NATO STANDARD

AEDP-18

NATO STANDARD ISR STREAMING SERVICES

Edition A Version 1
MARCH 2018

NORTH ATLANTIC TREATY ORGANIZATION

ALLIED ENGINEERING DOCUMENTATION PUBLICATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

28 March 2018

1. The enclosed Allied Engineering Documentation Publication AEDP-18, Edition A, Version 1, NATO STANDARD ISR STREAMING SERVICES, which has been approved by the Nations in the NATO AIR FORCE ARMAMENTS GROUP, is promulgated herewith. The agreement of nations to use this publication is recorded in STANAG 4559.

2. AEDP-18, Edition A, Version 1, is effective upon receipt.

3. No part of this publication may be reproduced, stored in a retrieval system, used commercially, adapted, or transmitted in any form or by any means, electronic, mechanical, photo-copying, recording or otherwise, without the prior permission of the publisher. With the exception of commercial sales, this does not apply to member nations or NATO commands and bodies.

4. This publication shall be handled in accordance with C-M(2002)60.

Zoltán GULYÁS
Brigadier General, HUNAF
Director, NATO Standardization Office

INTENTIONALLY BLANK

RESERVED FOR NATIONAL LETTER OF PROMULGATION

INTENTIONALLY BLANK

RECORD OF RESERVATIONS

CHAPTER	RECORD OF RESERVATION BY NATIONS
Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.	

INTENTIONALLY BLANK

RECORD OF SPECIFIC RESERVATIONS

[nation]	[detail of reservation]

Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.

INTENTIONALLY BLANK

TABLE OF CONTENTS

This table of contents provides the structure and content for AEDP-18. AEDP-17, NATO Standard ISR Library Interface, and, NATO Standard ISR Workflow Architecture, complement AEDP-18 in these related respects.

Tables

Figures

Annexes

INTENTIONALLY BLANK

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

1. This document sets forth AEDP-18, NATO Standard ISR Streaming Services, proposed for agreement under STANAG 4559 Edition 4.

2. AEDP-18 complements AEDP-17, NATO Standard ISR Library Interface, proposed for agreement under STANAG 4559 Edition 4, with a library streaming interface.

3. Sources for AEDP-18 are

a) MAJIIC2 Baseline Bravo.1

(1) DOP-MAJIIC2-078, MAJIIC2 CSD Stream Server Specification, Service Version 4.4, Document Version 5.0

(2) DOP-MAJIIC2-104, MAJIIC2 CSD Stream Controller Service Specification, Service Version 4.4, Document Version 2.0.

(3) DOP-MAJIIC2-093, MAJIIC2 WS-Notification Implementation Guide, Document Version 2.0

(4) DOP-MAJIIC2-094, MAJIIC2 Boolean Query Syntax (BQS) Semantics, Document Version 1.2

(5) DOP-MAJIIC2-096, MAJIIC2 SOA Services Profile, Document Version 2.0

(6) DOP-MAJIIC2-100, MAJIIC2 Catalogue Entry Documentation, Document Version 2.0

(7) DOP-MAJIIC2-101, MAJIIC2 Core Interface Documentation, Document Version 2.0

(8) DOP-MAJIIC2-102, MAJIIC2 CSD Data Model Version 2.7, Document Version 1.0

1.2 AIM

The aim of this standard is to promote interoperability for the exchange of NATO Intelligence, Surveillance and Reconnaissance (ISR) streaming data and products. The NATO Standard ISR Streaming Services provide standard interfaces for querying and accessing ISR streaming data and products through suitable applications maintained by NATO and NATO Nations.

1.3 DESCRIPTION

1. AEDP-18 describes the *CSD Stream Server* and its interfaces. The *CSD Stream Server* is responsible for streaming data, i.e. data generated by sensors and which is periodically updated, e.g. motion imagery or ground moving target indicator (GMTI).

2. The *CSD Stream Server* allows a sensor to declare that a stream is available and to provide periodic metadata updates, allows an exploitation system to query for recorded and live streaming data, and it allows an exploitation system to request the replay of recorded streaming data, or the relay of live streaming data. One *CSD Stream Server* may connect to other *CSD Stream Servers* to provide a coherent coalition enterprise view, using metadata replication.

3. This AEDP-18 describes the high-level functional requirements of the *CSD Stream Server*, and references the appropriate individual service specifications

1.4 RFC 2119 COMPLIANCE

1. 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 [IETF, RFC 2119, 1997].

2. The key word “INFORMATIVE” is used to denote strictly informative content of the document.

1.5 REFERENCES AND RELATION TO OTHER STANDARDS

1.5.1 Referenced Documents

1. The following STANAGS, Allied Publications, Specifications and Standards contain provisions contain provisions, which through references in this text constitute provisions of this AEDP:

AEDP-17 Annex B	NATO Standard ISR Library Interface, NSIL SPECIFICATION AND REQUIREMENTS
AEDP-17 Annex C	NATO Standard ISR Library Interface, NSIL CORBA INTERFACE
AEDP-17 Annex D	NATO Standard ISR Library Interface, NSIL WEB SERVICE CORE INTERFACE
AEDP-17 Annex E	NATO Standard ISR Library Interface, NSIL WEB SERVICE INTERFACES
AEDP-17 Annex F	NATO Standard ISR Library Interface, NSIL META-MODEL
AEDP-17 Annex G	NATO Standard ISR Library Interface, NSIL METADATA MODEL
AEDP-17 Annex J	NATO Standard ISR Library Interface, BQS SYNTAX AND SEMANTICS
AEDP-19 Annex A	NATO Standard ISR Workflow Architecture, CORE SERVICE INTERFACES
AEDP-19 Annex H	NATO Standard ISR Workflow Architecture, WORKFLOW SERVICES BUSINESS RULES
OASIS WS-Topics 1.3, 2006	OASIS Standard Specification, “Web services Topics”, Version 1.3, 1 October 2006
OASIS WS-BaseNotification 1.3, 2006	OASIS Standard Specification, “Web services Base Notification”, Version 1.3, 1 October 2006
OASIS WS-BrokeredNotification 1.3, 2006	OASIS Standard Specification, “Web Services Brokered Notification”, Version 1.3, 1 October 2006

OASIS WS-Security UsernameToken 1.1, 2006	OASIS Standard Specification, “Web Services Security UsernameToken Profile“, Version 1.1, Web Service Security, February 2006
W3C WS transfer 1.0, 2011	W3C Standard Specification, “Web Services Transfer”, Version 1.0, December 2011.

2. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this AEDP are encouraged to investigate the possibility of applying the most recent editions of the STANAG, Recommendations and Standards listed above. NATO maintains registers of currently valid STANAGs.

1.5.2 Related Documents

The following STANAGS, Allied Publications, Specifications and Standards relate to this AEDP:

RFC 2119	Key words for use in RFCs to Indicate Requirement Levels, IETF 1997
NATO C3 Classification Taxonomy, 2015	C3 Taxonomy, at https://tide.act.nato.int/em/index.php?title=C3_Taxonomy, viewed 12 October 2015
STANAG 4607	NATO Ground Moving Target Indicator (GMTI) Format Edition 3
STANAG 4609	NATO Digital Motion Imagery Standard Edition 4
STANAG 5516	NATO Tactical Data Exchange – Link 16 Edition 4
AEDP-8	NATO Motion Imagery (MI) STANAG 4609 (Edition 3) Implementation Guide Edition 3
DOP-MAJIIC-071 GMTI	MAJIIC2 STANAG 4607 GMTI Implementation Guide Version 3.0
DOP-MAJIIC-009 MI	MAJIIC2 STANAG 4609 Motion Imagery Implementation Guide Version 5.0
DOP-MAJIIC-068 LINK16	MAJIIC2 STANAG 5516 LINK16 Implementation Guide Version 5.0

A copy of the MAJIIC2 Implementation Guides can be requested from the STANAG 4559 Custodian.

