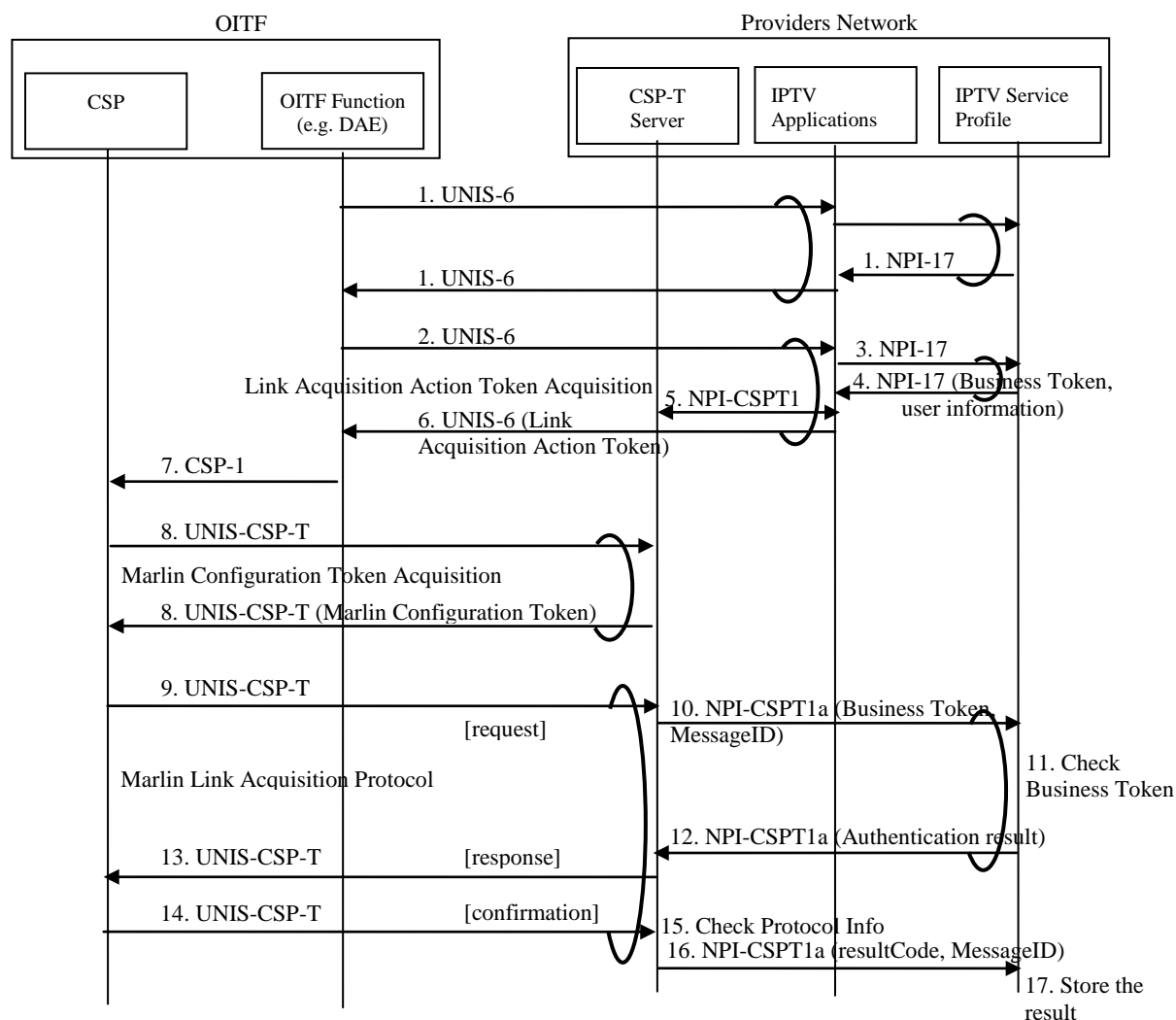


Figure 3: Link Acquisition Sequence

In Link Acquisition Sequence, the following steps are performed:

1. The OITF function (e.g. DAE) communicates with IPTV Applications and IPTV Service Profile function via UNIS-6 and NPI-17 for Link Acquisition².

² Note that, although NPI-17 is assumed as the interface for communication between IPTV Applications and IPTV Service Profile, in the case of the managed network model, NPI-2 and NPI-6 may be used instead.

2. Given the Marlin Action Token URL (e.g. embedded into the webpage obtained in step 1), the OITF Function (e.g. DAE application) sends the request for the Link Acquisition Action Token to IPTV Applications by UNIS-6.
3. When receiving the request from the OITF Function, the IPTV Applications sends a request to the IPTV Service Profile function by NPI-17 to get necessary information to generate the Link Acquisition Action Token,
4. Receiving the request from IPTV Applications, the IPTV Service Profile function sends Business Token and user information to IPTV Applications.
5. Given the user information from the IPTV Applications, the CSP-T Server finds the information of Octopus Node which corresponds to “From Node” and “To Node”. Then the CSP-T Server correlates the Business Token with “From Node” and “To Node” so that the CSP-T Server can check the information in (Marlin Link Acquisition Protocol) request.
6. IPTV Applications sends the Link Acquisition Action Token to the OITF Function by UNIS-6.
7. Given the Link Acquisition Action Token, the OITF Function sends it to the CSP by CSP-1.
8. When the CSP does not have a corresponding Marlin Configuration Token, the CSP gets the Marlin Configuration Token from the CSP-T Server by referring to the URL specified in the Link Acquisition Action Token by UNIS-CSP-T.
9. Given the Link Acquisition Action Token, the CSP sends a (Marlin Link Acquisition Protocol) request to the CSP-T Server.
10. To check the request from the CSP, when the request includes the correct combination of Business Token, “From Node”, and “To Node”, the CSP-T Server sends a Business Token and MessageID to the IPTV Service Profile function by NPI-CSPT1a. The MessageID is a unique id, and the same MessageID is set among request, response, and confirmation, so that IPTV Service Profile function can use the MessageID to correlate request, response, and confirmation.
11. The IPTV Service Profile function validates the data received from the CSP-T Server.
12. If validation succeeds, the IPTV Service Profile function returns to CSP-T Server the data necessary to fulfil the CSP request. If validation fails, an error is returned to the CSP-T Server.
13. The CSP-T Server sends a Marlin (Registration) response message to the CSP. This response includes either the registration agent correlated to the Business Token sent in the original CSP request, or a fault message as defined in [MRL BNSP].
14. The CSP sends a (Marlin Link Acquisition Protocol) Confirmation to the CSP-T Server by UNIS-CSP-T.
15. The CSP-T Server checks the resultCode (i.e. success or failure for registration in CSP), and then stores the “From Node” and “To Node” information by correlating with the user information so that CSP-T Server can manage Marlin domain information for the user.
16. The CSP-T Server sends the resultCode and the MessageID to the IPTV Service Profile function by NPI-CSPT1a.
17. The IPTV Service Profile function stores the resultCode in connection with the user information from step 4.

4.1.1.4.2 Marlin Deregistration (informative)

Marlin Deregistration provides functions which enable Marlin Client Function in CSP to deregister from a Marlin domain.

Note that this sequence assumes that the corresponding Node Acquisition and Link Acquisition have already been performed between the CSP and CSP-T Server.

Marlin Deregistration Protocol is triggered by a Marlin Action Token for Deregistration (hereafter Deregistration Action Token) from CSP. The OITF Function acquires the Deregistration Action Token, and then the OITF Function feeds it to CSP. After CSP acquires corresponding Marlin Configuration Token from CSP-T Server, CSP executes Marlin Deregistration Protocol with CSP-T Server.

The sequence of deregistration messages is shown in Figure 4.

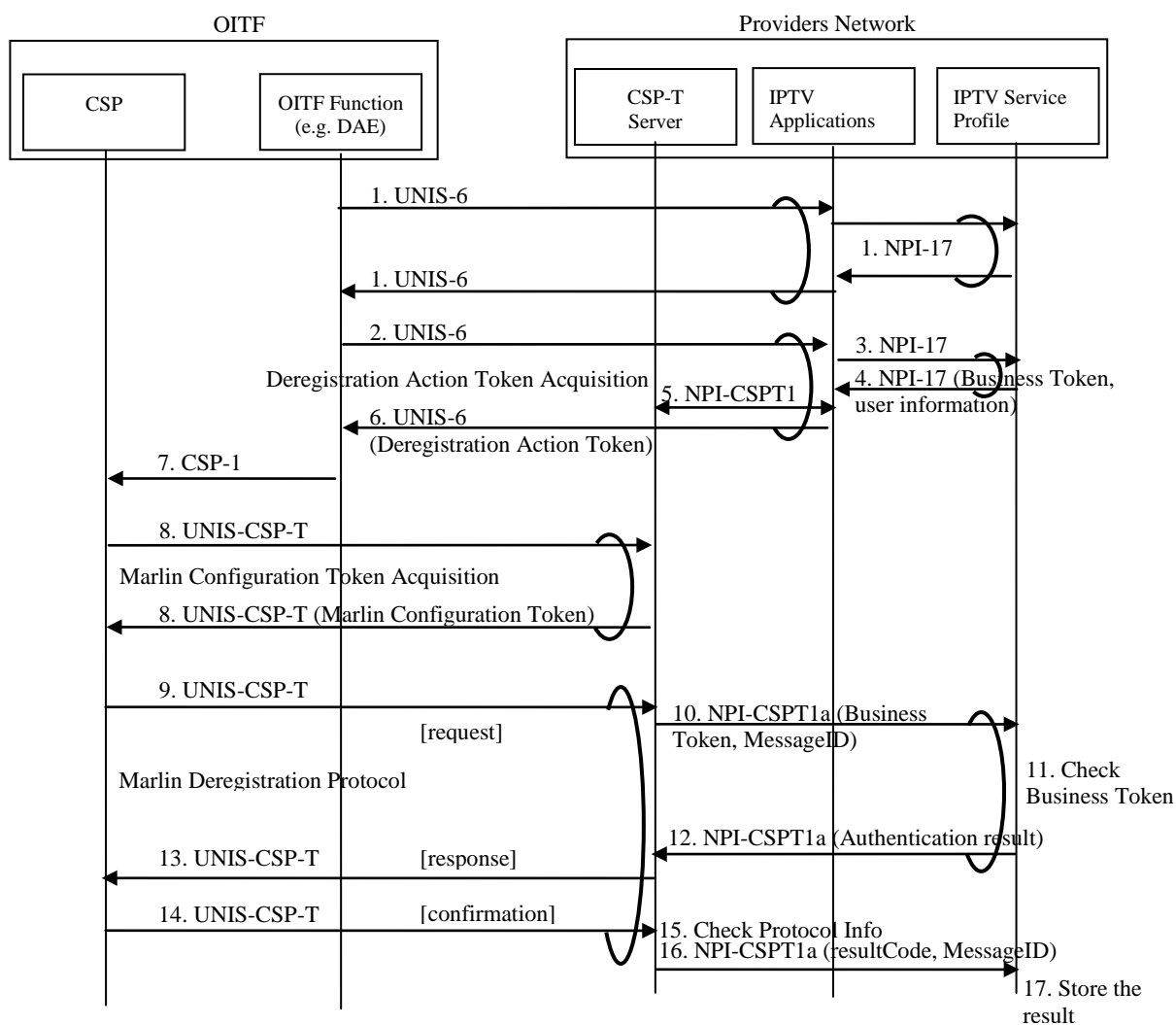


Figure 4: Deregistration Sequence

In this deregistration sequence, the following steps are performed:

1. The OITF function (e.g. DAE) communicates with IPTV Applications and IPTV Service Profile function via UNIS-6 and NPI-17 for Marlin Deregistration³.

³ Note that, although NPI-17 is assumed as the interface for communication between IPTV Applications and IPTV Service Profile, in the case of the managed network model, NPI-2 and NPI-6 may be used instead.

2. Given the Marlin Action Token URL (e.g. embedded into the web page obtained in step 1), the OITF Function (e.g. DAE application) sends the request for the Deregistration Action Token to the IPTV Applications by UNIS-6.
3. When receiving the request from the OITF Function, the IPTV Applications sends a request to the IPTV Service Profile function by NPI-17 to get necessary information to generate the Deregistration Action Token.
4. Receiving the request from IPTV Applications, the IPTV Service Profile function sends Business Token and user information to IPTV Applications.
5. Given the user information from the IPTV Applications, the CSP-T Server finds the information of Octopus Node which corresponds to "From Node" and "To Node". Then the CSP-T Server correlates the Business Token with "From Node" and "To Node" so that the CSP-T Server can check the information in (Marlin Deregistration Protocol) request.
6. IPTV Applications sends the Deregistration Action Token to the OITF Function by UNIS-6.
7. Given the Deregistration Action Token, the OITF Function sends it to the CSP by CSP-1.
8. When the CSP does not have a corresponding Marlin Configuration Token, the CSP gets the Marlin Configuration Token from the CSP-T Server by referring to the URL specified in the Deregistration Action Token.
9. Given the Deregistration Action Token, the CSP sends a (Marlin Deregistration Protocol) request to the CSP-T Server by UNIS-CSP-T.
10. To check the request from the CSP by the IPTV Service Profile function, when the request includes the correct combination of Business Token, "From Node", and "To Node", the CSP-T Server sends a Business Token and MessageID to the IPTV Service Profile function by NPI-CSPT1a. The MessageID is a unique id and the same MessageID is set among request, response, and confirmation so that IPTV Service Profile function can use the MessageID to correlate request, response, and confirmation.
11. The IPTV Service Profile function validates the data received from the CSP-T Server.
12. If validation succeeds, the IPTV Service Profile function returns to CSP-T Server the data necessary to fulfil the CSP request. If validation fails, an error is returned to the CSP-T Server.
13. The CSP-T Server sends a Marlin (Deregistration) response message to the CSP. This response includes either the deregistration agent correlated to the Business Token sent in the original CSP request, or an error message as defined in [MRL BNSP].
14. The CSP sends a (Marlin Deregistration Protocol) Confirmation to the CSP-T Server by UNIS-CSP-T.
15. The CSP-T Server checks the resultCode (i.e. success or failure for deregistration in CSP) and Message ID, and then stores the "From Node" and "To Node" information by correlating it with the user information, so that CSP-T Server can manage Marlin domain information for the user.
16. The CSP-T Server sends the resultCode and the MessageID to the IPTV Service Profile function by NPI-CSPT1a.
17. The IPTV Service Profile function stores the resultCode in connection with the user information from step 4.

4.1.1.4.3 Marlin License Acquisition (informative)

License Acquisition provides functions which enable Marlin Client Function in CSP to obtain a Marlin License.

Marlin License Acquisition Protocol is triggered by a Marlin Action Token for License Acquisition (hereafter License Acquisition Action Token) from CSP. The OITF Function acquires the License Acquisition Action Token, and then the OITF Function feeds it to the CSP. After CSP acquires corresponding Marlin Configuration Token from CSP-T Server, CSP executes Marlin License Acquisition Protocol with CSP-T Server.

The sequence of license acquisition messages is shown in Figure 5.

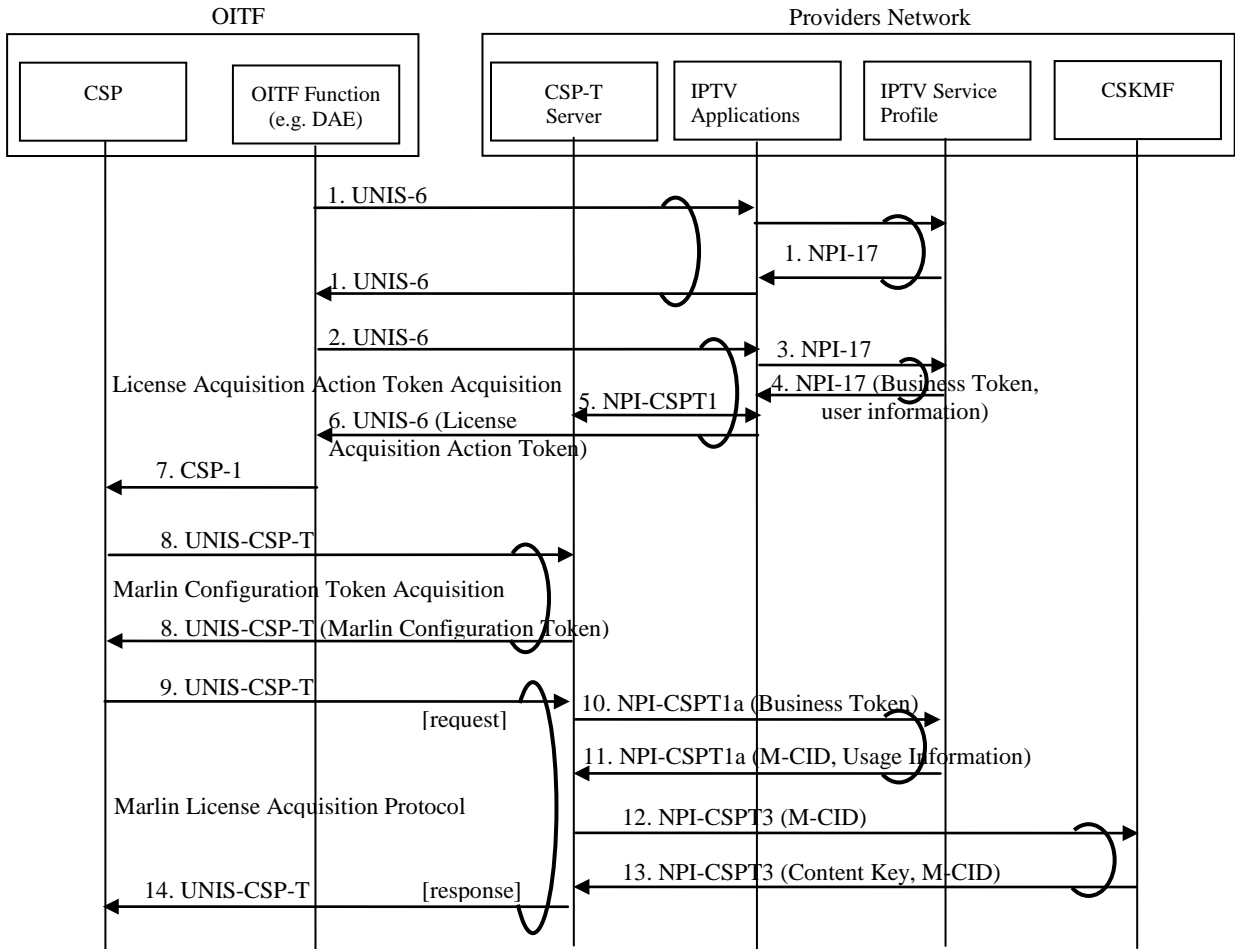


Figure 5: License Acquisition Sequence

In this sequence, the following steps are performed:

1. The OITF Function (e.g. DAE) communicates with IPTV Applications and IPTV Service Profile function via UNIS-6 and NPI-17 for License Acquisition⁴.
2. Given the Marlin Action Token URL (e.g. embedded into the webpage obtained in step1), the OITF Function (e.g. DAE application) sends the request for the License Acquisition Action Token to the IPTV Applications by UNIS-6.
3. When receiving the request from the OITF Function, the IPTV Applications sends a request to the IPTV Service Profile function by NPI-17 to get the information necessary to generate the License Acquisition Action Token.
4. Receiving the request from IPTV Applications, the IPTV Service Profile function sends Business Token and user information to IPTV Applications. This user information for License Acquisition also indicates “Bound to Node” of the Marlin License.

⁴ Note that, although NPI-17 is assumed as the interface for communication between IPTV Applications and IPTV Service Profile, in the case of the managed network model, NPI-2 and NPI-6 may be used instead.

5. Given the information from the IPTV Applications, the CSP-T Server correlates the Business Token with the “Bound to Node” so that the CSP-T Server can check the information in a (Marlin License Acquisition Protocol) request.
6. IPTV Applications sends the License Acquisition Action Token to the OITF Function by UNIS-6.
7. Given the License Acquisition Action Token, the OITF Function sends it to the CSP by CSP-1.
8. When the CSP does not have a corresponding Marlin Configuration Token, the CSP obtains the Marlin Configuration Token from the CSP-T Server by referring to the URL specified in the License Acquisition Action Token.
9. Given the License Acquisition Action Token, the CSP sends a (Marlin License Acquisition Protocol) request to the CSP-T Server by UNIS-CSP-T.
10. To check the request from the CSP, when the request includes the correct combination of Business Token and “Bound to Node”, the CSP-T Server sends a Business Token to the IPTV Service Profile function by NPI-CSPT1a.
11. The IPTV Service Profile function validates the data received from the CSP-T Server. If validation fails, an error is returned to the CSP-T Server. If validation succeeds, the IPTV Service Profile function returns to CSP-T Server the data necessary to generate the Marlin License, consisting at a minimum of:
 - M-CID (Marlin Content ID)
 - Usage Information which includes the content usage rules
12. To get the corresponding Content Key, the CSP-T Server sends the M-CID to the CSKMF by NPI-CSPT3.
13. When receiving the information, the CSKMF looks for the corresponding Content Key (*) by M-CID, and then sends the Content Key (*) and M-CID to the CSP-T Server by NPI-CSPT3.
14. The CSP-T Server sends a Marlin (License) response message to the CSP. This response includes either the License correlated to the Business Token sent in the original CSP request, or an error message as defined in [MRL BNSP].

* When the content is protected by Scramble Key and Service Key (or Program Key), the Service Key (or Program Key) is provided from CSKMF to CSP-T Server instead of Content Key. See section 4.1.3, for a brief explanation of such encryption scheme.

4.1.2 Protected Content Usages

Protected content usages include: playback, recording, time shifting.

Protected content can be played from a native application or from a DAE application using A/V plug-in or video/broadcast object as defined in [OIPF_DAE2]

Protected content can be time-shifted from a native application or from a DAE application using video/broadcast object as defined in [OIPF_DAE2]

Protected content can be recorded from a native application or from a DAE application video/broadcast object as defined in [OIPF_DAE2]

CSP SHALL control protected content usages as defined in [MRL BNSP] for License evaluation, [MRL MS3] for evaluation of Stream Access Statements, and [MRL BBTS] for ECM control. See also section 4.1.3 for an overview.

If usages are not allowed, CSP SHALL block the consumption of program (i.e. stop descrambling) and SHALL generate the appropriate event (no rights, parental control locking) to the calling application, i.e. native application or DAE object. The DAE AV or video/broadcast object SHALL trigger the event to the calling DAE application as specified in 4.1.1.3.

For MPEG-2 TS, usages SHALL be controlled at each ECM change. For other file formats, usages are controlled only when requesting the usage.

4.1.2.1 Marlin License Evaluation (informative)

4.1.2.1.1 License Evaluation (for (P)DCF [OMARLIN] or Marlin IPMP [MRL FF]) (informative)

This section describes the informative overview of how Marlin data objects acquired via Marlin Protocols are used for consumption of protected contents, such as rendering or exporting.

Figure 6 shows the message flow of License Evaluation.

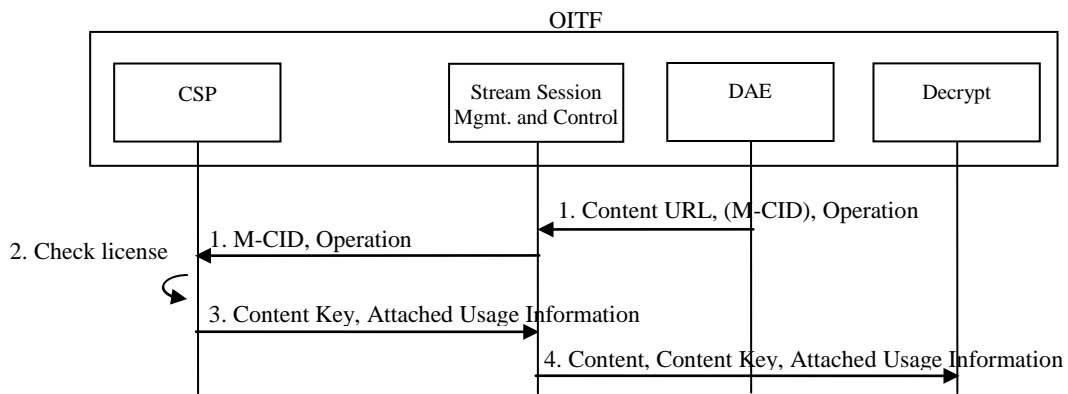


Figure 6: License Evaluation Sequence

In order to gain access to a protected content, steps below are performed:

1. OITF Function such as DAE⁵ triggers the evaluation of a corresponding Marlin License at CSP via Stream Session Management and Control by providing following information:
 - Content URL: the protected content to be accessed.
 - Optionally, M-CID (Marlin Content ID): Id of the protected content to be accessed. The M-CID can also be retrieved from the content, ContentAccessDescriptor [OIPF_DAE2], BCG, or SD&S [OIPF_META2].
 - Operation: operation to perform with the protected content (e.g., render, export).
2. The Marlin Client Function in CSP is required to check the following:
 - The PKI signatures on the Marlin data objects related to the protected content are validated. For trust management of Marlin, see section 9 of [MRL CORE].
 - The usage rule specified in the Marlin data objects for the protected content is valid for CSP.
3. If the license evaluation succeeds, the CSP returns the corresponding Content Key and attached usage information such as Output Control Information (if any) to the Stream Session Management and Control. Otherwise, the CSP responds with an error.
4. The Stream Session Management sends the received Content, Content Key and attached usage information, to the Decrypt function.

⁵ Note that, although DAE is used as a function to trigger the License Evaluation, this is only for illustrative purposes and other OITF function can be used, such as OITF embedded application depending on the design of the OITF.

4.1.2.1.2 License Evaluation (for MPEG-2 Transport Stream) (informative)

Figure 7 shows the message flow of License Evaluation with Scramble Key Decryption.

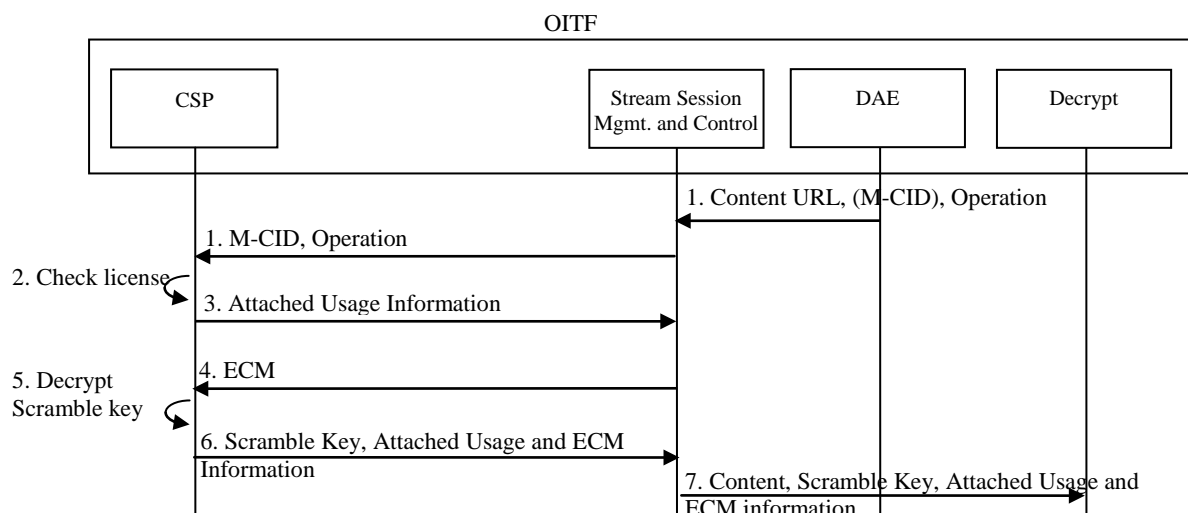


Figure 7: Scramble Key Decryption Sequence

When the content is encrypted by Scramble Key, License Evaluation and Scramble Key Decryption sequence below is followed:

1. OITF Function such as DAE⁶ triggers the evaluation of a corresponding Marlin License at CSP via Stream Session Management and Control by providing following information:
 - Content URL: the protected content to be accessed (Local or remote URL).
 - Optionally, M-CID (Marlin Content ID): Id of the protected content to be accessed. The M-CID can also be retrieved from the content, ContentAccessDescriptor [OIPF_DAE2], BCG, or SD&S [OIPF_META2].
 - Operation: operation to perform with the protected content (e.g., render, export).
2. The Marlin Client Function in CSP is required to check the following:
 - The PKI signatures on the Marlin data objects related to the protected content are validated (signature O.K. and certificate chain is successfully chained up to the Marlin Trust Anchors).
 - The usage rule specified in the Marlin data objects for the protected content is valid for CSP.
3. If the license evaluation succeeds, the CSP returns attached usage information such as Output Control Information (if any) to the Stream Session Management and Control. Otherwise, the CSP responds with an error.
4. The Stream Session Management and Control provides an ECM to the CSP. The ECM includes a Scramble Key encrypted by a Service or Program key and attached ECM information including the Encryption Algorithm Type, Parental Control Information, recording control Information and Output Control Information.
5. The CSP checks the ECM on integrity. If this is OK, the CSP decrypts encrypted Scramble Key with the appropriate key, and, based on the combined Output Control Information in the ECM and license, the CSP determines updated Output Control Information as specified in [MRL BBTS].
6. The CSP sends Scramble Key and attached usage and ECM information to the Stream Session Management and Control.
7. The Stream Session Management sends the received Content, Scramble Key and attached usage and ECM information to the Decrypt.

Note that the sequence assumes that Stream Session Management and Control is trusted by the CSP and that the Scramble Key, permission to perform the requested operation and attached usage information are transferred over a secure channel.

⁶ Note that, although DAE is used as a function to trigger the License Evaluation, this is only for illustrative purposes and can be performed by another OITF function, such as OITF embedded application, depending on the design of the OITF.

4.1.3 Content Encryption (informative)

This section contains an informative overview of Content Encryption to clarify sequences related to Content Key, Service or Program key, and Scramble Key in section 4.1.1.4.3 and 4.1.2.1.

Figure 8, Figure 9 and Figure 10 show the message flows of Content Encryption.

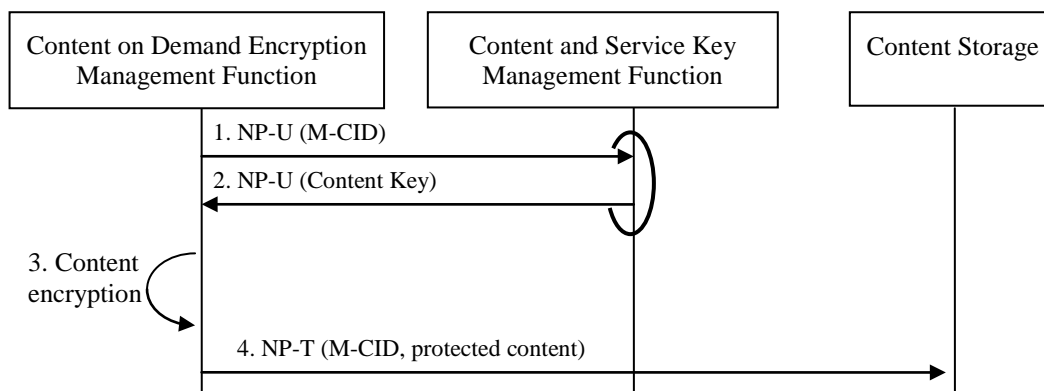


Figure 8: Content on Demand Encryption Sequence using Content Key (for (P)DCF [OMARLIN] or Marlin IPMP [MRL FF])

When (P)DCF [OMARLIN] or Marlin IPMP [MRL FF] Content On Demand is encrypted using Content Key, the following steps are performed in Content Encryption sequence:

1. The Content on Demand Management requests for a content specified by M-CID (Marlin Content ID), the Content Key to use.
2. The Content and Service Key Management function returns the Content Key.
3. The Content on Demand Encryption Management Function launches encryption of the content in the clear using the M-CID (Marlin Content ID) and Content Key. The protected content is generated.
4. The Content on Demand Management stores the protected content in the Content Storage.

