ETSI TS 126 238 V15.1.0 (2018-07)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS);
LTE;
5G;
Uplink streaming
(3GPP TS 26.238 version 15.1.0 Release 15)

Reference

RTS/TSGS-0426238vf10

Keywords

5G,LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The Framework for Live Uplink Streaming (FLUS) is an enabler for live media streaming from a source entity to a sink entity. FLUS offers an IMS-based and a non-IMS-based instantiation. The IMS/MTSI-based instantiation enables the establishment of live media streaming between two UEs or between a source entity and a sink entity, within and across operator networks. Compared with MTSI, where limited types of QoS for speech or video media are used, FLUS can provide a wider range of QoS operation, e.g., in the maximum delay, available bandwidth or target packet loss rate.

In the non-IMS-based instantiation, it is possible to operate FLUS as a more generic framework that is controlled through a RESTful API and that supports other media plane protocols (i.e. not based on IMS or MTSI).

In addition to providing a wider range of QoS operation over radio links, other advanced functionalities of FLUS, such as the signalling of immersive media, can be used to complement existing 3GPP services.

1 Scope

The present document defines a FLUS source entity and a FLUS sink entity that can support point-to-point transmission of speech/audio, video, and text. It defines media handling (e.g., signalling, transport, packet-loss handling, and adaptation). The goal is to ensure a reliable and interoperable service with a predictable media quality while allowing for flexibility in the service offerings.

A FLUS source entity, which may be embedded in a single UE, or distributed among a UE and separate audio-visual capture devices, may support all or a subset of the features specified in this document.

When used as a generic framework, only the F-C procedures for establishing the FLUS session are required to be supported by the source and sink entities, and no other feature or procedure specified in this document is mandated. Impact on the service quality and network capacity is left to the discretion of the implementation and the service utilizing the framework. For example, configuration of media formats and codecs follows the requirements of the respective service.

When offered as part of a 3GPP IMS/MTSI service, the source and sink are required to support the IMS control plane and media plane procedures, and the service quality is determined by the MTSI service policy.

The specification is written in a forward-compatible way in order to allow additions of media components and functionality in releases beyond Release 15.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] Recommendation ITU-R BS.2051-1 (06/2017): "Advanced Sound System for Programme Production".

[3] ISO/IEC DIS 23090-2: "Omnidirectional Media Format. N16824".

[4] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[5] 3GPP TS 26.235: "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs".

[6] 3GPP TS 24.147: "Conferencing using the IM Multimedia (IM) Core Network (CN) subsystem; Stage 3”.

[7] 3GPP TS 23.003: "Numbering, addressing and identification".

[8] 3GPP TS 33.203: "3G security; Access security for IP-based services".

[9] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[10] 3GPP TS 33.328: "IP Multimedia Subsystem (IMS) media plane security".

[11] IETF RFC 7231: "Hypertext transfer protocol (HTTP/1.1): Semantics and Content".

[12] 3GPP TS 24.229: "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

FLUS session: A logical association between a source and a sink within which media content can be sent from the source to the sink.

Media session: A subset or part of a FLUS session including the duration to establish the media session, the time period during which media content can be sent from FLUS source to FLUS sink and the duration to terminate the media session. One or more media sessions are delivered during a FLUS session. A media session may be established and controlled by a well-defined control protocol.

Media stream: The content sent from a FLUS source to a FLUS sink within a media session.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

<symbol>	<Explanation>

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

FLUS	Framework for Live Uplink Streaming
HMD	Head Mounted Display
IMS	IP Multimedia Subsystem

4 System architecture

4.1 General

This clause introduces the system architecture for FLUS.

4.2 System

4.2.1 General

Figure 4.2-1 depicts the architecture showing the relevant entities for providing an uplink streaming service.

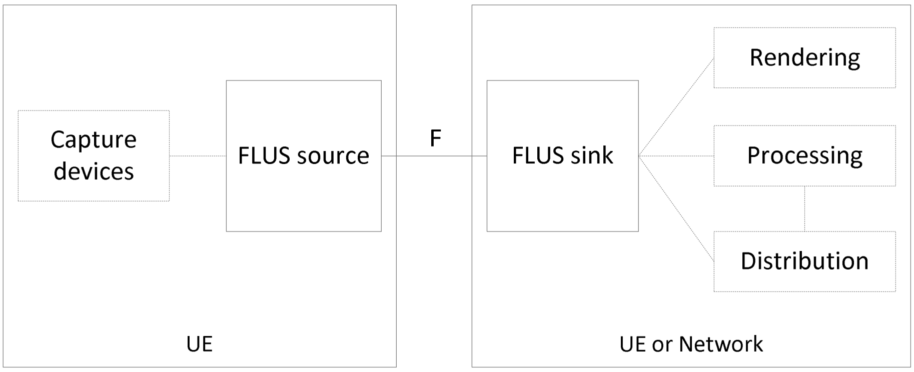


Figure 4.2-1: FLUS architecture

The uplink streaming service architecture is based on a FLUS source located in a UE and a FLUS sink located in either another UE or in the network.

The FLUS source receives media content from one or more capture devices. In the context of this specification, the capture devices are considered as parts of a UE or are connected to it.

When the FLUS sink is located in a UE, the FLUS sink shall forward media content to a decoding and rendering function.

When the FLUS sink is located in the network, the FLUS sink may forward media content to a processing or distribution sub-function. The processing and distribution sub-functions are not in scope of the present specification. The FLUS sink may act as a Media Gateway Function (MGW) and/or an Application Function (AF).

The F reference point connects a FLUS source and a FLUS sink. The F reference point enables the FLUS source to establish and control a single FLUS session.

The F reference point also enables the FLUS sink and the FLUS source to mutually authenticate and authorize each other.

The F reference point shall support security function for confidentiality protection of both the FLUS control plane (F-C) and FLUS user plane (F-U).

Details of the FLUS control plane and FLUS user plane functions are shown in Figure 4.2-2.

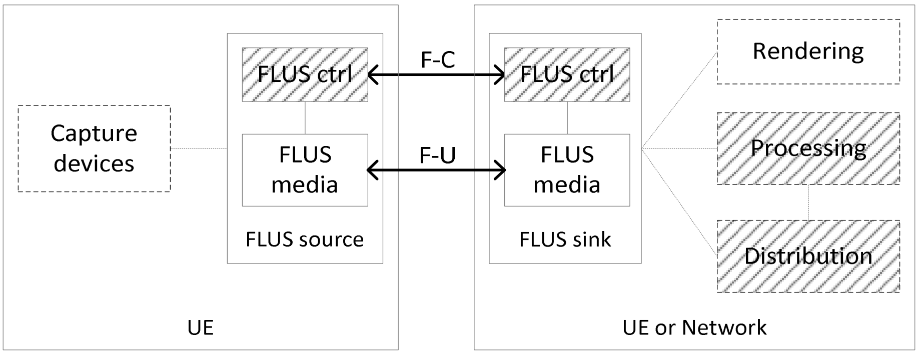


Figure 4.2-2: Sub-functions of FLUS

Note that in the above diagram, hatched-filled boxes represent FLUS control plane functionality, solid line boxes represent mandatory functionality, and dashed-line boxes corresponds to optional functionality. Also, note that F-C and F-U denote FLUS control and FLUS user plane functionalities, respectively, and do not represent reference points.

FLUS ctrl and F-C: FLUS control plane functionality including the associated processing by FLUS sink of the uploaded media for subsequent downstream distribution, plus FLUS media instantiation selection. F-C may also support configuration of static metadata for the session.