1.6 ABBREVIATIONS

AEDP	Allied Engineering Documentation Publication
BPMN	Business Process Model and Notation
BQS	Boolean Query Syntax
BRUC	Business Rules and Use Cases
COI	community of interest
CORBA	Common Object Request Broker Architecture
CRL	Collection Request List
CSD	Coalition Shared Data
CSL	Core Services Layer
CTL	collection task list
CXP	Collection and Exploitation Plan
E2RS	Enterprise Event Relay Service
EOB	electronic order of battle
EO/IR	electro-optical/infra-red
FMN	Federated Mission Networking
FMV	full motion video
FQN	fully qualified name
GAOI	geographic area of interest
GIS	geographic information system
GMTI	ground moving target indicator
HTML	HyperText Markup Language
HUMINT	human intelligence
ICP	intelligence collection plan
IP	Internet Protocol
IPL	ISR Product Library
IRM&CM	Information Requirement Management and Collection Management
ISR	intelligence, surveillance and reconnaissance
JCSL	JISR COI-Specific Services Layer
JISR	Joint Intelligence Surveillance and Reconnaissance
JPEG	Joint Photographic Experts Group
KLV	Key-Length-Value
KML	Keyhole Markup Language
LAN	local area network

MAJIIC2	Multi Intelligence All Source Joint Intelligence Surveillance and Reconnaissance Interoperability Coalition 2
MARD	MAJIIC 2 Architecture Requirements Document
MDA	model-driven architecture
MoD	Ministry of Defence
MOM	Message-oriented Middleware
NATO	North Atlantic Treaty Organization
NIIA	NATO ISR Interoperability Architecture
NSILI	NATO Standard ISR Library Interface
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
PED	processing, exploitation, and dissemination
PIM	Platform Independent Model
PSM	Platform Specific Model
SOA	service-oriented architecture
SOAP	Simple Object Access Protocol
SPS++	Simple Persistence as a Service
STANAG	NATO Standardization Agreement
TDL	Tactical Data Link
TTP	tactics, techniques and procedure
UAV	unmanned aerial vehicle
UDP	User Datagram Protocol
UML	Unified Modelling Language
URL	uniform resource locator
W3C	World Wide Web Consortium
WAN	wide-area network
WKT	well-known text
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition

CHAPTER 2 TERMS AND DEFINITIONS

This document uses the following terms as defined in Table 2-1.

Table 2-1 Definition of Terms

Stream	A continual flow of data whose properties vary over time.
Substream	A segment of a Stream bounded by a time interval. For example the result of an advanced query.
Live Stream	The immediate **Stream** output of a sensor.
Recorded Stream	A **Live Stream** which has been persisted. The Live Stream MAY still be transmitting.
Relayed Stream	A **Live Stream** which is being disseminated across a Wide Area Network (WAN) via a chain of one or more CSD Stream Servers.
Replayed Stream	A **Recorded Stream** which is being disseminated across a Wide Area Network (WAN) via a chain of one or more CSD Stream Servers.
Phase 0/1 Exploitation	Exploitation of **Live Streams** creating basic intelligence products, for example, chat messages and ISRSPOTREPs
Phase 2/3 Exploitation	Exploitation of **Live Streams** and **Recorded Streams,** creating more advanced intelligence products such as annotated still images, motion imagery clips, rich reports etc. Usually also includes collateral data from other sources.
NSILI	A NATO Standard ISR Library Interface (NSILI) server adhering to AEDP-17.
CSD Stream Metadata	Metadata represented using the data model defined in CHAPTER 4. This MUST be subsequently encoded as a Catalogue Entry in accordance with [*AEDP-17 Edition A Version 1, Annex E-4 CATALOGUE ENTRIES*].
Stream Discovery	The process of determining which **Live Streams** are

	currently available.
Stream Metadata	Metadata embedded within a **Stream** itself, e.g. key length value (KLV) for motion imagery **Streams**.
Stream Search	The process of determining which **Live Streams** or **Recorded Streams** satisfy a particular set of criteria. Note: superset of **Stream Discovery**.
Motion Imagery Clip	A motion imagery file that an analyst has created during exploitation. For example, it may show a target entering a building.

CHAPTER 3 CONCEPT OF EMPLOYMENT

3.1 GENERAL

1. A CSD Stream Server provides a set of interfaces for handling both Live and Recorded Streams. Streams MAY be:

- Ground Moving Target Indicator (GMTI)
- Motion imagery
- Link16[1].

2. Each CSD Stream Server MUST support at least one of the above stream types. Supported stream types are listed in the CSD stream controller service interface getCapabilities response, see Annex B STREAM CONTROLLER SERVICE INTERFACE.

3. The CSD Stream Server interacts with the following actors (see Figure 3-1):

- Sensors i.e. Stream producers
- Exploitation systems i.e. Stream consumers.
- An associated NSILI server for publishing Stream metadata to support the Information Requirements Management and Collection Management (IRM&CM) process.
- Other CSD Stream Server instances, for sharing Streams across sites and organizations through the WAN.

[1] Link16 is a special type of stream because it has multiple producers and multiple consumers, whereas a motion imagery stream or GMTI stream only has a single producer and one or more consumers.

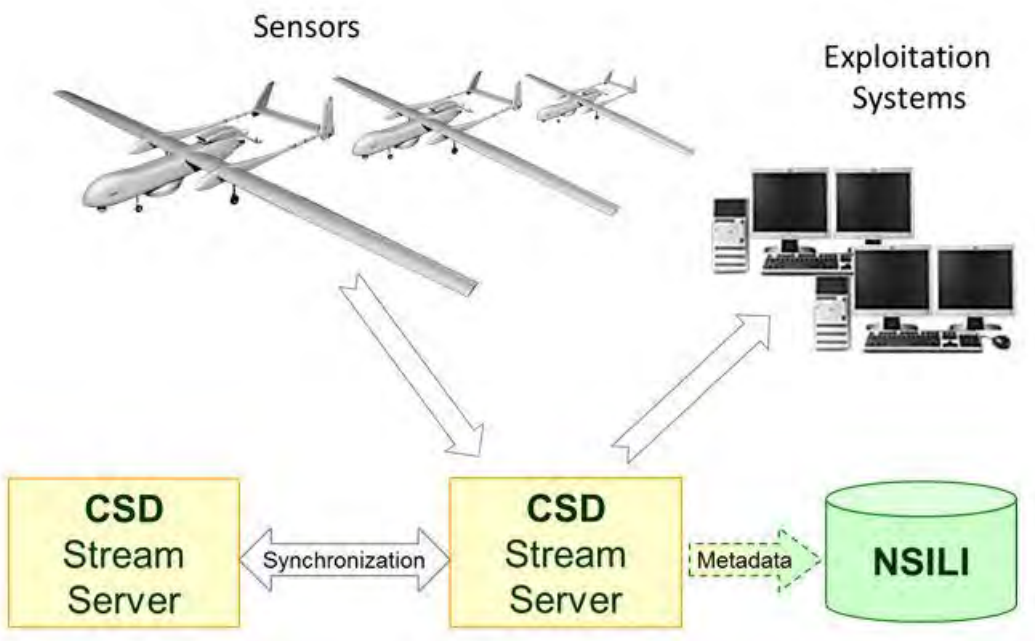

Figure 3-1 Actors around the CSD Stream Server

3.2 EXCHANGE OF STREAMING DATA

1. The CSD Stream Server provides interfaces for:

   - Stream publishing
   - Stream querying
   - Stream control
   - NSILI Bridge
   - Stream notification
   - Stream replication

   The interfaces are summarized in Annex A Table A-1 and illustrated in Annex A Figure A-1, followed by detailed descriptions of the interfaces.

2. Figure 3-2 shows how the actors interact with these interfaces:

   - Sensors emit Live Streams through their usual mechanism (mostly user datagram protocol (UDP) broadcast or multicast). In parallel, sensors declare and describe the Live Stream through the CSD Stream Publish Interface (red arrow).
   - The CSD Stream Server receives notifications coming from Sensors via the CSD Stream Publish Interface and records the corresponding Stream data and metadata describing the Live Stream.

- The CSD Stream Server makes Live Streams discoverable and Recorded Streams searchable through the CSD Stream Query Service Interface (blue arrow).
- Exploitation systems can discover Live Streams through the CSD Stream Query Service Interface (blue arrow). The metadata returned gives a description of the Live Stream (e.g. sensor, type, location etc.).
- Exploitation systems can request relay of a Live Stream using the CSD Stream Controller Service interface (green arrow).
- Exploitation systems can also discover Live Streams or replayed Recorded Streams through notifications made via the CSD Stream Notification Interface.
- Exploitation systems can search for Recorded Streams through the CSD Stream Query Interface (blue arrow). The metadata returned gives a description of the Recorded Stream (e.g. sensor, type, location etc.).
- Exploitation systems can request replay of Recorded Streams using the CSD Stream Controller Service Interface (green arrow). For example, this allows Phase 2/3 exploitation of a location with synchronized GMTI and motion imagery.