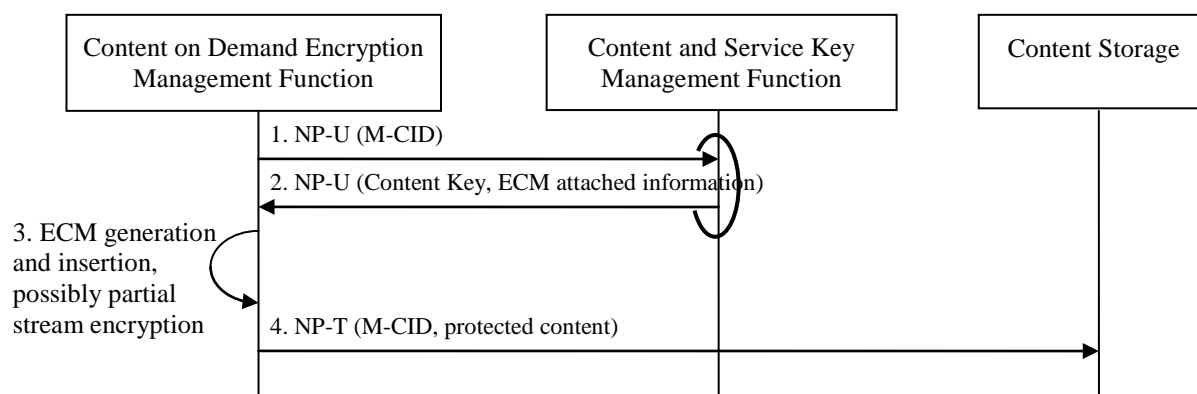


Figure 9: Content on Demand Encryption Sequence using Content Key (for MPEG-2 TS)

When MPEG-2 TS Content on Demand is encrypted using Content Key, the following steps are performed in Content Encryption Sequence:

1. The Content on Demand Management requests for the content item specified by M-CID (Marlin Content ID), the Content Key and ECM attached information, including the Encryption Algorithm, Parental Control Information and Output Control Information, to use.
2. The Content and Service Key Management function returns the Content Key and ECM attached information.
3. The Content on Demand Encryption Management Function launches encryption of the content in the clear. Scramble Keys are generated and the content is encrypted using these Scramble Keys. The Scramble Keys are encrypted using the Content Key. ECMs that include Scramble Keys and provided ECM attached information are inserted into the protected content.
4. The Content on Demand Encryption Management Function stores the protected content in the Content Storage.

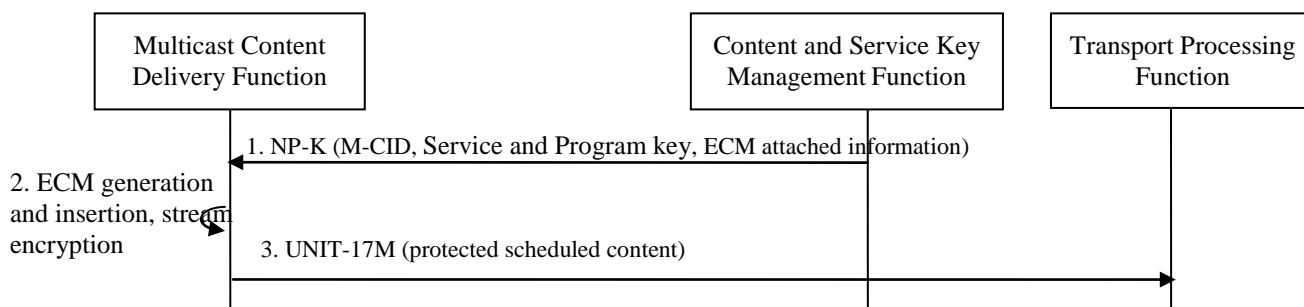


Figure 10: Scheduled Content Encryption Sequence using Scramble Key (for MPEG-2 TS)

When MPEG-2 TS Scheduled Content is encrypted by Scramble Keys then the Scramble Keys are encrypted by a Service or Program key, the following steps are performed in Content Encryption Sequence:

1. The Content and Service Key Management function sends the M-CID (Marlin Content ID), Service key and possibly a Program key, ECM attached information including Encryption Algorithm, Parental Control Information and Output Control Information to the Multicast Content Delivery Function.
2. The Multicast Content Delivery Function generates Scramble Keys and then encrypts clear content using these Scramble Keys. Then the Multicast Content Delivery Function encrypts the Scramble Keys using the Service Key or Program Key. When the Program Key is used for encryption of Scramble Keys, the Multicast Content Delivery Function encrypts the Program Key using the Service Key. An ECM that includes the encrypted Scramble Keys and provided ECM attached information is inserted into the protected content.
3. Protected scheduled content is sent to the Transport Processing Function through UNIT-17M.

4.1.4 Protected File Formats

The protected file formats supported in the present specification are:

- the MP4 file format as defined in section 4.2 of [OIPF_MEDIA2]
 - encrypted according to the OMA (P)DCF file formats, including Marlin specific extensions in an OMA compatible way, as defined in section 4 of [OMARLIN],
 - encrypted according to the Marlin IPMP file format as specified in section 2.3 of [MRL FF],
 - encrypted as specified in [CENC],
- the MPEG-2 TS file format as specified in section 4.1.5.

NOTE: this section lists four different protected file formats supported by this specification. The criteria that determine under which circumstances which one or more of these is implemented are out of the scope of the present document.

4.1.5 Protection of MPEG-2 Transport Streams

If the OITF supports the unprotected MPEG-2 TS format, the OITF SHALL support the Marlin protected MPEG-2 TS format, as defined in this section and its sub-sections. Otherwise, the support of the Marlin protected MPEG-2 TS format as defined in this section and its sub-sections is OPTIONAL.

If the OITF supports the unprotected time stamped MPEG-2 TS format, the OITF SHALL support the Marlin protected time stamped MPEG-2 TS format, as defined in this section and its sub-sections. Otherwise, the support of the Marlin protected time stamped MPEG-2 TS format as defined in this section and its sub-sections is OPTIONAL.

4.1.5.1 Context

Transport of conditional access messages in MPEG-2 TS is defined by DVB. CA_descriptors (Conditional access descriptor) are used to signal the presence of conditional access information in the stream. Conditional access messages are transported in short MPEG-2 TS private section (section_syntax_indicator = 0). Two types of messages are considered:

- ECM messages, which are linked to descrambling, access criteria and Control Words (TEK). These messages are signalled in the CA_descriptor in the PMT. ECM Messages should have a high repetition rate in order to allow quick programme access.

- EMM messages, which are linked to rights management. These messages are signalled in the CA_descriptor in the CAT. These messages' repetition rate should be set at head end level in order to comply with the operator QoS requirements.

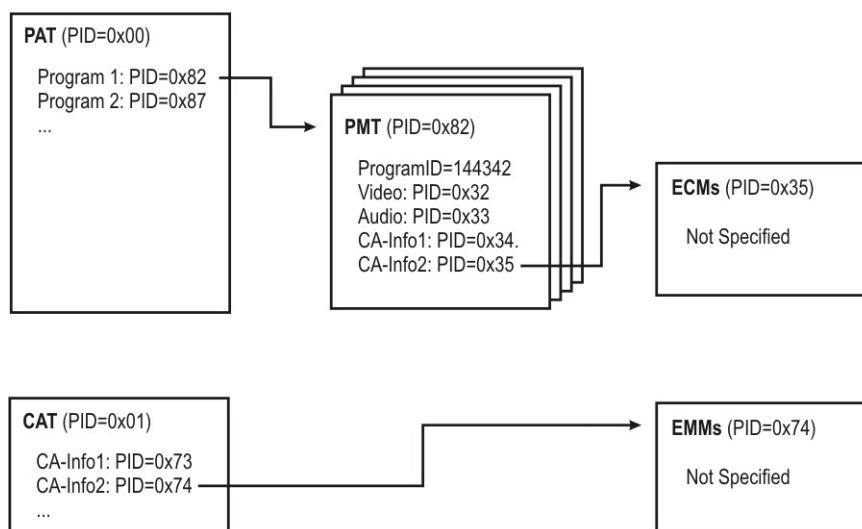


Figure 11: Conditional Access Descriptors Signalling ECM and EMM Messages

This specification uses Marlin to protect MPEG-2 Transport Streams and time stamped Transport Streams as specified in [MRL BBTS]. If an OITF supports Marlin protected MPEG-2 TS, the OITF SHALL implement the functions of the DRM Client as specified in [MRL BBTS]. For Marlin protected MPEG-2 TS the Content Delivery function SHALL deliver Transport Streams or Time-stamped Transport Streams that are formatted as specified in [MRL BBTS].

If an OITF supports Marlin protected MPEG-2 TS, the OITF SHALL support the Parental_rating access_criteria_descriptor as specified in [IEC62455], the recording control access_criteria_descriptor as specified in section 4.1.5.2 and SHALL support at least the rating_type 0 in these criteria, which maps to the parental rating system in DVB Systems [DVB-SI].

For the recording control, refer also to section 4.1.5.2, the OITF SHALL compare the required operation with the allowed operations (PVR and time shifting) in the recording control criteria and refuse the requested operation to the calling application (native or DAE) if the requested operation is not allowed.

For the parental rating control, the OITF SHALL compare the program's rating from the parental rating access_criteria_descriptor with the current parental rating criterion set in the OITF by the application (either native application or DAE) and SHALL block the consumption of programme (i.e. stop descrambling), if the parental rating system is supported by the OITF and the programme's rating does not meet the parental rating criterion (e.g. rating is at or above a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age). The OITF SHALL raise an event to the application controlling the playback or other operation, whenever a parental rating is discovered for the A/V content that does not meet the parental rating criterion that is set for the parental system in use, which has lead to blocking of the consumption of the content. The event SHALL provide the programme's rating. In case the application is a native application and if the MPEG-2 TS stream provides a Parental Control URL, as defined in section 0, the native application SHOULD launch the DAE with the Parental Control URL for management of parental control. In case the application is a DAE application, the event is called onParentalRatingChange and is defined in sections 7.13.5 and 7.14.6 of [OIPF_DAE2].

If the OITF does not support the particular parental rating system used in the program, the OITF SHALL raise an event to the application controlling the playback or other operation. The event SHALL provide the programme's rating. In case the application is a DAE application, the event is called onParentalRatingError and is defined in sections 7.13.5 and 7.14.6 of [OIPF_DAE2]. The event MAY be managed via the DAE application (see section 4.5 of [OIPF_DAE2] for more information). In case the application is a native application, the event is managed through an OITF vendor dependent user interface. In both cases, consumption MAY be unblocked by setting a new parental rating threshold. This threshold setting is usually restricted to privileged users, e.g. parents and a successful PIN input by a user may be used to control the parental rating threshold setting. The OITF SHOULD continue monitoring the MPEG-2 TS, taking into account parental rating criteria changes in ECM streams or new settings for the parental rating threshold in the OITF, and SHALL unblock consumption (i.e. re-start descrambling) if the current programme's rating becomes lower than the current parental rating threshold.

When no valid rights are available for the MPEG-2 TS, the OITF SHALL block the consumption of the programme (i.e. stop descrambling) and SHALL raise an event to the application controlling the playback or other operation. In case the application is a DAE application, the event is called onDRMRightsError and is defined in sections 7.13.6 and 7.14.7 of [OIPF_DAE2]. The OITF SHOULD continue monitoring the MPEG-2 TS, taking into account criteria changes in ECM streams or rights changes in OITF and SHALL unblock consumption (i.e. re-starting descrambling) if there are valid rights for the requested operation.

For the avoidance of doubt, the OITF SHALL support the presence of descriptors (for a general description of descriptors, see [MPEG2TS] which are not defined in this specification) but SHALL ignore these descriptors. In particular, to allow DVB-SimulCrypt with other CA systems as defined in [DVB-SC] and gateway-centric approach, the presence of following descriptors SHALL be supported: CA descriptor for other CA systems than Marlin and than CA systems supported in a CSPG, scrambling descriptor [DVB-SI], and copyright descriptor [MPEG2TS].

4.1.5.2 Recording Control Access Criteria

This section defines an access_criteria_descriptor that MAY be present in the IEC62455 ECM as defined in [MRL BBTS].

Table 1: Recording Control access_criteria_descriptor

recording control information access_criteria_descriptor	Tag	Length (in bits)	Type
recording_control_information_byte	0x010	8	bslbf

Table 2: Bit Assignments of recording_control_information_byte

Bit #	7	6	5	4	3	2	1	0
Assignment	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	DNTS	DNR

The DNR (Do Not Record) bit signals that a BBTS is not allowed to be stored for PVR function. The OITF SHALL NOT store for PVR function the TS packets of a BBTS that are received after receipt of a BBTS packet carrying an IEC62455 ECM that includes a recording control access_criteria_descriptor in which the DNR bit is set to 1.

The OITF MAY store for PVR function the TS packets of a BBTS that are received after receipt of a BBTS packet carrying an IEC62455 ECM that does not include a recording control access_criteria_descriptor or does include a recording control access criteria in which the DNR bit is set to 0.

The DNTS (Do Not Time Shift) bit signals that a BBTS is not allowed to be stored for time shifting. The OITF SHALL NOT store for time shifting the TS packets of a BBTS that are received after receipt of a BBTS packet carrying an IEC62455 ECM that includes a recording control access_criteria_descriptor in which the DNTS bit is set to 1.

The OITF MAY store for time shifting the TS packets of a BBTS that are received after receipt of a BBTS packet carrying an IEC62455 ECM that does not include a recording control access_criteria_descriptor or does include a recording control access criteria in which the DNTS bit is set to 0.

The time shifting period SHALL not exceed 90 minutes in case the DNR bit is set to 1 and the DNTS bit is set to 0.

The combination of DNR equals 0 (PVR allowed) and DNTS equals 1 (time shift not allowed) SHOULD NOT be set.

For an overview of the combinations, see Table 3.

Table 3: DNR and DNTS Combinations

DNR	DNTS	Description
0	0	Time shifting allowed for infinite period; PVR allowed
0	1	SHOULD NOT occur
1	0	Time shift limited to 90 minutes; PVR NOT allowed
1	1	Time shift NOT allowed; PVR NOT allowed

For this version of the specification, the rsvd (reserved for future use) bits SHALL be set to 0.

4.1.5.3 PMT Table

When creating transport streams that are formatted as specified in [MRL BBTS], the Content Delivery SHALL include a BBTS CA_descriptor [MRL BBTS] in each PMT pointing to a stream protected by Marlin and SHALL include the serviceBaseCID, see [MRL BBTS], into the BBTS CA_descriptor. The socID [MRL BBTS] used by the Content Delivery SHALL be “marlin” (without the double quotes).

In case DVB-SimulCrypt is used with other CA systems as defined in [DVB-SC] and/or with the gateway-centric approach then the content_key_index field in the IEC62455 ECM as defined in [MRL BBTS] SHALL match the scrambling_mode of the other CA system. If the cipher_mode field is 0x1 (CBC) then the initial_vector and next_initial_vector fields in the IEC62455 ECM SHALL be set to 0 as specified in [ATIS-IDSA].

4.1.5.4 CAT Table

When creating transport streams that are formatted as specified in [MRL BBTS], the Content Delivery function MAY include a BBTS CA_descriptor [MRL BBTS] in the CAT for streams protected by Marlin, in order to provide Marlin Rights URLs. If several Marlin Rights URL sets are provided for different service operators, the Content Delivery SHALL include several BBTS CA_descriptor and each BBTS CA_descriptor SHALL include a different serviceBaseCID.

The Rights Issuer URL section, defined in [MRL BBTS] MAY contain a Parental Control URL, as defined in this section. Use of the Parental Control URL is described in section 4.1.5.1.

The coding of the Parental Control URL parameter in the TLV format is the following:

Table 4: Parental_Control_URL Parameter Syntax

Syntax	Mnemonic	No. of bits
Parental_Control_URL () { Parental_Control_URL_tag = 0x05 Parental_Control_URL_length For (i=0; i<N; i++){ Parental_Control_URL_data_byte } }	uimsbf uimsbf bslbf	8 8 8

Parental_Control_URL_tag This specification has defined the value of 0x05 for the Parental Control URL parameter.

Parental_Control_URL_length Specifies the length of the Parental_Control_URL_data_byte in bytes (N).

Parental_Control_URL_data_byte The Parental Control URL for this content.

NOTE: The syntax of Table 4 and similar tables in subsequent sections follows conventions outlined in [MPEG2TS] (e.g. mnemonics, use of C-language like loop descriptors).

Before accessing the Rights Issuer URL specified in [MRL BBTS], the OITF, or the DAE application that receives an “onDRMRightsError” event as defined in sections 7.13.6 and 7.14.7 of [OIPF_DAE2], SHALL obtain user consent to access the web page. When a service receives an HTTP request to the Rights Issuer URL, the service SHOULD respond with an HTML page and not with a Marlin Action Token or with a Marlin License. This HTML SHALL comply with CE-HTML. After user interaction via the HTML pages, the service MAY return a Marlin Action Token or Marlin License.

Before accessing the Parental Control URL specified in this section, OITF SHALL obtain user consent to access the web page. When receiving an HTTP request to the Parental Control URL, the service SHOULD respond with an HTML page This HTML SHALL comply with CE-HTML.

4.1.5.5 System Renewability Messages

In the scope of this specification, DTCP and HDCP System Renewability Messages (SRM) can be transported in a Marlin protected MPEG-2 TS. The signalling and transport of SRM in Marlin protected MPEG-2 TS SHALL comply with [DVB-SRM] specification.

If an OITF supports HDCP output and receives Marlin protected MPEG2-TS format, the OITF SHALL detect the presence of HDCP System Renewability Messages and install them, as defined in [HDCP].

If an OITF supports DTCP output and receives Marlin protected MPEG2-TS format, the OITF SHALL detect the presence of DTCP System Renewability Messages and install them, as defined in [DTCP].

4.1.6 Operation of Marlin Technologies

This section specifies the operation of Marlin technologies to support certain type of use cases.

4.1.6.1 Status of Marlin License Support

A usage rule which uses status information (such as count) is supported in the Marlin License (e.g. burn usage rule by allowing Export action to a certain target system). When the Marlin License requires status management in the client, the corresponding Marlin License SHOULD also have a 'not after' condition specified in the absolute validity period. The value specified by 'not after' SHOULD be no later than 1 month from the issuance of the Marlin License. For example, when the Marlin License issued on 24 November 2008 00:00 allows 3 times burn to Blu-ray, this Marlin License should only be valid until 24 December 2008 00:00.

4.1.6.2 Subscription Support

A CSP function that implements this specification MUST support BNS Extended Topology for Subscription Nodes as defined in [MRL BNSP]. It means that CSP MUST support following Marlin Protocols for Subscription Node which are originally defined as OPTIONAL functions in [MRL BNSP]:

- Marlin License Acquisition to bind Marlin License to Subscription Node
- Marlin Deregistration from a domain represented by Subscription Node where a corresponding Subscription Link SHALL have the following properties:
 - LinkFrom: Personality Node or User Node
 - LinkTo: Subscription Node

The CSP MUST signal this support of the BNS Extended Topology by using the mechanism defined in [MRL BNSP], section 6.2.

4.1.7 DRM Data

4.1.7.1 DRMSystemID

DRMSystemID, used to signal the type of DRM, is defined in [OIPF_META2]. DRMSystemID is used in metadata structures, defined in [OIPF_META2], in APIs defined in [OIPF_DAE2] and in protocols defined in [OIPF_PROT2]. For Marlin, since the DVB CA_System_ID is assigned as 0x4AF4, the value for the DRMSystemID to signal Marlin SHALL be set to the following value: "urn:dvb:casystemid:19188". In addition to the DRMSystemID "urn:dvb:casystemid:19188" signaling Marlin, URN "urn:marlin:ms3:1-0" defined in [MRL MS3] section 1.3.1 MAY be used as a second DRMSystemID to signal that Marlin Simple Secure Streaming feature is supported.

4.1.7.2 Metadata – DRM Control Information

A DRM Control Information structure to hold DRM dependant control parameters is defined in [OIPF_META2] as an extended element included in Content Access Descriptor, defined in [OIPF_DAE2] and extension of PurchaseItem element of BCG and SD&S metadata, defined in [OIPF_META2].

For Marlin protected content, the element of DRMControlInformation SHALL be mapped as specified in the following table:

Table 5: DRMControlInformation Mapping for Marlin

Element / Attribute Name	Element / Attribute Mapping for Marlin
DRMControlInformation	
DRMSystemID	SHALL be set to the value defined for the Marlin System ID, in section 4.1.7.1.
DRMContentID	SHALL be set Marlin Content ID. For Marlin protected MPEG-2 TS or timespamped MPEG2-TS, the content ID is derived from the socID; together

	with serviceBaseCID as defined in [MRL BBTS].
RightsIssuerURL	SHOULD be set to the RightsIssuerURL present in Marlin protected content formats, defined in section 4.1.4 and 4.1.5.
SilentRightsURL	SHOULD be set to the SilentRightsURL present in Marlin protected content formats, defined in section 4.1.4 and 4.1.5. When accessing to this SilentRightsURL, Marlin Action Token or MIPPVControlMessage MAY be returned.
PreviewRightsURL	SHOULD be set to the PreviewRightsURL present in Marlin protected content formats, defined in section 4.1.4 and 4.1.5.
DoNotRecord	SHOULD be set to the same value as the DNR (Do Not Record) bit in recording control access criteria defined in section 4.1.5.2.
DoNotTimeShift	SHOULD be set to the same value as the DNTS (Do Not Time Shift) bit in recording control access criteria defined in section 4.1.5.2.
DRMGenericData	Placeholder element for which currently no mapping is defined.
DRMPrivateData	DRMPrivateData SHALL be an instance of a MarlinPrivateDataType structure, see B.1.
contentType	SHALL be set to the mime type of the DRMPrivateData. For Marlin, it SHALL therefore be set to MIME type of a Marlin License, see [MRL BNSP] or to the MIME type of a Marlin Token, see [MRL BNSP].

If Marlin Simple Secure Streaming feature is supported, in addition to the DRMControlInformation structure given in Table 5, a second DRM Control Information structure MAY be present with parameters mapped as specified in the following table:

Table 5b: DRMControlInformation Mapping for Marlin Simple Secure Streaming

Element / Attribute Name	Element / Attribute Mapping for Marlin
DRMControlInformation	
DRMSystemID	SHALL be set to the value defined for the Marlin Simple Secure Streaming feature, in section 4.1.7.1.
DRMContentID	SHALL be set Marlin Content ID. In case of scheduled content over IP, the content ID is derived from the socID; together with serviceBaseCID as defined in [MRL BBTS].
RightsIssuerURL	Placeholder element for which currently no mapping is defined.
SilentRightsURL	Placeholder element for which currently no mapping is defined.
PreviewRightsURL	Placeholder element for which currently no mapping is defined.
DoNotRecord	SHOULD be set to the same value as the DNR (Do Not Record) bit in recording control access criteria defined in section 4.1.5.2.
DoNotTimeShift	SHOULD be set to the same value as the DNTS (Do Not Time Shift) bit in recording control access criteria defined in section 4.1.5.2.
DRMGenericData	Placeholder element for which currently no mapping is defined.
DRMPrivateData	Placeholder element for which currently no mapping is defined.

mimeType	SHALL be set to the mime type of the DRMPrivateData. For Marlin Simple Secure Streaming feature, it SHALL be set to MIME type of a Marlin Token, [MRL MS3].
----------	---

Both MarlinPrivateDataType and HexBinaryPrivateDataType extend DRMPrivateDataType, defined in [OIPF_META2]; and so the element DRMPrivateData can be substituted by either MarlinPrivateData or HexBinaryPrivateData as defined below:

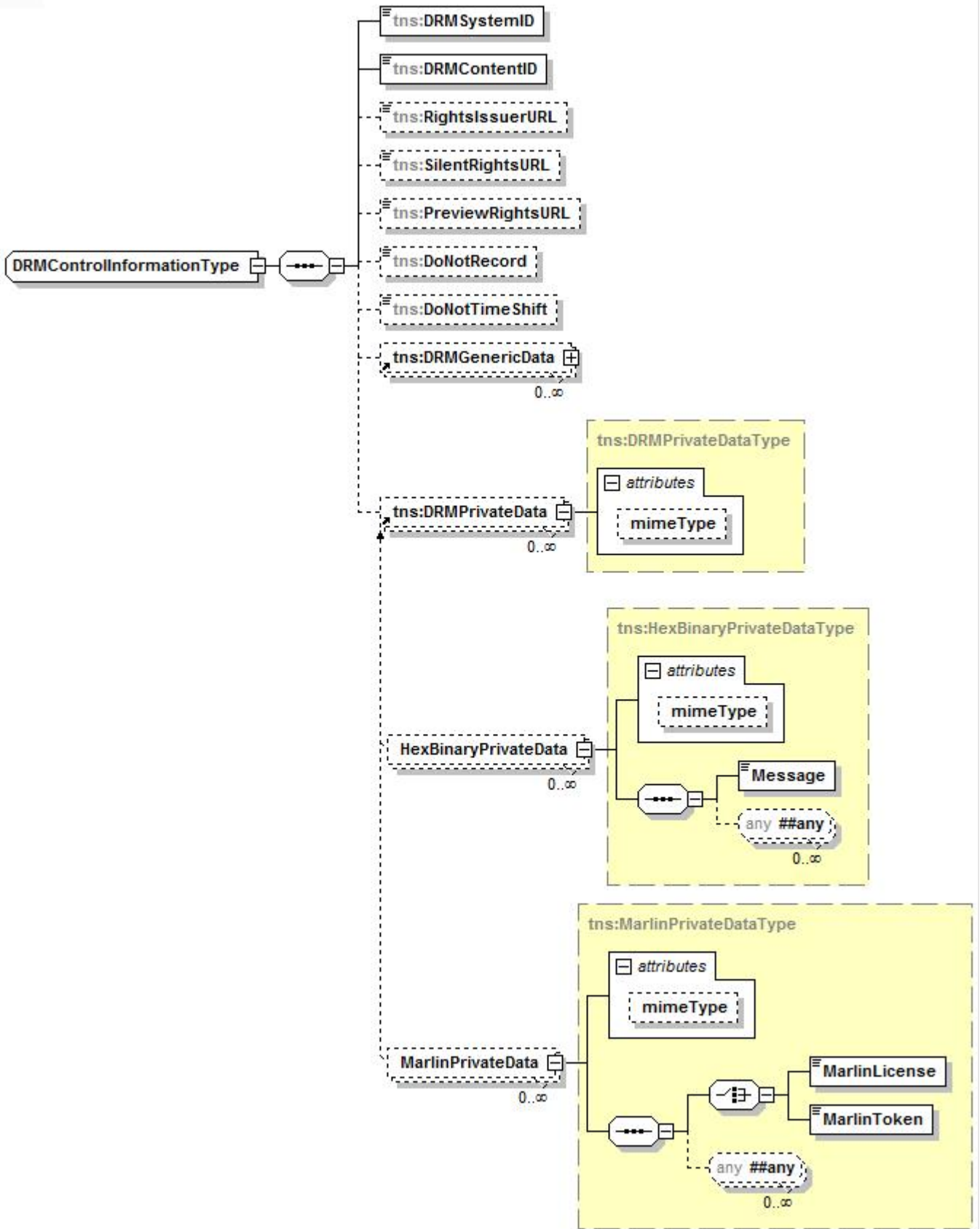


Figure 12: Outline of DRMControlInformationType with MarlinPrivateData

The XML schema for the MarlinPrivateDataType is defined in section B.1.

Table 6: MarlinPrivateData Structure

Element / Attribute Name	Element / Attribute Description
MarlinPrivateData	
MarlinLicense	A Base64 encoded XML Document containing an instance of a Marlin License, typically used for channel preview.
MarlinToken	A Base64 encoded XML Document containing an instance of a Marlin Token, to be used for triggering Marlin Protocol.

4.1.7.3 DAE Marlin Messages

The CSP SHALL support receiving the following messages via the sendDRMMMessage API defined in [OIPF_DAE2], section 7.6.1.

- Marlin Action Token, format and mime type defined in [MRL BNSP].
- MIPPVControlMessage, format and mime type defined in section 4.1.7.3.1
- Marlin License, format and mime type defined in [MRL BNSP].

For these messages, the DRMSystemID SHALL be set to the value defined for the Marlin System ID in section 4.1.7.1.

4.1.7.3.1 MIPPVControlMessage Format

This section describes the usage of MIPPVControlMessage, which is used for pay per view case, and defines the message structure of MIPPVControlMessage. MIPPVControlMessage is used in a pay per view use case where a large number of users try to acquire Marlin License just before a pay per view program begins. To avoid such simultaneous accesses to CSP-T (DRM Server), a service can apply MIPPVControlMessage which includes common Marlin License (i.e. common for OITFs), Marlin Action Token, and Marlin License acquisition timing information. These three data items are used as follows in a typical pay per view case:

1. When an OITF Function (e.g. DAE application) receives a MIPPVControlMessage, the OITF Function (e.g. DAE application) passes the MIPPVControlMessage to CSP. CSP uses common Marlin License embedded in MIPPVControlMessage to play a pay per view program until it gets the Marlin License that is valid only for that OITF. Since a common Marlin License is valid for any OITF, the common Marlin License expires during the pay per view program.
2. By following the timing information in MIPPVControlMessage, the client executes the Marlin Action Token in MIPPVControlMessage, and then it acquires the Marlin License for the OITF.
3. After acquisition of the Marlin License for the OITF, the OITF can play the pay per view program even after the expiration of the common Marlin License.

The MIPPVControlMessage includes Marlin License, which is common among clients, Marlin Action Token, which is used to acquire the unique Marlin License, and timing information, which indicates the timing to initiate Marlin License Acquisition protocols.

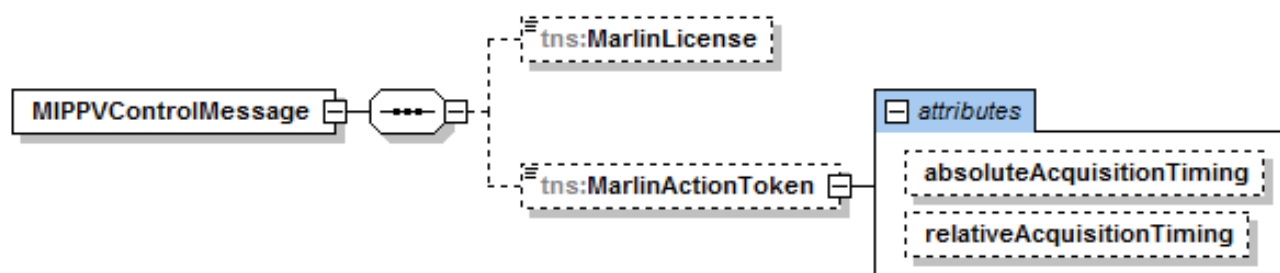


Figure 13: Outline of MIPPVControlMessage

The XML schema for the MIPPVControlMessage is defined in section B.2.

Each element has the following semantics:

Table 7: MIPPVControlMessage Format

Element / Attribute Name	Element / Attribute Description
MIPPVControlMessage	
MarlinLicense	A Base64 encoded XML Document containing an instance of a Marlin License.
MarlinActionToken	A Base64 encoded XML Document containing an instance of a Marlin Action Token.
absoluteAcquisitionTiming	License acquisition timing in absolute time
relativeAcquisitionTiming	License acquisition timing in relative time from the start of the content

MIME type of MIPPVControlMessage is defined as follows:

```
appl i cat i on/vnd. oi pf. mi ppvcontrol message+xml
```

4.2 Gateway-Centric Approach

This section specifies the functionality for the OIPF Gateway-Centric Approach to Content and Service Protection. It elaborates on the CSPG functional entity and UNIS-CSP-G, HNI-CSP, HNI-AGC reference points introduced in the Functional Architecture described in [OIPF_ARCH2].