FLUS media and F-U: FLUS user plane functionality which includes setup of one or more media sessions and subsequent media data transmission via media streams. In some cases, a media session establishment protocol (e.g. IMS session set-up for MTSI-based FLUS instantiation) is necessary.

NOTE : F-C is not needed when the FLUS sink is an MTSI client that is only capable of rendering. In such event, logical control plane functions such as media and session descriptions and support for FLUS session establishment are encapsulated in FLUS user plane functionality.

F-C is used to establish and control the FLUS session. F-C allows the FLUS source to

- select a FLUS media instantiation,
- provision static metadata associated with each media session present in the FLUS session,
- select and configure the processing and distribution sub-functions.

The FLUS media instantiation is defined as part of the FLUS session. The user plane (F-U) may also contain the media stream establishment procedures when needed. Multiple media streams may be established for one FLUS session.

A media stream may contain media components of a single content type, e.g., audio, or media components of different content types, e.g., audio and video. A FLUS session may be composed of more than one media stream containing the same content type, e.g., multiple media streams of video.

4.2.2 Uplink streaming for MTSI

The architecture of uplink streaming for MTSI is depicted in Figure 4.2-3.

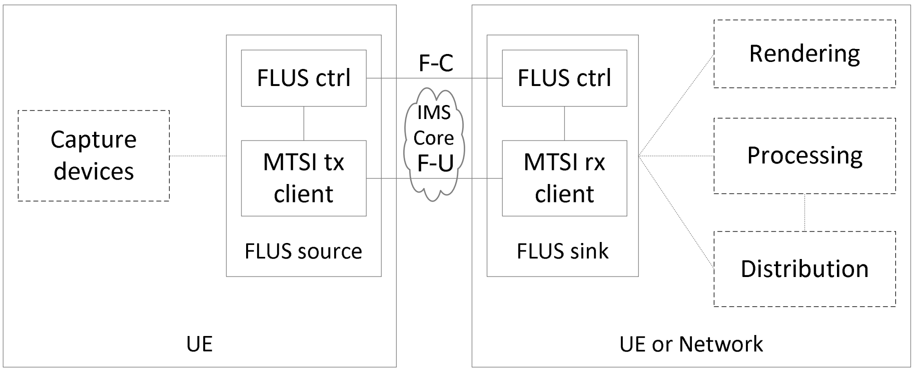


Figure 4.2-3: Uplink streaming for MTSI

The reception function of an MTSI client, i.e., the “MTSI rx client” as shown in Figure 4.2-3, is used to realize the FLUS media receiver components in the FLUS sink. The FLUS sink may be instantiated as another UE or as an MTSI rx client function in the network.

F-U contains all MTSI-related signalling.

The transmission function of an MTSI client, MTSI tx client, is used to realize the FLUS media sender components in the FLUS source.

4.2.3 Uplink streaming for PSS-based distribution

The architecture of uplink streaming for subsequent PSS distribution is depicted in Figure 4.2-4.

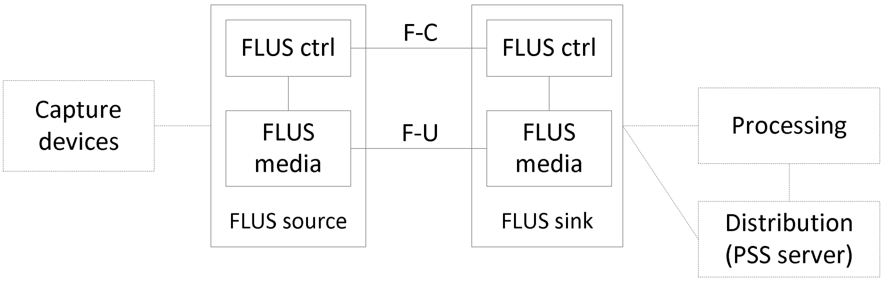
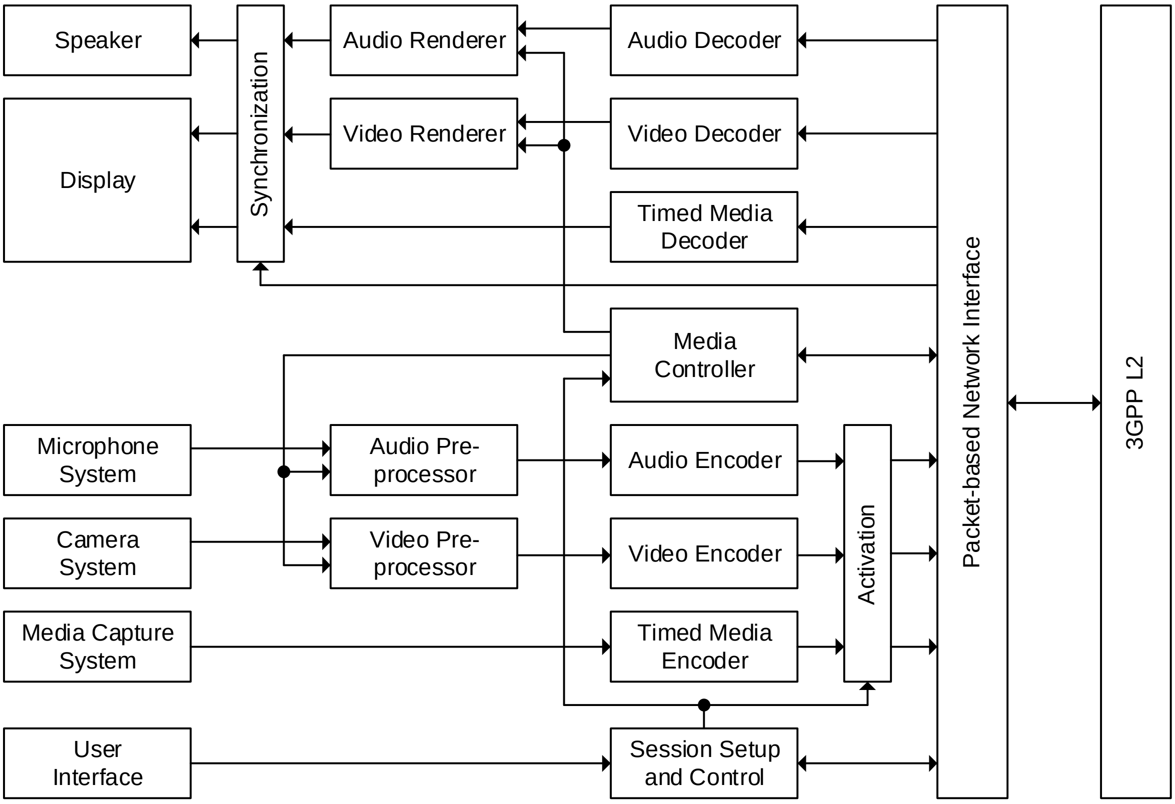


Figure 4.2-4: Uplink streaming for PSS

The PSS Content Source is located on the UE side and contains the FLUS source.

4.3 Terminal

The functional components of a terminal including both a FLUS source and a FLUS sink that uses 3GPP access are shown in figure 4.3-1.



NOTE 1: It is not required that all components of a FLUS source or a FLUS sink are included in a terminal.

Figure 4.3-1: Functional components of a FLUS source and a FLUS sink

The scope of the present document is to specify media handling and interaction, which includes media control, as well as transport of media and control data.

As indicated in figure 4.3-1, whether or not to employ or how to realize the audio/video renderer, media controller, or audio/video pre-processor is left to the configuration of a FLUS source and a FLUS sink, and their implementation. For these functional components, this document only defines the signalling of the media controller, which will be used to define the handling of media.