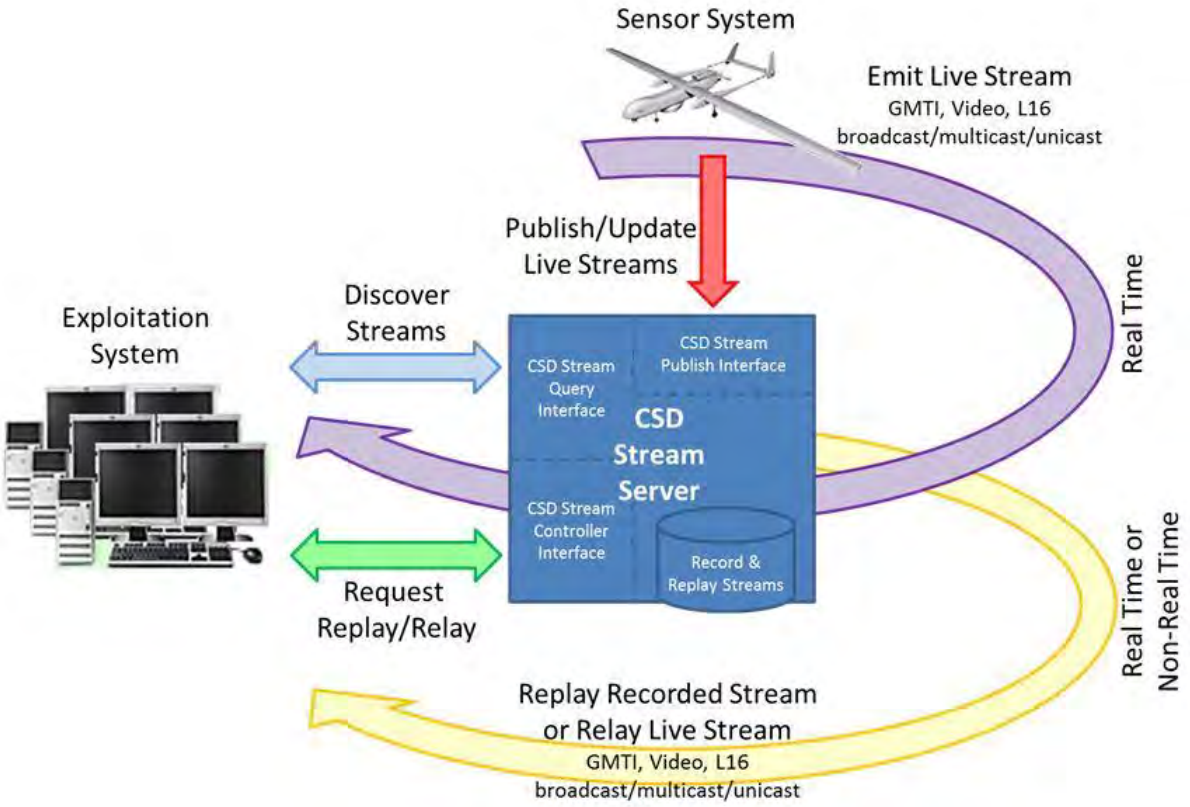

Figure 3-2 Interactions with the CSD Stream Server

3. The overall system of systems is thus more flexible and requires fewer configuration rules compared to legacy systems, by providing a coherent means to discover Live Streams and search Recorded Streams. It also handles the different types of Streams (e.g. GMTI, motion imagery etc.) in a uniform manner.

3.3 OPERATIONAL SCENARIOS

1. The CSD Stream Server supports the following operational scenarios:

- **Phase 0/1 exploitation of Live Streams**: The CSD Stream Server links Sensors and Exploitation Systems by providing the means to declare Live Streams from Sensors and by providing discovery capabilities for Exploitation Systems. Whenever a Sensor starts producing a Live Stream, it declares the Stream to the CSD Stream Server with a set of metadata. Exploitation Systems can then discover Live Streams of operational interest using this metadata.
- **Phase 2/3 exploitation of Recorded Streams**: The CSD Stream Server provides features to search for and replay relevant Recorded Streams. Multiple different Stream types can be synchronously replayed for multi-intelligence exploitation.
- **Training**: A set of pre-recorded Streams can be replayed synchronously for exploitation operator training.

2. Note that these scenarios can occur at the same time. For instance, while analyzing a Live Motion Imagery Stream, an exploitation operator might need support from a Recorded GMTI Stream to analyze what happened before an observed event occurs.

3.4 AEDP-18 XML SCHEMA and WSDL DEFINITIONS

The data model underlying the STANAG 4559 Streaming Services is established by CSD STREAM SERVER DATA MODEL, see CHAPTER 4, and the following schema and service definitions, which are maintained and managed (versioning) in a NATO Metadata Registry and Repository under a “NATO/NIIA/STANAG 4559 Edition 4” namespace:

Table 3-1 XSDs and WSDLs for CSD Stream Server Services (services)

Name	WSDL	Version
Catalog Entry Schema	CatalogEntry.xsd	4.4
Common Messages Schema	CommonMessages.xsd	4.4

(CommonMessages)		
Country Code Schema (common)	CountryCode1059.xsd	4.4
CSD Stream Server Replication Service Web-service Definition Language (WSDL) File (CSDStreamServerReplication)	CSDStreamServerReplication Service.wsdl	4.4
CSD Stream Server Synchronization Schema (CSDStreamServerReplication)	CSDStreamServerSync.xsd	4.4
ISR Common Schema (common)	isrcommon.xsd	4.4
MAJIIC Notification Schema	MAJIICNotification.xsd	4.4
MAJIIC2 Services Replication Web-service Definition Language (WSDL) File (CSDStreamServerReplication)	majiic-services-replication.wsdl	4.4
MAJIIC2 Services Replication Schema (CSDStreamServerReplication)	majiic-services-replication.xsd	4.4
Core Service Web-service Definition Language (WSDL) File	services.wsdl	4.4
Stream Controller Messages Schema (StrmCtrl)	StreamControllerMessages.xsd	4.4
Stream Controller Service Web-service Definition Language (WSDL) File (StrmCtrl)	StreamControllerService.wsdl	4.4
Stream Controller Web-service Definition Language (WSDL) File (StrmCtrl)	StrmCtrlr.wsdl	4.4