The gateway-centric approach provides a content protection solution whereby the service provider is able to deploy any preferred protection system to deliver protected content to the user, but the delivery protection is terminated in the CSP Gateway (CSPG) function and a common local protection solution is used to maintain protection of the content on the final link between the CSPG and the OITF. CSPG are specified for CI+ (section 4.2.3) and for DTCP-IP (section 4.2.4).

It is permitted that the CSPG and OITF functional entities are implemented in the same device. In this case the CA/DRM system used for content delivery will be terminated directly at the terminal device. Also, the OITF-CSPG communication is a device-internal interface that does not need to conform to the HNI-CSP interface, i.e. there is a “virtual” CSPG embedded in the terminal device. This is conceptually equivalent to the implementation of any chosen CA/DRM in the device hosting the OITF. This approach is documented informatively in Annex F.

4.2.1 Capabilities

DAE SHALL signal which CA_System_ID values [MPEG2TS] and optionally the type of CSP Gateway are supported in the OITF including those available via Gateway-Centric Approach as defined in section 9 of [OIPF_DAE2] document.

The list of supported CA_System_ID values and optionally the type of CSP Gateway SHALL also be retrieved by the Service Platform provider using one of the following methods:

- The OITF remote management interface [OIPF_PROT2]
- As part of the Service Provider Discovery SUBSCRIBE message [OIPF_PROT2]

4.2.2 CSPG-DAE Interface

When a DAE application uses the DRM Agent API and event, sendDRMMMessage and onDRMMMessageResult, defined in [OIPF_DAE2] section 7.6.1, to handle a DRM Message (see section 7.6.1 in [OIPF_DAE2]) for a given CA_System_ID that is supported by a CSPG, the OITF SHALL forward these messages to the appropriate function, CSP or CSPG.

When protected content is used (played, time-shifted, recorded) from a DAE application, the OITF SHALL forward events (no rights or parental control locking) from the CSPG to the DAE application via the A/V or video/broadcast object. The DRM events onDRMRightsError, onParentalRatingChange and onParentalRatingError are defined in [OIPF_DAE2] sections 7.13.5, 7.13.6, 7.14.6 and 7.14.7. The DRM events (no rights or parental control locking) SHALL include the CA_System_ID information.

4.2.3 CI+ based Gateway

All normative statements in this section and its sub-sections apply only in case the CI+ based Gateway-Centric Approach is supported.

NOTE: The criteria that determine under which circumstances the CI+ based Gateway-Centric Approach is implemented are out of the scope of the present document.

4.2.3.1 Overview

The CSPG-CI+ is an optional entity handling security for the OITF. It SHALL make any specific content protection solution transparent to the OITF. This is achieved by the use of a standard secure channel between the OITF and the CSPG-CI+. The CSPG-CI+ acts as a bridge between a specific protection solution and one standard secure channel. Once the OITF and the CSPG-CI+ are mutually authenticated, the OITF is seamlessly able to receive any content that was initially secured by the different content protection solutions that the CSPG-CI+ handles. The incoming stream to the CSPG-CI+ is associated with the generic media format label "PF", as defined in [OIPF_MEDIA2].

The protected content stream is sent from the OITF to the CSPG-CI+ and then sent back to the OITF protected in such a way that only authenticated OITF can gain access to it. Incoming and outgoing streams format are based on MPEG-2 Transport Stream. Protected file formats based on MP4 file format (i.e. OMA (P)DCF and Marlin IPMP) are not supported.

The definition of the interfaces is based on the DVB CI specification ([DVB-CI]) and the CI+ specification ([CI+]).

Figure 14 presents an overview of the functions and interfaces of the CSPG-CI+.

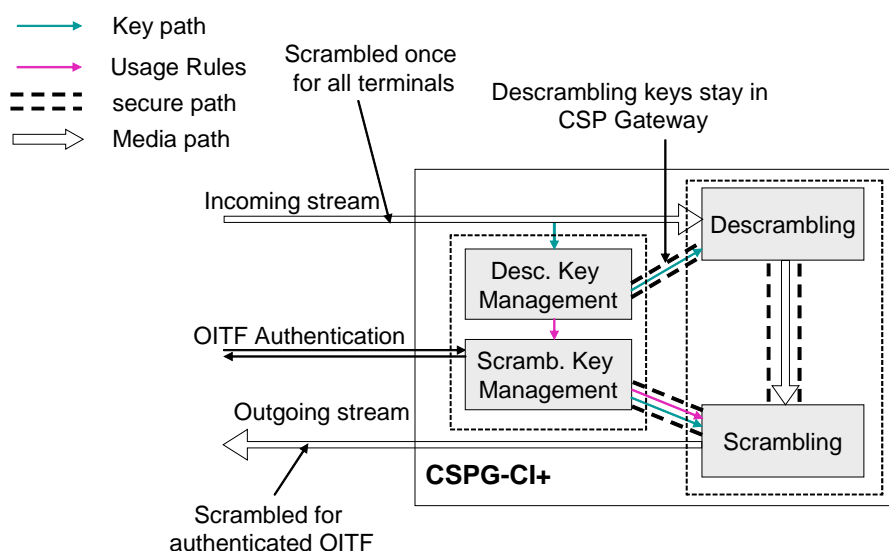


Figure 14: CSPG-CI+ Overview

In order to provide seamless behaviour to the end user (e.g. for service selection operation), the incoming stream in Figure 14 must be delivered through the UNIT-17 reference point as for the Terminal-Centric Approach. Figure 15 describes CSPG-CI+ in the home network context and maps interfaces from Figure 14 to Home Network interfaces defined in [OIPF_ARCH2].

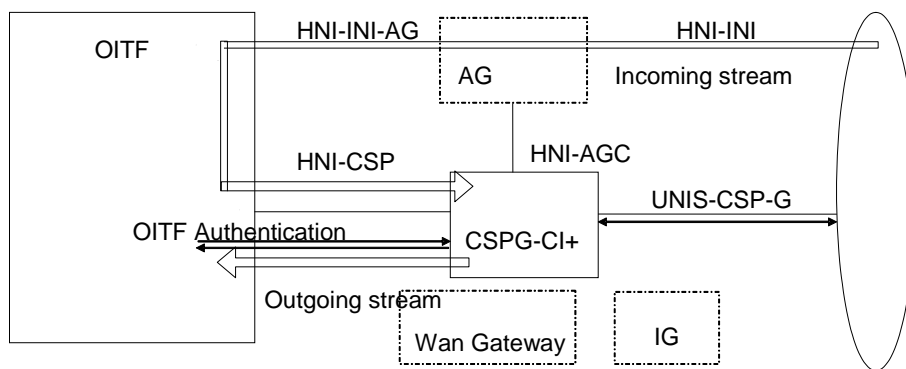


Figure 15: CSPG-CI+ Context

The OITF and CSPG-CI+ SHALL comply with CI+ specifications ([CI+]).

4.2.3.2 CSPG-CI+ Connectivity

The physical interface is based on a PCMCIA slot as specified by the DVB-CI specification ([DVB-CI]) and the CI+ specification ([CI+]).

4.2.3.3 CSPG-CI+ Discovery

The CSPG-CI+ discovery SHALL be performed at OITF start-up and CSPG-CI+ initialization. The setup of the session to the [CI+] Specific Application Support (SAS) resource and the connection to the Open IPTV Forum private application are described in section 4.2.3.4.1.1. A successful connection means that a CSPG-CI+ has been discovered.

4.2.3.4 Residential Network Interfaces

4.2.3.4.1 HNI-CSP

HNI-CSP is an interface to exchange control information and media between the CSPG-CI+ and the OITF.

4.2.3.4.1.1 Control Channel

OITF controls the CSPG-CI+ using resources defined in [DVB-CI] as well as resources as defined in section 11 of [CI+].

OITF and CSPG-CI+ SHALL use the SAS resource, defined in [CI+], section 11.4, to handle messages as specified in this section.

The OITF SHALL send a *SAS_connect_rqst()* APDU [CI+] to the CSPG-CI+ with the specific Open IPTV Forum *private_host_application_ID* defined in Table 8. The CSPG-CI+ SHALL acknowledge the connection by sending back a *SAS_connect_cnf()* APDU [CI+]. The highest *private_host_application_ID* value known by the host shall be tried first. If the *SAS_session_status* returned by the *SAS_connect_cnf()* APDU is not 0x00 (“Connection established”) then a lower value shall be tried and so on until success.

Table 8: Open IPTV Forum *private_host_application_ID*

private_host_application_ID	Value (64bits)
OIPF_APPLICATION_ID for version up to V2.2 included	0x0108113101190000
OIPF_APPLICATION_ID for version above V2.2	0x0108113101190001

Then any further exchanges between the OITF and the CSPG-CI+ are completed through the use of the *SAS_async_msg()* APDU. Syntax of this APDU is reminded in Table 9.

Table 9: *SAS_async_msg()* APDU syntax

Syntax	No. of Bits	Mnemonic
---------------	--------------------	-----------------

Syntax	No. of Bits	Mnemonic
SAS_async_msg() { SAS_async_msg_tag length_field() message_nb message_length for (i=0; i<message_length; i++) { message_byte } }	24 8 16 8	uimsbf uimsbf uimsbf uimsbf

4.2.3.4.1.1.1 Specific messages

The OITF and CSPG-CI+ SHALL support the messages listed in Table 11. For each of the messages the message_byte payload takes the generic syntax given in Table 10. The message data may be broken into a number of records containing the same or different types of data identified by the datatype_id.

Table 10: Generic message_byte() syntax

Syntax	No. of Bits	Mnemonic
message_byte() { command_id ca_system_id transaction_id send_datatype_nbr for (i=0; i<send_datatype_nbr; i++) { datatype_id datatype_length data_type() } }	8 16 32 8 8 16 8 * datatype_length	uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf bslbf

command_id	8-bit value that identifies the message. The values are defined in Table 11.
ca_system_id	16-bit integer that identifies the CA system being queried.
transaction_id	A 32-bit value, generated by the OITF, provided in a message to the CSPG-CI+ that will be returned in any corresponding reply message from the CSPG-CI+. The transaction_id allows the OITF to match the CSPG-CI+'s replies with the corresponding requests. The OITF SHOULD increment the value, modulo 2^{32} , with every message it sends. The transaction_id should be ignored in messages sent spontaneously (events) by the CSPG-CI+ (i.e. rights_info, parental_control_info, system_info).
send_datatype_nbr	8-bit integer that gives the number of data type items included in the message.
datatype_id	8-bit integer that identifies the type of the data contained in the data type loop. The values are defined in Table 12.
datatype_length	16-bit integer that gives the length of the data_type() field in bytes.
data_type	Data type payload. The data type loop shall only contain the specified data type, but may contain multiple records of the same type, the number of records may be determined by computation of the datatype_length field.

Table 11: OIPF specific messages and command_id values

Message	command_id value (hexadecimal)	Direction

		OITF	CSPG-CI+
send_msg	0x01		→
reply_msg	0x02		←
parental_control_info	0x03		←
rights_info	0x04		←
system_info	0x05		←
can_play_content_req	0x06		→
can_play_content_reply	0x07		←
can_record_content_req	0x08		→
can_record_content_reply	0x09		←
(reserved)	0x0A-0x7F		
(user defined)	0x80-0xFF		

Table 12: OIPF specific datatype_id values

Data type	datatype_id value (hexadecimal)
oipf_ca_vendor_specific_information	0x01
oipf_country_code	0x02
oipf_parental_control_url	0x03
oipf_rating_type	0x04
oipf_rating_value	0x05
oipf_rights_issuer_url	0x06
oipf_access_status	0x07
oipf_status	0x08
oipf_drm_private_data	0x09
oipf_can_play_status	0x0A
oipf_can_record_status	0x0B
(reserved)	0x0C-0x7F
(user defined)	0x80-0xFF

4.2.3.4.1.1.2 Mapping of messages to DAE API or Events

The OITF SHALL map the specific messages listed in Table 11 to DAE API or Events as described in Table 13:

Table 13: Mapping to DAE API or Events

Message	DAE API or Event
send_msg	sendDRMMessage
reply_msg	onDRMMessageResult
parental_control_info	onParentalRatingChange, onParentalRatingError
rights_info	onDRMRightsError
system_info	onDRMSystemMessage
can_play_content_req	canPlayContent
can_play_content_reply	canPlayContent
can_record_content_req	canRecordContent
can_record_content_reply	canRecordContent

The DRMSystemID attribute in DAE API or Events are mapped to the ca_system_id field in the SAS_async_msg APDU. The ca_system_id field is filled by extracting the numeric value from the DRMSystemID string, such that "urn:dvb:casystemid:" is removed and the remaining number is converted from a string to a 16 bit integer. The DRMSystemId is build by prefixing the 16 bit integer converted to a decimal number string with "urn:dvb:casystemid:" as described in [OIPF_META2].

Private data are array of bytes encoded for DAE API or Events attributes in a string using a hexadecimal representation, as defined for xs:hexBinary type used in XML schemas. In CI+ SAS_async_msg fields, the private data is encoded in bytes.

Precise mapping of DAE API or Events and attributes are described in the following sections.

4.2.3.4.1.1.3 send_msg

A native application or DAE application SHOULD use the *send_msg* message to provide DRM specific messages to the CSPG-CI+.

When requested by either a native or DAE application, the OITF SHALL send the *send_msg* message to the CSPG-CI+ to exchange DRM messages. Examples of usage are:

- Service Provider handles the purchase of content at the server side and then uses the *send_msg* message via a DAE application to ask the CSPG-CI+ to retrieve the associated license.
- Service provider sends the *send_msg* message via a DAE application to the CSPG-CI+ to force the CSPG-CI+ to purchase a specific program.

The data types for the *send_msg* message are listed in the following table.

Table 14: send_msg message data types

Syntax	Occurrence number
oipf_ca_vendor_specific_information	1

oipf_ca_vendor_specific_information Vendor specific information. The maximum length is 65000 bytes.

When a DAE application calls the sendDRMMMessage API with msgType set to the MIME type "application/vnd.oipf.cspg-hexbinary" and a DRMSYSTEMID set to a ca system id supported by the CSPG-CI+, the OITF SHALL send a *send_msg* message to the CSPG-CI+.

The prototype of the sendDRMMMessage API defined in [OIPF_DAE2] is recalled here:

String sendDRMMMessage(String msgType, String msg, String DRMSYSTEMID)

The OITF SHALL map the attributes of the called DAE API as follows:

- the DRMSYSTEMID attribute is mapped to the ca_system_id field as described in section 4.2.3.4.1.1.2.
- the private data in msg attribute encoded in a string using a hexadecimal representation, as defined for xs:hexBinary type used in XML schemas is decoded to bytes before passing it to *send_msg* message in the oipf_ca_vendor_specific_information field as described in section 4.2.3.4.1.1.2.

4.2.3.4.1.1.4 reply_msg

The CSPG-CI+ SHALL send the *reply_msg* message to the OITF to provide the status of the *send_msg* message.

The data types for the *reply_msg* message are listed in the following table.

Table 15: reply_msg message data types

Syntax	Occurrence number
oipf_status	1
oipf_ca_vendor_specific_information	0..1

oipf_status

If equal to 0, the *send_msg* message has been successfully handled by the CSPG-CI+ and a oipf_ca_vendor_specific_information may be available.
If equal to 1, the *send_msg* message failed because an unspecified error occurred.

If equal to 2, the *send_msg* message failed because the CSPG-CI+ was unable to complete the necessary computations in the time allotted.

If equal to 3, the *send_msg* message failed because oipf_ca_vendor_specific_information has a wrong format.

If equal to 4, the *send_msg* message failed because user consent is needed for

that action.

If equal to 5, the *send_msg* message failed because the specified CA system in *ca_system_id* is unknown.

Unspecified status values SHOULD be considered as, message failed because an unspecified error occurs.

oipf_ca_vendor_specific_information Vendor specific information. The maximum length is 65000 bytes.

NOTE: A service provider should not provide a DRM Message in metadata (BCG, SD&S, CAD) and expect a response in *oipf_ca_vendor_specific_information* of *reply_msg* message, if these metadata are handled by a native application. The native application sending the DRM message to the CSPG-CI+ will not know how to handle a response.

When receiving a *reply_msg* message with a *transaction_id* mapping to a *send_msg* message issued from a DAE application call to *sendDRMMMessage*, the OITF SHALL issue an *onDRMMMessageResult* event to the DAE application

The prototype of the *onDRMMMessageResult* event defined in [OIPF_DAE2] is recalled here:

function onDRMMMessageResult(String msgID, String resultMsg, Integer resultCode)

The OITF SHALL set the attributes of the issued DAE event as follows:

- the *msgID* attribute set to the value returned to the called *sendDRMMMessage*.
- the *resultCode* attribute is mapped to *oipf_status* field as follows:

oipf_status field	Description	resultCode attribute	Description
0	Successful	0	Successful
1	Unspecified error	1	Unknown error
2	Out of time	2	Cannot process request
3	Wrong format	6	Wrong format
4	User Consent Needed	4	User Consent Needed
5	Unknown DRM system	5	Unknown DRM system

- the *resultMsg* attribute set to the private data in *oipf_ca_vendor_specific_information* encoded in a string as described in section 4.2.3.4.1.1.2.

4.2.3.4.1.1.5 parental_control_info

The CSPG-CI+ SHALL send a *parental_control_info* message to advise the OITF whenever the selected program's rating changes. If the new rating does not meet the parental rating criterion (e.g. rating is at or above a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age), the program is not descrambled anymore. If the new rating meets the parental rating criterion (e.g. rating is under a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age), the program is descrambled again.

The data types for the *parental_control_info* message are listed in the following table.

Table 16: parental_control_info message data types

Syntax	Occurrence number
--------	-------------------

Syntax	Occurrence number
oipf_access_status	1
oipf_rating_type	1
oipf_rating_value	1
oipf_country_code	0..n
oipf_parental_control_url	0..1

oipf_access_status	If equal to 0, the program is no longer being descrambled, access conditions to the program are no longer being met. A oipf_parental_control_url may be provided. If equal to 1, the program is descrambled again.
oipf_rating_type	Rating_type as defined in the parental_rating access_criteria_descriptor in [IEC62455].
oipf_rating_value	1-byte rating_value as defined in the parental_rating access_criteria_descriptor in [IEC62455].
oipf_country_code	2-byte optional country_codes as defined in the parental_rating access_criteria_descriptor in [IEC62455].
oipf_parental_control_url	Optional url for connecting to the service provider, for unlocking the parental control.

The OITF SHALL support at least the parental rating system identified by the oipf_rating_type 0, which maps to the parental rating system in DVB Systems [DVB-SI].

If an oipf_parental_control_url is provided and the event is raised to a native application, the native application SHOULD launch the DAE with the oipf_parental_control_url that might allow to unlock parental control in the CSPG-CI+.

When the *parental_control_info* message is received and a DAE application is launched, the OITF SHALL issue the relevant event to the DAE application:

- onParentalRatingChange event, if the parental rating system specified by the oipf_rating_type is supported by the OITF.
- onParentalRatingError event, if the parental rating system specified by the oipf_rating_type is not supported by the OITF.

The prototype of the onParentalRatingChange and onParentalRatingError events defined in [OIPF_DAE2] are recalled here:

function onParentalRatingChange(String contentID, ParentalRatingCollection ratings, String DRMSystemID, Boolean blocked)

function onParentalRatingError(String contentID, ParentalRatingCollection ratings, String DRMSystemID)

The OITF SHALL set the attributes of the issued event as follows:

- the contentId attribute is set to null or undefined.
- the ratings attribute (ParentalRatingCollection object) is filled out with a single ParentalRating object. This ParentalRating object is initialized as follows:
 - o If the oipf_rating_type is supported by the OITF, the oipf_rating_type field is mapped into the scheme property of the ParentalRating object. If the oipf_rating_type is not supported by the OITF, the scheme is set to null or undefined.
 - o The oipf_rating_value field is mapped into the value property of the ParentalRating object. If the oipf_rating_type is supported by the OITF, the name property of the ParentalRating object is filled with the string representation of the parental rating value. If the oipf_rating_type is not supported by the OITF, the name property is set to null or undefined.

- o The oipf_country_code field is mapped into the region property of the ParentalRating object
- the DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The blocked attribute is mapped to oipf_access_status as follows

oipf_access_status field	Description	Blocked attribute	Description
0	program not descrambled	True	Content blocked
1	Program descrambled	False	Content not blocked

A DAE application SHOULD use a proprietary method using sendDRMMMessage to unlock parental control.

If the program is no longer being descrambled (oipf_access_status=0), the native or DAE application SHOULD not stop playing the program, as the program may become descrambled again later (access criteria change, parental unlocking etc).

4.2.3.4.1.1.6 rights_info

The CSPG-CI+ SHALL send a *rights_info* message to advise the OITF that access conditions or rights changed and that the CSPG-CI+ is no longer able or is able again to descramble all requested elementary streams. Once this message is received and if a DAE application is launched, the OIPF SHALL send the relevant event onDRMRightsError, as defined in [OIPF_DAE2] sections 7.13.6 and 7.14.7, to the DAE application.

If the program is descrambled again, the OITF SHOULD display the program again. If the program is no longer being descrambled, the OITF MAY decide to stop the program and SHOULD use the oipf_rights_issuer_url, which may provide for the CSPG-CI+ information to let it retrieve missing rights.

The data types for the *rights_info* message are listed in the following table.

Table 17: rights_info message data types

Syntax	Occurrence number
oipf_access_status	1
oipf_rights_issuer_url	0..1

oipf_access_status If equal to 0, the program is no longer being descrambled, access conditions to the program are no longer being met. A oipf_rights_issuer_url may be provided. If equal to 1, the program is descrambled again.

oipf_rights_issuer_url Optional url for connecting to the service provider.

The prototype of the onDRMRightsError event defined in [OIPF_DAE2] is recalled here:

```
function onDRMRightsError( Integer errorState, String contentID, String DRMSystemID, String rightsIssuerURL )
```

When the *right_info* message is received and a DAE application is launched, the OITF SHALL issue the onDRMRightsError event to the DAE application.

The OITF SHALL set the attributes of the issued event as follows:

- The errorState attribute is mapped to oipf_access_status field as follows:

oipf_access_status field	Description	errorState attribute	Description
0	program not descrambled	0	No license

1	Program descrambled	2	Valid license
---	---------------------	---	---------------

- The contentId attribute is set to null or undefined.
- The DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The rightsIssuerURL is mapped to oipf_rights_issuer_url if this field is present. If the oipf_rights_issuer_url is not present, rightIssuerURL is set to null or undefined.

If the program is no longer being descrambled (oipf_access_status=0), the native or DAE application SHOULD not stop playing the program, as the program may become descrambled again later (access criteria change, rights update etc).

4.2.3.4.1.1.7 system_info

The CSPG-CI+ SHALL send a *system_info* message to advise the OITF of any DRM related event, e.g. the removal of a smartcard. Once this message is received and if a DAE application is launched, the OIPF SHALL send the relevant event onDRMSystemMessage, as defined in [OIPF_DAE2] section 7.6.1, to the DAE application.

The data types for the *system_info* message are listed in the following table.

Table 18: system_info message data types

Syntax	Occurrence number
oipf_ca_vendor_specific_information	1

oipf_ca_vendor_specific_information Vendor specific information. The maximum length is 65000 bytes.

When the *system_info* message is received and if a DAE application is launched, the OITF SHALL issue the onDRMSystemMessage event to the DAE application.

The prototype of the onDRMSystemMessage event defined in [OIPF_DAE2] is recalled here:

function onDRMSystemMessage(String DRMSystemID, String msg)

The OITF SHALL set the attributes of the issued event as follows:

- The DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The msg attribute set to the private data in oipf_ca_vendor_specific_information encoded in a string as described in section 4.2.3.4.1.1.2.

4.2.3.4.1.1.8 can_play_content_req and can_play_content_reply

Note: The following messages are only supported for a private_host_application_ID greater or equal to 0x0108113101190001.

When requested by either a native or DAE application, the OITF SHALL send the *can_play_content_req* message to the CSPG-CI+ to check the local availability of a valid license for playing a content protected by a DRM integrated in the CSPG-CI+.

The data types for the *can_play_content_req* message are listed in the following table.

Table 19: can_play_content_req message data types

Syntax	Occurrence number
oipf_drm_private_data	1

oipf_drm_private_data DRM proprietary private data. The maximum length is 16384 bytes.

When a DAE application calls the canPlayContent API with a DRMSystemId set to a ca system id supported by the CSPG-CI+, the OITF SHALL send a *can_play_content_req* message to the CSPG-CI+ and wait for a *can_play_content_reply* message from the CSPG-CI+.

The data types for the *can_play_content_reply* message are listed in the following table.

Table 20: *can_play_content_reply* message data types

Syntax	Occurrence number
oipf_can_play_status	1

oipf_can_play_status If equal to 1, the CSPG-CI+ has a valid license available that may allow playing the content associated to the DRM metadata. If equal to 0, the CSPG-CI+ has no license available.

When the CSPG-CI+ receives a *can_play_content_req* message, then it shall check whether it owns a valid license for playing the protected content which oipf_drm_private_data relates to and reply to the OITF with a *can_play_content_reply* message with an oipf_can_play_status that indicates whether or not the CSPG-CI+ has a valid license available.

The prototype of the canPlayContent API defined in [OIPF_DAE2] is recalled here:

Boolean canPlayContent (String DRMPprivateData, String DRMSystemID)

The OITF SHALL map the attributes of the called DAE API as follows:

- the DRMSystemId attribute is mapped to the ca_system_id field as described in section 4.2.3.4.1.1.2.
- the DRMPprivateData is mapped to oipf_drm_private_data.
- the returned Boolean is mapped from the oipf_can_play_status.

4.2.3.4.1.1.9 can_record_content_req and can_record_content_reply

Note: The following messages are only supported for a private_host_application_ID greater or equal to 0x0108113101190001.

When requested by either a native or DAE application, the OITF SHALL send the *can_record_content_req* message to the CSPG-CI+ to check the local availability of a valid license for recording a content protected by a DRM integrated in the CSPG-CI+.

The data types for the *can_record_content_req* message are listed in the following table.

Table 21: *can_record_content_req* message data types

Syntax	Occurrence number
oipf_drm_private_data	1

oipf_drm_private_data DRM proprietary private data. The maximum length is 16384 bytes.

When a DAE application calls the canRecordContent API with a DRMSystemId set to a ca system id supported by the CSPG-CI+, the OITF SHALL send a *can_record_content_req* message to the CSPG-CI+ and wait for a *can_record_content_reply* message from the CSPG-CI+.

The data types for the *can_record_content_reply* message are listed in the following table.

Table 22: *can_record_content_reply* message data types

Syntax	Occurrence number
oipf_can_record_status	1

oipf_can_record_status If equal to 1, the CSPG-CI+ has a valid license available that may allow recording the content associated to the DRM metadata. If equal to 0, the CSPG-CI+ has no license available.

When the CSPG-CI+ receives a *can_record_content_req* message, then it shall check whether it owns a valid license for recording the protected content which oipf_drm_private_data relates to and reply to the OITF with a *can_record_content_reply* message with an oipf_can_record_status that indicates whether or not the CSPG-CI+ has a valid license available.

The prototype of the canRecordContent API defined in [OIPF_DAE2] is recalled here:

Boolean canRecordContent (String DRMPrivateData, String DRMSystemID)

The OITF SHALL map the attributes of the called DAE API as follows:

- the DRMSystemId attribute is mapped to the ca_system_id field as described in section 4.2.3.4.1.1.2.
- the DRMPrivateData is mapped to oipf_drm_private_data.
- the returned Boolean is mapped from the oipf_can_record_status.

4.2.3.4.1.2 Media Channel

Media are exchanged as defined in the [CI+] specification.

For streamed content, in either Scheduled Content case or Content on Demand case, the transmission of the protected content from the OITF to the CSPG-CI+ is performed by using MPEG-2 Transport Stream.

For downloaded content, the OITF SHALL stream the content to the CSPG-CI+ at consumption time.

4.2.3.4.2 UNIS-CSP-G

This reference point is used to exchange with the network. Since the CSPG-CI+ does not have network connectivity, it uses the OITF to reach the network.

4.2.3.4.2.1 Low-Speed Communication Resource

The OITF SHALL support the Low-Speed Communications resource with IP extension as specified in [CI+], section 14.2.

4.2.3.4.3 HNI-AGC

In case there is an Application Gateway, control flow is handled through the OITF, via HNI-INI-AG and HNI-CSP control channel. The HNI-AGC reference point introduced in [OIPF_ARCH2] is not used.

4.2.3.5 Provider Network Interfaces

The scrambler on network side SHALL have an interface with the CSP-G Server functional entity so that ECMs can be provided during content encryption. This interface is not described in the present specification.

4.2.3.6 Protected Streaming and File Formats

The CSPG-CI+ supports the MPEG-2 Transport Stream format. The CSPG-CI+ supports the MPEG-2 Transport Stream format. The generically protected incoming transport stream to the CSPG-CI+ is associated with the media format label "PF", as defined in [OIPF_MEDIA2].

The CSPG-CI+ does not support the time stamped MPEG-2 Transport Stream format.

However, in the case content is received by the OITF under a time stamped MPEG-2 Transport Stream format and if the OITF supports the unprotected time stamped MPEG-2 TS format,

- the OITF MAY first use the timestamps provided through the 4 additional bytes of each time stamped MPEG-2 TS (as defined in [OIPF_MEDIA2]) packet to eliminate network jitter and restore the original packet arrival times before sending the content to the CSPG-CI+,
- and the OITF SHALL remove the 4 additional bytes from each time stamped MPEG-2 TS (as defined in [OIPF_MEDIA2]) packet before sending the content to the CSPG-CI+.

If the OITF does not support the unprotected time stamped MPEG-2 TS format, the support of the above two operations is OPTIONAL.

4.2.3.6.1 Protection of MPEG-2 Transport Streams

MPEG-2 Transport Stream can be streamed or downloaded. Based on the CA_descriptor found in the PMT table, the OITF knows if it can handle the stream or if it has to send it to the CSPG-CI+.

If the CA_descriptor found in the PMT is a Marlin CA_descriptor (with CA_system_ID value assigned for Marlin) and the Terminal-Centric Approach is supported by the OITF, then the OITF SHALL manage the content with CSP function described in section 4.1.

If the CA_descriptor found in the PMT is a Marlin CA_descriptor and the Terminal-Centric Approach is not supported by the OITF, then the OITF SHALL ignore it unless Marlin is supported by a CSPG-CI+ in which case the OITF SHALL provide the protected content to the relevant CSPG-CI+.

If the CA_descriptor found in the PMT is not a Marlin CA_descriptor, then the OITF SHALL compare the CA_system_ID value with the CA_system_ID supported by the CSPG-CI+. A CSPG-CI+ might support more than one CA_system_ID. If a CA_system_ID value matches then the OITF SHALL provide the protected content to the CSPG-CI+. In case several CSPG-CI+ gateways are connected to the OITF, the OITF SHALL provide the protected content to only one CSPG-CI+.

If there are several CA_descriptors in the PMT, i.e. referring to different content protection systems (Marlin and/or those offered by the CSPG-CI+ gateways), and if the user is already granted with a valid right or license through one of these content protection systems, the OITF SHALL select the corresponding content protection system as a priority.

NOTE: If simulcrypting with the Terminal-Centric solution is desired, the algorithm used for content encryption in the Gateway-Centric Approach has to be the same as for the Terminal-Centric Approach.

The scrambling algorithm SHALL be signalled in the PMT at program loop level by the scrambling_descriptor specified in [DVB-SI]. Within the scrambling_descriptor, the algorithm is specified by the scrambling_mode field. The following scrambling_modes are referenced by the Open IPTV Forum:

Table 23: Scrambling Modes

scrambling_mode	Description
0x01	DVB-CSA1
0x02	DVB-CSA2
0x70	AES 128-bit key using the Cipher Block Chaining (CBC) encryption mode with the IV setting and the residual termination block process as specified in [ATIS-IDSA].