Timed media may include text, graphics, etc.

If a terminal including a FLUS source uses 3GPP access only for uplink media transmission, its functional components are shown in figure 4.3-2.

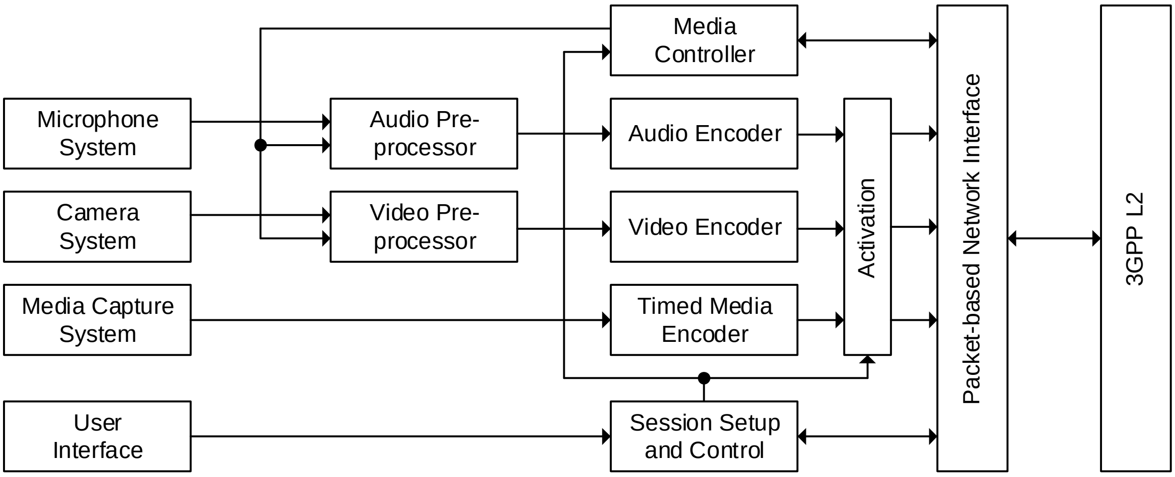


Figure 4.3-2: Functional components of a FLUS source for uplink media transmission

4.4 Procedures

4.4.1 General

A FLUS session may include one or more media streams. Media streams are time bounded to the FLUS session they belong to. When a media stream is active, the FLUS source can send media content to the FLUS sink.

Figure 4.4-1 depicts an example relationship of a FLUS session with a single media session containing two media streams.

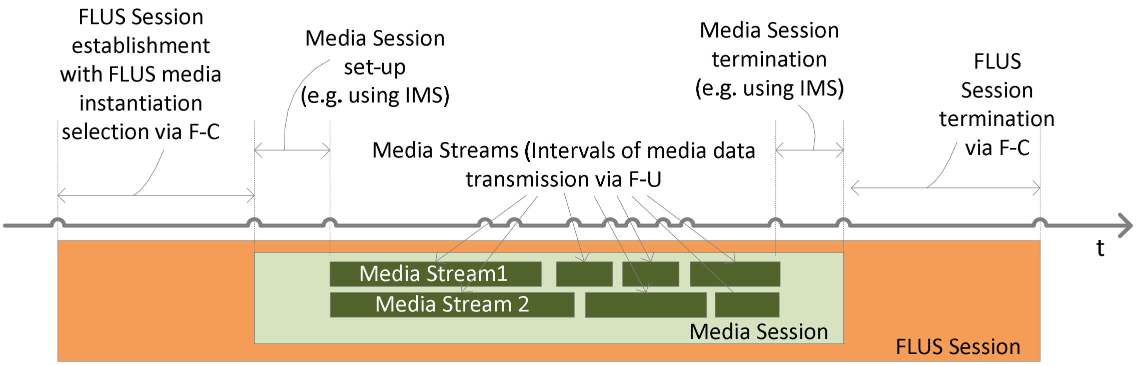

FLUS Session establishment with FLUS media instantiation selection via F-C
Media Session set-up (e.g. using IMS)
Media Session termination (e.g. using IMS)
FLUS Session termination via F-C
Media Streams (Intervals of media data transmission via F-U
t
Media Stream1
Media Stream 2
Media Session
FLUS Session

Figure 4.4-1: FLUS, media sessions and media streams

When the FLUS sink is located in a UE and the UE renders the received media content directly, the FLUS session may be implicitly present, e.g. it may be realized through IMS/MTSI.

When the FLUS sink is located on the network side and provides Media Gateway functionality, the FLUS session is used to select the media session instantiation and to configure any processing and distribution related sub-functions.

4.4.2 FLUS session establishment

It is assumed that the FLUS source has the necessary information to establish an F-C connection to a FLUS sink, e.g., in terms of a SIP-URI or an HTTP URL.

4.4.3 FLUS session update

The FLUS session establishment procedure creates a FLUS session resource, which is then configured through the FLUS session update procedure, for example, resulting in the selection of the media session instantiation.

The FLUS session update procedure includes the following FLUS session configuration parameters:

- Selection of the media session instantiation
- Provision of session specific metadata,
- Setting a description of the processing of the media data that the FLUS sink is to perform,
- Configuration of the distribution and storage options for the media data

4.4.4 FLUS sink capability discovery

This procedure allows a FLUS source to discover capabilities of a FLUS sink.

The FLUS sink capabilities include

- Processing capabilities
    - Supported input formats, codecs and codec profiles / levels, resolutions, frame rates
    - Transcoding with formats, output codecs, codec profiles / levels, bitrates, etc,
    - Reformatting with output format,
    - Combination of input media streams, e.g. network based stitching, mixing,
- Distribution capabilities
    - Storage capabilities,
    - CDN Origin capabilities and CDN Origin Server base URLs,
    - Forwarding, including supported forwarding protocol and associated security procedures

4.4.5 FLUS session termination

The FLUS source may explicitly terminate a FLUS session and all its provisioned and active media sessions. Alternatively, the FLUS session is automatically terminated, when the last media session of the FLUS session is terminated.

4.5 FLUS source systems

4.5.1 Introduction

FLUS supports different source systems. A source system describes the captured media sources and their inter-relationships, such that the FLUS sink can make use of the received media streams. FLUS can be deployed with source systems as defined in this specification, or with proprietary source systems. A source system is identified by a unique identifier that can be declared during the session establishment procedure.

A set of 3GPP-defined source systems is specified in this clause.

4.5.2 General source systems

A source system provides a collection of media streams that can be combined by the FLUS sink to create a full media experience. FLUS provides the ability to stream continuous media data from a source to a sink.

Media streams within one source system are time-synchronized and self-descriptive in the source system. Media streams are identified by a unique identifier in a source system.

Each media stream shall be self-declarative in the announced system.

Media streams may be of the following types: video, audio, text or metadata.

A source system may declare additional metadata in the namespace of the source system.