INTENTIONALLY BLANK

CHAPTER 4 CSD STREAM SERVER DATA MODEL

4.1 SCOPE

1. The data model of the CSD Stream Server (informally known as the NSIL_STREAM view) is shown in Figure 4-1.

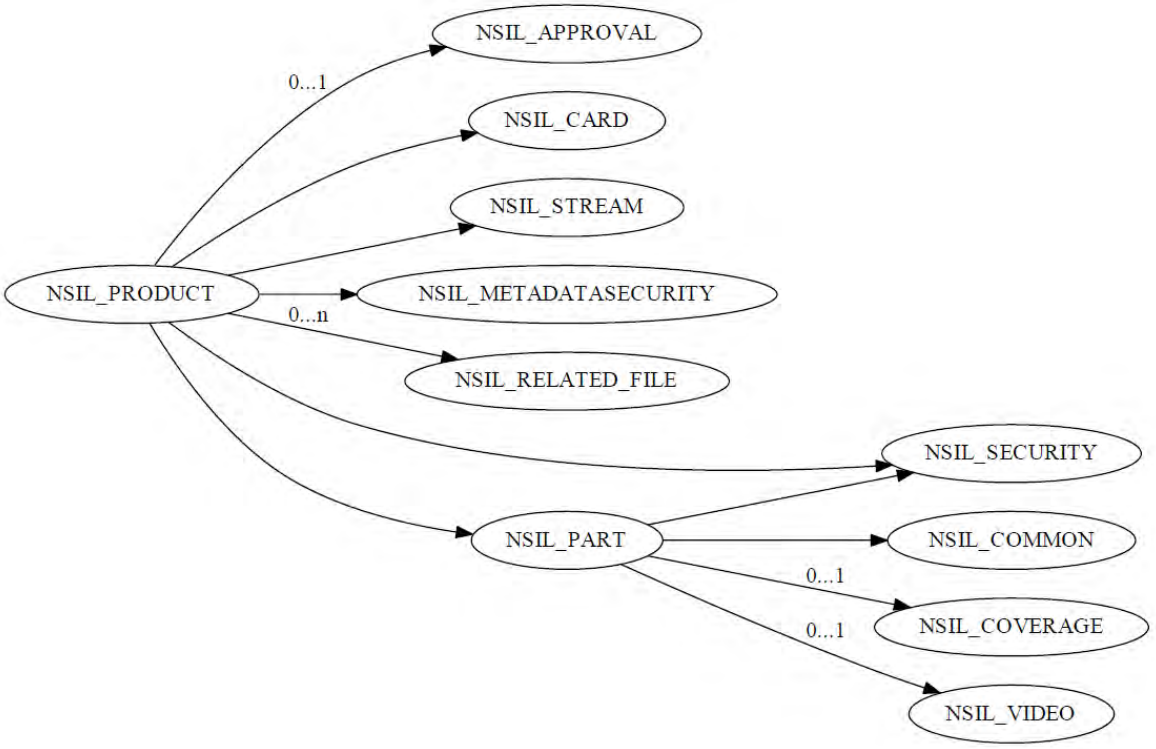

Figure 4-1 CSD Stream Server Data Model

2. The usage and requirements of all attributes and entities MUST be in accordance with AEDP-17 Annex G NSIL METADATA MODEL, unless explicitly overridden in this chapter.

3. The CSD Stream Server MUST NOT use multiple part products.

4. The NSIL_VIDEO entity MUST only be used for motion imagery Streams.

5. The NSIL_GMTI and NSIL_TDL entities MUST NOT be used, since they are only appropriate for a single GMTI revisit or TDL message, rather than streaming GMTI or TDL data, which contains multiple GMTI revisits or TDL messages.

4.2 CSD STREAM METADATA CATEGORIZATION

1. CSD Stream Metadata attributes can be divided into three categories:

- Static (Definition) – metadata attribute which is constant throughout the lifetime of a stream due to the formal definition of the attribute, e.g. stream identifier.

- Static (BR) – metadata attribute which is constant throughout the lifetime of the stream due to the application of reasonable business rules, e.g. stream classification.

- Dynamic (Stream) – metadata attribute which can reasonably change during the lifetime of the stream, e.g. stream geographic coverage.

2. Table 4-1 shows the categorization of each attribute in the CSD Stream Server Data Model.

Table 4-1 Categorization of CSD Stream Server Data Model attributes

Entity	Attribute	Category
NSIL_APPROVAL	approvedBy	Static (BR)
	dateTimeModified	Static (BR)
	status	Static (BR)
NSIL_CARD	identifier	Static (BR)
	sourceDateTimeModified	Static (Definition)
	dateTimeModified	Static (Definition)
	status	Static (Definition)
	numberOfParts	Static (Definition)
	publisher	Static (BR)
	sourceLibrary	Static (Definition)
NSIL_COMMON	descriptionAbstract	Static (BR)
	identifierMission	Static (BR)
	identifierUUID	Static (BR)
	identifierJC3IEDM	Static (BR)
	language	Static (BR)
	source	Static (BR)
	subjectCategoryTarget	Dynamic (Stream)
	targetNumber	Dynamic (Stream)
	type	Static (Definition)

Entity	Attribute	Category
NSIL_COVERAGE	spatialCountryCode	Dynamic (Stream)
	spatialGeographicReferenceBox	Dynamic (Stream)
	temporalEnd	Dynamic (Stream)
	temporalStart	Dynamic (Stream)
NSIL_METADATASECURITY	classification	Static (BR)
	policy	Static (BR)
	releasability	Static (BR)
NSIL_PART	partIdentifier	Static (BR)
NSIL_RELATED_FILE	extent	Dynamic (Stream)
	fileType	Dynamic (Stream)
	URL	Dynamic (Stream)
	isFileLocal	Dynamic (Stream)
NSIL_SECURITY	classification	Static (BR)
	policy	Static (BR)
	releasability	Static (BR)
NSIL_STREAM	archived	Static (Definition)
	archivalInformation	Static (Definition)
	creator	Static (BR)
	dateTimeDeclared	Static (BR)
	standard	Static (BR)
	standardVersion	Static (BR)
	sourceURL	Static (BR)
	programID	Static (BR)
NSIL_VIDEO	averageBitRate	Static (BR)
	category	Dynamic (Stream)
	encodingScheme	Static (BR)
	frameRate	Static (BR)
	numberOfRows	Static (BR)
	numberOfColumns	Static (BR)
	metadataEncodingScheme	Static (BR)
	MISMLevel	Static (BR)
	scanningMode	Static (BR)

4.3 CSD STREAM METADATA USE CASES

CSD Stream Metadata is used in five separate (but closely related) use cases as detailed below. It should be noted that use of some attributes varies slightly from that defined in AEDP-17 Annex G NSIL METADATA MODEL. The differences in the use of attributes are detailed in the following paragraphs.

4.3.1 CSD Stream Publish Interface

Sensors use the CSD Stream Publish Interface to notify the CSD Stream Server of the start of a new stream, provide updated metadata for an existing stream, and to notify the CSD Stream Server that an existing stream is ending. Note that there is no “update and replace” operation – metadata which may change during the lifetime of a stream (i.e. dynamic metadata) augments previous metadata, it does not replace it.

a. The CSD Stream Server SHALL discard a notification about streams sent to the CSD Stream Publish Interface when any mandatory attribute of the CSD Stream Server Data Model is not provided.

b. The CSD Stream Server SHALL discard a notification about a stream when any static metadata attribute has been changed from the value provided in the initial notification.

c. The CSD Stream Server SHALL support <forward compatibility>, as specified in Chapter 4.5, for the catalog entries in the notifications provided by the Sensor.

d. INFORMATIVE: For support of <forward compatibility> as per Paragraph a of Chapter 4.5, the Sensor is expected to provide to the CSD Stream Server all metadata model extensions it is publishing.

e. NSIL_CARD.status (although IPL-assigned in the CSD Server Data Model) SHALL be used to indicate the Stream status as follows:

    i. NEW: the Sensor is starting transmitting the Stream

    ii. CHANGED: the Sensor is providing a metadata update

    iii. OBSOLETE: the Sensor is stopping transmitting the Stream

f. NSIL_COMMON.identifierUUID SHALL uniquely identify the Stream from the Sensor’s perspective.

g. The CSD Stream Server SHALL regard stream notifications with the same NSIL_COMMON.identifierUUID value as belonging to the same logical stream.

h. NSIL_COVERAGE.spatialGeographicReferenceBox MUST contain the geographic area covered since the last notification.

i. NSIL_STREAM.dateTimeDeclared SHALL contain the time at which the stream started i.e. this value will not change with subsequent notifications.

j. A notification with status NEW using the CSD Stream Publish represents a mark of the start of the stream. The value of its NSIL_COVERAGE.temporalStart attribute SHOULD match the value in its NSIL_STREAM.dateTimeDeclared.

k. Each notification with status CHANGED using the CSD Stream Publish Interface represents a set of metadata corresponding to a certain period of time. The end of this period SHALL be specified by the NSIL_COVERAGE.temporalStart attribute of this notification. The start of this period SHOULD be inferred from the NSIL_COVERAGE.temporalStart value contained in the previous notification.

l. NSIL_COVERAGE.temporalEnd SHALL be omitted if NSIL_CARD.status is NEW or CHANGED.

m. When NSIL_CARD.status is OBSOLETE, NSIL_COVERAGE.temporalStart SHALL have the same value as NSIL_STREAM.dateTimeDeclared, and NSIL_COVERAGE.temporalEnd SHALL have the value of the end of the period defined in point j. above.

n. NSIL_STREAM.sourceURL SHALL be provided by the Sensor.

o. When the stream notification contains an NSIL_RELATED_FILE entity whose attribute ‘fileType’ is set as THUMBNAIL and whose attribute ‘url’ is pointing to a 128x128 JPEG image file, the CSD Stream Server MAY resolve the provided url and pull the file content from the referenced location.