4.2.3.6.2 Downloaded Content Usage

Downloaded content SHALL be stored locally as it is received by the OITF not going through the CSPG-CI+.

Downloaded content SHALL be provided to the CSPG-CI+ at consumption time only. Consequently, any conversion from e.g. time stamped MPEG-2 TS as defined in [OIPF_MEDIA2] to TS is performed at consumption time as well.

4.2.3.7 Personal Video Recorder

PVR functionality is supported by using URI (Usage Rule Information) as defined in [CI+], section 5.7.

When the OITF is asked to store content, it SHALL send the content to CSPG-CI+. The content is returned from CSPG-CI+ and recorded in accordance with the URI associated with the content.

4.2.3.8 Time Shifting

Time Shifting functionality is supported by using URI (Usage Rule Information) as defined in [CI+], section 5.7.

When the OITF is asked to time shift content, it SHALL store the content returned from CSPG-CI+ before rendering in accordance with the URI associated to the content.

4.2.3.9 CI+ Specification Usage

4.2.3.9.1 Module Deployment

As the network offered in the Open IPTV Forum context is a bi-directional communication channel, the optional Registered Service Mode (RSM) in the CI+ specification [CI+] is recommended in the CSP specification. The RSM SHOULD be supported by CSPG-CI+.

4.2.3.9.2 Host Service Shunning

As no DVB-CI backward compatibility is needed, the OITF SHALL make the CSPG-CI+ operate in a CI+ mode [CI+] only (thus preventing CSPG-CI+ gateways from operating with the unencrypted DVB-CI link). CI+ Protected Service Signalling defined in section 10.1 of [CI+] is not used.

4.2.3.10 DRM Data

4.2.3.10.1 DRMSystemID

DRMSystemID, used to signal the type of DRM, is defined in [OIPF_META2]. DRMSystemID is used in metadata structures in APIs defined in [OIPF_DAE2] and in protocols defined in [OIPF_PROT2]. For CSPG-CI+, the DVB CA_System_ID in DRMSystemID SHALL be the one of the specific content protection solution in the CSPG-CI+.

4.2.3.10.2 Metadata – DRM Control Information

A DRM Control Information structure to hold DRM dependant control parameters is defined in [OIPF_META2] as an extended element included in Content Access Descriptor, defined in [OIPF_DAE2] and extension of PurchaseItem element of BCG and SD&S metadata, defined in [OIPF_META2].

For specifically protected content, the element of DRMControlInformation SHALL be mapped as specified in the following table:

Table 24: DRMControlInformation Mapping for CSPG-CI+

Element / Attribute Name	Element / Attribute Mapping for CSPG-CI+
DRMControlInformation	
DRMSystemID	SHALL be set to the value defined for the specific protection system in the CSPG-CI+, in section 4.2.3.10.1
DRMContentID	Vendor specific information.
RightsIssuerURL	SHOULD be set to the RightsIssuerURL which is provided in the <i>rights_info</i> message defined in section 4.2.3.4.1.1.6.
SilentRightsURL	MAY be set to an URL allowing retrieval of a message to be forwarded to the CSPG-CI+ in order to silently get updated rights. The MIME type or the HTTP response SHALL be "application/vnd.oipf.cspg-hexbinary" and the body of the HTTP response SHALL be a hexadecimal string as described in section 4.2.3.4.1.1.2.
PreviewRightsURL	MAY be set to an URL allowing retrieval of a message to be forwarded to the CSPG-CI+ in order to get preview rights. The MIME type or the HTTP response SHALL be "application/vnd.oipf.cspg-hexbinary" and the body of the HTTP response SHALL be a hexadecimal string as described in section 4.2.3.4.1.1.2.
DoNotRecord	Vendor specific mapping
DoNotTimeShift	Vendor specific mapping
DRMPrivateData	DRMPrivateData structure SHALL be substituted by the HexBinaryPrivateData structure.
contentType	SHALL be set to the mime type of the DRMPrivateData. For CSPG-CI+, it SHALL therefore be set to the following MIME type: "application/vnd.oipf.cspg-hexbinary"

Both MarlinPrivateDataType and HexBinaryPrivateDataType extend DRMPrivateDataType which is defined in [OIPF_META2], and so the element DRMPrivateData can be substituted by either MarlinPrivateData or HexBinaryPrivateData as described in DRMControlInformation outline in Figure 12.

The XML schema for HexBinaryPrivateData is defined in section B.3.

Table 25: HexBinaryPrivateData Structure

Element / Attribute Name	Element / Attribute Description
HexBinaryPrivateData	
Message	A hexadecimal encoded sequence of bytes to be sent to the CSPG-CI+ using <i>send_msg</i> message

4.2.4 DTCP-IP based Gateway

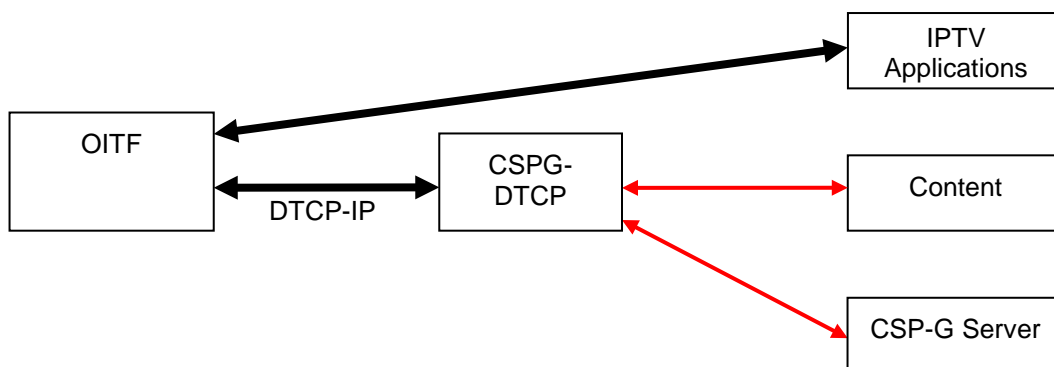
All normative statements in this section and its sub-sections apply only in case the DTCP-IP based Gateway-Centric Approach is supported.

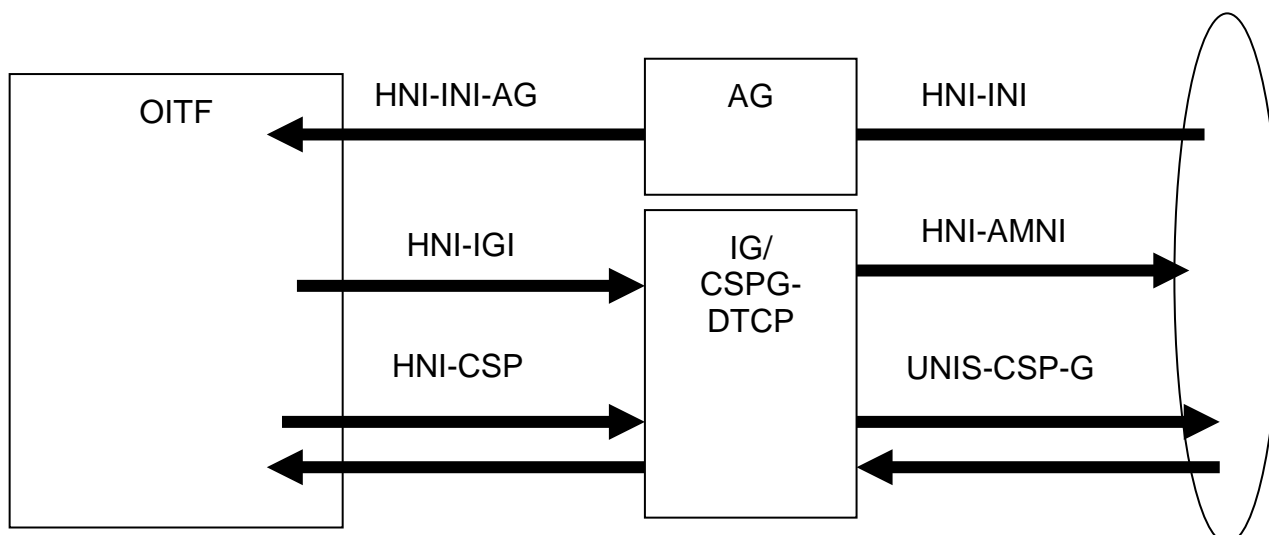
NOTE: The criteria that determine under which circumstances the DTCP-IP based Gateway-Centric Approach is implemented are out of the scope of the present document.

4.2.4.1 Overview

The CSP Gateway based on DTCP-IP (CSPG-DTCP) is an optional entity handling security for the OITF. The CSPG-DTCP resides in the residential network and makes any specific content protection solution transparent. This is achieved by transforming a service proprietary content protection format into standard protection formats which are sent by a secure channel. OITF and CSPG-DTCP mutually authenticate each other, and CSPG-DTCP transfers content and its usage rule information to OITF in a secure manner. The definition of this interface is based on DTCP ([DTCP]) and DTCP over IP ([DTCP-IP]).

- Browsing interactions are executed between DAE and IPTV Applications.
- OITF discovers CSPG-DTCP in a home IP network by the use of the UPnP device discovery protocol as specified in [OIPF_PROT2], section 10.1.1.3.
- For managed network relying on IMS, CSPG-DTCP is co-located with IG to share session management information between IG and CSPG-DTCP. If it supports multicast IPTV services, it is co-located with WAN Gateway to intercept IGMP messages from OITF.
- CSPG-DTCP acts as an HTTP proxy or RTSP proxy. CSPG-DTCP identifies the location of the content through an input URL from OITF.
- CSPG-DTCP transforms service specific content protection formats and usage information format to DTCP over IP content protection format and usage information format respectively.

**Figure 16: CSPG-DTCP Overview**



NOTE: HNI-AGC and HNI-AGI are not involved for the CSPG-DTCP.

Figure 17: Overview of Involved Reference Points

4.2.4.2 CSPG-DTCP Connectivity

The CSPG-DTCP is an IP connected device, and uses the same physical interface used for other IP devices such as IG, AG or home router.

4.2.4.3 HNI-CSP

The main functionalities of the HNI-CSP are to provide:

- CSPG-DTCP discovery as described in [OIPF_PROT2]),
- Content access through CSPG-DTCP,
- DTCP AKE, content stream and usage rule transmission

4.2.4.3.1 Content Access Through CSPG-DTCP

When an OITF determines, e.g. by inspecting the information in the DRMTyp element, see [OIPF_META2], of the content guide, content access descriptor or SD&S, that the content is protected by a service specific protection scheme, it SHALL access the content through the CSPG-DTCP, which acts as an HTTP proxy or RTSP proxy. CSPG-DTCP receives content and SHALL transform the protection scheme to DTCP-IP. When OITF receives error code of 403 from CSPG-DTCP, the error code is interpreted as DRM rights error. Then a DAE application accesses the error handling web page as an action of onDRMRightsError event defined in [OIPF_DAE2], or a native application accesses the RightsIssuerURL described in BCG or SD&S metadata [OIPF_META2].

Refer to Annex E for examples of session setup sequences with a CSPG-DTCP.

For HTTP streaming and download, the OITF SHALL send HTTP GET request through the HTTP proxy in CSPG-DTCP. Note that other HTTP transactions SHALL not use the HTTP proxy in CSPG-DTCP.

4.2.4.3.2 DTCP AKE, Content Streaming and Usage Rule Transmission

DTCP AKE (Authentication and Key Exchange), DTCP content stream and DTCP usage rule are defined in [DTCP] and [DTCP-IP]. The usage rule is provided to the OITF from the CSPG-DTCP considering appropriate mapping, which depends on the service provider's business models. Content type of HTTP response/request SHALL be set to DTCP application media type as defined by [DTCP-IP].

4.2.4.4 UNIS-CSP-G

This interface is out of scope because of applied service specific protection scheme.

4.2.4.5 Protected Streaming and File Formats

The CSPG-DTCP supports either or both of the following formats protected by DTCP-IP encryption on HNI-CSP. The supported format depends on the CA system supported by the CSPG-DTCP. Media format on UNIS-CSP-G is out of scope of this specification.

- MPEG-2 TS and/or time stamped MPEG-2 TS
- MP4 File Format

If the OITF supports the unprotected MPEG-2 TS, the OITF SHALL support the DTCP-IP protected MPEG-2 TS format, as defined in this section and its sub-sections. Otherwise, the support of the DTCP-IP protected MPEG-2 TS format as defined in this section and its sub-sections is OPTIONAL.

If the OITF supports the unprotected time stamped MPEG-2 TS format, the OITF SHALL support the DTCP-IP protected time stamped MPEG-2 TS format, as defined in this section and its sub-sections. Otherwise, the support of the DTCP-IP protected time stamped MPEG-2 TS format as defined in this section and its sub-sections is OPTIONAL.

If the OITF supports the unprotected MP4 file format, the OITF SHALL support the DTCP-IP protected MP4 file format, as defined in this section and its sub-sections. Otherwise, the support of the DTCP-IP protected MP4 file format as defined in this section and its sub-sections is OPTIONAL.

4.2.4.5.1 Protection of MPEG-2 Transport Streams

An MPEG-2 Transport Stream can be streamed or downloaded through CSPG-DTCP. CSPG-DTCP SHALL transmit the content in the DTCP PCP format. The DTCP PCP format encapsulates the MPEG-2 Transport Stream format, which is defined by [OIPF_MEDIA2]. For the avoidance of doubt, Transport Stream level scrambling or PES level scrambling are not used. Both `transport_scrambling_control` bits and `pes_scrambling_control` bits SHALL be set "00".

For content with parental rating control, CSPG-DTCP SHALL transmit MPEG-2 Transport Stream with CA descriptor and KSM table as specified in 4.2.4.5.1.1 and 4.2.4.5.1.2. The `access_criteria_descriptor` carries information for parental rating control.

If the OITF supports the DTCP-IP based Gateway-Centric Approach, the OITF SHALL support the parental rating `access_criteria_descriptor`, specified in [IEC62455], and SHALL support at least the `rating_type 0` within these criteria, which maps to the parental rating system in DVB Systems [DVB-SI]. Other descriptors in the key stream message SHOULD be ignored.

For the parental rating control, the OITF SHALL compare the program's rating from the parental rating `access_criteria_descriptor` with the current parental rating criterion set in the OITF by the application (either native application or DAE) and SHALL block the consumption of the programme if the parental rating system is supported by the OITF and the programme's rating does not meet the parental rating criterion (e.g. rating is at or above a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age). The OITF SHALL raise an event to the application controlling the playback or other operation whenever a parental rating for the A/V content is detected that does not meet the parental rating criterion that is set for the parental system in use, and which has led to blocking of the consumption of the content. The event SHALL provide the programme's rating. In case the application is a DAE application, the event is called `onParentalRatingChange` and is defined in sections 7.13.5 and 7.14.6 of [OIPF_DAE2].

If the OITF does not support the particular parental rating system used in the programme, the OITF SHALL raise an event to the application controlling the playback or other operation. The event SHALL provide the programme's rating. In case the application is a DAE application, the event is called `onParentalRatingError` and is defined in sections 7.13.5 and 7.14.6 of [OIPF_DAE2]. The event MAY be managed via the DAE application (see section 4.5 of [OIPF_DAE2] for more information). In case the application is a native application, the event is managed through an OITF vendor dependent user interface. In both cases, consumption MAY be unblocked by setting a new parental rating threshold, the setting of which is usually restricted to privileged users, e.g. parents. A successful PIN input by a user MAY be used to control the parental rating threshold setting. The OITF SHOULD continue monitoring the MPEG-2 TS, taking into account parental rating criteria changes in ECM streams or new settings for the parental rating threshold in the OITF, and SHALL unblock consumption if the current program's rating becomes lower than the current parental rating threshold.

4.2.4.5.1.1 CA_descriptor

Content with parental rating control SHALL include the CA descriptor in PMT with the following restrictions:

Table 26: CA_descriptor

Syntax	No. of bits	Mnemonic	Value
CA_descriptor() {			
descriptor_tag	8	uimsbf	9
descriptor_length	8	uimsbf	
CA_system_ID	16	uimsbf	0x0007
MPEG2_Reserved	3	bslbf	
CA_PID	13	uimsbf	
for (i=0; i<N; i++) {			
private_data_byte	8	uimsbf	
}			
}			

descriptor_tag MPEG has defined the tag value of 9 for the CA-descriptor.

descriptor_length The length of the descriptor.

CA_system_ID 0x0007

CA_PID The PID on which the KSM table can be found

MPEG2_reserved Bits reserved by [MPEG2TS].

private_data_byte Not used and SHALL be ignored.

4.2.4.5.1.2 Key Stream Message and KSM Table

Content with parental rating control SHALL include Key Stream Message in KSM table ([IEC62455], [DVB-CA]).

Key Stream Message is defined in section 7.2 of [IEC62455] and the following usage restrictions SHALL be applied:

- access_criteria_flag is set to KSM_FLAG_TRUE for the content with parental rating control.
- traffic protection protocol is set to KSM_ALGO_MPEG2_TS_CRYPT.
- traffic_authentication_flag is set to KSM_FLAG_FALSE (traffic authentication is not used).
- next_traffic_key_flag is set to KSM_FLAG_FALSE.
- timestamp_flag is set to KSM_FLAG_FALSE.
- programme_flag is set to KSM_FLAG_FALSE.
- service_flag is set to KSM_FLAG_FALSE.
- content_key_index MAY be set to any value defined in [IEC62455]. The OITF SHALL ignore this field.
- odd_even_flag MAY be set to any value defined in [IEC62455]. The OITF SHALL ignore this field.
- cipher_mode MAY be set to any value defined in [IEC62455]. The OITF SHALL ignore this field.
- encrypted_traffic_key_material_length is set to 0.
- traffic_key_lifetime is set to 0.

For content with parental rating control, the access_criteria_descriptor loop in the Key Stream Message SHALL have at least one parental_rating access_criteria_descriptor. The OITF SHALL ignore other access_criteria_descriptors.

4.2.4.5.2 Protection of MP4 File Format

MP4 file format can be downloaded through CSPG-DTCP. CSPG-DTCP SHALL transmit the content in DTCP PCP format which encapsulates MP4 file format which is defined by [OIPF_MEDIA2].

4.2.4.6 Downloaded Content Usage

For downloaded content, content SHALL be transformed to DTCP-IP protection by CSPG-DTCP when content is being downloaded. Content SHALL be stored and played back by OITF in a manner compliant to DTCP compliance rules [DTCP-AA].

4.2.4.7 PVR Usage

For PVR usage for scheduled content service, content SHALL be transformed to DTCP-IP protection by CSPG-DTCP when content is being streamed or multicast. Content SHALL be stored and played back by OITF in a manner compliant to DTCP compliance rules [DTCP-AA].

5 User Identification, Authentication, Authorisation and Service Access Protection

For the syntax of the messages mentioned in section 5, see Volume 4 [OIPF_PROT2].

5.1 General Principals

This section presents the general principles that govern Service Access Protection and User authentication. In this section the **requested service** represents for example Service Provider Discovery (SPD), Service Discovery (SD), or IPTV Application.

This section also applies to services on the IG requested from the OITF over the HNI-IGI. In this case the equivalent of SAA function and Service Function are co-located on the IG.

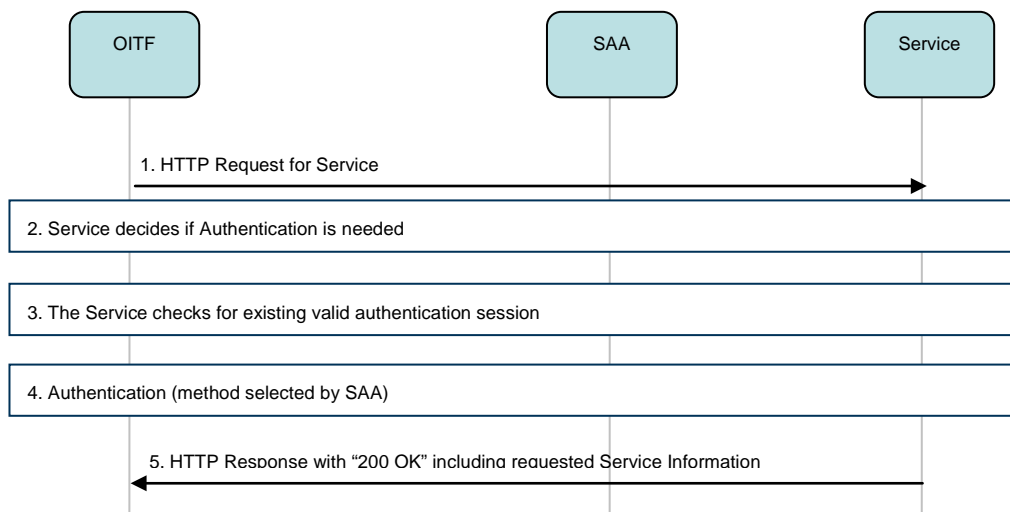


Figure 18: General Message Flow for Service Access Protection and User Authentication

1. The OITF requests a service.
2. The requested service decides whether the request needs to be authenticated or not.
 - If not, the service directly serves the request, go to step 5.
 - If so, go on with step 3.
3. The requested service checks if the request is part of an existing valid authenticated service session (see section 5.6, Session Management).
 - If so, it directly serves the request, go to step 5.
 - If not, go on with step 4.
4. The requested service triggers SAA authentication. There are two cases: the SAA function is co-located with the requested service or the SAA function is standalone (see section 5.3, Service Access Protection). The SAA decides what authentication mechanisms it uses (see section 5.2, Interfaces, and section 5.4, OITF Authentication Mechanisms).
 - If the authentication is successful, go on with step 5.
 - If not, the OITF may e.g. retry step 4 or display an error message, or return an HTTP error.
5. The requested service serves the request.

The **requested service** decides what security is needed for the service delivery: Authentication needed or not, Confidentiality needed (TLS/SSL) or not.

The **SAA** decides what authentication mechanisms it uses and what security is needed for the performed authentication: TLS/SSL or not.

5.2 Interfaces

This section describes the impact of User Identification, Authentication, Authorisation and Service Access Protection on the HNI-INI and HNI-IGI interfaces.

5.2.1 HNI-INI

The following authentication mechanisms are supported for HTTP protocol on HNI-INI interface between OITF and Network (see section 5.4 for their specification):

- No authentication;
- HTTP authentication, see 5.4.1;
- Network based authentication (this requires no action on the OITF), see 5.4.2;
- Web based authentication, see 5.4.3;
- HTTP Digest authentication using an IG (this requires an IG to be present in the home network), see 5.4.4;
- GBA authentication using an IG (this requires an IG to be present in the home network), see 5.4.5;

The OITF **SHALL** support all the mechanisms listed above.

The SAA **MAY** use any of the mechanisms listed above.

Note that GBA authentication can be achieved using either the mechanism in section 5.4.5 GBA Authentication using IMS Gateway or the, more general, mechanism in section 5.4.4 HTTP Digest Authentication using IMS Gateway. Section 5.4.4 allows the use of different authentication mechanism in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that section 5.4.5 GBA Authentication using IMS Gateway will be deprecated and removed in future versions of this specification.

5.2.2 HNI-IGI

In this case the equivalent of SAA function and Service Function are co-located on the IG. The following authentication mechanisms are supported for HTTP protocol on HNI-IGI interface between OITF and IG:

- No authentication
- HTTP authentication, see 5.4.1
- Web based authentication, see 5.4.3

The OITF **SHALL** support all the mechanisms listed above.

On the HNI-IGI interface, the IG **SHALL** support at least one of the following authentication mechanisms:

- No authentication
- HTTP authentication, see 5.4.1

The IG **MAY** use any of the above listed mechanisms (No authentication, HTTP authentication or Web based authentication).

The OITF and IG **SHALL** support and perform IMS registration as specified in section 5.4.6 in [OIPF_PROT2] and described in section 5.5. They **SHALL** do so prior to any service access attempt in managed network relying on IMS.

5.2.3 Common Requirements

On both HNI-INI and HNI-IGI interface, the **OITF SHALL** support all of the following mechanisms, redirection, and security for the HTTP protocol and HTML support:

- standard HTTP requirements: HTTP redirection, HTTP cookies
- URL parameters
- HTML forms and HTTP Post in forms
- TLS/SSL – TLS 1.2 SHOULD be supported, if not then TLS 1.1 SHOULD be supported, otherwise TLS 1.0 SHALL be supported. The OITF SHALL support TLS Renegotiation Extension as described in [RFC5746].

Note: The requirements above ensure the support of SAML web-based SSO, see section 5.6.4.

To avoid extra message exchanges, the **OITF SHALL** provide in requests, when available (see section 5.6):

- HTTP authentication header (Authorization)
- HTTP cookie header (Cookie)

5.3 Service Access Protection (informative)

5.3.1 SAA Co-located with Service (informative)

The following figure describes the sequences when the SAA function is co-located with the requested service.

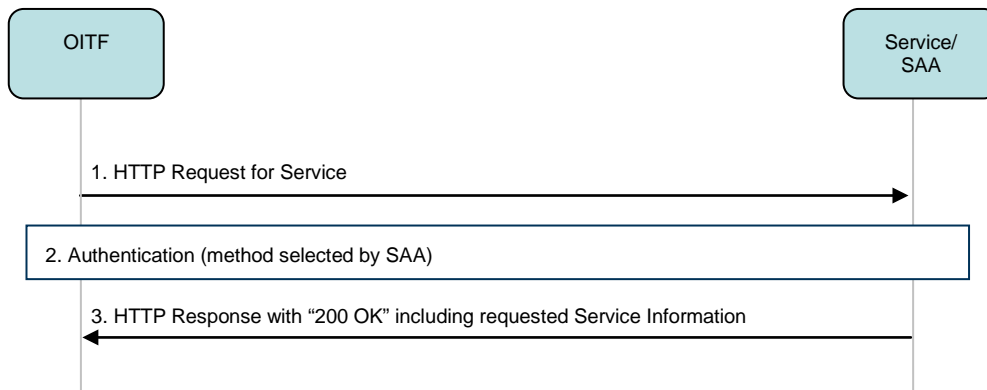


Figure 19: SAA Co-located with Requested Service

1. The OITF requests a service. Authentication is needed and there is no valid authenticated service session.
2. The service/SAA performs authentication.
3. The requested service serves the request.

5.3.2 SAA Standalone (informative)

The following figure describes the sequences when the SAA function is standalone, the OITF is redirected to the SAA for authentication.

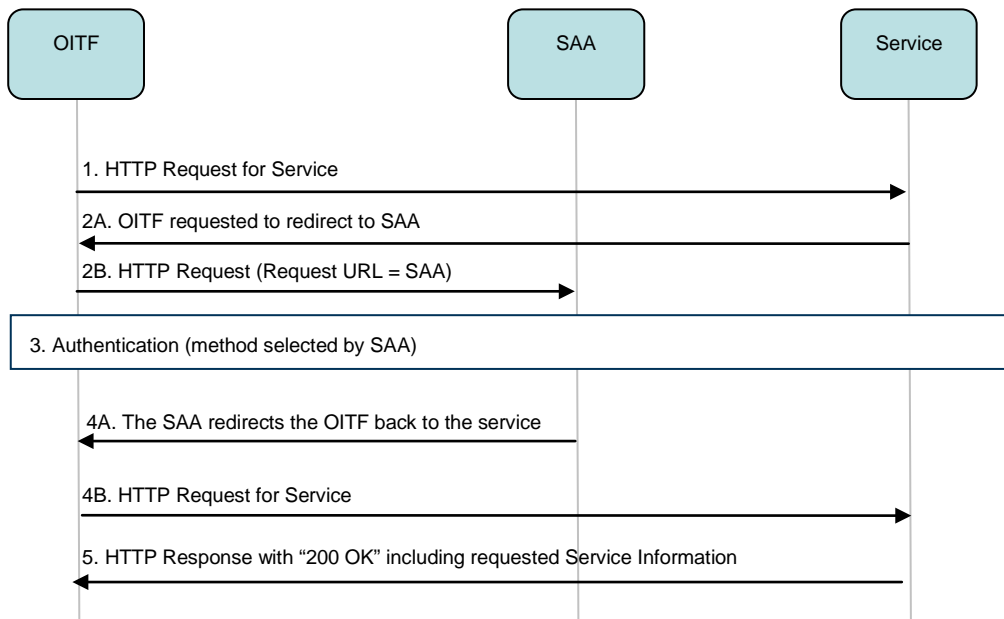


Figure 20: Standalone SAA, Redirection Mode

1. The OITF requests a service. Authentication is needed and there is no valid authenticated service session.
- 2A. The requested service triggers SAA authentication. The service redirects the OITF to the SAA (e.g. using HTTP redirection (Location = SAA)).
- 2B. The OITF connects to the SAA, using the redirection obtained in step 2A.
3. The SAA performs authentication.
- 4A. The SAA redirects the OITF back to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection).
- 4B. The OITF requests the service again, using the redirection obtained in step 4A.
5. The requested service checks that authentication succeeded and serves the request.

5.4 OITF Authentication Mechanisms

5.4.1 HTTP Basic and Digest Authentication

The OITF SHALL support HTTP basic and digest authentication as specified in [RFC2617]. A possible message flow for HTTP basic and digest authentication is described in Figure 21. When HTTP basic or digest authentication [RFC2617] is used, it is assumed that user identifier and its secret information (e.g. password) are shared between OITF and Providers Network (SAA) in advance of the described sequence.

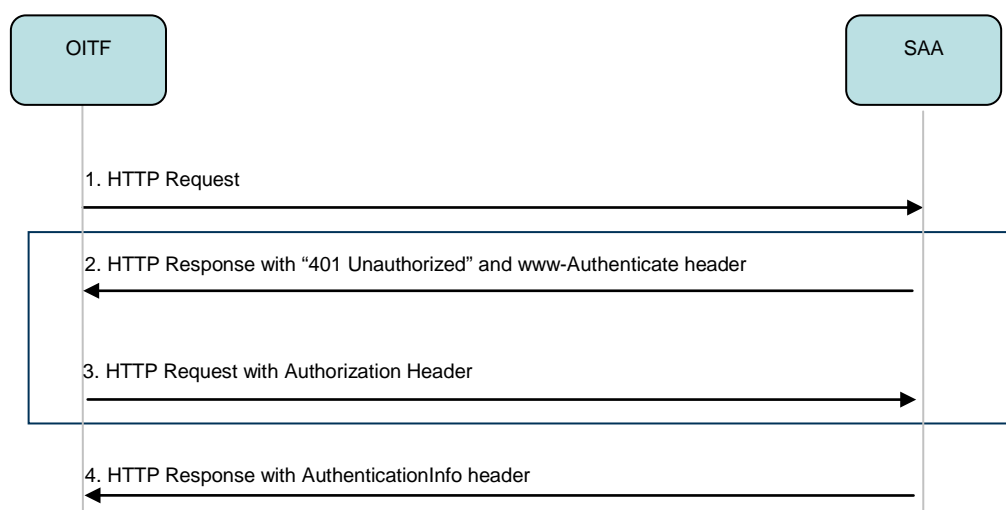


Figure 21: HTTP Basic and Digest Authentication

1. The OITF requests a service co-located with the SAA function or has requested a service and has been redirected to SAA function.
2. The SAA responds with a "401 Unauthorized" status code with a WWW-Authenticate header defined in [RFC2617].
3. The OITF re-sends the request with an Authorization header as defined in [RFC2617]. The user identifier and its secret information are used as username-value and password for the generation of the Authorisation header.
4. The SAA checks the Authorisation header. If the verification succeeds, the SAA/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection). The response contains an AuthenticationInfo header. The response may contain session management information (cookie, URL parameter).

If no user and password are available at the OITF, a window may be displayed to the user for entering his credentials between step 2 and 3. This is the standard working in a DAE application. As described in general principles, this situation shall occur only if no valid authentication session or credentials are available in the OITF.