Table 4.5-1: General source system description

Source System	Description	IMS-based Instantiation SDP Parameters	Non-IMS-based Instantiation Parameters
Source System Identifier	A URI that uniquely identifies the source system	a=3gpp-flus-system:<urn>	JSON Object (SourceSystemIdentifier)
FLUS media stream to session mapping	Grouping description needed to identify the FLUS media streams when a FLUS session is part of an MTSI session	a=group:FLUS <mid_1> <mid_2>...	Not applicable
Configuration	Provides source system-specific configuration parameters for the source system.	a=3gpp-flus-configuration:<base 64 encoded>	Not applicable
Media Stream	Descriptions of the media streams that are defined by the source system	m= (one per stream)	Defined by the F-U instantiation
Identifier	A unique identifier shall be associated with every media stream. If not present, then the source system description should contain sufficient information to uniquely identify the media stream in the system.	a=mid:<media_id>	Defined by the F-U instantiation (may be implicit)
Stream Configuration	Describes the details of the media stream, including the encoding, the metadata, etc. The media stream is self-describing in the system. The definition of a single media stream, in the context of session description information, is provided during	a=3gpp-flus-media-configuration:<base 64 encoded>	Defined by the F-U instantiation

				session establishment, e.g. by a media line in SDP.		
		Bandwidth Requirement		Indication of the required network bandwidth to transport the media streams of the source system. Mandatory if QoS is required	b=AS:<bw> a=bw-info:	Defined by the F-U instantiation
		Transmission Direction		Optional indication of the transmission direction of the media streams	a=sendonly (mandatory)	Defined by the F-U instantiation
		Codec		Codec Identifier	a=rtpmap: (mandatory)	Defined by F-U instantiation.
		Codec configuration		Codec-specific configuration information for the set of media streams in the source system	a=fmtp:	Defined by F-U instantiation
			Media Type	Media type of each media stream: audio, video, text, etc...	m=<type> (mandatory)	Defined by F-U instantiation
			Media Transport and Control	Identification of the media transport protocol, operating over IP, for this media stream in the source system	m=<type> <port> RTP/AVP (mandatory) RTP/AVPF (if video rate adaptation is desired)	Defined by F-U instantiation.

4.5.3 Vendor-specific source system

A vendor-specific source system shall be identified by a unique vendor-specific urn identifier such as `urn:[com]:[vendor_x]:[system_y]`

4.5.4 Default source system

The default source system shall be identified by the unique identifier `urn:org:3gpp:flus:default:2017`. This source system is used for planar media formats.

4.5.5 3DOF FLUS source system

4.5.5.1 Introduction

The three-degrees of freedom system is defined to fully describe content that enables the user to look around from a single observation point in 3D space.

A 3DOF source system is identified with the unique identifier `urn:org:3gpp:flus:3dof:2017`.

The coordinate system is defined, as well as the basic structure to describe this source system.

4.5.5.2 Coordinate system

The direction in which a FLUS sink is interested can be represented using a coordinate system shown in figure 4.5-1, which consists of three axes, X, Y, and Z [3]. The X axis is equal to back-to-front axis, Y axis is equal to side-to-side (or lateral) axis, and Z axis is equal to vertical (or up) axis. Three angles, pitch, yaw, and roll, are measured around the X, Y, and Z axes respectively. As the torso is assumed to be fixed, the angles are assumed to be generated by the movement of the head. If seen from the origin to the positive direction of each axis, indicated by the arrows, each angle increases clockwise. The X-Z plane is aligned with the ground. When looking in the positive direction of Z axis, all angles are zero. The value ranges of yaw and roll are both −180.0, inclusive, to 180.0, exclusive, degrees. The value range of pitch is −90.0 to 90.0, inclusive, degrees.

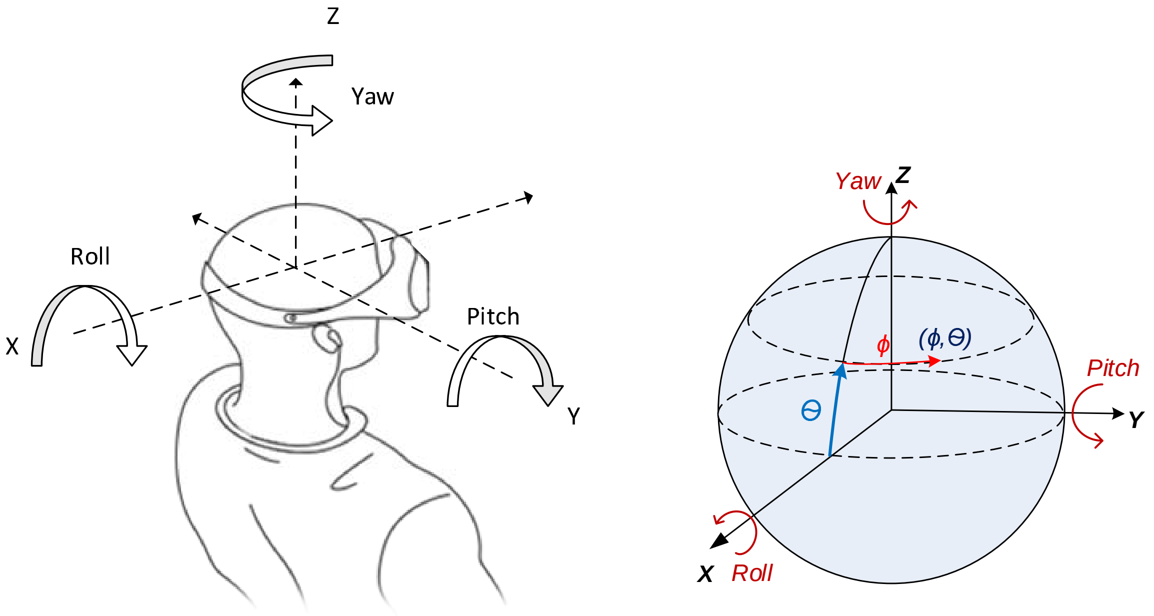


Figure 4.5-1: Coordinate system at media receiver

A source system is defined as a capturing device that has a center point from which the content is captured.

The captured content is represented in a reference system that enables a consumer of FLUS captured content to be able to look around from a single observation point in 3D space defined by the position of central capturing device(s).

This ability to look around and listen from a center point in 3D space is defined as 3 degrees of freedom (3DOF).

- Tilting side to side on the X-axis referred to as *Roll*ing
- Tilting forward and backward on the Y-axis, referred to as *Pitch*ing
- Turning left and right on the Z-axis, referred to as *Yaw*ing

It is worth noting that this center point is not necessarily static - it may be moving. The center point is defined by a physical location.

The content may be combined with simultaneously captured audio that may be 3D.

Additional timed media sources may be captured and streamed to the FLUS sink.

Signals may represent a 2D signal or a spherical video. Spherical signals defined in this specification are represented in a spherical coordinate space in angular coordinates (ϕ, θ) for use in omnidirectional video applications and 3D audio. The viewing and listening perspective is from the origin sensing/looking/hearing outward toward the inside of the sphere. The spherical coordinates are defined so that ϕ is the azimuth (longitude, increasing eastward) and θ is the elevation (latitude, increasing northward) as depicted in figure 4.5-1.

Depending on the applications or implementations, not all angles may be necessary or available at the FLUS source. The 360 video and the 3D audio may have a restricted coverage. For example, it may not be possible for a FLUS source to control media encoding parameters related to roll or pitch, e.g., when a wheel-shaped camera with a series of lenses is located on a flat surface, in which case only yaw needs to be fed back. The means to measure these angles is outside the scope of this document.

For video, such a center point may exist for each eye, referred to as stereo signal. And obviously, the video consists of three color components, typically expressed by the luminance (Y) and two chrominance components (U and V).

Although this coordinate system is defined for FLUS sink, and a FLUS source may use a typical spherical coordinate system as defined in [2], where the layout of loud speakers is represented with azimuth and elevation, the FLUS source should take appropriate actions based on the geometric information it receives from the far-end. If such a spherical coordinate system is used, the direction at the FLUS source, when both azimuth and elevation are zero, corresponds to the direction at FLUS sink when pitch, yaw, and roll are all zero.