4.3.2 CSD Stream Query Service Interface (Query)

Exploitation systems use the CSD Stream Query Service Interface to pose queries using the Boolean Query Syntax (BQS) encoded parameters of the getHitCount and getQueryResultSegment operations.

a. The CSD Stream Server MUST support querying by any CSD Stream Server Data Model attribute which is defined as queryable in the CSD-Server Data Model [AEDP-17 Annex G NSIL METADATA MODEL].

b. The CSD Stream Server MUST support sorting by any CSD Stream Server Data Model attribute(s) which are defined as sortable in the CSD-Server Data Model [AEDP-17 Annex G NSIL METADATA MODEL].

c. In addition, the CSD Stream Server MUST support sorting by the NSIL_COVERAGE.temporalEnd and NSIL_COVERAGE.temporalStart attributes.

d. INFORMATIVE: The Exploitation System is expected to limit its query terms to the attributes and entities known to be supported by the CSD Stream Server (i.e. the CSD Stream Server Data Model together with any known extensions).

e. When using Advanced Query (see A-3.3), the CSD Stream Server SHALL interpret the NSIL_COVERAGE.temporalStart attribute as the start time of the substream.

f. When using Advanced Query (see A-3.3, the CSD Stream Server SHALL interpret the NSIL_COVERAGE.temporalEnd attribute as the end time of the substream.

4.3.3 CSD Stream Query Service Interface (Query Results)

Exploitation systems receive metadata from the CSD Stream Server as a catalog entry according to [AEDP-17 Annex E-4 CATALOGUE ENTRIES], corresponding to query results using the getMetadata operation of the CSD Stream Query Service Interface.

a. The CSD Stream Server SHALL return all attributes in the catalog entry that are defined in its CSD Stream Server Data Model.

b. When supporting <forward compatibility> as per Paragraph b of Chapter 4.5, the CSD Stream Server SHALL also return any received additional attributes and/or entities in the catalog entry that are not defined in its CSD Stream Data Model.

c. INFORMATIVE: The Exploitation System is expected to ignore any unrecognized attributes and/or entities returned in CSD Stream Server Metadata which are present due to supporting the <forward compatibility> feature (see Chapter 4.5).

d. When using Advanced Query (see A-3.3), the CSD Stream Server MUST fuse dynamic metadata in accordance with Chapter 4.4.

e. The CSD Stream Server MAY populate the NSIL_STREAM.sourceURL attribute with an HTTP URL through which the Stream or Substream data can be downloaded.

f. The CSD Stream Server MAY include an NSIL_RELATED_FILE entity of fileType THUMBNAIL, whose attribute ‘url’ points to an HTTP URL which contains a single 128x128 JPEG image file representing the query result.

g. When providing the THUMBNAIL of the stream result, the CSD Stream Server SHOULD match this thumbnail with the first THUMBNAIL provided by the sensor in a notification for this stream (see also 4.3.1.o).

h. The CSD Stream Server MAY include two NSIL_RELATED_FILE entities of fileType THUMBNAIL_FIRST_FRAME and THUMBNAIL_LAST_FRAME, whose attribute 'url' points to an HTTP URL which contains a single 128x128 JPEG image file representing the query result.

i. When providing thumbnails for the first and last frame, the CSD Stream Server SHOULD match the THUMBNAIL_FIRST_FRAME and THUMBNAIL_LAST_FRAME with the THUMBNAILs provided in the first and last sensor notifications covered by the returned substream result (see also 4.3.1.o).

4.3.4 CSD Stream Notification Interface (OPTIONAL)

Exploitation systems may subscribe for notifications emitted by the optional CSD Stream Notification Interface in order to discover Live Streams and replayed Recorded Streams which may be of interest. These notifications are submitted to and received from the local notification broker. Since the CSD Stream Notification Interface is optional, the following requirements only apply if it is implemented.

a. The CSD Stream Server SHALL provide all attributes in the catalog entry that are defined in its CSD Stream Server Data Model and were provided by the Sensor.

b. When supporting <forward compatibility> as per Paragraph b of Chapter 4.5, the CSD Stream Server SHALL also provide any received additional attributes and/or entities in the catalog entry that are not defined in its CSD Stream Data Model.

c. The CSD Stream Server SHALL provide any attributes in the catalog entry that are defined as IPL-assigned according to its CSD Stream Server Data Model.

d. INFORMATIVE: The Exploitation System is expected to ignore any unrecognised attributes and/or entities returned in CSD Stream Server Metadata which are present due to supporting the <forward compatibility> feature (see Chapter 4.5).

4.3.5 CSD Stream Replication Service Interface

CSD Stream Servers SHALL exchange CSD Stream Metadata using the CSD Stream Replication Service Interface so that each CSD Stream Server holds a representation of the entire Stream metadata catalogue for the coalition enterprise.

a. The CSD Stream Server SHALL include all CSD Stream Server Data Model attributes, as provided by the Sensor or an incoming replication

message and which are defined in its Stream Server Data Model, in the corresponding outgoing replication messages.

b. When supporting <forward compatibility> as per Paragraph b of Chapter 4.5, the CSD Stream Server SHALL include received additional attributes and/or entities that are not in its CSD Stream Server Data Model and were provided by the Sensor or an incoming replication message in the corresponding outgoing replication messages.

c. When present in incoming messages, the NSIL_STREAM.sourceURL and NSIL_RELATED_FILE.URL attributes SHALL be replaced to a local URL, in the same way that NSIL_FILE.productURL is replaced in the CSD-Server synchronization, see [AEDP-17 Annex B-3.8 Synchronization Using CORBA Specific Requirements].

d. NSIL_CARD.status SHALL indicate the Stream status as follows:

    i. NEW: the Sensor is starting transmitting the Stream

    ii. CHANGED: the Sensor is providing a metadata update

    iii. OBSOLETE: the source CSD Stream Server is indicating that the Stream data has been removed so may no longer be replayed/relayed or downloaded.

e. At the end of a Live Stream, the NSIL_COVERAGE.temporalEnd attribute MUST be populated in the corresponding outgoing replication message indicating that the Stream has finished.

4.4 CSD STREAM METADATA FUSION

Multiple CSD Stream Data Model attributes may be fused in order to form a single query result, or a single outgoing replication message. All the dynamic attributes already support multiple values as follows:

a. NSIL_COMMON.subjectCategoryTarget, NSIL_COMMON.targetNumber, NSIL_COVERAGE.spatialCountryCode and NSIL_VIDEO.category are comma-separated text lists and multiple values SHALL be added to the list.

b. NSIL_COVERAGE.spatialGeographicReferenceBox represents a geospatial bounding box and multiple values SHALL be fused by the geospatial union operator.

c. NSIL_COVERAGE.temporalEnd represents the latest date/time and multiple values SHALL be fused by the maximum date/time operator.

d. NSIL_COVERAGE.temporalStart represents the earliest date/time and multiple values SHALL be fused by the minimum date/time operator.

e. THUMBNAIL_FIRST_FRAME and THUMBNAIL_LAST_FRAME Related Files SHALL be fused by using the earliest and latest THUMBNAIL respectively.

4.5 CSD STREAM SERVER <FORWARD COMPATIBILITY>

The feature <forward compatibility> allows different CSD Stream Servers with unique metadata extensions to interoperate, subject to the constraints detailed in the AEDP-17 Annex B NSIL SPECIFICATION AND REQUIREMENTS.

1. A CSD Stream Server SHALL support the feature <forward compatibility>.

   a. A CSD Stream Server MAY realize support for the feature <forward compatibility> by providing the feature <extending its metadata model>.

   b. A CSD Stream Server MAY realize support for the feature <forward compatibility> by providing the feature <accepting catalogue entries not covered by its metadata model>.

2. A CSD Stream Server SHALL have the ability to provide to other CSD Stream Servers all metadata model extensions it is supporting.

3. INFORMATIVE: The receiving CSD-Server is not obliged to use the same metadata model extensions provided by another CSD-Server. For example, there may exist differences in the set of extensions supported by each individual CSD-Server based on (national) security reservations. This should be prevented as much as possible from an operational point of view.

4. Metadata attributes defined in extensions not explicitly covered by the CSD Stream Server Data Model pursuant to 4.1 and AEDP-17 Annex G NSIL METADATA MODEL MUST NOT be nor treated as dynamic (as defined in 4.2)

5. The requirements under each use case above have already provided functional support and further clarification for the CSD Stream Server feature <forward compatibility>.