NOTE: To protect the password that is in the clear in HTTP basic authentication; the SAA may additionally require TLS/SSL as stated in the general principles.

5.4.2 Network Based Authentication (informative)

This section describes the message flows for network based authentication. Network based authentication is a silent authentication based on network information. This authentication is transparent to the OITF.

In the case of a managed network, the SAA can rely on (proprietary) network specific information, which information is out of scope of this specification, to authenticate an incoming request. The sequences are depicted in the following figure:

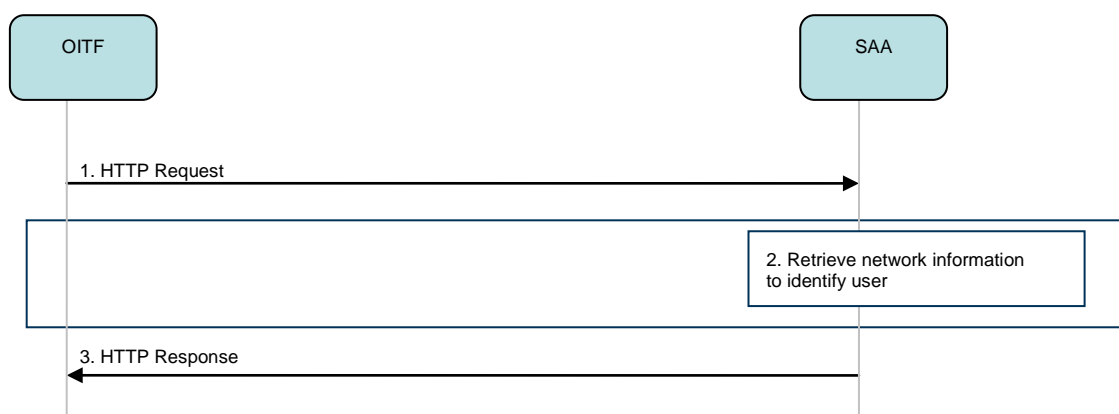


Figure 22: Network Based Authentication

1. The OITF requests a service co-located with the SAA function or has requested a service and has been redirected to an SAA function.
2. The SAA links the request to the user based on network information.
3. If the operation succeeds, the SAA/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection). The response may contain session management information (cookie, URL parameter).

5.4.3 Web Based Authentication

The calling function in the OITF SHOULD support receiving a CE-HTML response for a service HTTP request and SHOULD launch the DAE for displaying it. If the calling function does not support receiving an CE-HTML, XHTML or HTML compatible response, it SHALL signal it to the server by including its acceptable media types without “application/xhtml+xml”, “application/ce-html+xml”, and “text/html” in the request's HTTP "accept" header explicitly, and by also not including CE-HTML/1.0 as part of the User-Agent header. If the calling function does not support receiving an CE-HTML, XHTML or HTML compatible response, the SAA SHALL return a "403 Forbidden" HTTP error.

As described in general principles, this situation shall occur only if no valid authentication session is available in the OITF (e.g. no cookie available).

The DAE within the OITF SHALL support CE-HTML forms and HTTP Post in forms.

The remainder of this section describes the message flows for web based authentication. Web based authentication can be explicit or implicit/silent.

- explicit authentication: the user is prompted with a web page form to fill-in with a login and password: the result of the authentication can be persistent for later re-use (implicit/silent authentication)
- implicit/silent authentication: the user is not prompted with any form but s/he is silently authenticated based on persistent data (session management)

Web based authentication mechanisms do not add requirements to the OITF besides supporting a DAE. They are based on optionally HTML forms and HTTP Post, HTTP redirection and HTTP cookies.

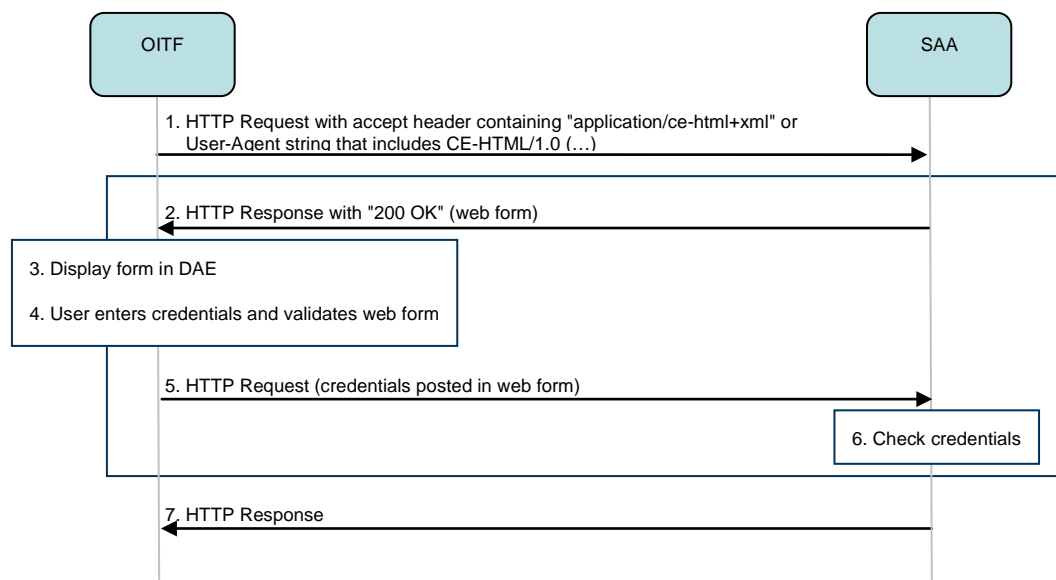


Figure 23: Web Based Authentication with Form

1. The OITF requests a service co-located with the SAA function or has requested a service and has been redirected to SAA function.
2. If the HTTP request to the SAA has a User-Agent string that includes CE-HTML/1.0 as defined in [CEA-2014-A], or the "accept" HTTP header includes (explicitly or implicitly) a CE-HTML accept header (“application/ce-html+xml”), the SAA responds with a CE-HTML compatible web form for requesting user credentials. User credentials provisioning are out of scope of this specification.

3. The web form is displayed in the DAE.
4. The user enters his credentials and validates the form.
5. The form validation posts the user credentials to the SAA.
6. The SAA checks the credentials.
7. If verification is successful, the SAA/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection). The response MAY contain session management information (cookie, URL parameter).

5.4.4 HTTP Digest Authentication – Using IMS Gateway

This section specifies optional functionality by which an OITF can use HTTP Digest credentials in an IG, if present in the home network, for user authentication to HTTP services relying on IMS credentials. The mechanism specified here allows the use of different types of credentials, depending on the capabilities of the IG, and in a way transparent to the OITF, including an extension mechanism to future authentication mechanisms. The OITF discovers the authentication mechanisms supported by IG and the associated credentials stored in the IG, and offers them towards an application server. The application server selects one of the offered authentication mechanisms.

The IG SHALL signal that it supports HTTP Digest Authentication in its description during UPnP discovery as specified in [OIPF_PROT2], section 10.1.1.1.3.

HTTP Basic authentication SHALL NOT be used.

NOTE: The criteria that determine under which circumstances the functionality by which an OITF can use the HTTP Digest credentials in a Gateway Function is implemented in an OITF are out of the scope of the present document.

5.4.4.1 Initial procedure

When the OITF is powered up and if the IG supports HTTP Digest Authentication, the OITF SHALL request supported HTTP Digest authentication realms from the IG as described in [OIPF_PROT2], section 5.4.6.3.1. Receiving this request:

- If the IG supports GBA as defined in [3GPP33.220], the IG SHALL perform a GBA bootstrapping for the current IMS registered user towards the GBA Single Sign-On Function (acting like a BSF in [3GPP33.220]). The GBA registration is based on secrets shared between the ISIM and the network provider. The result of a successful GBA run is the establishment of a session identifier, B-TID, and a shared key, Ks. The decision on running GBA_U or GBA_ME is based on subscription information (i.e; UICC capabilities) as described in [3GPP33.220]. Thus if the ISIM supports GBA, GBA_U bootstrap SHALL be run and in this case the key Ks is computed by the ISIM on the IG side and doesn't leave the UICC. If the ISIM doesn't support GBA, GBA_ME SHALL be run. The support of GBA by the ISIM is indicated in the ISIM Service Table as defined in [3GPP31.103]. This Ks key can later be re-used to derive server side application (NAF) specific keys. These keys can also be passed on to trusted applications in the home network, and can later be used for authentication based on the GBA authentication, but without further need for IG-provider network communication.
- The IG SHALL provide the list of supported realms for HTTP Digest authentication – using IMS Gateway. If the IG supports GBA, it SHALL include in this list the realm for GBA authentication, as defined in [3GPP24.109].
- The IG MAY provide a token to append to the HTTP User-Agent of the OITF for signalling support of specific authentication scheme. The IG SHALL provide the token "3gpp-gba", as specified in [3GPP24.109], if it supports GBA.

The OITF MAY check the returned User-Agent token. The OITF SHALL accept unknown User-Agent tokens, in order to allow evolution of the authentication procedure.

The OITF SHALL append the returned User-Agent token to its User-Agent.

Note: If the IG supports GBA Authentication, as the IG adds "3gpp-gba" to the returned User-Agent token, the OITF acts as a User Equipment in [3GPP24.109] and signals in its User Agent that it supports GBA Authentication.

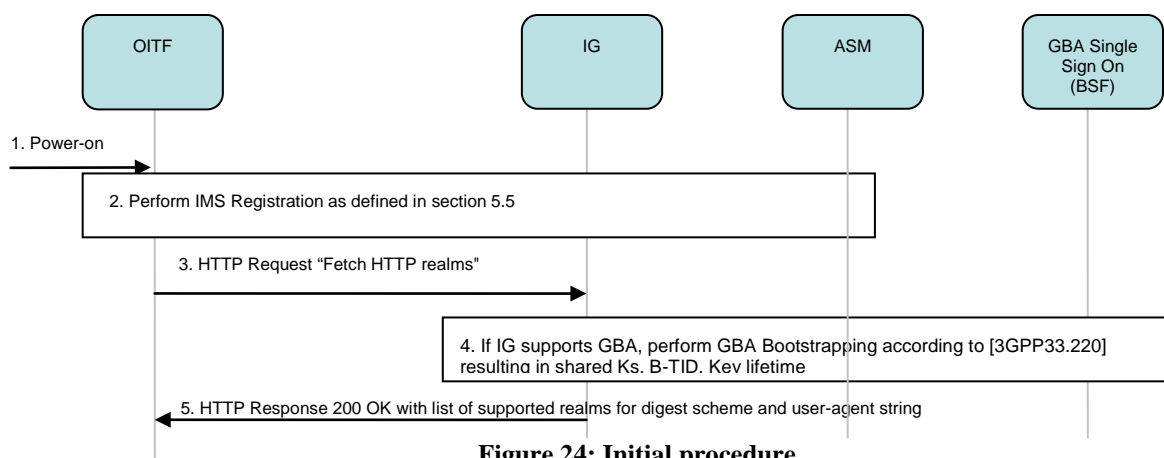


Figure 24 shows the message sequence for initial procedure to ensure HTTP Digest authentication using IG. It contains the following steps:

1. The OITF is powered on.
2. The OITF performs a user registration as defined in section 5.5.
3. The OITF sends a Fetch HTTP realms request to IG as defined in [OIPF_PROT2], section 5.4.6.3.1, step 1. The IG validates the request. The IG may require at that stage any authentication mechanism specified in section 5.2.2 and/or any mechanism and security (i.e. TLS/SSL) specified in section 5.2.3. For simplification, none of this mechanism is shown in Figure 24.
4. If IG supports GBA, the IG performs GBA Bootstrapping procedure according to [3GPP33.220] towards the GBA Single Sign-on (BSF) function in the provider's network. If successful, this results in establishing a shared key Ks on both ends. The GBA Single Sign-on function also sends the lifetime of the key Ks and a session identifier B-TID to the IG.
5. The IG returns the list of supported realm and user-agent string to the OITF as defined in [OIPF_PROT2], section 5.4.6.3.1, step 2.

5.4.4.2 Authentication procedure

If the OITF has registered to an IG which supports HTTP Digest Authentication, each time the OITF needs to access a service offered by an application server that requires HTTP Digest authentication, the OITF SHALL check the realm against the realms retrieved from IG in the initial procedure. If the realm matches to one of the IG supported realms, the OITF SHALL retrieve HTTP credentials and HTTP headers from the IG, as specified in [OIPF_PROT2], section 5.4.6.3.2.

As a pre-requisite to this procedure, the IMS registration MUST have been successfully completed.

The IG MAY provide the following HTTP header:

- For 3GPP GBA Authentication, a "X-3GPP-Intended-Identity" containing the identity of the current user, as specified in [3GPP24.109]
- For HTTP Digest Authentication, a "X-OIPF-Intended-Identity" containing the identity of the current user.

The SAA MAY verify that the intended identity matches to the authenticated identity.

Note: The intended identity is used to identify the user when credentials are shared among users. The service provider should define and enforce policies for sharing of credentials among users.

The OITF MAY check the returned HTTP Headers. The OITF SHALL accept unknown User-Agent tokens, in order to allow evolution of the authentication procedure.

The OITF SHALL use the returned credentials towards the application server, using HTTP Digest authentication as specified by [RFC2617] and SHALL add the returned HTTP headers to the outgoing HTTP requests for this realm.

Note: The service provider should define and enforce policies for sharing of credentials among application servers.

5.4.4.2.1 Authentication procedure using stored credentials

The same credentials and realm as for SIP digest MAY be used, this is an operator security and deployment choice (managed in the IG and the network). In this case:

- the userid SHALL be set to the value of the private user identity;
- the realm SHALL be set to the domain name of the home network;

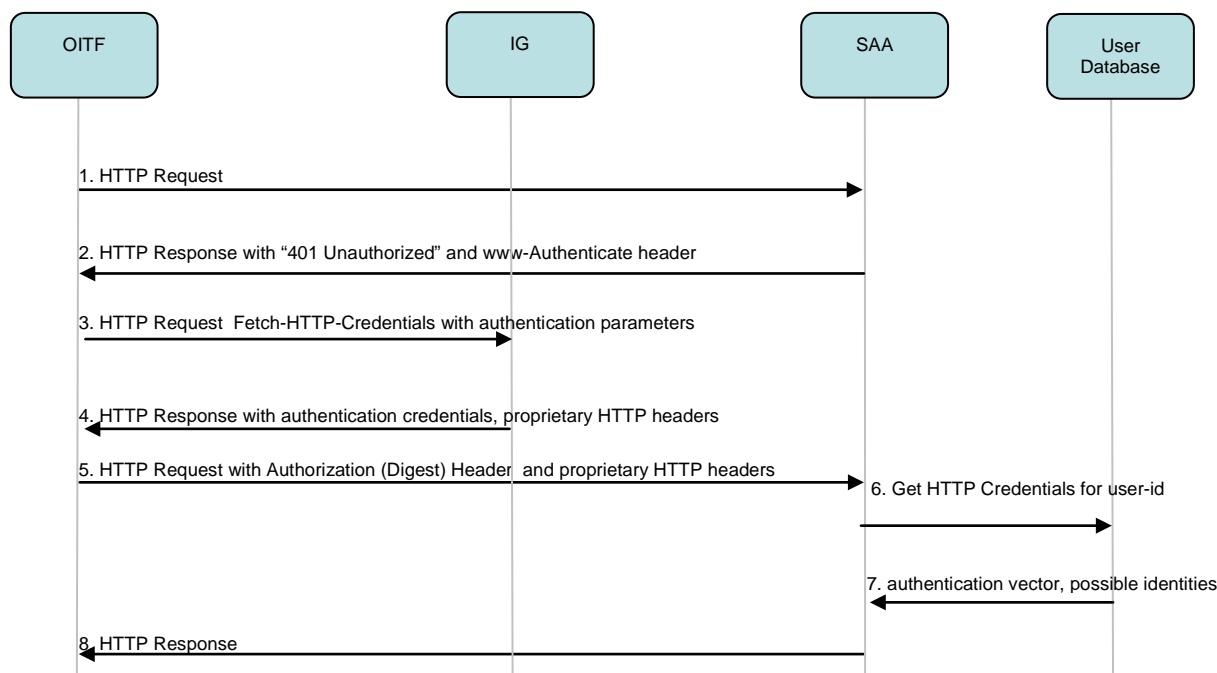


Figure 25: Authentication between an OITF and an SAA based on HTTP credentials stored in IG

Figure 25 shows the message sequence for authentication between an OITF function and an SAA based on HTTP credentials retrieved from the IG. It contains the following steps:

1. OITF function sends a request for a resource (e.g., service) to the SAA. It is assumed here that the resource requires authentication.
2. The SAA returns a 401 Unauthorized message as defined in [RFC2617]
3. The OITF checks the realms. The realm is one of the realms supported by the IG for HTTP Digest authentication. The OITF sends a request including the IMPU, the auth-scheme and realm and additional authentication parameters in case of digest authentication to the IG to retrieve HTTP credentials for the registered user. The request format is specified in [OIPF_PROT2], section 5.4.6.3.2, step 1.
4. IG returns the authentication credentials and optionally HTTP Headers. The nature of the authentication credentials and the response format are specified in [OIPF_PROT2], section 5.4.6.3.2, step 2. The IG may require at that stage any authentication mechanism specified in section 5.2.2 and/or any mechanism and security (i.e. TLS/SSL) specified in section 5.2.3 for access control and/or protection of the credentials. For simplification, none of this mechanism is shown in Figure 25.
5. The OITF function repeats the request 1. with an Authorisation header, using returned authentication credentials. The OITF adds the returned HTTP headers, if any, to the request.
6. SAA requests from the User Database for the subscriber specified via its user-id, its HTTP credentials (authentication vector) and possible identities.

7. The SAA gets the authentication vector and possible identities from the User Database. The SAA checks the user-id and password. The SAA may verify that the intended identity provided in the HTTP header belongs to the possible identities of the subscriber. Note: it is assumed that there exists a trust relation between SAA and User Database. Details are out of scope of this specification.
8. Upon successful authentication, the SAA/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection). The response may contain session management information (cookie, URL parameter).

The message format for steps 3 and 4 are specified in the section 5.4.6.3.2 of [OIPF_PROT2].

5.4.4.2.2 Authentication procedure using GBA credentials

The key K_s that was established during the GBA registration MAY be used later on for authentication between OITF functions and services (i.e., Application Servers). Each time an OITF needs to access a service offered by an AS (i.e., NAF) that requires GBA Authentication, a specific key K_{s_NAF} in case of GBA-ME or $K_{s_ext_NAF}$ in case of GBA-U SHALL be derived by the IG or ISIM in IG respectively and the server side GBA Single Sign-on function (acting like a BSF in [3GPP24.109]). For clarity this specific key is named in the rest of the document $K_{s_}(ext)_NAF$ and will refer to K_{s_NAF} in case of GBA_ME and $K_{s_ext_NAF}$ in case of GBA_U. This generated key SHALL be conveyed to the OITF function in the residential network by the IG, and to the AS by the server side GBA Single Sign-on function (BSF). The key $K_{s_}(ext)_NAF$ SHALL then be used for authentication between the OITF function and the AS, using HTTP Digest authentication as specified by [3GPP24.109].

When a SAA (acting like a NAF in [3GPP24.109]) requests GBA Authentication (perceived as regular HTTP Digest authentication by the OITF), the OITF SHALL retrieve HTTP credentials, in this case GBA Credentials, and HTTP Headers and SHALL perform HTTP Digest authentication.

As a pre-requisite to this procedure, the GBA registration MUST have been successfully completed by the IG in the initial procedure (cf. 5.4.4.1).

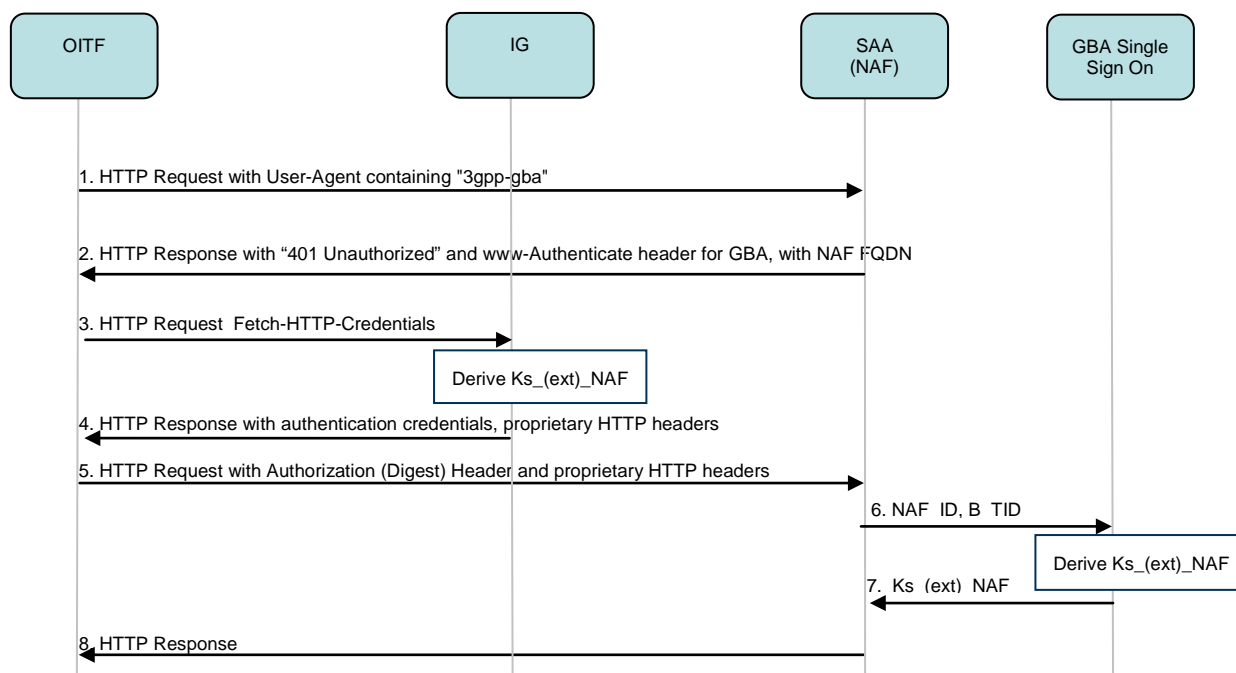


Figure 26: Authentication between an OITF and an SAA Based on GBA Credentials

Figure 26 shows the message sequence for authentication between an OITF function and an SAA based on the previously established GBA bootstrapping. It contains the following steps:

1. OITF function sends a request for a resource (e.g., service) to the SAA (NAF). It is assumed here that the resource requires authentication. The User-Agent string in the HTTP request contains "3gpp-gba" indicating to the SAA that it supports GBA authentication. Note: the user-agent string has previously been sent from IG to OITF,

2. The SAA (NAF) returns a 401 Unauthorized message, the realm indicates that 3GPP bootstrapping is used and provides the NAF FQDN as defined in [3GPP24.109].
3. The OITF checks the realms. The realm is one of the realms supported by the IG for HTTP Digest authentication. The OITF sends a request including the IMPU, the auth-scheme and realm and additional authentication parameters for digest authentication to the IG to retrieve HTTP credentials for the registered user. The request format is specified in [OIPF_PROT2], section 5.4.6.3.2, step 1. The IG identifies from the realm that GBA authentication is requested. IG generates Ks_NAF in case of GBA_ME or Ks_ext_NAF with the cooperation of the ISIM in case of GBA_U (Ks_ext_NAF). Note: according to [3GPP33.220], the NAF_ID is constructed as follows: $NAF_ID = FQDN\ of\ the\ NAF \parallel Ua\ security\ protocol\ identifier$. The FQDN of the NAF is included in the realm. The identifier for Ua security protocol HTTP Digest authentication according to [3GPP24.109] is (0x01,0x00,0x00, 0x00,0x02).

Ks_ext_NAF is computed as $Ks_ext_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_ID)$, where KDF is the key derivation function as specified in Annex B of [3GPP33.220], and the key derivation parameters consist of the user's IMPI, the NAF_ID and RAND.

4. IG returns the authentication credentials and optionally HTTP Headers. The B-TID is used as username and Ks_ext_NAF as password. The IG may return a "X-3GPP-Intended-Identity" HTTP header containing the identity of the current user, as specified in [3GPP24.109]. The response format is specified in [OIPF_PROT2], section 5.4.6.3.2, step 2.
5. The OITF function repeats the request 1. with an Authorisation header, using authentication credentials returned from IG in step 4. The OITF adds the returned HTTP headers, if any, to the request.
6. SAA (NAF) sends B-TID and its NAF_ID to the GBA Single Sign-on function (BSF) in provider network, the GBA Single Sign-on function retrieves Ks and calculates Ks_ext_NAF .
7. The GBA Single Sign-on function (BSF) in provider network returns Ks_ext_NAF , together with its lifetime, to SAA (NAF).

Note the key lifetime returned by the GBA Single Sign-on function (BSF) is equal to the lifetime of the corresponding Ks . But the SAA (NAF) may choose a shorter key lifetime based on local policy and/or application-specific needs.

8. If Ks_ext_NAF has expired, the SAA (NAF) shall send a suitable bootstrapping renegotiation request to the OITF, according to [3GPP33.220] and [3GPP24.109]. Otherwise the SAA (NAF) uses Ks_ext_NAF to authenticate the request. Upon successful authentication, the SAA (NAF)/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection).. The response may contain session management information (cookie, URL parameter).

The message format for steps 3 and 4 are specified in the section 5.4.6.3.2 of [OIPF_PROT2].

5.4.5 GBA Authentication – Using IMS Gateway

Note that GBA authentication can be achieved using either the mechanism in section 5.4.5 GBA Authentication using IMS Gateway or the, more general, mechanism in section 5.4.4 HTTP Digest Authentication using IMS Gateway. 5.4.4 allows the use of different authentication mechanism in a way that is transparent to the OITF, including possible future authentication mechanisms, and should preferably be used. It is expected that section 5.4.5 GBA Authentication using IMS Gateway will be deprecated and removed in future versions of this specification.

This section specifies optional functionality by which an OITF can use the ISIM in an IG, if present in the home network, for user authentication to services relying on IMS credentials. This section is based on the principles described in [OIPF_ARCH2], Appendix B, but extends that section.

The IG SHALL signal that it supports GBA Authentication in its description during UPnP discovery as specified in [OIPF_PROT2], section 10.1.1.1.3.

NOTE: The criteria that determine under which circumstances the functionality by which an OITF can use the ISIM in a Gateway Function is implemented in an OITF are out of the scope of the present document.

5.4.5.1 Initial GBA Registration

When the OITF is powered up or when the user initiates a registration, i.e. when the OITF requests a User Registration from the IG, and if the IG supports GBA Authentication, the OITF SHALL, after the User Registration from the IG,

request a GBA Registration from the IG as described in [OIPF_PROT2]. Receiving this request, the IG SHALL perform a GBA registration for the current IMS registered user towards the GBA Single Sign-On Function (acting like a BSF), according to [3GPP33.220]. The GBA registration is based on secrets shared between the ISIM and the network provider. The result of a successful GBA run is the establishment of a session identifier, B-TID, and a shared key, Ks. This key Ks can later be re-used to derive server side application (NAF) specific keys. These keys can also be passed on to trusted applications in the home network, and can later be used for authentication based on the GBA authentication, but without further need for IG-provider network communication.

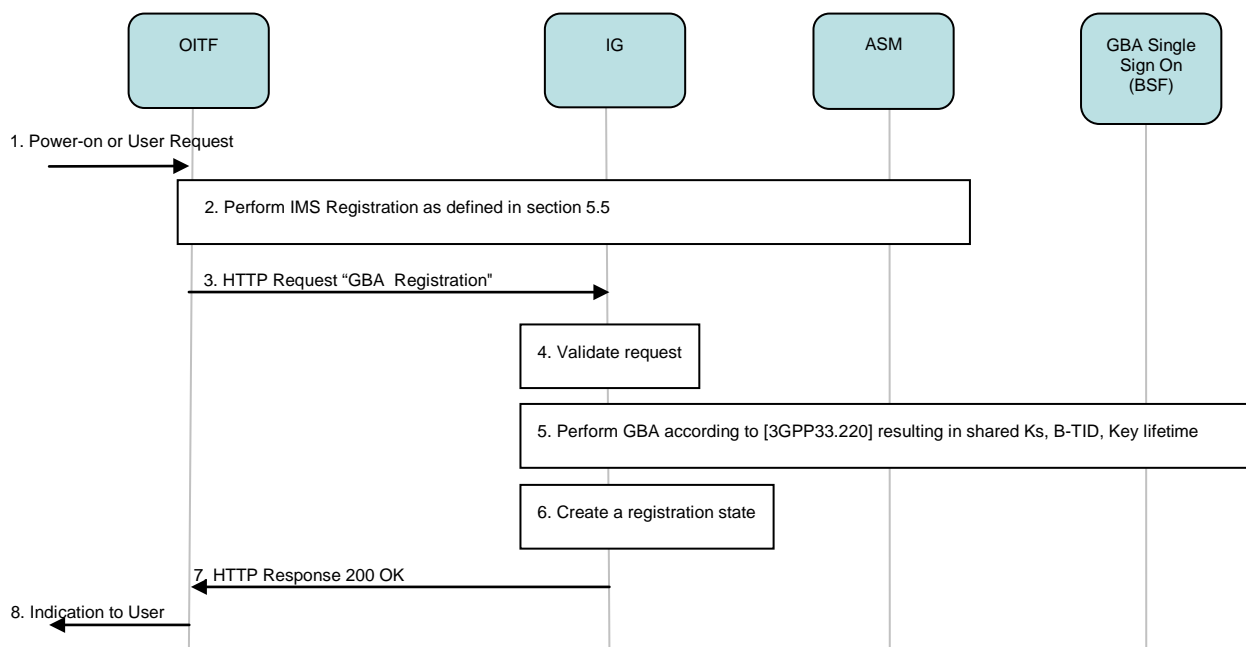


Figure 27: Initial GBA Registration

Figure 27 shows the message sequence for initial GBA registration. It contains the following steps:

1. The OITF is powered on (automatic default registration) or the user requests a personalised registration.
2. The OITF performs a user registration as defined in section 5.4.5.
3. The OITF sends a GBA registration request to IG as defined in [OIPF_PROT2], section 5.4.6.2.1, step 1.
4. The IG validates the request. The IG may require at that stage any authentication mechanism specified in section 5.2.2 and/or any mechanism and security (i.e. TLS/SSL) specified in section 5.2.3. For simplification, none of this mechanism is shown in Figure 27.
5. The IG performs GBA bootstrapping procedure according to [3GPP33.220] towards the GBA Single Sign-on function (BSF) in the provider's network. If successful, this results in establishing a shared key Ks on both ends. The GBA Single Sign-on function (BSF) also sends the lifetime of the key Ks and a session identifier B-TID to the IG.
6. The IG returns the outcome of the GBA registration process to the OITF as defined in [OIPF_PROT2], section 5.3.6.2.1, step 2.
7. If the result of the registration procedure is successful, a registration state is created and maintained in IG.
8. An indication is sent to the user that includes the outcome of the registration process.

5.4.5.2 Re-use of GBA Authentication – Using HTTP Digest Authentication

The key Ks that was established during the GBA registration MAY be used later on for authentication between OITF functions and services (i.e., Application Servers). Each time an OITF needs to access a service offered by an AS (i.e., NAF) that requires GBA Authentication, a specific key Ks_(ext)_NAF SHALL be derived by the IG and the server side GBA Single Sign-on function (acting like a BSF in [3GPP24.109]). This generated key SHALL be conveyed to the OITF function in the residential network by the IG, and to the AS by the server side GBA Single Sign-on function (BSF). The

key $Ks_(\text{ext})_NAF$ SHALL then be used for authentication between the OITF function and the AS, using HTTP Digest authentication as specified by [3GPP24.109].

If the OITF has registered to an IG which supports GBA Authentication, the OITF SHALL act as a User Equipment in [3GPP24.109] and therefore SHALL signal in its User Agent that it supports GBA Authentication.