4.5.5.3 Descriptive Parameters

In order to describe the media and metadata components of the system, the description language needs to provide information on the source system as well as each media component. The 3DOF FLUS system is described in Table 4.5-2.

Table 4.5-2: 3DOF FLUS source System Description

3DOF FLUS source system			Cardinality	Description
	Source System Identifier		URI	Identifier of the source system as a URI, and in this specification shall be set to: `urn:org:3gpp:flus:3dof:2017`
	Description			Detailed description of the source system including static metadata, etc. , and which shall be conveyed in a separate namespace from that of the Source System Identifier.
	Video Stream		0 … 1	At most one video stream is contained. If present, the media stream is identified by the media type “video”, such that no additional identifier is needed. The video stream shall contain sufficient information to be self-declarative in the coordinate system as defined in clause 4.5.5.2.
		Identifier	0 … 1	Unique identifier of the video stream in the source system. If not present, the source system should contain sufficient information to uniquely identify the video stream in the source system.
		Type	1	Provides the type of the contained video stream - 360 mono: The video represents a single spherical video. - 360 stereo: The video contains two spherical videos, one for each eye. - Unknown: will be provided by source system.
		Description	1	Describes the details of the video stream including the encoding, the metadata, etc. The media stream is self-describing in the source system.

	Audio Stream		0 … 1	If present, the media stream is identified by the media type "audio", such that no additional identifier is needed. The audio stream shall contain sufficient information to be self-describing in the coordinate system as defined in clause 4.5.4.2.
		Identifier	0 … 1	Unique identifier of the audio stream in the source system. If not present, the source system should contain sufficient information to uniquely identify the audio stream in the source system.
		Type	1	Defines the type of the contained audio stream: - 2D: the audio describes a planar version of the surrounding audio. No heights are included. - 3D: the audio includes 3 dimensions with heights. - Unknown: will be provided by source system.
		Description	1	Describes the details of the audio stream including the encoding, the metadata, etc. The media stream is self-describing in the source system
	Other Media Stream		0 … N	If present, each media stream shall include an identifier that provides an exact description of each (non- video or audio) media stream. Note that metadata is also considered as a separate media stream. Each such media stream shall contain sufficient information to be self-describing in the coordinate system defined in clause 4.5.5.2.
		Identifier	1	Unique identifier (for each instance) of the media stream in the source system. The identifier value shall be provided to describe the metadata. Examples for metadata description are 4CC codes in the file format.
		Description	1	Describes the details of this media stream including the encoding, the metadata, etc. The media stream is self-describing in the source system

5 Protocols

5.1 General

Protocol instantiations that can be configured by the functionalities in clause 7 may be used in the context of this specification.

5.2 IMS-based system

5.2.1 System configuration

5.2.1.1 Introduction

In an IMS-based FLUS system, the FLUS control plane or F-C enables the configuration and selection of a FLUS sink for the reception of media streams delivered over the FLUS user plane or F-U during a FLUS session.

5.2.1.2 FLUS sink configuration and selection

The means for discovery or configuration, and selection by the FLUS source of the FLUS sink depend on the deployment scenario and location of the FLUS sink, as described in sub-clauses 5.2.1.2.1 and 5.2.1.2.2.

5.2.1.2.1 UE-based FLUS sink

In this deployment scenario, the FLUS sink is a directly-targeted recipient of the media content streamed by the FLUS source, The FLUS source is expected to be aware of the identity of the FLUS sink, in the form of a SIP URI, which will be included in the Request-Line of the SIP INVITE message. There is no need for a separately-defined FLUS sink configuration and selection procedure in the delivery of the SIP INVITE from the source to the sink.

5.2.1.2.2 Network-based FLUS sink

In this deployment scenario, the FLUS sink is a network ingest server that performs post-processing of the media streams(s) uploaded from the FLUS source, for subsequent distribution to the recipient UEs. It acts as a SIP Application Server (AS) in the IMS network architecture, and its identity shall be in the form of a "FLUS Factory URI.

The FLUS source may be pre-provisioned with the FLUS Factory URI of the FLUS sink. In that case, the FLUS source shall send the INVITE message with the Request-Line-URI set to the value of that URI. If the FLUS Factory URI is not pre-provisioned in the UE, the UE shall construct a default FLUS Factory URI (similar to the procedures defined in clause 13.10 of TS 23.003 [7] regarding default Conference Factory URI construction) in either of the following formats:

a) sip:flus@flus-factory.operator.com, when the UE of the FLUS source contains the ISIM application and assumes that the name of its home network domain is 'operator.com', or

b) sip:flus@flus-factory.ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org, when the UE of the FLUS source does not contain the ISIM application, and has a home network domain name of 'ims.mnc<MNC>.mcc<MCC>.3gppnetwork.org'.

Such a FLUS Factory URI shall in turn be used by the S-CSCF to select an appropriate FLUS sink (AS) to which the INVITE message from the FLUS source is routed for IMS session establishment.

5.2.1.3 FLUS management object

The UE may be pre-provisioned with a FLUS sink URI through a 3GPP FLUS Management Object.

The FLUS Management Object shall have a Management Object Identifier: “urn:oma:mo:ext-3gpp-flus:1.0”. The MO shall be compatible with OMA Device Management protocol specification version 1.2 and above as described in [OMA-ERELD_DM-VI]. The following nodes for FLUS configuration are defined:

Node: /<*X*>

This interior node specifies the unique object id of a FLUS management object. The purpose of this interior node is to group together the parameters of a single object.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

The following interior nodes shall be contained if the FLUS sink in the terminal supports the FLUS Management Object.

/<X>/Sink/<X>

This node is a collection of information about a FLUS sink

- Occurrence: OneOrMore
- Format: node
- Minimum Access Types: Get

/<X>/Sink/<X>/SIP_URI

This leaf node provides the SIP URI for the FLUS sink that is described by the parent node.

- Occurrence: One
- Format: string
- Minimum Access Types: Get

/<X>/Sink/<X>/Capabilities

This leaf node provides a URL to a JSON or XML document that describes the capabilities of the FLUS sink. The format of the document is outside the scope of this specification.

- Occurrence: ZeroOrOne
- Format: string
- Minimum Access Types: Get

The structure of this FLUS MO is depicted in Figure X.

/<*X*>/Sink/<*X*>/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<*X*>/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

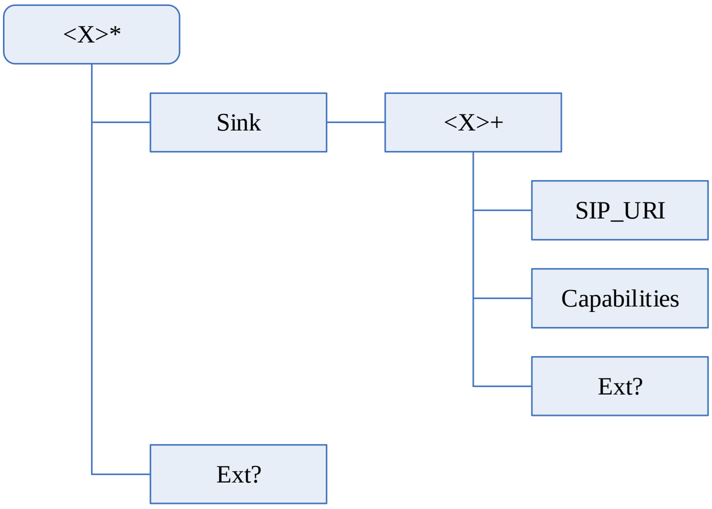


Figure 5.2-1: FLUS management object tree

5.2.2 Session management

The IMS-based FLUS System uses the SIP protocol for all session management. The FLUS Sink and FLUS Source shall support the IMS procedures as defined by TS 24.229 [12].