4.5.1 <Forward compatibility> behavior examples [INFORMATIVE]

1. Figure 4-2 illustrates <forward compatibility> as per a of 4.5, which operates as follows:

   a. A sensor operating on Baseline Y makes the definition of its metadata model extensions known to the community.

   b. The middle CSD Stream Server, operating on Baseline X, takes the definitions of the extensions for Baseline Y and updates its data model based on them (possibly, after the usage of these Baseline Y extensions has received approval from an external process). This CSD Stream Server

may now accept and index both Baseline X and Y extensions (as long as they are not in conflict).

The two former steps are expected to be performed only once, and upon joining of the sensor or the CSD Stream Server to the network.

c. A sensor operating on Baseline Y generates a notification message containing Core Metadata, together with the Baseline Y Extension Metadata.

d. The content of this notification message is stored and indexed by the sensor's local CSD Stream Server.

e. The local CSD Stream Server for the sensor generates an outgoing replication message containing both the Core Metadata and the Baseline Y Extension Metadata.

f. The middle CSD Stream Server, operating now on Baseline X + Y, receives the incoming replication message and indexes both the content of the Core Metadata and the Baseline Y extension.

g. The middle CSD Stream Server, operating in Bridgehead mode (see Section A-7.8), also generates an outgoing CSD Stream Server replication message containing both the Core Metadata and the Baseline Y Extension Metadata.

h. The right-hand CSD Stream Server, operating on Baseline Y, receives the incoming replication message, stores and indexes the content of both the Core Metadata and the Baseline Y Extension Metadata.

i. The exploitation system connected to the right-hand CSD Stream Server can therefore search for Core Metadata and Baseline Y Extension Metadata, even though a different baseline CSD Stream Server was present in the replication topology originally.

j. Furthermore, any exploitation system connected to the middle CSD Stream Server can still search for both Core Metadata and Baseline Y extensions, originating from sensors connected to either the left-hand or right-hand CSD Stream Server, and will receive Baseline Y Extension metadata in any matching results.

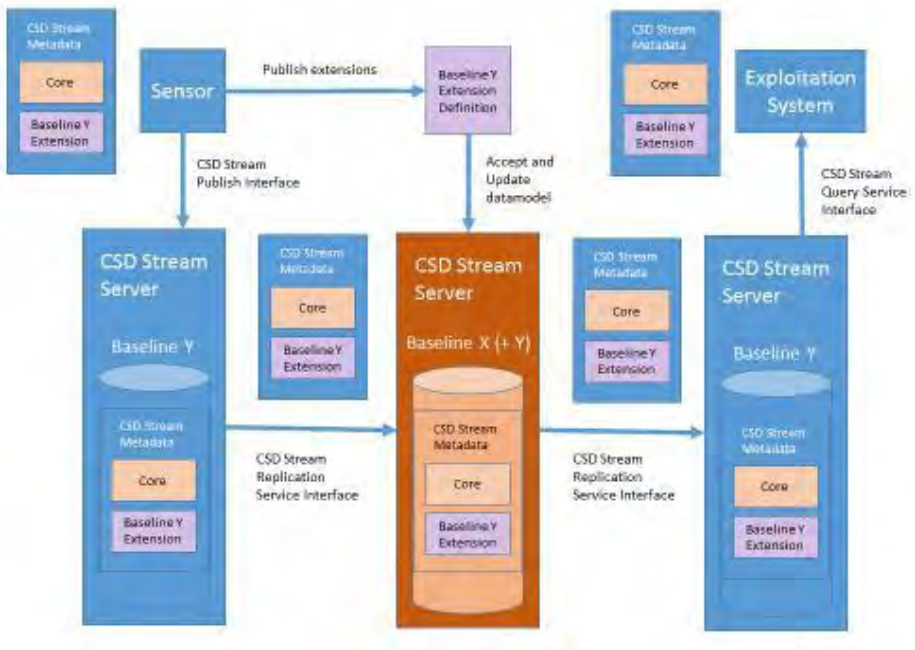

Figure 4-2 Example 1 of multiple baseline CSD Stream Server deployment

2. Figure 4-3 illustrates <forward compatibility> as per b of 4.5, which operates as follows:

a. A sensor operating on Baseline Y generates a notification message containing Core Metadata, together with the Baseline Y Extension Metadata.

b. The content of this notification message is stored and indexed by the sensor's local CSD Stream Server.

c. The local CSD Stream Server for the sensor generates an outgoing replication message containing both the Core Metadata and the Baseline Y Extension Metadata.

d. The middle CSD Stream Server, operating on Baseline X, receives the incoming replication message and indexes the content of only the Core Metadata, ignoring the Baseline Y Extension Metadata. The Baseline Y Extension Metadata is persisted, but not indexed.

e. The middle CSD Stream Server, operating in Bridgehead mode (see Section A-7.8), also generates an outgoing CSD Stream Server replication message containing both the Core Metadata and the Baseline Y Extension Metadata.

f. The right-hand CSD Stream Server, operating on Baseline Y, receives the incoming replication message, stores and indexes the content of both the Core Metadata and the Baseline Y Extension Metadata.

g. The exploitation system connected to the right-hand CSD Stream Server can therefore search for Core Metadata and Baseline Y Extension Metadata, even though a different baseline CSD Stream Server was present in the replication topology.

h. Furthermore, any exploitation system connected to the middle CSD Stream Server can still search for Core Metadata originating from sensors connected to either the left-hand or right-hand CSD Stream Server, and will also receive Baseline Y Extension metadata in any matching results

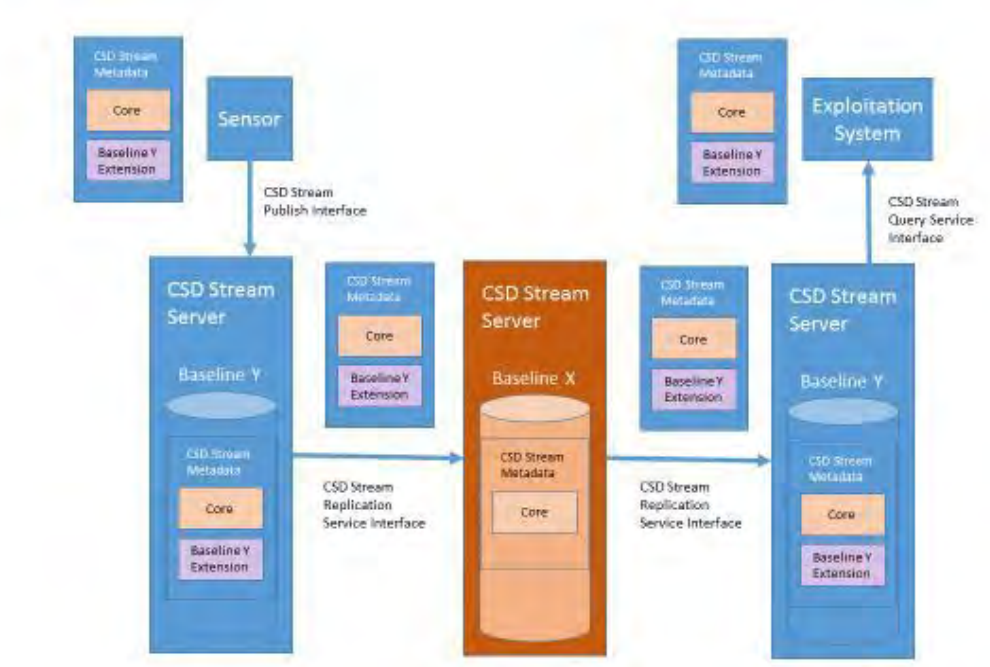

Figure 4-3 Example 2 of multiple baseline CSD Stream Server deployment

4.6 NSILI BRIDGE METADATA

1. The proxy Stream product created in the associated NSILI library is intended to be as minimal as possible. This metadata is shown in Table 4-2.

2. Note that NSIL_STREAM.sourceURL is deliberately omitted – using the NSILI metadata, the local CSD Stream Server should be queried by Stream identifier to find the full metadata for the Stream.