When a SAA (acting like a NAF in [3GPP24.109]) requests GBA Authentication, the OITF SHALL retrieve GBA Credentials for the specified SAA (NAF) from the IG as specified in [OIPF_PROT2], and SHALL perform HTTP Digest authentication as specified by [3GPP24.109].

If the OITF retrieves an X-HNI-IGI-Intended-Identity HTTP header from the IG, it SHALL use it as intended user identity and SHALL add an "X-3GPP-Intended-Identity" HTTP header to the outgoing HTTP requests to the SAA (NAF); as specified in [3GPP24.109]. The SAA MAY verify that the intended identity belongs to the user (i.e. the identity matches one of the user's public identities indicated in the user security setting that was retrieved from the GBA Single Sign-On Function (BSF)).

As a pre-requisite to this procedure, the GBA registration (cf. 5.4.5.1) MUST have been successfully completed.

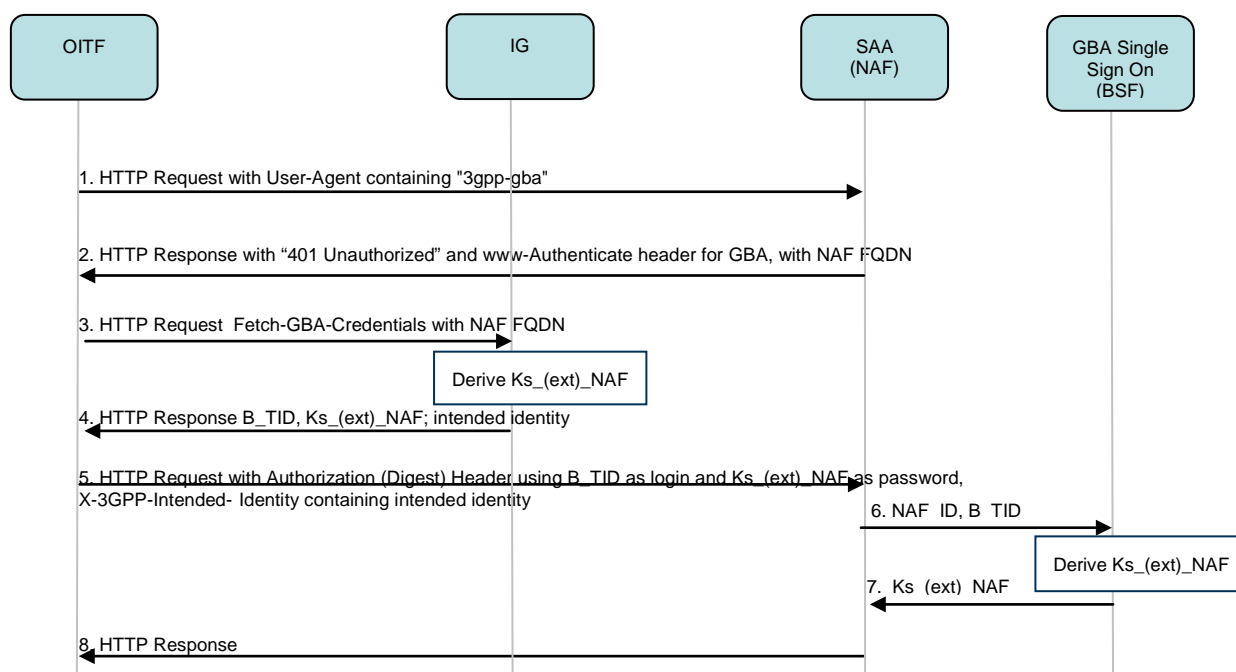


Figure 28: Authentication between an OITF and an SAA Based on GBA Keys

Figure 28 shows the message sequence for authentication between an OITF function and an SAA based on the previously established GBA key. It contains the following steps:

1. OITF function sends a request for a resource (e.g., service) to the SAA (NAF). It is assumed here that the resource requires authentication. The User-Agent string in the HTTP request contains "3gpp-gba" indicating to the SAA (NAF) that it supports GBA authentication.
2. The SAA (NAF) returns a 401 Unauthorized message, the realm indicates that 3GPP bootstrapping is used and provides the NAF FQDN as defined in [3GPP24.109].
3. OITF sends a request including the NAF FQDN to the IG to retrieve GBA credentials, and IG generates Ks_NAF in case of GBA_ME or Ks_ext_NAF with the co-operation of the ISIM in case of GBA_U ($Ks_(\text{ext})_NAF$). Note: according to [3GPP33.220], the NAF_ID is constructed as follows: $NAF_ID = FQDN$ of the NAF || Ua security protocol identifier. The identifier for Ua security protocol HTTP Digest authentication according to [3GPP24.109] is (0x01,0x00,0x00, 0x00,0x02). The request format is specified in [OIPF_PROT2], section 5.3.6.2.2, step 1.

$Ks_(\text{ext})_NAF$ is computed as $Ks_(\text{ext})_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_ID)$, where KDF is the key derivation function as specified in Annex B of [3GPP33.220], and the key derivation parameters consist of the user's IMPI, the NAF_ID and RAND.

4. IG returns $Ks_{(ext)}_{NAF}$, B-TID, the lifetime of the key $Ks_{(ext)}_{NAF}$ and optionally the intended identity to OITF. The lifetime indicates the expiry time of the key $Ks_{(ext)}_{NAF}$ and is equal to the lifetime of the key Ks (which was specified by the BSF during the GBA bootstrapping procedure). The response format is specified in [OIPF_PROT2], section 5.3.6.2.2, step 2.
5. The OITF function repeats the request with an Authorisation header, using B-TID as username and $Ks_{(ext)}_{NAF}$ as password. If a non empty intended identity is returned from the IG, the OITF adds an X-3GPP-Intended-Identity HTTP Header containing the intended identity. If no intended identity is returned from the IG, the OITF shall not add an X-3GPP-Intended-Identity.
6. SAA (NAF) sends B-TID and its NAF_ID to the GBA Single Sign-on function (BSF) in provider network, the GBA Single Sign-on function (BSF) retrieves Ks and calculates $Ks_{(ext)}_{NAF}$.
7. The GBA Single Sign-on function (BSF) in provider network returns $Ks_{(ext)}_{NAF}$, together with its lifetime, to SAA (NAF).

Note the key lifetime returned by the GBA Single Sign-on function (BSF) is equal to the lifetime of the corresponding Ks . But the SAA (NAF) may choose a shorter key lifetime based on local policy and/or application-specific needs.

8. If $Ks_{(ext)}_{NAF}$ has expired, the SAA (NAF) shall send a suitable bootstrapping renegotiation request to the OITF, according to [3GPP33.220]. Otherwise the SAA (NAF) uses $Ks_{(ext)}_{NAF}$ to authenticate the request. Upon successful authentication, the SAA (NAF)/service serves the request or redirects the OITF to the service (e.g. by using SAML HTTP-POST binding, SAML HTTP Post SimpleSign binding or HTTP redirection). The response may contain session management information (cookie, URL parameter).

The message format for steps 3 and 4 are specified in the section 5.3.6.2.2 of [OIPF_PROT2].

5.4.5.3 Binding Between GBA User Authentication and DRM Device Authentication (Informative)

GBA authenticates ISIM/IMPI, not the device. On the other hand, DRM (e.g. Marlin) relies on device authentication; the device must have a valid certificate issued by the DRM trust authority. To avoid security issues e.g. allowing a legitimate (from a DRM point of view) device that is however not in fact authorised by a user accessing services, the GBA (user) authentication and the DRM device authentication need to be securely linked together.

5.5 IMS Registration – OITF

This section specifies the message flows for IMS Registration using SIP Digest⁷ authentication or IMS AKA authentication by means of which Service Platform Providers and IMS Gateways located in Residential Networks can authenticate each other. These message flows are based on [3GPP33.203] and [3GPP24.229] (stage 3 specification).

5.5.1 Relevant Functional Entities and Reference Points

Figure 29 extracts the functional entities and reference points relevant for IMS Registration from the OIPF Provider and Residential Network Architectures (see Figures 5-2 and 5-4 in [OIPF_ARCH2]):

⁷ This section specifies authentication-related details of certain SIP messages. Elsewhere, for example at ETSI TISPAN, this SIP authentication method is often called “HTTP Digest” as SIP Digest [RFC3261] is identical to HTTP Digest [RFC2617] – despite the fact that the protocol in question is SIP and not HTTP. The authentication method treated in this section is referred to as “SIP Digest” since the name “HTTP Digest” might lead to the wrong impression that the protocol in question is HTTP.

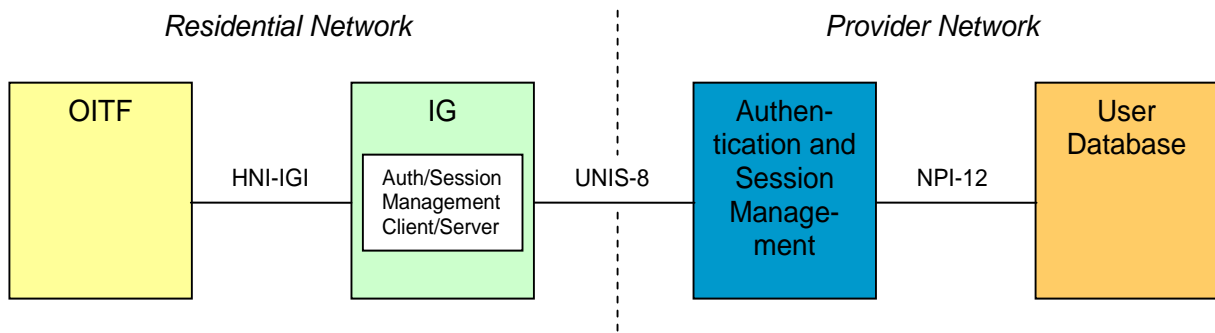


Figure 29: OIPF Functional Entities and Reference Points Involved in IMS Registration

SIP Digest authentication, and respectively IMS AKA authentication is interlaced into the IMS Registration message exchange between the IMS Gateway (IG) and the Authentication and Session Management (ASM) functional entities. IMS Registration occurs either when the IG is powered up or when the IG receives a corresponding request from an OITF. The User Database supplies the ASM with authentication vectors needed for SIP Digest authentication, and respectively IMS AKA authentication.

5.5.2 Prerequisites

Prior to the first IMS Registration (and hence prior to the first SIP Digest or IMS AKA) protocol execution, the following parameters MUST be provisioned:

- to the IG⁸:
 - for SIP Digest:
 - one or more IP Multimedia Private Identities (IMPI),
 - one or more IP Multimedia Public Identities (IMPU), each associated to one or more IMPIs,
 - one or more passwords, each assigned to one and only one of the IMPIs provisioned to the IG,
 - a Service Platform Provider Network Domain Name.
 - for IMS AKA, an ISIM or a USIM application shall always be used for authentication, as described in [3GPP33.203]. For the purpose of this document, the ISIM is a term that indicates a collection of IMS security data and functions on a UICC.
 - The ISIM SHALL include :
 - one IMPI.
 - one or more IP Multimedia Public Identities (IMPU), associated with the IMPI
 - a SPP Network Domain Name referred as Home Network Domain Name in 3GPP specifications
 - Support for sequence number checking in the context of IMS Domain
 - An Authentication key
 - The same framework for algorithms as specified for USIM
 - There shall only be one ISIM for each IMPI.

⁸ In case of IMS AKA, these parameters are in a UICC with an ISIM or USIM application.

- and to the User Database, the IMS subscription information comprising:
 - the IMPI(s) and IMPU(s) provisioned to the IG,
 - the association of the IMPU(s) to the IMPI(s),
 - and for SIP Digest the password(s) provisioned to the IG. The User Database stores each password against the IMPI it is assigned to.
 - And for IMS AKA the Authentication Key contained and protected within the UICC in the IG. The User Database stores each Authentication Key against the IMPI it is assigned to.

Methods for provisioning these parameters to IG and User Database functional entities are out of scope of this specification.

5.5.3 SIP Digest Message Flows

Figure 30 shows the message flow for SIP Digest authentication, which is interlaced into IMS Registration messages:

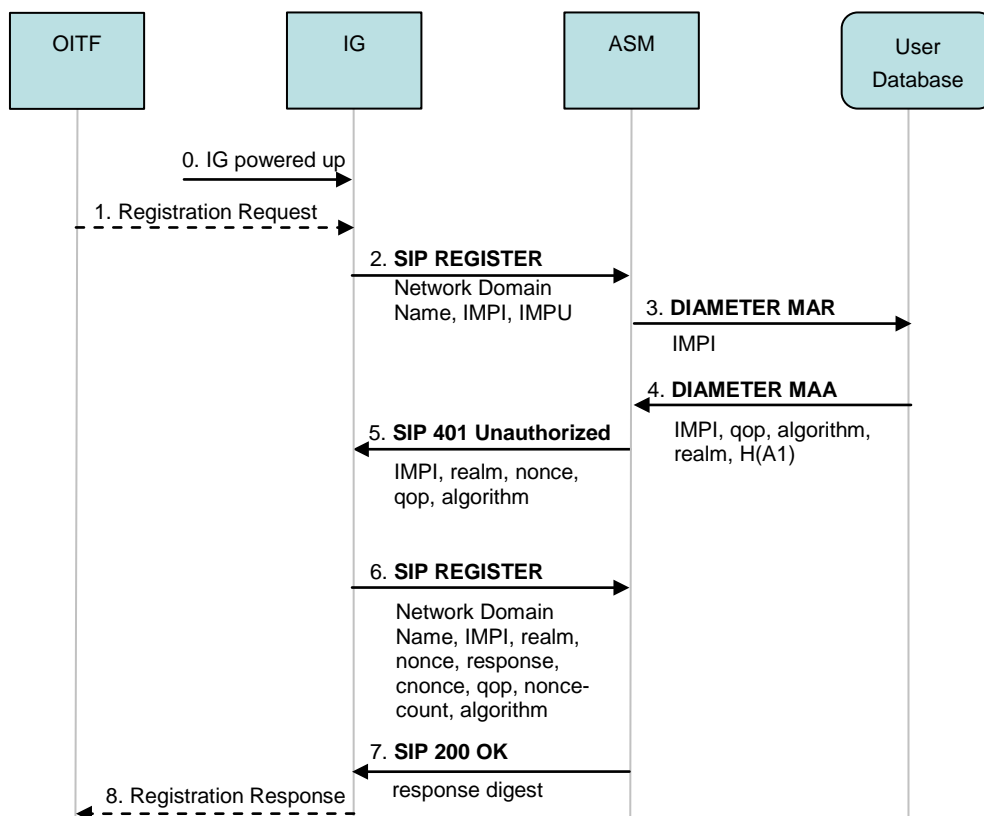


Figure 30: SIP Digest Message Flow Interlaced into IMS Registration

0. The IG is powered up. This can initiate the execution of steps 2 – 7.
1. **OITF to IG: Registration Request**
The OITF sends a request for registration to the IMS Gateway (IG), when needed (the end user explicitly logs on for personalized services).
2. **IG to ASM: SIP REGISTER**
This request contains the SPP Network Domain Name of the IG's IMS home network, an IMPI and an IMPU. If the ASM has a valid SIP Digest authentication vector (SD-AV) for the specific IMPI, steps 3, 4 and 5 are omitted.
3. **ASM to User Database: DIAMETER MULTIMEDIA AUTH REQUEST (MAR)**
The ASM requests a SD-AV from the User Database with respect to the IMPI received in step 2.

4. **User Database to ASM: DIAMETER MULTIMEDIA AUTH ANSWER (MAA)**
 Along with the IMPI, the User Database sends a SD-AV to the ASM containing the following data: qop value (quality of protection), the authentication algorithm, realm, and a hash value H(A1) of the IMPI, realm, and password. [RFC2617] provides additional information on the values in the authentication vector for SIP Digest based authentication. Upon reception of the MAA message, the ASM stores the H(A1) value and generates the nonce value needed to challenge the IG.
5. **ASM to IG: SIP 401 Unauthorized**
 The ASM denies the IG authentication but sends a SIP 401Unauthorized message to the IG in order to challenge the IG. This message contains the IMPI, the nonce, the authentication algorithm, and the realm and qop values.
6. **IG to ASM: SIP REGISTER**
 After reception message 5, the IG generates a client nonce (cnonce) and calculates an authentication response value using this cnonce and other values received in step 5 (see [RFC2617]). The IG sends a new SIP REGISTER request to the ASM, this time with the authentication response along with the parameters IMPI, realm, nonce, response, cnonce, qop, nonce-count, and algorithm.
7. **ASM to IG: SIP 200 OK** (successful case)
 After reception of the SIP REGISTER message containing the authentication response value, the ASM calculates the *expected* response value using the previously stored H(A1) and the stored nonce value together with other parameters (see [RFC2617]). If the response value received from the IG equals the expected response value, the IG has been authenticated and the IMPU is registered in the ASM. In this successful case, the ASM sends the SIP 200 OK from ASM to the IG, enabling the IG to authenticate the SPP Network. This SIP 200 OK message contains a response digest calculated using the cnonce value generated by the IG prior to sending message 6.
8. **IG to OITF: Registration Response**
 The IG informs the OITF about the result of the registration procedure (when step 1 was executed).

The details of the messages 2 – 7 are specified in [3GPP24.229].

5.5.4 IMS AKA Message Flows

To support IMS AKA, a UICC with an ISIM or USIM application must be integrated into the IMS Gateway (IG). From the IMS point of view, the IG thereby takes the role of an IMS Subscriber. The UICC stores a long-term secret key K which is shared between the ISIM or USIM application and a User Database belonging to the network operator that provides the ISIM or the USIM. Figure 31 shows the high-level message flows for user identification and authentication based on the IMS AKA procedure

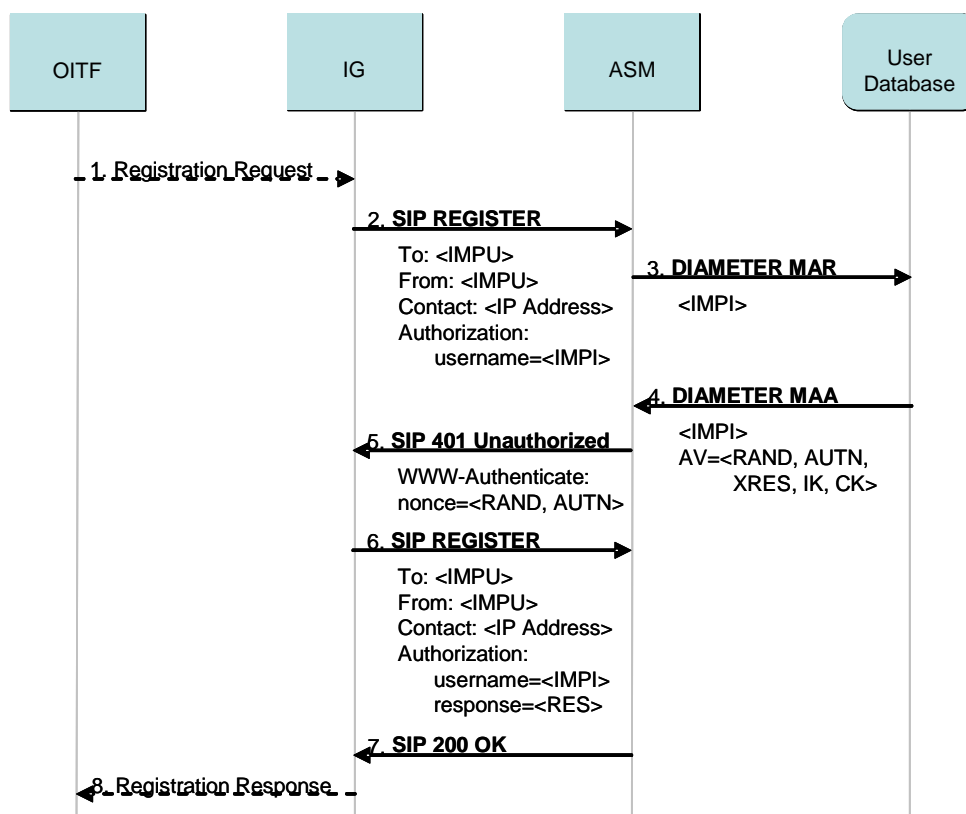


Figure 31: User Identification and Authentication based on the IMS AKA procedure

0. The IG is powered up. This can initiate the execution of steps 2 .7
1. **OITF to IG: Registration Request**
The OITF sends a request for registration to the IMS Gateway (IG), when needed (the end user explicitly logs on for personalized services)
2. **IG to ASM: SIP REGISTER**
This request contains the SPP Network Domain Name of the IG's IMS home network, the IMPI and the IMPU. All this data is read from the ISIM.
3. **ASM to User Database: DIAMETER MULTIMEDIA AUTH REQUEST (MAR)**
ASM requests authentication data from the User Database with respect to the IMPI received in step 2.
4. **User Database to ASM: DIAMETER MULTIMEDIA AUTH ANSWER (MAA)**
The User Database sends an Authentication Vectors (AV) to the ASM containing the following data: random challenge RAND, answer XRES expected by the IG in step 6, network authentication token AUTN, integrity key IK, and ciphering key CK. The authentication token AUTN contains a message authentication code (MAC) enabling the IG to authenticate the SPP Network (see step 5).
5. **ASM to IG: SIP 401 Unauthorized**
At this point in time, the ASM denies the IG authentication. Instead, it sends a SIP Unauthorized message with a WWW-Authenticate header to the IG. This header contains RAND and AUTN. After reception of this message, the IG verifies the message authentication code contained in AUTN thereby authenticating its SPP Network.
6. **IG to ASM: SIP REGISTER**
ISIM computes the value RES on input of its version of the secret key K stored on the UICC of the IG. The IG sends a new SIP REGISTER request to the ASM, this time with RES as response to the challenge the ASM initiated in step 5.
7. **ASM to IG: SIP 200 OK**
If RES = XRES (successful case), ASM considers the IG as authenticated, and binds IMPU to the IP address <IP address>.

8. IG to OITF: Registration Response

The IG informs the OITF about the result of the registration procedure. (when step 1 was executed)

In case of success, the ISIM of the IG is able, based on its knowledge of the secret key K and the authentication token AUTN, to calculate the same values of the integrity key IK and the ciphering key CK as those that the ASM received in step 4 from the User Database. The IG and the ASM use IK and CK to establish IPsec Security Associations for protecting SIP signaling messages over the IG – ASM reference point

The details of the messages 2 -7 are specified in [3GPP24.229].

5.6 Session Management and Single Sign On

User authentication does not need to be performed with each request. In order to avoid re-authentication at each request, a Service (and/or SSA) can rely on authentication session management and Single Sign On. The following authentication session management can be used: cookies, URL parameters and HTTP authentication session, if HTTP or GBA authentication has been used. SAML Web-based Single Sign On can be used.

5.6.1 Cookie Session

The OITF SHALL support HTTP session management using cookies as described in [RFC2109]. The cookie is opaque data to the OITF.

Persistent cookies SHALL be stored in non-volatile memory (Flash, HDD, etc.) in the OITF.

All OITF applications using HTTP (not only DAE) SHALL be able to create, read and delete **persistent** cookies with respect to domain restriction as specified in [RFC2109]. Persistent cookies SHOULD be shared between all components in an OITF.

User SHALL have the possibility to delete **persistent** cookies in OITF.

The following figure shows an example of sequences based using cookie session:

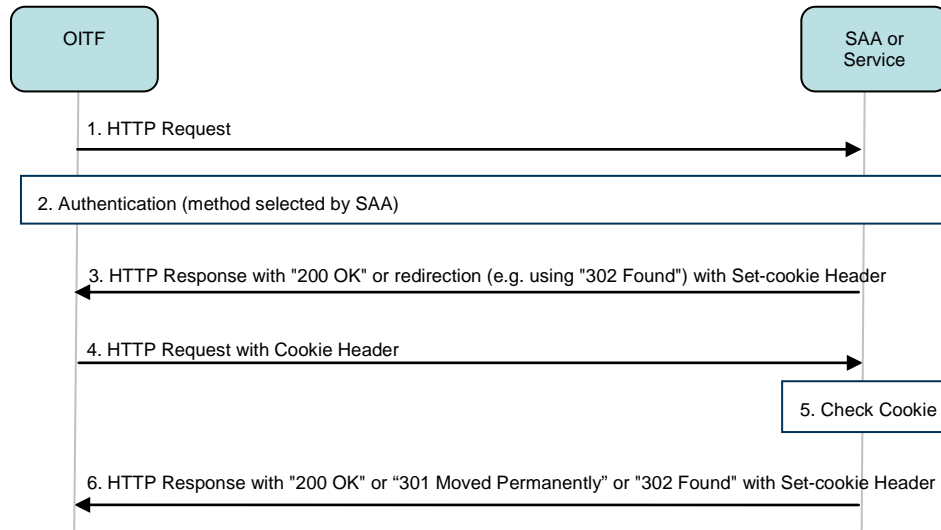


Figure 32: Session Management Using Cookie

1. The OITF requests a service with no valid cookie.
2. The service triggers the SAA authentication and the SAA performs the wanted authentication.
3. The service or SAA sets a cookie using Set-Cookie response header as specified in [RFC2109].
4. The OITF requests a service. Applicable cookies are provided in each HTTP request as specified in [RFC2109] (domain-match, port-match, path-match, Max_Age-match, etc.).
5. The service checks the cookie. Cookie checking is out of scope of this specification.
6. The service optionally refreshes the cookie and sets it again using Set-Cookie response header as specified in [RFC2109].

Steps 4 to 6 are performed for each new HTTP request according to cookie matching.

5.6.2 URL Parameters (informative)

An alternative to cookies for passing session data is the use of hidden input fields in forms or URL parameters in requests passed to the server. These mechanisms are transparent to the OITF. Below is an example message flow using URL parameters. Note that the use of hidden input fields can also be achieved with HTTP POST. The mechanism of using HTTP POST is not described in this section.

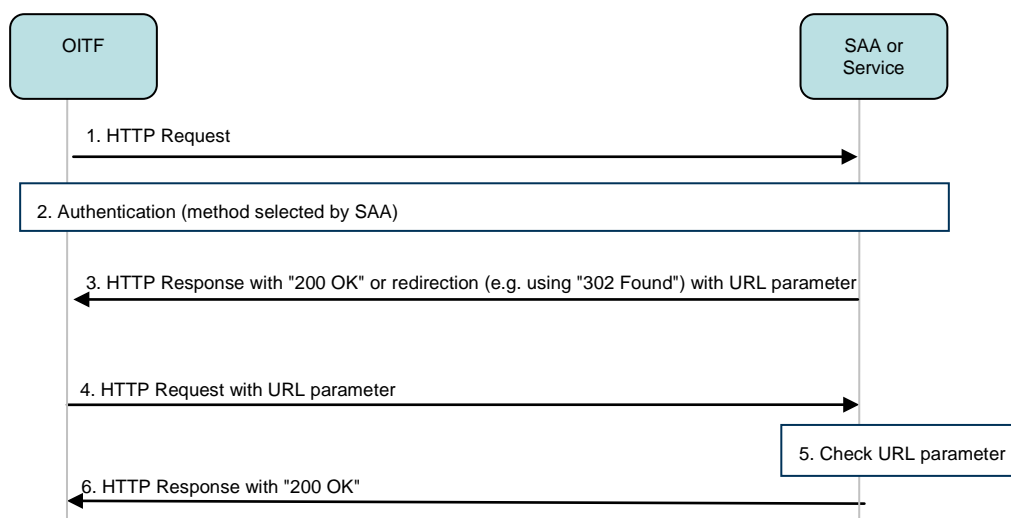


Figure 33: Session Management Using URL Parameters

1. The OITF requests a service with no valid authentication session.
2. The service triggers the SAA authentication and the SAA performs the wanted authentication.
3. The service or SAA redirects to the service with a new URL parameter for session data.
4. The OITF requests a service with the URL parameter.
5. The service checks the session data in the URL parameter. Session data is opaque data and out of scope of this specification.
6. The service serves the request.

NOTE: URL parameters are often used to pass session information from an HTTP session to a session using another protocol (e.g. RTSP).

NOTE: a web server (service or SAA) can maintain an HTTP session using this technique. But the server is responsible for modifying every link URL, so that the session data is posted in a form or appended to the request.

NOTE: Passing information through URL parameters is highly insecure.

5.6.3 HTTP Authentication Session

When using HTTP authentication, a server can rely on HTTP authentication session as specified in [RFC2617].

The User MAY be prompted to allow OITF to store HTTP authentication parameters, i.e. username and password, in non-volatile memory.

All OITF applications using HTTP (not only DAE) SHOULD have access to HTTP authentication parameters, i.e. username and password.

All OITF applications using HTTP (not only DAE) SHOULD share the current HTTP authentication session (e.g. B-TID, Ks_NAF, nonce, cnonce, nonce-count and opaque values).

If username and password can be stored, the user SHALL have the possibility to change stored username and passwords in OITF for a given protection space as specified in [RFC2617].

The following figure shows an example of sequences based on HTTP authentication session:

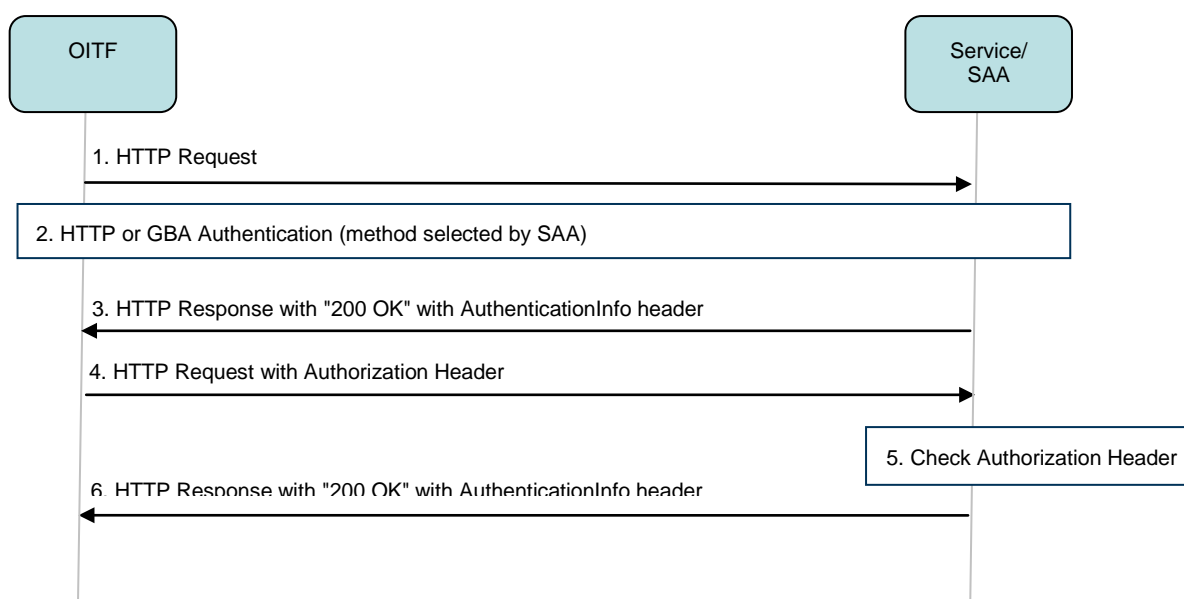


Figure 34: HTTP Authentication Session

1. The OITF requests a service with no valid HTTP authentication session.
2. The service/SAA performs HTTP or GBA authentication.
3. The service/SAA serves the request including an AuthenticationInfo header as specified in [RFC2617].
4. The OITF requests again a service. Appropriate HTTP Authorisation headers are provided in each HTTP request within the protection space (specified by domain) as specified in [RFC2617].
5. The Service/SAA checks the Authorisation header.
6. The Service/SAA serves the request including an AuthenticationInfo header.

Step 4 to 6 can be performed for each new HTTP request within the protection space.