5.2.3 Data transport

The FLUS Source and Sink that implement the IMS-based FLUS system shall support data transport as specified in section 7 of TS 26.114 [4].

5.3 Generic FLUS system

5.3.1 System configuration

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

5.3.2 Session management

5.3.2.1 FLUS sink capability discovery

This procedure allows a FLUS source to discover capabilities of the FLUS sink.

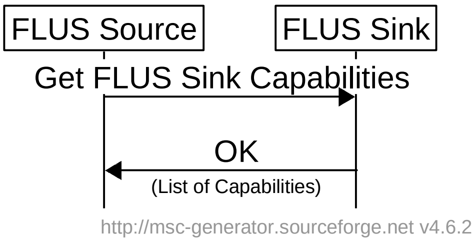

FLUS Source
FLUS Sink
Get FLUS Sink Capabilities
OK
(List of Capabilities)
http://msc-generator.sourceforge.net v4.6.2

Figure 4.4-5: Get current FLUS sink capabilities

1. The FLUS source sends the FLUS sink capability request to a known URL of the FLUS sink.

2. The FLUS sink provides a list of FLUS sink capabilities in response.

5.3.2.2 FLUS session establishment

It is assumed that the FLUS source has selected a FLUS sink and acquired the necessary information to establish an F-C connection to that FLUS sink (i.e. the HTTPS URL).

The procedure allows a FLUS source to create a new FLUS session. Session configuration properties and in particular FLUS media instantiation selection is added in subsequent procedures.

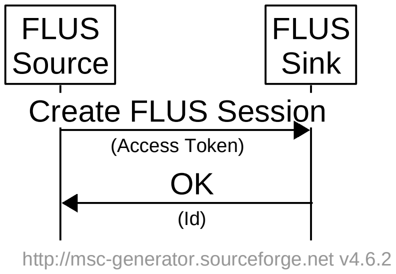


Figure 4.4-2: FLUS session creation

1. The FLUS session is created. The FLUS source provides a valid access token.

2. On successful creation, the FLUS sink responds with the resource id of the session. FLUS session properties are fetched and modified with subsequent transactions.

5.3.2.3 Get FLUS session properties

The procedure allows a FLUS source to fetch the current configuration of the FLUS session.

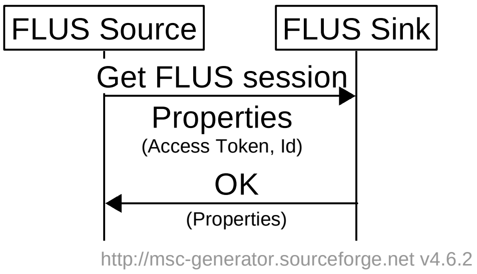

FLUS Source
FLUS Sink
Get FLUS session Properties
(Access Token, Id)
OK
(Properties)
http://msc-generator.sourceforge.net v4.6.2

Figure 4.4-3: Get current FLUS session properties

1. The FLUS source sends along with the session property request, the access token and the resource id of the session.

2. The FLUS sink provides the FLUS session properties in response.

5.3.2.4 FLUS session update

The procedure allows a FLUS source to update the current configuration of the FLUS session.

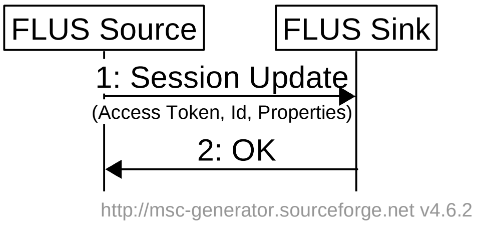

FLUS Source
FLUS Sink
1: Session Update
(Access Token, Id, Properties)
2: OK
http://msc-generator.sourceforge.net v4.6.2

Figure 4.4-4: FLUS session update

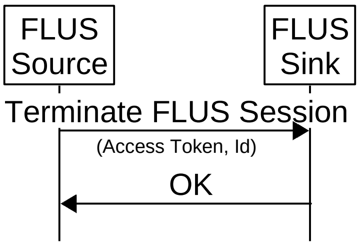
The FLUS source may first fetch the current FLUS session configuration using the Get FLUS session properties procedure.

1. The FLUS source modifies the properties of the session resource. The procedure may allow modification of individual properties or all properties.

2. The FLUS source updates the resource identified by the id of the session.

5.3.2.5 FLUS session termination

The FLUS source may explicitly terminate a FLUS session and all its provisioned and active media sessions. Alternatively, the FLUS session is automatically terminated, when the last media session of the FLUS session is terminated. The procedure allows a FLUS source to terminate a FLUS session. All media streams will be terminated automatically with the termination of the FLUS session.



http://msc-generator.sourceforge.net v4.6.2

Figure 4.4-6: FLUS session termination

1. The FLUS source sends the terminate FLUS session command. The access token and the resource id of the FLUS session is provided as input.

2. The FLUS sink terminates the FLUS session, including all active media streams and acknowledges the reception of the request.

5.3.2.6 List of FLUS sink capabilities

The FLUS sink capabilities should include the features that are listed under 4.4.4.

5.3.2.7 List of FLUS session properties

All FLUS session properties, except for the resource id, are always carried in an HTTP message body. The access-token is always carried as part of HTTP headers. Except for the FLUS session creation request (where the id is not present), the resource id shall be present in the URL of all requests that relate to a specific FLUS session.

In the table below, the following assertions are made:

- Table header: C stands for Create FLUS session procedure, G is for Get FLUS session properties procedure, U is for Update FLUS session properties procedure and T is for Terminate FLUS session procedure. “I”, and “O” respectively denote “request” (going **I**nto the FLUS sink), and response (going **O**ut of the FLUS sink).

- Optional (“O”) means that the property may or may not be sent/received during a REST transaction. It does not necessarily mean that the property is optional. It is possible, for example, that a session is not yet active because the FLUS source has not set the property in any previous update transaction using the PUT or PATCH HTTP method, as opposed to representing a hint on the importance of the property for the FLUS sink.

- A property marked as optional (O) in a request message may be present in the request. When not present in the request body, the property, if present in the FLUS sink, will not be updated.

- A property marked as optional (O) in a response message is only present in the response when a value is assigned or changed by the FLUS sink.

- A property marked as mandatory (M) in a response message is always present in the response. The FLUS sink provides defaults, which may be modified subsequently by the content provider.

- A blank cell in the table means “forbidden” (the property cannot be added to the request or returned by the FLUS sink, depending on the transaction direction)..