3. Note also that it is the NSIL_COMMON.identifierUUID attribute which links Stream metadata in the NSILI library with the CSD Stream Server. It is not the

NSIL_CARD.identifier attribute since that is assigned by the NSILI library and may be different from the NSIL_CARD.identifier attribute value provided by the CSD Stream Server.

Table 4-2 NSILI Bridge Metadata

Entity	Attribute	Value
NSIL_CARD	publisher	As per business rules agreed for a particular mission
NSIL_STREAM	creator	From CSD Stream Publish Interface metadata
	dateTimeDeclared	From CSD Stream Publish Interface metadata
NSIL_METADATA_SECURITY	classification	From CSD Stream Publish Interface metadata
	policy	
	releasability	
NSIL_SECURITY, NSIL_PART:NSIL_SECURITY	classification	From CSD Stream Publish Interface metadata
	policy	
	releasability	
NSIL_PART	partIdentifier	Not defined
NSIL_PART: NSIL_COMMON	identifierUUID	From CSD Stream Publish Interface metadata
	type	

INTENTIONALLY BLANK

CHAPTER 5 MOTION IMAGERY SPECIFIC CONSIDERATIONS

5.1 CSD STREAM METADATA (INFORMATIVE)

5.1.1 CSD Stream Publish Interface (INFORMATIVE)

Table 5-1 provides an example of CSD Stream Metadata generated by a motion imagery sensor and submitted via the CSD Stream Publish Interface. Note that subsequent Stream updates will use the same NSIL_COMMON.identifierUUID value, but will have an NSIL_CARD.status value of CHANGED. When the Stream ends, a similar notification should be made with an NSIL_CARD.status value of OBSOLETE and NSIL_COVERAGE.temporalEnd value specifying the time at which the Stream stopped.

5.1.2 CSD Stream Query Service Interface (INFORMATIVE)

Table 5-2 provides an example of CSD Stream Metadata for a motion imagery stream returned by a CSD Stream Server via the getMetadata operation of the CSD Stream Query Service Interface.

5.2 STREAM DOWNLOAD (INFORMATIVE)

Downloaded Motion Imagery stream data must consist of STANAG 4609 NATO Digital Motion Imagery Standard format data.

5.3 STREAM REPLAY

When replaying motion imagery Streams, the CSD Stream Server SHALL ensure that the client can immediately decode and process the Stream by transmitting a Stream following the AEDP-8 rules for clipping [AEDP-8 Edition 3, 2009].

Table 5-1 Example of CSD Stream Metadata sent by a motion imagery sensor

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	**status**	**NEW**	Set to NEW since this is the first notification of the Stream
	publisher	ACC XXN	
NSIL_PRODUCT: NSIL_STREAM	creator	USA pred-sim	
	dateTimeDeclared	2008/03/20 15:20:05.000	Indicates the date/time of the start of the Stream
	standard	STANAG4609	
	standardVersion	3.0	
	sourceURL	**udp://@239.0.0.1:1234**	Provide the technical parameters necessary to connect to the Stream
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	**identifierUUID**	**0339b4de-613e-4b55-9d47-164151674c06**	Provide a unique Stream identifier. Subsequent updates must use the same identifier for this Stream.
	source	Predator	
	type	VIDEO	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	spatialGeographic-ReferenceBox	POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))	This is the area covered from <temporalStart of previous notification> to <temporalStart of this notification>
	temporalStart	**2012/12/06 13:47:55.000**	Instantaneous date/time of the Stream. **temporalEnd** is not present since the Stream is still being transmitted.

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART: NSIL_VIDEO	averageBitRate	2.0	
	category	VIS	
	encodingScheme	264ON2	
	frameRate	29.97	
	numberOfRows	480	
	numberOfColumns	640	
	metadataEncodingScheme	KLV	
	MISMLevel	4	
	scanningMode	PROGRESSIVE	
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_RELATED_FILE	fileType	THUMBNAIL	
	URL	http://web-repository/thumbnail_predator_001.jpg	The file provided in this URL will be downloaded by the CSD Stream Server to serve it as thumbnail of the stream.

Table 5-2 Example of CSD Stream Metadata returned by a CSD Stream Server via the CSD Stream Query Service Interface for a motion imagery stream

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	**identifier**	**3c816a55-0fe3-4508-8a68-5bfa86d5492e**	The Stream may be replayed using this identifier value via the CSD Stream Controller Service Interface
	sourceDateTimeModified	2012/12/06 14:08:28.413	
	dateTimeModified	2012/12/06 14:08:28.413	
	status	CHANGED	
	numberOfParts	1	
	publisher	ACC XXN	
	sourceLibrary	wsstream-catalina	Identifies the source CSD Stream Server at which this Stream was recorded.
NSIL_PRODUCT: NSIL_STREAM	creator	USApred-sim	
	dateTimeDeclared	2008/03/20 15:20:05.000	
	standard	STANAG4609	
	standardVersion	3.0	
	sourceURL	**http://wsstream-catalina/download?id=3c816a55-0fe3-4508-8a68-5bfa86d5492e**	Optional URL allowing the stream contents to be downloaded.

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	identifierUUID	0339b4de-613e-4b55-9d47-164151674c06	
	source	Predator	
	type	VIDEO	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	spatialGeographic-ReferenceBox	POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))	
	temporalEnd	2012/12/06 14:06:28.413	
	temporalStart	2012/12/06 13:47:55.000	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART: NSIL_VIDEO	averageBitRate	2.0	
	category	VIS	
	encodingScheme	264ON2	
	frameRate	29.97	
	numberOfRows	480	
	numberOfColumns	640	
	metadataEncodingScheme	KLV	
	MISMLevel	4	
	scanningMode	PROGRESSIVE	
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT NSIL_RELATED_FILE	**fileType**	THUMBNAIL	
	url	**http://wsstream-catalina/download?id=d0a3baed-1e90-43fa-bbf6-475c99f264dc**	This is the thumbnail provided in the first notification of the stream
	extent	0.1	
	isFileLocal	true	

Entity	Attribute	Value	Remark
NSIL_PRODUCT NSIL_RELATED_FILE	**fileType**	THUMBNAIL_FIRST_FRAME	
	URL	**http://wsstream-catalina/download?id=3890cc9e-d570-47c9-9e11-d4e241fd2b86**	This is the thumbnail provided in the notification for temporalStart 2012/12/06 13:47:55.000 (the start of the substream represented by this query result)
	extent	0.1	
	isFileLocal	true	
NSIL_PRODUCT NSIL_RELATED_FILE	**fileType**	THUMBNAIL_LAST_FRAME	
	URL	**http://wsstream-catalina/download?id=0fb2b1ec-a10a-4e1c-bf67-dc692d4b134d**	This is the thumbnail provided in the notification for temporalStart 2012/12/06 14:06:28.413 (the start of the substream represented by this query result)
	extent	0.1	
	isFileLocal	true	

CHAPTER 6 GMTI SPECIFIC CONSIDERATIONS

6.1 CSD STREAM METADATA (INFORMATIVE)

6.1.1 CSD Stream Publish Interface (INFORMATIVE)

Table 6-1 provides an example of CSD Stream Metadata generated by a GMTI sensor and submitted via the CSD Stream Publish Interface. Subsequent Stream updates will use the same NSIL_COMMON.identifierUUID value, but will have an NSIL_CARD.status value of CHANGED. When the Stream ends, a similar notification should be made with an NSIL_CARD.status value of OBSOLETE and an NSIL_COVERAGE.temporalEnd value specifying the time at which the Stream stopped.

6.1.2 CSD Stream Query Service Interface (INFORMATIVE)

Table 6-2 provides an example of CSD Stream Metadata for a GMTI stream returned by a CSD Stream Server via the getMetadata operation of the CSD Stream Query Service Interface.

6.2 STREAM DOWNLOAD (INFORMATIVE)

Downloaded GMTI stream data MUST consist of STANAG 4607-format [NATO, NATO Ground Moving Target Indicator (GMTI) Format, 2010] binary data containing all packets in the time range satisfying the query. There MUST NOT be any padding bytes between packets.

6.3 STREAM REPLAY

When replaying GMTI Streams, the CSD Stream Server MUST ensure that the relevant releasability, mission and job segments are transmitted ahead of the relevant dwell segments so that the client can immediately begin to decode and process the Stream.