5.6.4 SAML Web-based SSO

This section specifies the functionality and possible message flows for basic SAML web-based single sign-on.

SAML Web-based single sign-on SHALL adhere to section 4.1 of [SAMLPROF], whereby either a SAML HTTP POST or a SAML HTTP SimpleSign binding of a SAML <Response> message from the SAA SHALL use MIME-type “application/ce-html+xml” as defined in [CEA-2014-A]. A standard CEA-2014-A compatible browser is able to handle the SAML HTTP redirect and POST bindings defined in this section, without requiring any extensions to CEA-2014-A. This profile of SAML therefore does not add requirements to the OITF besides supporting DAE functionality.

The remainder of this section describes sequences of how SAML Web-based single sign-on is handled between the different relevant entities, i.e. the service, the SAA, and the OITF.

The sequences assume that the SAA and service provider share a logical identification of the user in advance of the described sequence. The user is known to the SAA. The SAA maintains knowledge of the user’s authentication credentials.

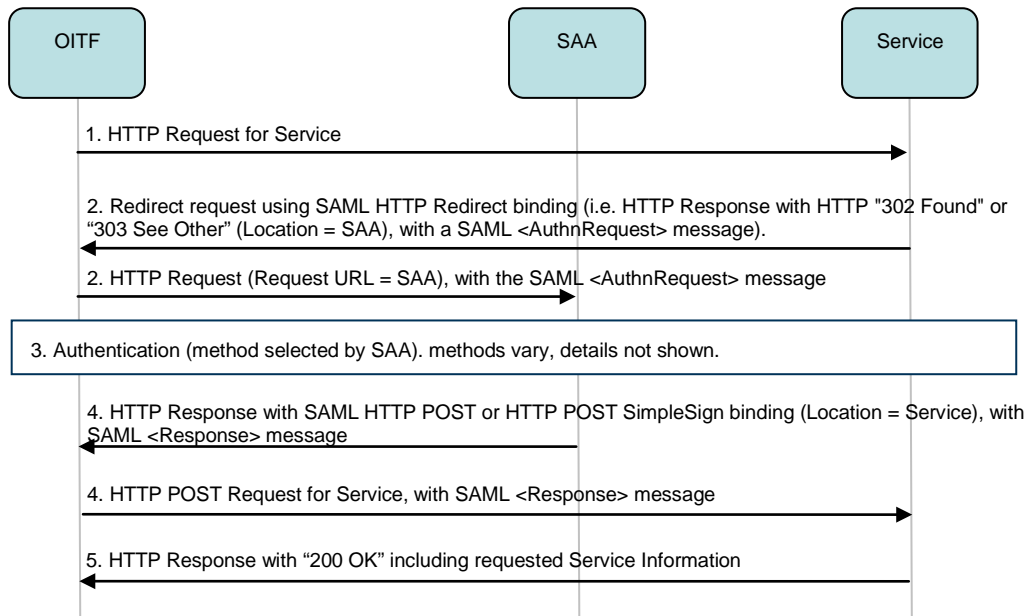


Figure 35: SAML Web-based SSO

1. The OITF requests a service. Authentication is needed and there is no valid authenticated service session.
2. The requested service triggers SAA authentication by issuing a redirect request using SAML HTTP Redirect binding, i.e. an HTTP Response with HTTP "302 Found" or "303 See Other" (Location = SAA) , with a SAML <AuthnRequest> message (as defined in section 3.4.1 of [SAML CORE]).
3. The SAA authenticates the user. Various methods exist for this. Valid methods include the authentication methods as defined in sections 5.4.1 through 5.4.5 of this document.
4. The SAA responds with either a SAML HTTP POST or HTTP POST SimpleSign binding of a SAML <Response> message (as defined in section 3.3.3 of [SAML CORE]). Since the browser of the OITF is CE-HTML compliant, the SAA response message must use MIME-type "application/ce-html+xml" as defined in [CEA-2014-A]. The CE-HTML browser will load the CE-HTML page with the SAML POST binding, after which it issues an HTTP POST request to the target service with the SAML <Response> message as payload.
5. The requested service checks the SAML <Response> message to see if authentication succeeded. If succeeded, the service serves the request.

6 Forced Play Out Using Media Zones

Content may contain navigation constraints for forced playout, see [OIPF_MEDIA2] sections 4.1 and 4.2.

If an OITF supports DMZ navigation constraints signalled in zone maps within MP4 files or MPEG-2 TS, it SHALL indicate this via the appropriate capability signalling [OIPF_DAE2]. If an OITF does not understand the navigation constraints, this capability description is either absent or set to “false”. If the capability description to support such DMZ navigation constraints is set to “true”, an OITF SHALL obey the signalled constraints and SHALL NOT ignore the presence of navigation constraints.

Note: When this capability description is not sent or is set to “false”, it is the choice of the service provider whether the content shall be sent to the OITF as there is no guarantee whether the navigation constraints will be obeyed.

For navigation constraints pertaining to protected content, the zone map information MAY be integrity protected using an included signature as described in [MRL_DMZ]. If the zone map is integrity protected using a signature, and if the terminal-centric approach is used for content protection, the key used for signature is derived as described in [MRL_DMZ] sections 2.1 and 2.3 for MP4 (using a key derived from the content key), and as described in [MRL_DMZ] section 7.2.2 for MPEG-2 TS (using a signature key signalled in ECMs). Note that the [MRL_DMZ] specification contains normative language on what should happen if the integrity of the signalled constraints cannot be verified. If an OITF supports DMZ navigation constraints and if integrity protection is used, the OITF SHALL verify the integrity of the signalled constraints, If the integrity of the signalled constraints cannot be verified, the OITF SHALL NOT play the associated content.

Note: Server-based play out control is described in [OIPF_PROT2], section 6.1.2.3. The concept there is applicable to interactive streaming where the server may or may not grant requests for trick-mode commands like fast forward.

Annex A Link of User Authentication and DRM Device Authentication (informative)

This section describes the generic mechanism to link user authentication result with device authentication in OITF. Although the device authentication mechanism is provided by Marlin, the user authentication mechanism varies depending on the system environment.

The mechanism described in this section uses HTTP Digest Authentication [RFC2617] and assumes that user identifier and its secret information (e.g. password, Ks_NAF) are shared between OITF and Providers Network in advance of the sequences between CSP and CSP-T Server.

The sequence below explains how the user authentication and device authentication are securely correlated with each other by Marlin Action Token Acquisition and Marlin Protocol.

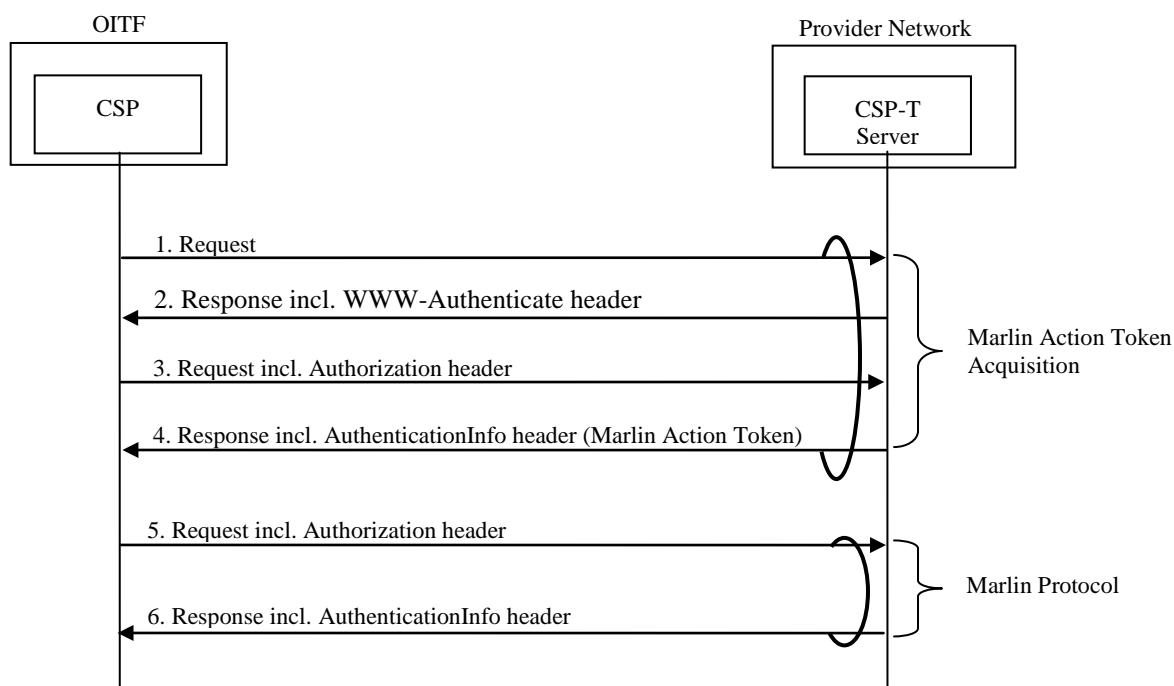


Figure 36: User Authentication for CSP, CSP-T Server communication

1. The CSP requests a Marlin Action Token to the CSP-T Server.
2. When the CSP-T Server receives the request from CSP for the Marlin Action Token, the CSP-T Server responds with a "401 Unauthorized" status code with a WWW-Authenticate header defined in [RFC2617].
3. When the CSP receives the response, the CSP sends the request which includes an Authorisation header defined in [RFC2617]. The user identifier and its secret information are used as username and password for generation of the Authorisation header.
4. When the CSP-T Server receives the Authorisation header,
 - The CSP-T Server verifies the Authorisation header.
 - When the verification succeeds, the CSP-T Server generates user information to be included into the Business Token, and stores the combination of user identifier from the Authorisation header and user information to be included into the Business Token.
 - The CSP-T Server then sends Marlin Action Token which includes the Business Token with AuthenticationInfo header defined in [RFC2617] to the CSP as the response.
5. Given the Marlin Action Token, the CSP sends a (Marlin Protocol) request to CSP-T Server which includes Authorisation header calculated from its username and password, and the Business Token.

6. When the CSP-T Server receives the Authorisation header in the (Marlin Protocol) request, which includes the Business Token,
 - The CSP-T Server verifies the Authorisation header.
 - When the verification succeeds, the CSP-T Server checks the combination of user identifier and Business Token in the request with its stored combination.
 - If the check succeeds, the CSP-T Server sends a (Marlin Protocol) response and correlates user identifier and its secret information (i.e. user authentication) with device identifier (i.e. device authentication).

Annex B XML Schemas

This appendix contains XML schemas relating to messages described in previous sections.

B.1 XML Schema for MarlinPrivateDataType Structure

This is the XML schema for MarlinPrivateDataType Structure (see section 4.1.7.2):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!-- schema filename is csp-MarlinPrivateData.xsd -->
<xs:include schemaLocation="csp-DRMPrivateDataType.xsd"/>
<xs:complexType name="MarlinPrivateDataType">
  <xs:complexContent>
    <xs:extension base="DRMPrivateDataType">
      <xs:sequence>
        <xs:choice>
          <xs:element name="MarlinLicense" type="xs:base64Binary"/>
          <xs:element name="MarlinToken" type="xs:base64Binary"/>
        </xs:choice>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

The DRMPrivateDataType structure is defined in the included file "csp-DRMPrivateDataType.xsd" as