Table 5.3.2.7-1: List of FLUS session properties

Property Name	Property Description	C I	C O	G I	G O	U I	U O	T I
id	Identifier of the FLUS session resource.		M					

<table>
<tr>
<td></td>
<td>Note, the id is only provided within an HTTP body during the Create FLUS session response. Otherwise, the id is present in the message URL to identify the resource in the FLUS sink.
<table>
<tr><td>Type</td><td>Unit</td><td>Default</td></tr>
<tr><td>Integer</td><td>None</td><td>N/A</td></tr>
</table>
</td>
<td></td><td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
<tr>
<td>F-U Instantiation</td>
<td>Identifier of the FLUS media instantiation that is used by this FLUS session.
Vendor specific enumeration values shall start with 'vnd-' followed by a unique vendor name and optionally followed by additional characters.
The F-U instantiation shall be provided as a globally unique URN.
<table>
<tr><td>Type</td><td>Unit</td><td>Default</td></tr>
<tr><td>URI</td><td>None</td><td>All</td></tr>
</table>
</td>
<td></td><td></td><td></td><td>M</td><td>O</td><td></td><td></td>
</tr>
<tr>
<td>Entrypoint URL</td>
<td>Entry point URL information (e.g., SIP URL) for establishing the F-U connection to start the media streaming. Details on the Entrypoint URL is F-U instantiation specific.</td>
<td></td><td></td><td></td><td></td><td></td><td></td><td></td>
</tr>
<tr>
<td>Processing Description</td>
<td>The property stores a media processing description document that defines the post processing pipeline that the FLUS sink shall apply to received media components. The pipeline description also may also set the distribution target (incl FLUS sink storage) for the media.
The format and semantics of this document are out of scope of the FLUS specification.</td>
<td></td><td></td><td></td><td>O</td><td>O</td><td></td><td></td>
</tr>
</table>

5.3.3 Data transport

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

6 Terminal capabilities

6.1 General

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

7 Uplink Streaming Control Interface

7.1 General

A UE with a FLUS source can connect to a FLUS sink and start live uplink streaming of content that is captured by the UE or by a set of connected capture devices. This section specifies REST API procedures for FLUS sink discovery, capability discovery, FLUS session setup, and session termination between the FLUS source and the FLUS sink.

7.1.1 Resources

F-C defines a set of resources managed at the FLUS sink that are controlled by the FLUS source. The set of resources are defined in the following table:

Table 7.1.1-1: Resources for managing FLUS sessions at FLUS sink

Resource Name	Resource Type	Description
Sink	Instance resource	Represents a single FLUS sink as described in section 7.1.1.1. The FLUS source can query the properties of that sink. However, a FLUS source cannot create, update, or terminate a FLUS sink. The creation, update, and termination of FLUS sinks are controlled exclusively by the network
Session	Instance resource	Represents a single FLUS session resource as described in section 7.1.1.2. The FLUS source can provision or modify a single session at the FLUS sink.
Sessions	Collection Resource	Represents a collection of FLUS session resources.

7.1.1.1 Sink Resource

A Sink resource provides a representation of the capabilities of a FLUS sink. It also contains a sessions property that represents a collection of FLUS sessions. A FLUS source can query sink resources using REST API methods to examine the properties of the sink resource. Different properties of sink resources represent the capabilities of the corresponding FLUS sink.

Each sink resource has the set of properties described in Table 7.1.1.1-1.

Table 7.1.1.1-1: Properties of Sink Resource

Property Name	Description	Example Values
capabilities	List of supported features and instantiations by the FLUS sink. Each capability is to be expressed using an URN.	urn:vnd:xzy:capability-name

As described in Table 7.1.1.1-1 above, each sink resource describes the capabilities of the corresponding FLUS sink. A FLUS source can retrieve the resource description of a sink resource and find out its capabilities and make a decision if it wants to use the corresponding FLUS sink as described using the capability exchange procedure in section 7.3.

7.1.1.2 Session Resource

A Session resource provides the representation of a session between a FLUS source and a FLUS sink. A FLUS source can create a session at a FLUS sink using REST API methods operating on the session resource described in this section.

Each session resource has the set of properties described in Table **5.2.2.7-1**.

FLUS sources use the above session resource representation to create and manage sessions at the selected FLUS sink. Different procedures that operate on the sink and session resources are discussed in the following sections.

7.1.2 Supported Methods

The F-C API follows the RESTful design principles. All operations shall be performed using HTTP 1.1 (IETF RFC 7231 [6]) over TLS (3GPP TS 33.246 [7]).

HTTP Method	**CRUD**	**Resource**	**PATH**
POST	Create	Session	/flus/v1.0/sessions
GET	Read	Session	/flus/v1.0/sessions/{session-res-id}
		Sink	/flus/v1.0/sinks/{sink-res-id}/
PUT	Replace	Session	/flus/v1.0/sessions/{session-res-id}
PATCH	Modify	Session	/flus/v1.0/sessions/{session-res-id}
DELETE	Delete	Session	/flus/v1.0/session/{session-res-id}

7.1.3 Error Handling

The Uplink Streaming Interface API shall use the HTTP status codes to indicate any errors that might occur in the processing of operations on FLUS resources. Unless defined otherwise, the HTTP status codes shall be interpreted as specified in IETF RFC 7231 [6]. API operations that are not successfully handled shall not leave the resource at an undefined state. The response should provide sufficient information for a human operator to understand and locate the error.

7.2 Discovery

The FLUS sink discovery procedure is used by the FLUS source to discover available FLUS sinks that are provided by the operator.

To query the list of available FLUS sinks, the FLUS REST client shall use one of the following procedures:

- The operator provides one or more FLUS sink URLs to the FLUS source.
- DNS discovery of the FLUS sink
- The FLUS Sources uses the pre-defined URL: http://flus.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org/flus/v1.0/sinks/ in an HTTP request. The response shall be a JSON document that represents an array of objects, each of which contains a URL to the entry point to the FLUS Sink.

7.3 Capability retrieval

GET /flus/v1.0/capabilities

Using this capability exchange procedure, the FLUS source enquires about the supported features and instantiations of the FLUS sink.

To retrieve the capabilities of a FLUS sink, the FLUS source shall use the HTTP GET method on the “sink” instance resource as follows:

- the request URI with the “path” is set to “/flus/v1.0/capabilities”
- the Host field is set to the FQDN of the FLUS sink

The URL in the request is set according to the information retrieved using the Discovery procedure described in section 7.2.

The sink resource information provides in detail the capabilities of the FLUS sink. The response from the FLUS sink to the FLUS source shall contain the following:

- The Content-Type field specifying the MIME type (JSON) using which the sink resource information is encoded.
- The Content-Length is the length of the content body.

The above header fields are followed by the content body in the format indicated by the Content-Type field. The content body includes the detailed representation of the sink resource as described in section 7.1.1.1 from which the capabilities of that FLUS sink can be inferred.

The possible response messages from the FLUS sink to the FLUS source are shown in Table 7.3-1.

Table 7.3-1: Response status code, message, and contents for sink resource retrieval using HTTP GET

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides the sink capabilities to the FLUS source.
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4 Uplink streaming configuration

7.4.1 FLUS session properties fetch procedure

GET /flus/v1.0/sessions/{session-res-id}

Sessions can be read by the FLUS source when it wishes to know the latest representation of the session resource at the FLUS sink. To fetch the properties of a FLUS session, the FLUS source sends a HTTP GET request to the FLUS sink as follows:

- the request URI with the "path" set to "/flus/v1.0/sessions/{session-res-id}"
- the Host field is set to the FQDN of the FLUS sink

The {session-res-id} in the request URI is the session resource identifier of the session previously created between the FLUS source and the FLUS sink.

Upon receiving HTTP GET request from the FLUS source, the FLUS sink checks to see if such a session exists that matches the given session resource identifier. If such a session exists, the FLUS sink shall respond to the FLUS source with a 200 OK message along with the complete representation of the session resource. The response from the FLUS sink to the FLUS source shall contain the following:

- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The content body of this response message shall be the representation of the session resource as described in sub clause 7.1.1.2

Alternatively, if such a session cannot be found by the FLUS sink, it shall send a 404 Not Found message to the FLUS sink. If the request cannot be fulfilled, the FLUS sink shall send a 403 Forbiddent message to the FLUS source.

The possible response messages from the FLUS sink, depending on whether the GET request is successful or unsuccessful, are shown in Table 7.4.1-1.

Table 7.4.1-1: Response status code, message, and contents for service modification using HTTPS GET

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides session properties of the session resource to the FLUS source
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4.2 FLUS session update procedure

The sessions at the FLUS sink can be modified partially or completely using the following two procedures

7.4.2.1 Partial modification of FLUS session

PATCH /flus/v1.0/session/{session-res-id}

To update some of the FLUS session parameters at the sink, the FLUS source sends an HTTP PATCH request as follows:

- the request URI with the “path” is set to “/flus/v1.0/sessions/{session-res-id}”
- the Host field is set to the FQDN of the FLUS sink
- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The {session-res-id} in the request URI is the session resource identifier of the session whose modification is sought.

The content body of the HTTP PATCH message shall contain the updated partial representation of the session resource.

Upon receiving the HTTP PATCH request from the FLUS source, the FLUS sink checks to see if such a session exists at the sink. If such a sink exists, FLUS sink shall update the session properties based on the values from the incoming request. Upon successful update of the requested session, the FLUS sink shall respond to the FLUS source with a 200 OK success message indicating that the session was successfully updated. The FLUS sink shall also include the session resource identifier of the session that is updated. As alternative to the 200 OK message, FLUS sink may send a 204 No Content success message without any message content to the FLUS source. If the session cannot be updated, the FLUS sink shall send a 403 message. If the session is not found, the FLUS sink shall send a 404 message.

The possible response messages from the FLUS sink, depending on whether the PATCH request is successful or unsuccessful, are shown in Table 7.4.2.1-1.

Table 7.4.2.1-1: Response status code, message, and contents for session modification using HTTP PATCH

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink provides the session resource identifier of the session that is modified
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled

404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.4.2.2 Full modification of FLUS session

PUT /flus/v1.0/sessions/{session-res-id}

For complete update of the session parameters at the FLUS sink, the FLUS source sends an HTTP PUT request as follows:

- the request URI with the “path” is set to “/flus/v1.0/sessions/{session-res-id}”
- the Host field is set to the FQDN of the FLUS sink
- The Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- The Content-Length is the length of the content body.

The {session-res-id} in the request URI is the session resource identifier of the session whose modification is sought.

The content body of the HTTP PUT message shall contain the updated complete representation of the session resource.

Upon receiving the HTTP PUT request from the FLUS source, the FLUS sink checks to see if such a session exists at the sink. If such a sink exists, FLUS sink shall update the session properties based on the values from the incoming request. Upon successful update of the requested session, the FLUS sink shall respond to the FLUS source with a 200 OK success message indicating that the session was successfully updated. The FLUS sink shall also include the session resource identifier of the session that is updated. As alternative to the 200 OK message, FLUS sink may send a 204 No Content success message without any message content to the FLUS source. If the session cannot be updated, the FLUS sink shall send a 403 message. If the session is not found, the FLUS sink shall send a 404 message.

The possible response messages from the FLUS sink, depending on whether the PATCH request is successful or unsuccessful, are shown in Table 7.4.2.2-1.

Table 7.4.2.2-1: Response status code, message, and contents for session modification using HTTP PATCH

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink shall send the session resource identifier of the session that is modified
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

7.5 Session establishment

POST /flus/v1.0/sessions

To create a session at the FLUS sink, the FLUS source shall use the HTTP GET method on the “sessions” collection resource as follows:

- the request URI with the “path” is set to “/flus/v1.0/sessions”

- the Host field is set to the FQDN of the FLUS sink
- the Content-Type field specifying the MIME type (JSON) using which the session resource information is encoded.
- the Content-Length is the length of the content body.

The content body of the POST request shall contain the representation of the session resource. The session resource representation includes the details such as the media streams, their formats and codecs, the relationship between them, and the selected user plane instantiation and user plane control protocol.

Upon receipt of HTTP POST to create a session, the FLUS sink shall create a session. Upon successful creation of session resource, the FLUS sink shall respond back to the FLUS source with a 201 success message indicating that the session is successfully created along with the session resource of the created session. The session information returned to the FLUS source along with the 201 message includes the parameters applied for the session, thus confirming the parameters, and any default values assigned to session parameters which were not supplied by the FLUS source in the HTTP POST message.

The session resource identifier is the identifier that uniquely identifies the session within the list of all sessions at that sink. When the FLUS source receives the session resource identifier, it shall use this identifier in subsequent requests to the FLUS sink to refer to this session.

Alternatively, if the creation of session is failed, the FLUS sink shall send a 403 message.

The possible response messages from the FLUS sink, depending on whether the POST request is successful or unsuccessful, are shown in Table 7.5-1.

Table 7.5: Response status code, message, and contents for session creation

Status Code	Message	Contents
201 Created	Session created successfully	The FLUS sink provides the session resource identifier of the created session and detailed session information
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [11] as applicable		

7.6 Session termination

DELETE /flus/v1.0/sessions/{session-res-id}

To terminate a FLUS session, the FLUS sink shall use the HTTP DELETE method on the “session” instance resource as follows:

- the request URL with the “path” is set to “/flus/v1.0/sessions/{session-res-id}”
- the Host field is set to the FQDN of the FLUS sink

The {session-res-id} is the session resource identifier for which the session termination is sought.

Upon reception of a HTTP DELETE request, the FLUS sink will check to see if the session with the given session resource identifier exists at the given sink. If such a session exists, the FLUS sink shall delete the session instance associated with the given session resource identifier. Further, the FLUS sink responds back to the FLUS source with a 200 success message along with the session resource identifier indicating that the session was successfully deleted. Alternately, if the session cannot be deleted, the FLUS sink shall send a 403 Forbidden message. If the session was not found, or if the sink was not found, the FLUS sink returns back with a 404 message.

The possible response messages from the FLUS sink, depending on whether the DELETE request is successful or unsuccessful, are shown in Table 5.2.2.2.4-1.

Table 7.6-1: Response status code, message, and contents for session deletion

Status Code	Message	Contents
200 OK	The request has succeeded	The FLUS sink shall send the session resource identifier of the session that is deleted
204 No Content	The request has succeeded	None
403 Forbidden	Request cannot be fulfilled	The FLUS sink may include optional text to indicate why the request could not be fulfilled
404 Not Found	Requested resource not found	None
Note: In addition to the above response codes, the FLUS sink can also send appropriate response codes described in IETF RFC 7231 [6] as applicable.		

8 FLUS Security

8.1 IMS-based FLUS

Security functionality for IMS is specified at both the control and media planes. From the FLUS system perspective, it is described in clause 4.2.1 that the FLUS media/user planes contain both the media session establishment and the subsequent media data transmission functions. In other words, in an IMS-based FLUS system, both the IMS control and media plane security functionalities are combined in FLUS media and F-U as shown in Figure 4.2-2. From a logical standpoint, security functionality pertaining to session establishment procedures in the IMS-based FLUS system corresponds to IMS control plane security, which in turn consists of access and core network related security functions. The former, on access security, is defined in TS 33.203 [8], while the core network security functionality is defined in TS 33.210 [9]. Media data transmission in the IMS-based FLUS system include RTP transport of continuous media content, and its associated security functionality is defined in TS 33.328 [10].

8.2 Non-IMS-based FLUS

In this version of the specification, this function is left for implementation, but expected to be specified in a future version.

Annex A (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-12	SA#78	SP-170884				Presented to TSG SA#78 (for approval)	1.0.0
2017-12	SA#78					Approved for Release 15 at TSG SA#78	15.0.0
2018-03	SA#79	SP-180024	0001	3	F	Corrections to FLUS Framework	15.1.0

History

Document history		
V15.1.0	July 2018	Publication