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:complexType name="DRMPrivateDataType" abstract="true">
    <xs:attribute name="mimeType" type="xs:string" use="optional"/>
    <!-- NOTE: DRMPrivateDataType is an abstract type that can be extended and replaced
by a specific instance type to carry messages for a particular DRM system.
Derived types of <DRMPrivateData> should include an <any>
construct to be prepared for future extensibility, as is done for
example for <MarlinPrivateData> -->
  </xs:complexType>
</xs:schema>
```

B.2 XML Schema for MIPPVControlMessage Format

This is the XML schema for MIPPVControlMessage (see section 4.1.7.3.1):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oi pf: csp: MI PPVControl Message: 2008"
xml ns: tns="urn:oi pf: csp: MI PPVControl Message: 2008"
xml ns: xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!-- schema filename is csp-MIPPVControlMessage.xsd -->
<xs:element name="MIPPVControlMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MarlinLicense" type="xs:base64Binary" minOccurs="0"/>
      <xs:element name="MarlinActionToken" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="absoluteAcquisitionTiming" type="xs:dateTime"
                use="optional"/>
              <xs:attribute name="relativeAcquisitionTiming" type="xs:duration"
                use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

B.3 XML Schema for HexBinaryPrivateDataType Structure

This is the XML schema for HexBinaryPrivateDataType Structure (see 4.2.3.10.2):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xml ns: xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!-- schema filename is csp-HexBinaryPrivateDataType.xsd -->
<xs:include schemaLocation="csp-DRMPrivateDataType.xsd"/>
<xs:complexType name="HexBinaryPrivateDataType">
  <xs:complexContent>
    <xs:extension base="DRMPrivateDataType">
      <xs:sequence>
        <xs:element name="Message" type="xs:hexBinary"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

The DRMPrivateDataType structure is defined in the included file csp-DRMPrivateDataType.xsd and is shown in section B.1.

Annex C DRM Messages used in DAE (informative)

The following table summarizes the DRM messages and their MIME-types used in sendDRMMMessageAPI defined in [OIPF_DAE2].

Table 27: DRM Messages used in DAE

Mime-type	Description
application/vnd.marlin.drm.actiontoken+xml	Marlin Action Token defined by [MRL BNSP]
application/vnd.oipf.mippvcontrolmessage+xml	MIPPVControl Message as described in section 4.1.7.3.1
application/vnd.oipf.cspg-hexbinary	<p>CSPG-CI+ message as described in section 4.2.3.4.1.1.2, “Mapping of messages to DAE API or Events”</p> <p>Note: This type may also be used for CSPG messages in the case the gateway is combined with the OITF as introduced in section 4.2.</p>
application/vnd.marlin.drm.actiontoken2+xml	Marlin Action Token defined by [MRL MS3]

Annex D CSPG-CI+ Usage Examples (informative)

CI+ Host is the function in the OITF responsible for managing the dialog with the CSPG-CI+. CSPG-CI+ is referred to as CI+ CAM in [CI+] specifications. It is an internal function in OITF, not identified in [OIPF_ARCH2]. It is shown on the following sequence diagrams to help understanding of interaction with other identified functions.

Management of content protection using CSPG-CI+ has no impact on the protocols used for service discovery, Scheduled Content or CoD session establishment and management and delivery, as defined in [OIPF_PROT2]. CSPG-CI+ protected services can be Scheduled Content services or CoD services (multicast streaming, unicast streaming or download IPTV services). Following sequence diagrams are only examples of services.

D.1 CSPG-CI+ Initial Power-on (informative)

During initial power-on, the CSPG-CI+ and the OITF mutually authenticate each other using the CI+ authentication mechanism. Figure 37 is an overview of the mechanism. For further detail, please refer to [CI+], section 6.

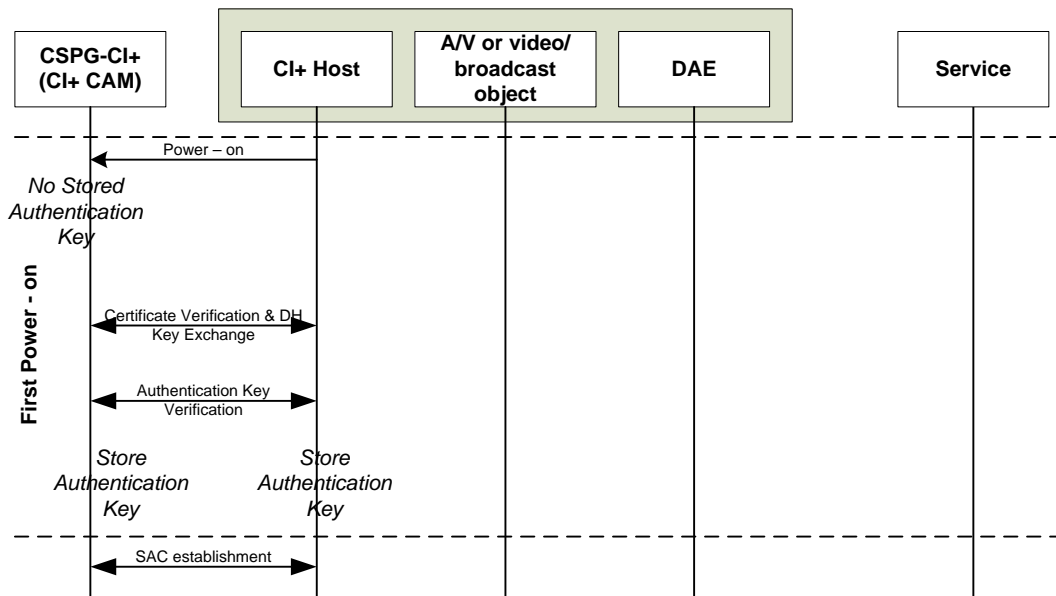


Figure 37: CSPG-CI+ First Power-on

D.2 CSPG-CI+ Normal Power-on (informative)

During initialization, if the CSPG-CI+ has stored authentication information, it only verifies that this authentication information is shared with the OITF. Figure 38 is an overview of the mechanism. For further detail, please refer to [CI+], section 6.

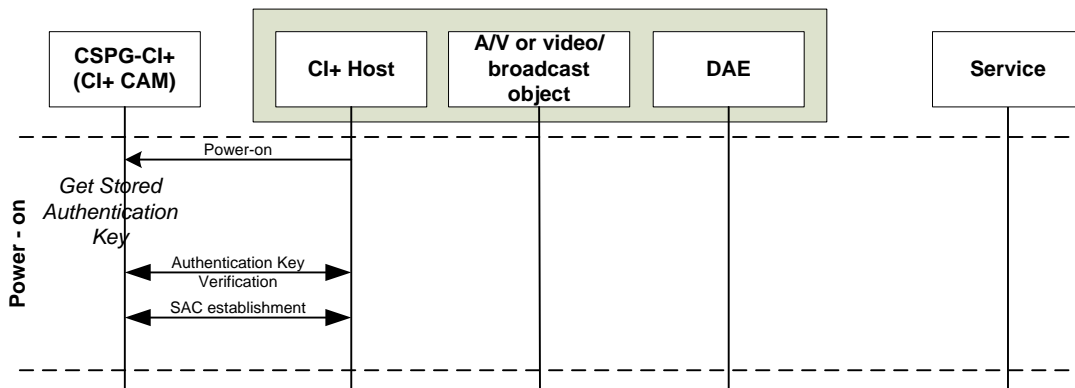


Figure 38: CSPG-CI+ Normal Power-on

D.3 Live Session Example (informative)

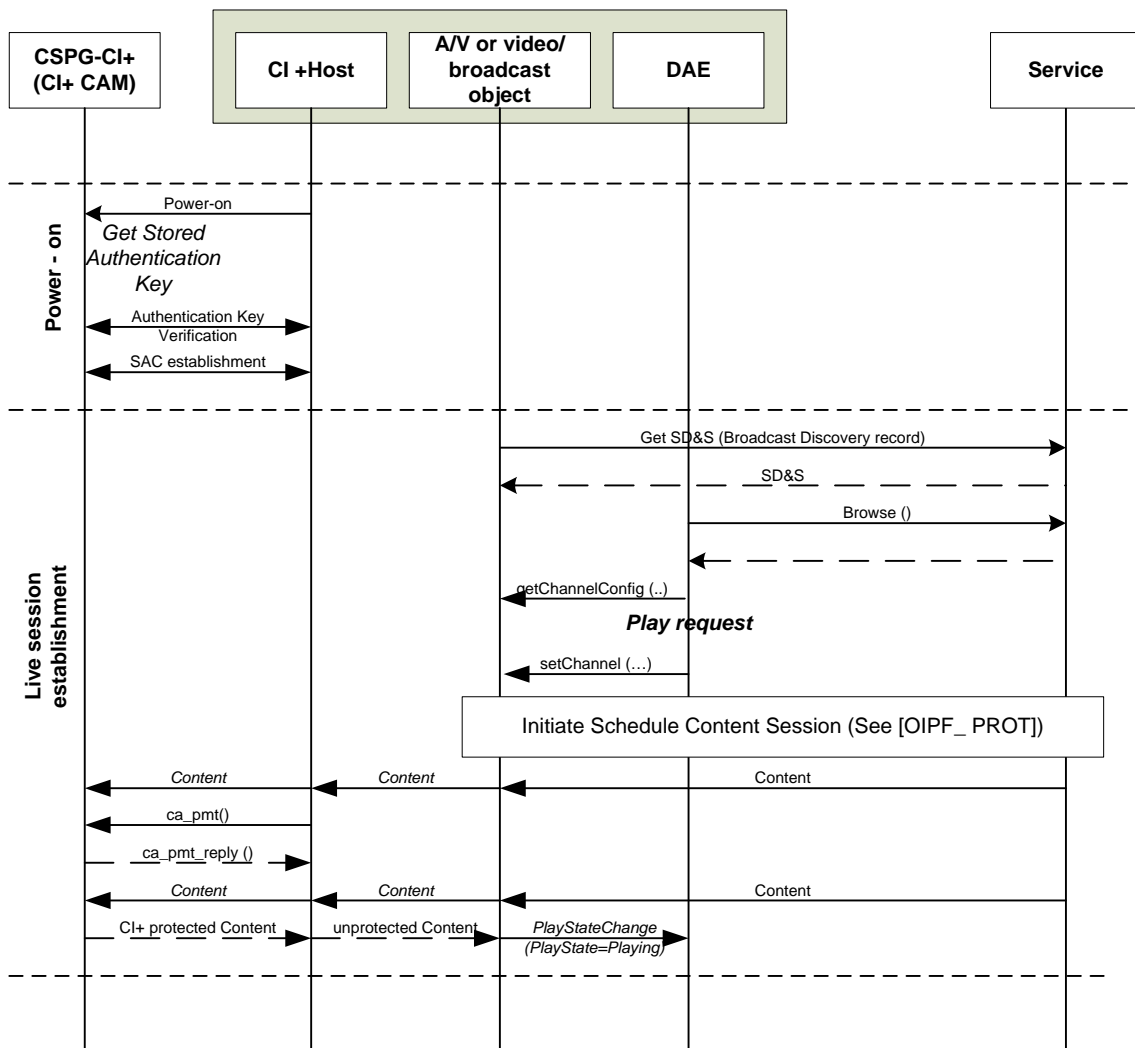


Figure 39: CSPG-CI+ Live Session Example

D.4 Parental Control Management Example (informative)

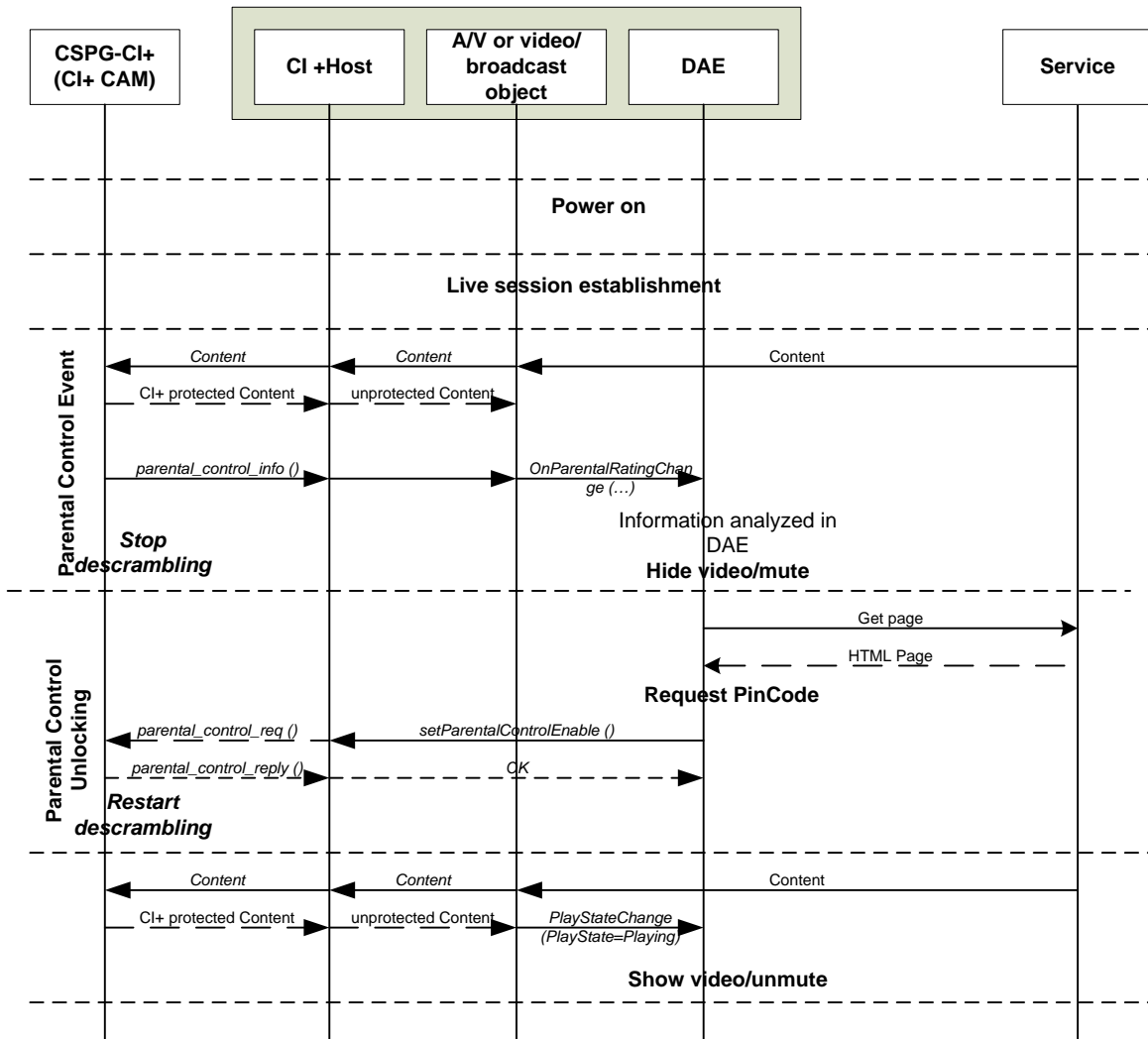


Figure 40: Parental Control Management Example

D.5 No Rights Event and Purchase Example (informative)

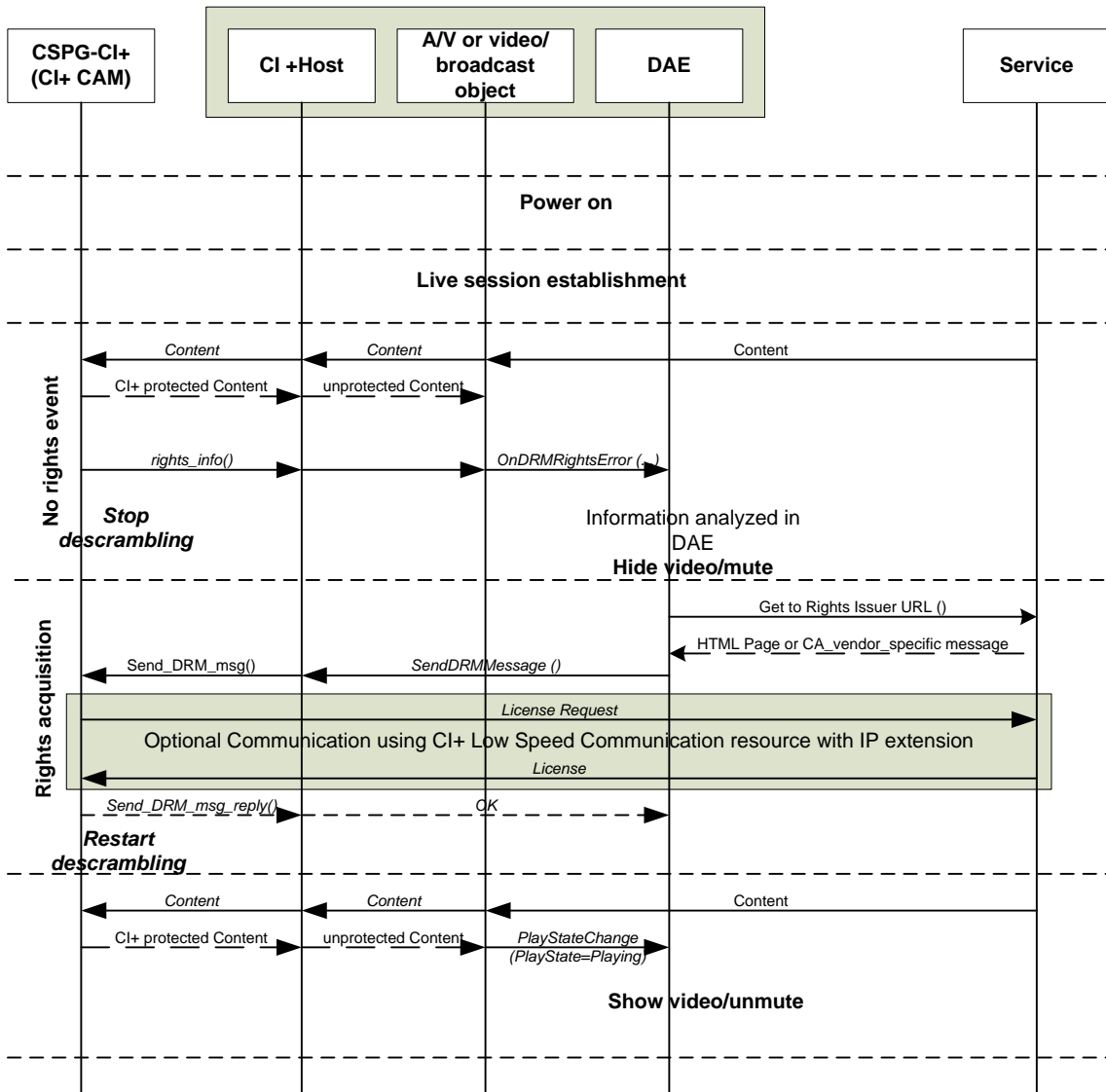


Figure 41: No Rights Event and Purchase Example

D.6 VOD Session Example (informative)

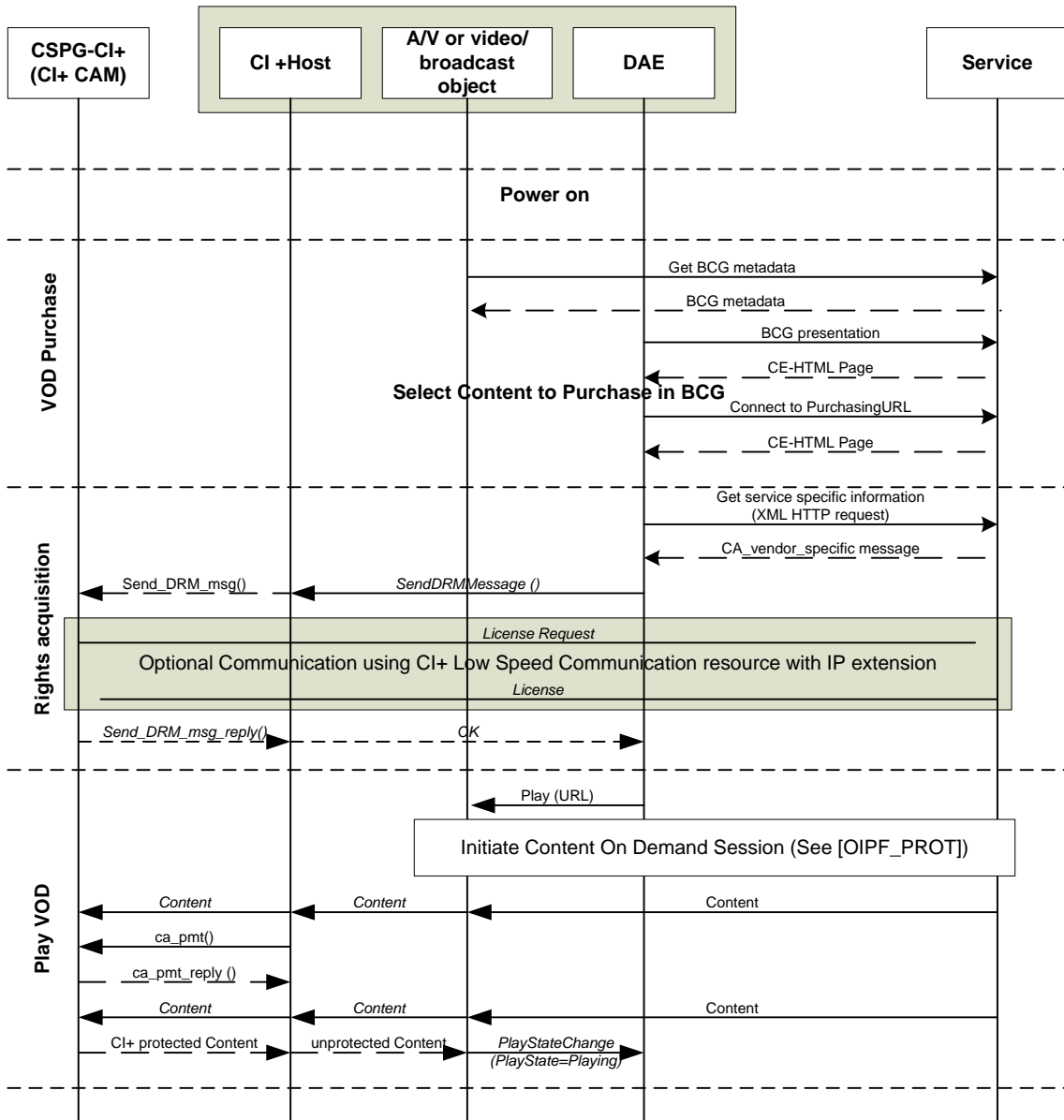


Figure 42: VOD Session Example

Annex E CSPG-DTCP Session Setup Sequence Examples (informative)

This appendix describes session setup sequences with CSPG-DTCP for following use cases:

- Multicast streaming with SIP session management,
- Unicast streaming with SIP session management,
- Unicast streaming with RTSP session management, and
- HTTP streaming and download.

Note that SIP messages over HNI-IGI (between OITF and IG) are delivered over HTTP as specified in [OIPF_PROT2].

E.1 Multicast streaming with SIP session management

Figure 43 describes session setup sequence for multicast streaming with SIP session management:

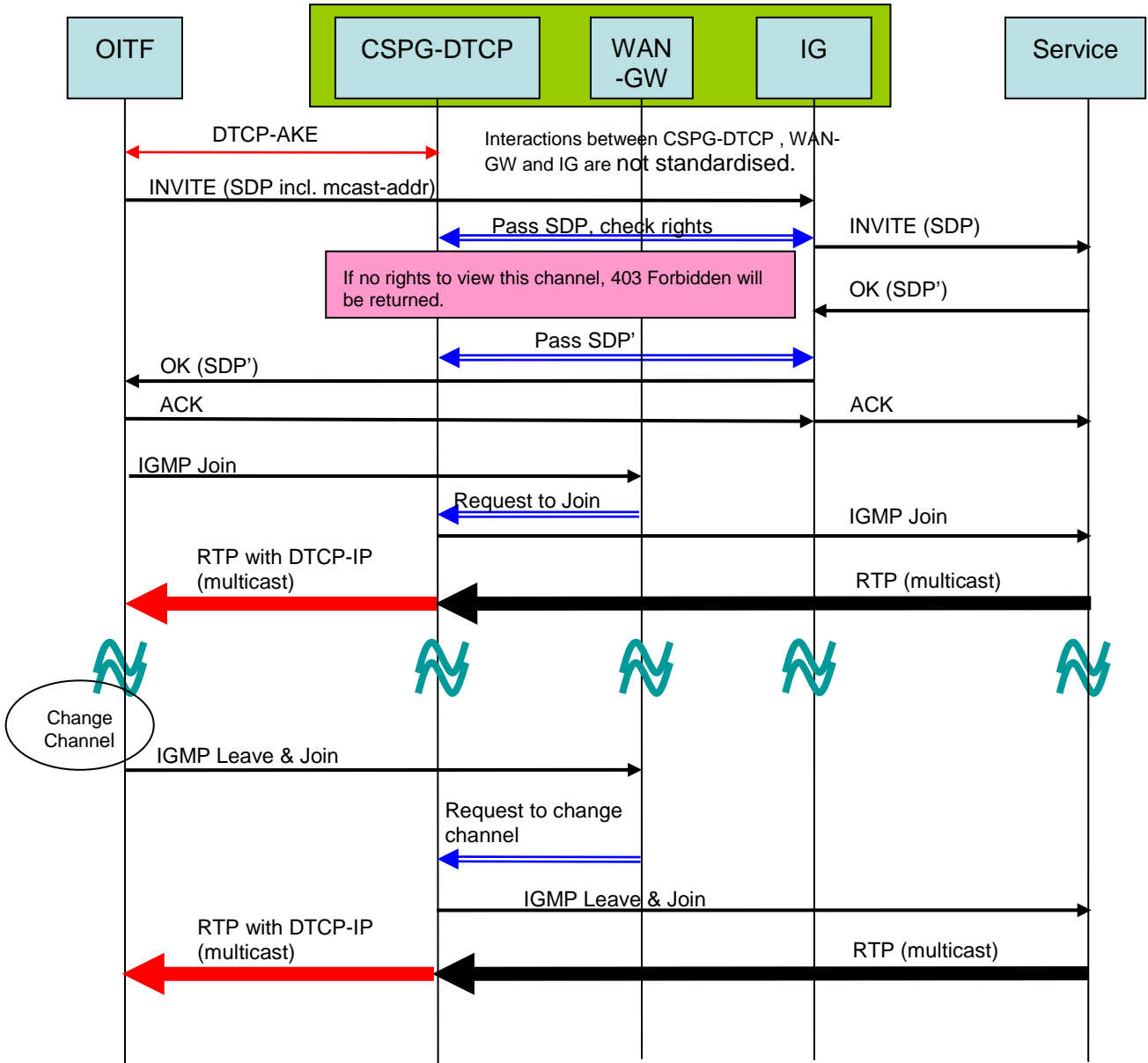


Figure 43: Session Setup Sequence for multicast streaming with SIP session management

Figure 44 describes CSPG-DTCP initiated teardown sequence for multicast IPTV service. Note that OITF and Network initiated teardown sequences are the same as defined in [OIPF_PROT2]:

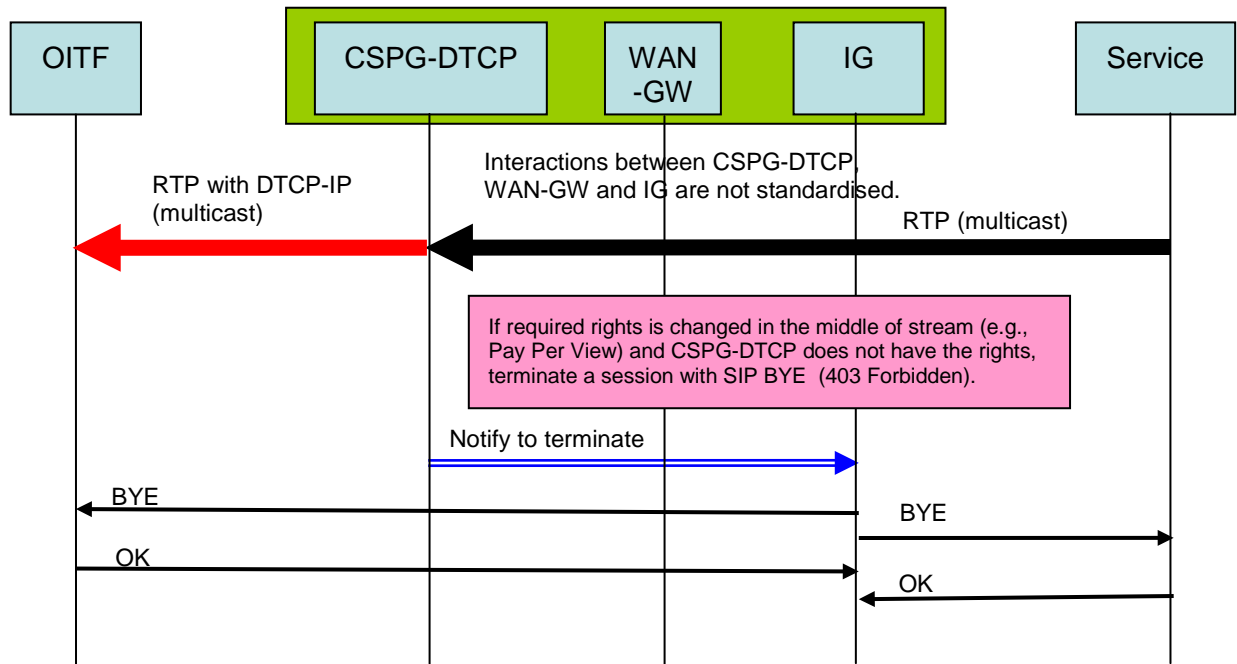


Figure 44: CSPG-DTCP Initiated Teardown Sequence for multicast streaming with SIP session management

E.2 Unicast Streaming with SIP session management

Figure 45 describes session setup sequence for unicast streaming with SIP session management:

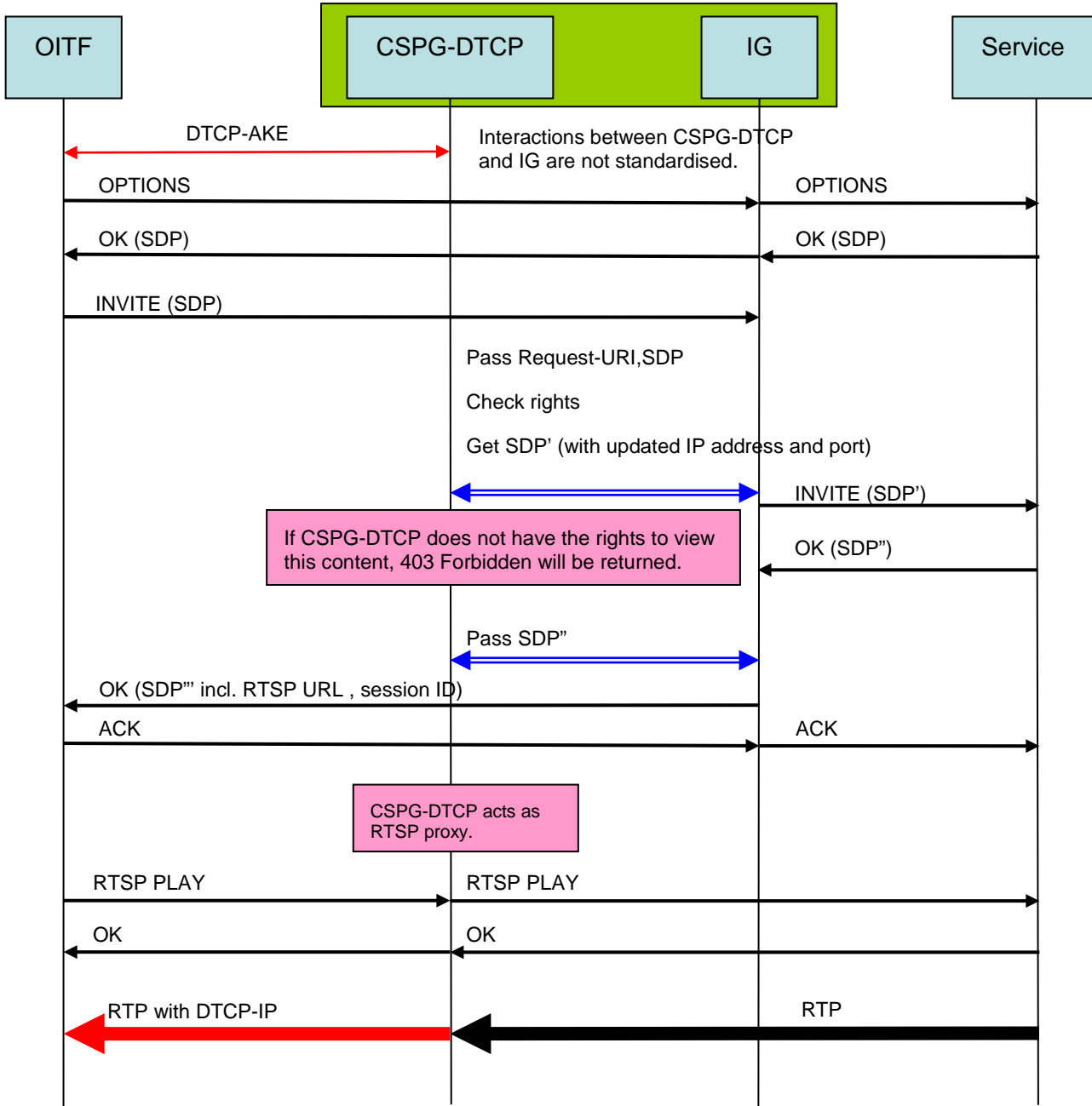


Figure 45: Session Setup Sequence for unicast streaming with SIP session management

E.3 Unicast Streaming with RTSP session management

Figure 46 describes session setup sequence for unicast streaming with RTSP session management:

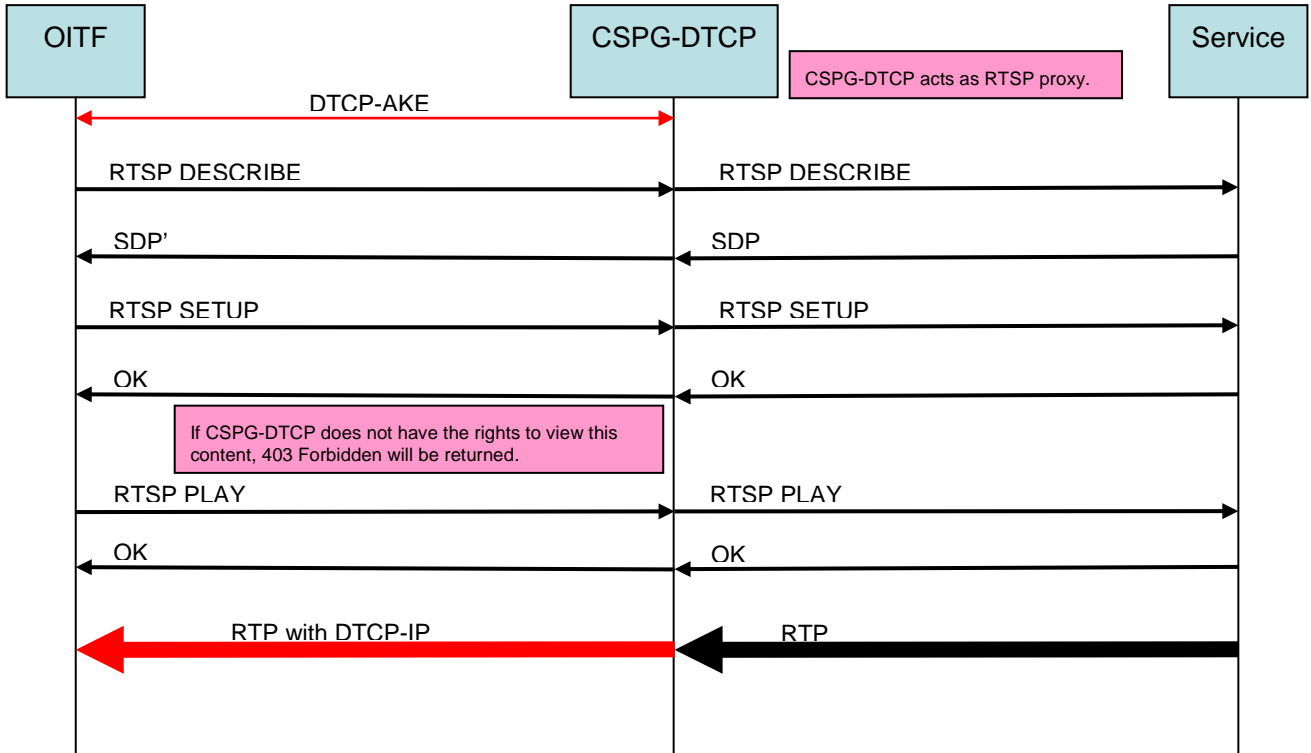


Figure 46: Session Setup Sequence for unicast streaming with RTSP session management

E.4 HTTP Streaming and Download

Figure 47 describes session setup sequence for HTTP streaming and download:

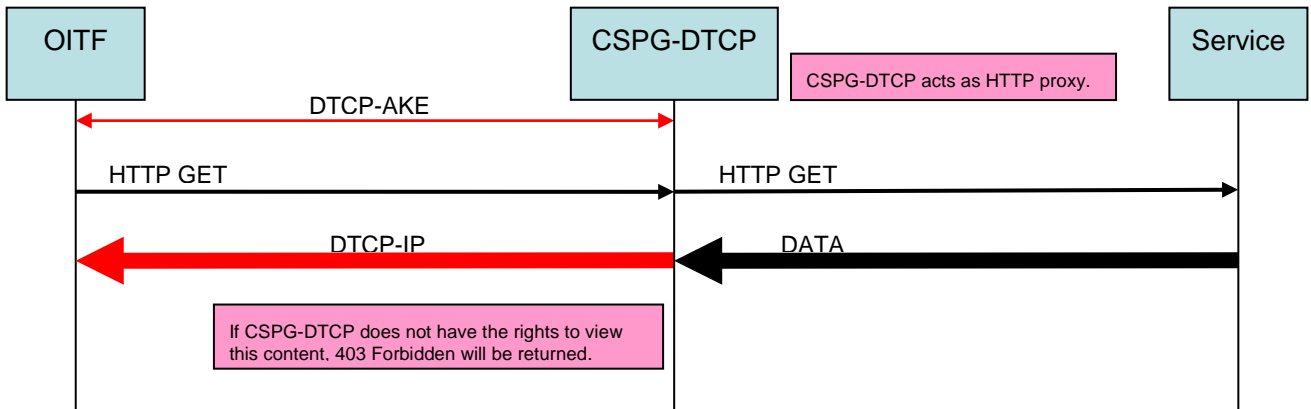


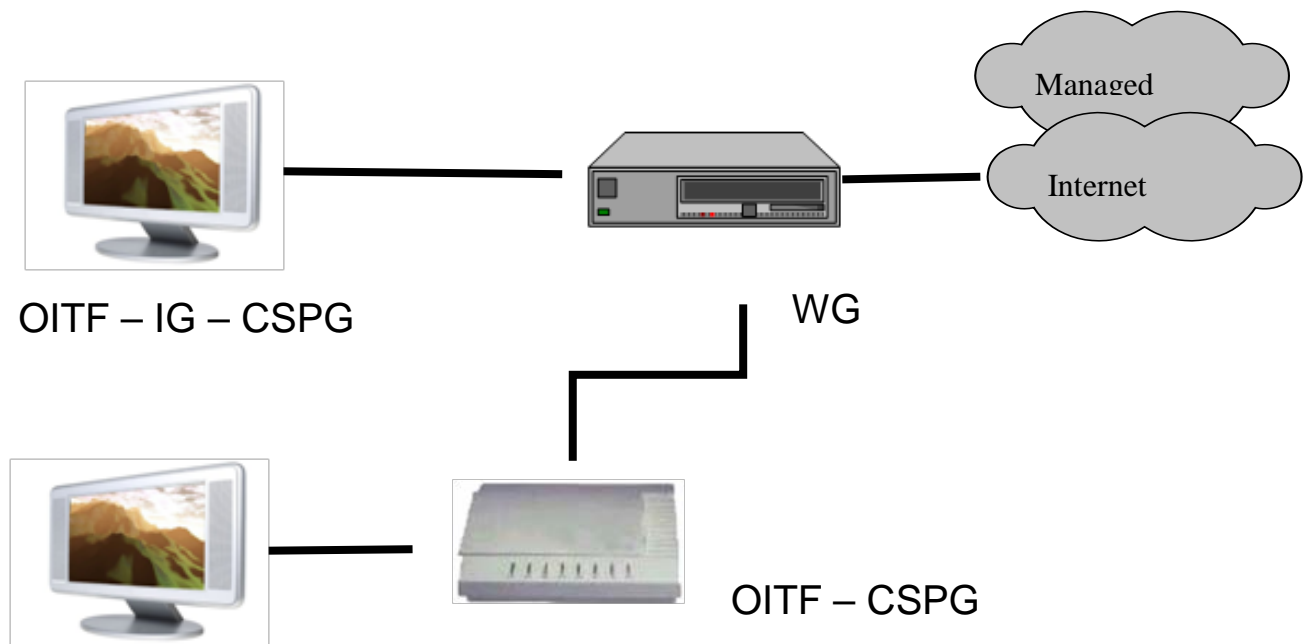
Figure 47: Session Setup Sequence for HTTP Streaming and Download

Annex F Embedded CSPG (informative)

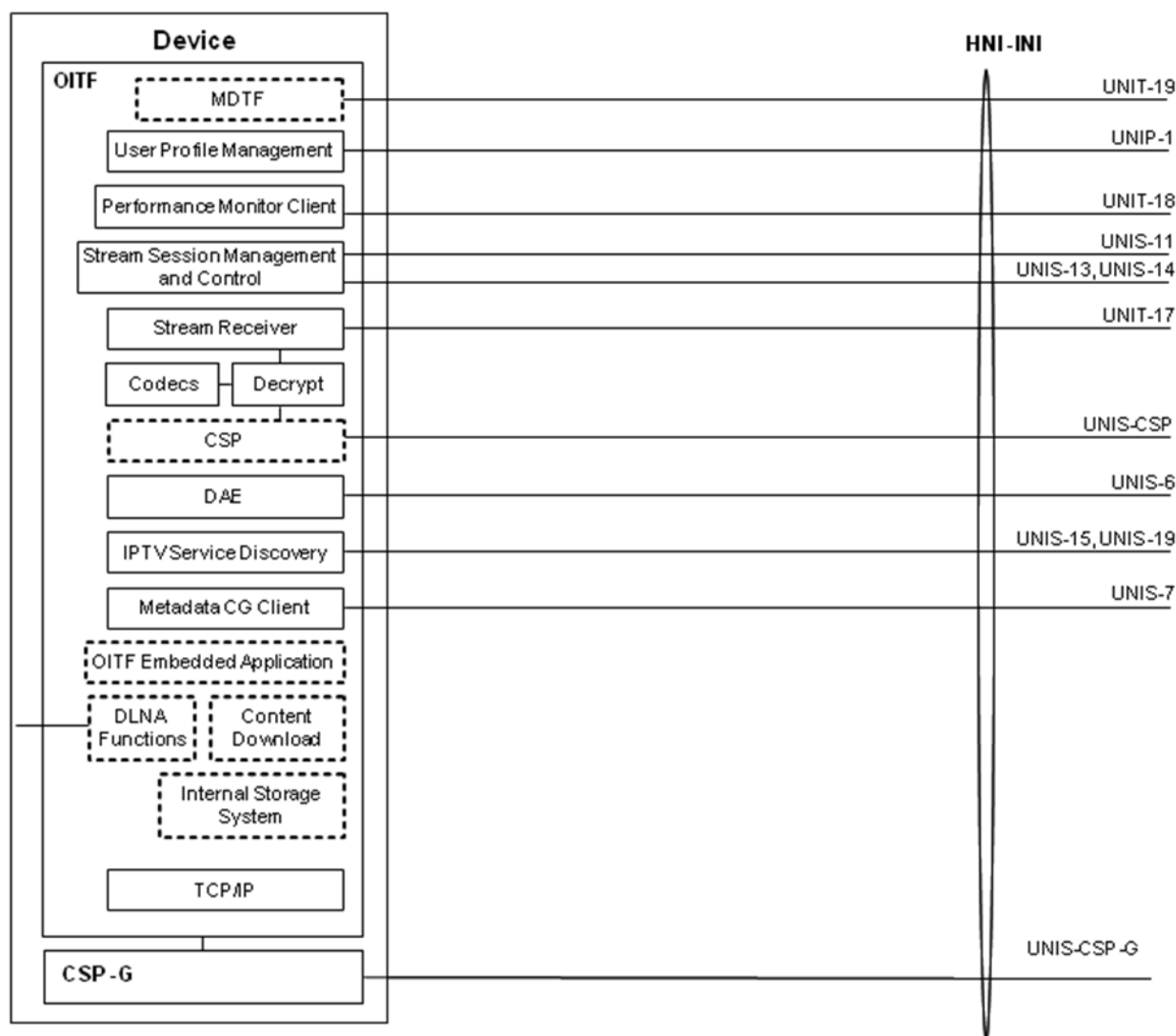
As introduced in section 4.2, the CSP Gateway-Centric Approach allows for co-location of an embedded (virtual) CSPG in the same physical device that hosts the OITF. This is a purely conceptual approach that in practice facilitates that the embedded CSPG can terminate any chosen CA or DRM solution deployed in that device, for the reception of services that implement the chosen CA/DRM system. This appendix provides an informative description of the embedded CSPG approach.

The following deployments are possible implementations of OIPF specifications:

- Combined IG, OITF and CSPG TV or STB: A TV or STB including IG, OITF and CSPG functionality.
- Combined OITF and CSPG TV or STB: A TV or STB including OITF and CSPG functionality.



For a CSPG embedded in the same device as OITF, the following figure applies:



As shown on the figure above:

- The interface between CSPG and OITF functions is internal to device implementation and is out of scope of the present specification.
- The external interfaces UNIS-xx between the device and the network which form the basis for network-device interoperability remain unchanged, compared to when CSPG and OITF are implemented in separate devices. Service behaviour remains the same when the CSPG is embedded.

With the embedded (virtual) CSPG approach, there is no normatively specified interface between the OITF and CSPG. The DAE application communicates directly with the chosen CA/DRM solution using the common DRM agent communication API's defined for the DAE [DAE section 7.6]. The embedded CSPG is not signalled as a CSPG implementation at all; all communications via the DRM agent API take place using the usual CA/DRM system identifier and DRM capability indication mechanisms, as described in section 4.2.1.

The following example shows the signaling for a device with both embedded (virtual) CSPG and (non-embedded) CSPG-CI+ capabilities as defined in section 9.3.10 DRM capability indication of [OIPF_DAE2].

Example: `<drm DRMSystemID="urn:dvb:casystemid:01535">TS_PF</drm>`
`<drm DRMSystemID="urn:dvb:casystemid:12348" protectionGateways="ci+">TS_PF TTS_PF</drm>`

For an embedded CSPG, the following interface mappings can be described:

- Connectivity and Discovery: internal to the device

- HNI-CSP:
 - Control Channel: The events and functions provided by the CSPG are mapped internally in the device to the DAE API and events, e.g. sendDRMMessage, onDRMMessageResult, onDRMRightsError, onParentalRatingChange or onParentalRatingError. This mapping is out of scope for the present specification.
 - Media Channel: This is an internal interface.
- UNIS-CSP-G: The device provides access to a network driver to the CSPG
- HNI-AGC: not used

F.1 Application to Simple and Secure Streaming

The call flow in Figure 48 depicts an example of sequence for getting credential and applies when secure streaming of content is triggered by the end-user.

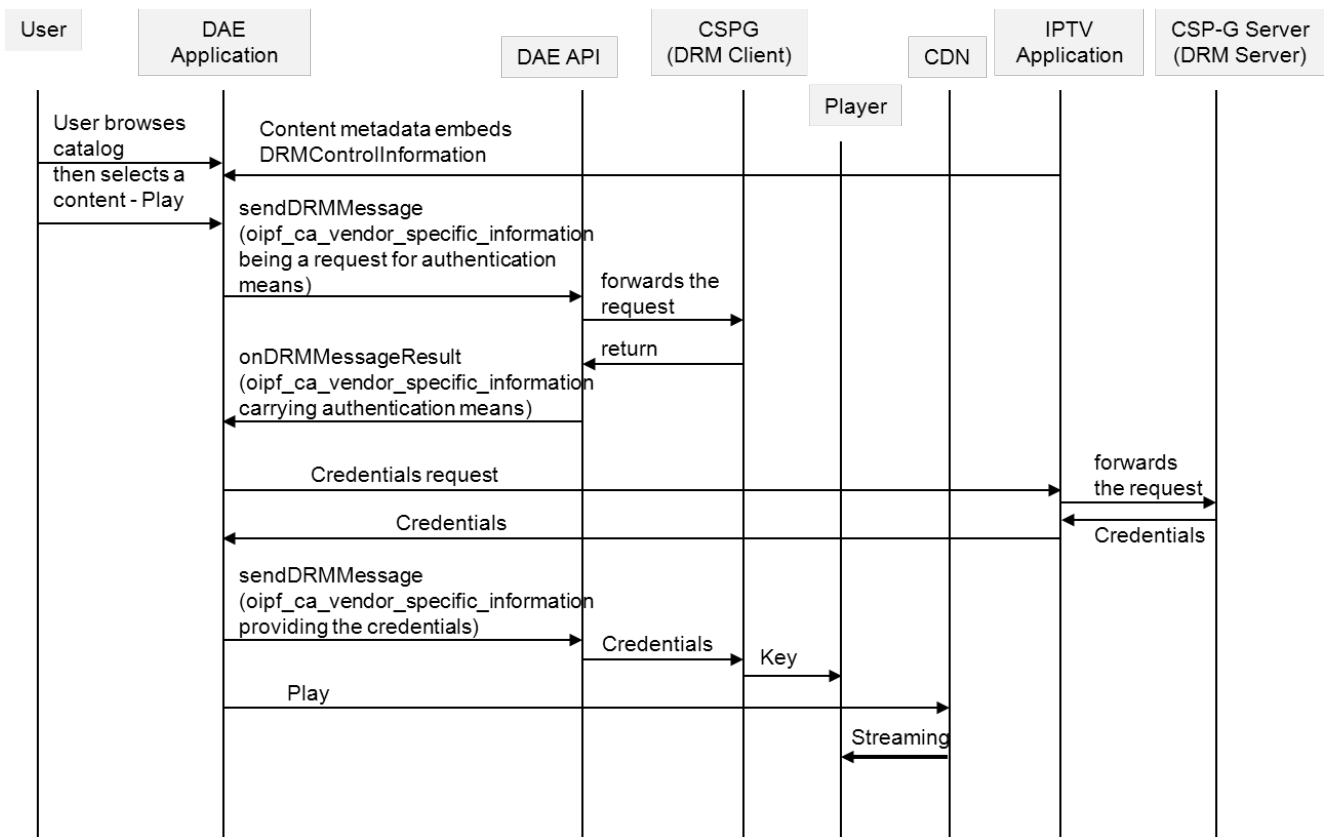


Figure 48: Simple and Secure Streaming with CSPG

1. The end-user browses the catalog and selects a content for playing.
2. The server provides DRMControlInformation associated to the content.
3. The DAE Application asks the DRM client to provide authentication means via the DAE API. Note that the full authentication sequence is not described here for simplification and because it is specific to the DRM implementation.
4. The DRM client replies to the requests via the DAE API.
5. The DAE Application requests credentials to the IPTV Application providing DRMControlInformation associated to the content and authentication means bound to the OITF DRM client.
6. The IPTV Application transmits the request to the DRM Server.

7. The DRM Server returns credentials to allow access to the content.
8. The IPTV Application returns the credentials to the DAE Application.
Note: the retrieval of credentials could also be performed through a proprietary communication between the CSPG and the CSP-G Server. In this case the reply to DAE application in step 4 above is limited to an acknowledgement that credentials are available for play.
9. The DAE Application provides the credentials to the DRM Client via the DAE API.
10. The DRM client checks the credential and extracts the content key. The DRM client provides the content key to the Player.
11. The DAE Application triggers the streaming of the content.
12. The CDN streams the content to the OITF. The decryption process associated to the Player uses the content key to render the content in clear.