Table 6-1 CSD Stream Metadata sent by GMTI sensor through WS-Notification

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	**status**	**NEW**	Set to NEW since this is the first notification of the Stream
	numberOfParts	1	
	publisher	ACC XXN	
NSIL_PRODUCT: NSIL_STREAM	creator	USA u2-sim	
	dateTimeDeclared	2008/03/20 15:20:05.000	Indicates the date/time of the start of the Stream.
	standard	STANAG4607	
	standardVersion	3.0	
	sourceURL	**udp://@:8204**	Provide the technical parameters necessary to connect to the Stream
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	**identifierUUID**	**0339b4de-beed-4b55-9d47-164151674c06**	Provide a unique Stream identifier. Subsequent updates must use the same identifier for this Stream
	source	U2	
	type	GMTI	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	spatialGeographic-ReferenceBox	POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))	This is the area covered from <temporalStart of previous notification> to <temporalStart of this notification>
	temporalStart	**2012/12/06 13:47:55.000**	Instantaneous date/time of the Stream. **temporalEnd** is not present since the Stream is still being transmitted
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

Table 6-2 Example of CSD Stream Metadata returned by a CSD Stream Server via the CSD Stream Query Service Interface for a GMTI stream

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	**identifier**	**3c816a55-0fe3-4508-8a68-5bfa86d5492e**	The Stream may be replayed using this identifier value via the CSD Stream Controller Service Interface
	sourceDateTimeModified	2012/12/06 14:08:28.413	
	dateTimeModified	2012/12/06 14:08:28.413	
	status	CHANGED	
	numberOfParts	1	
	publisher	ACC XXN	
	sourceLibrary	wsstream-fra-phy	Identifies the source CSD Stream Server at which this Stream was recorded.

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_STREAM	creator	USA u2-sim	
	dateTimeDeclared	2008/03/20 15:20:05.000	
	standard	STANAG4607	
	standardVersion	3.0	
	sourceURL	**http://wsstream-fra-phy/download?id=3c816a55-0fe3-4508-8a68-5bfa86d5492e&start=&end=**	Optional URL allowing the stream contents to be downloaded.
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	identifierUUID	0339b4de-613e-4b55-9d47-164151674c06	
	source	U2	
	type	GMTI	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	spatialGeographic-ReferenceBox	(POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))	
	temporalEnd	2012/12/06 14:06:28.413	
	temporalStart	2012/12/06 13:47:55.000	
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

CHAPTER 7 LINK 16 SPECIFIC CONSIDERATIONS

7.1 CSD STREAM METADATA (INFORMATIVE)

Unlike motion imagery and GMTI Streams, there is no single source of Link 16 Streams, hence recording of Link 16 Streams is performed automatically by the local CSD Stream Server. Furthermore, there is no support in the current specification for replication of Link 16 metadata (it is assumed that an external service will be used for distribution of Link16 streams across the network) and no support for late joiners to catch up on historic Link 16 metadata.

7.1.1 CSD Stream Publish Interface (INFORMATIVE)

Table 7-1 provides an example of CSD Stream Metadata for a Link 16 stream. There are no Link 16 sensors by its very nature, though this interface may be used by Link 16 aggregator systems, retrieving tracks from several sources and relaying them to the CSD Stream Server. Otherwise, the Publish Interface may not be used for Link 16 streams, but the local Stream Server be configured to automatically record any Link 16 streams.

7.1.2 CSD Stream Query Interface (INFORMATIVE)

Table 7-2 provides an example of CSD Stream Metadata for a Link 16 stream returned by a CSD Stream Server via the getMetadata operation of the CSD Stream Query Service Interface.

7.2 STREAM DOWNLOAD

Downloaded Link 16 stream data MUST consist of STANAG 5516 formatted binary data containing all messages in the time range satisfying the query. Each message MUST include a NACT header, see [DOP-MAJIIC-068 LINK16] for reference. There MUST NOT be any padding bytes between messages.

7.3 STREAM REPLAY

There are no specific Replay requirements for Link 16 data.

Table 7-1 CSD Stream Metadata sent by Link16 aggregator through WS-Notification

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	**status**	**NEW**	Set to NEW since this is the first notification of the Stream
	numberOfParts	1	
	publisher	ACC XXN	
NSIL_PRODUCT: NSIL_STREAM	creator	FRA stream-1	
	dateTimeDeclared	2008/03/20 15:20:05.000	Indicates the date/time of the start of the Stream.
	standard	STANAG5516_NACT	
	standardVersion	4.0	
	sourceURL	**udp://@:8204**	Provide the technical parameters necessary to connect to the Stream
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	**identifierUUID**	**df048f2c-a2f0-4475-a0ba-6fdb4bf84068**	Provide a unique Stream identifier. Subsequent updates must use the same identifier for this Stream
	source	FRA stream-1	
	type	TDL DATA	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	**temporalStart**	**2012/12/06 13:47:55.000**	Instantaneous date/time of the Stream. **temporalEnd** is not present since the Stream is still being transmitted
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	USA	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR, NATO	

Table 7-2 Example of CSD Stream Metadata returned by a CSD Stream Server via the CSD Stream Query Service Interface for a Link 16 stream

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_CARD	identifier	a73f52d3-c4fc-4428-94a6-9658a17cbdf7	The Stream may be replayed using this identifier value via the CSD Stream Controller Service Interface
	sourceDateTimeModified	2012/12/06 14:08:28.413	
	dateTimeModified	2012/12/06 14:08:32.315	
	status	NEW	
	numberOfParts	1	
	publisher	ACC XXN	
	sourceLibrary	wsstream-fra-phy	Identifies the source CSD Stream Server at which this Stream was recorded.

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_STREAM	creator	FRA stream-1	
	dateTimeDeclared	2008/03/20 15:20:05.000	
	standard	STANAG5516_NACT	
	standardVersion	4.0	
	sourceURL	http://wsstream-fra-phy/download?id= a73f52d3-c4fc-4428-94a6-9658a17cbdf7&start=&end=	Optional URL to download stream contents.
NSIL_PRODUCT: NSIL_METADATASECURITY	classification	UNCLASSIFIED	
	policy	NATO	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR,USA	
NSIL_PRODUCT: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	NATO	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR,USA	
NSIL_PRODUCT: NSIL_PART	partIdentifier	1	

Entity	Attribute	Value	Remark
NSIL_PRODUCT: NSIL_PART: NSIL_COMMON	identifierUUID	df048f2c-a2f0-4475-a0ba-6fdb4bf84068	
	source	FRA stream1	
	type	TDL DATA	
NSIL_PRODUCT: NSIL_PART: NSIL_COVERAGE	temporalEnd	2012/12/06 13:47:55.000	
	temporalStart	2012/12/06 12:47:55.000	
NSIL_PRODUCT: NSIL_PART: NSIL_SECURITY	classification	UNCLASSIFIED	
	policy	NATO	
	releasability	CAN,DEU,ESP,FRA,GBR,ITA, NLD,NOR,USA	

ANNEX A STREAM SERVER SERVICE INTERFACES

TABLE OF CONTENT

Tables

Figures

A-1 PURPOSE

1. This Annex defines the CSD Stream Server Service Interfaces.

2. The CSD Stream Server provides six interfaces which are summarized in Table A-1 and illustrated in Figure A-1.

Table A-1 CSD Stream Server interfaces

CSD Stream Publish Interface	Used by sensors (i.e. Stream producers) to declare Live Streams.
CSD Stream Query Service Interface	Used by exploitation systems (i.e. Stream consumers) to query for both Live Streams and Recorded Streams.
CSD Stream Controller Service Interface	Used by exploitation systems (i.e. Stream consumers) to request relay of Live Streams or replay of Recorded Streams.
NSILI Bridge	Used by the CSD Stream Server to store minimal Stream metadata in an associated NSILI Server, allowing associations to be made from tasks to Streams, or from intelligence products (e.g. reports or motion imagery clips) to Streams for the purpose of supporting the IRM&CM process.
CSD Stream Notification Interface (OPTIONAL)	Used to notify consumers of new relayed Live Streams or replayed Recorded Streams which they may be able to access.
CSD Stream Replication Service Interface	Used to exchange CSD Stream Metadata between CSD Stream Servers to provide a coherent enterprise coalition view of all Stream products.

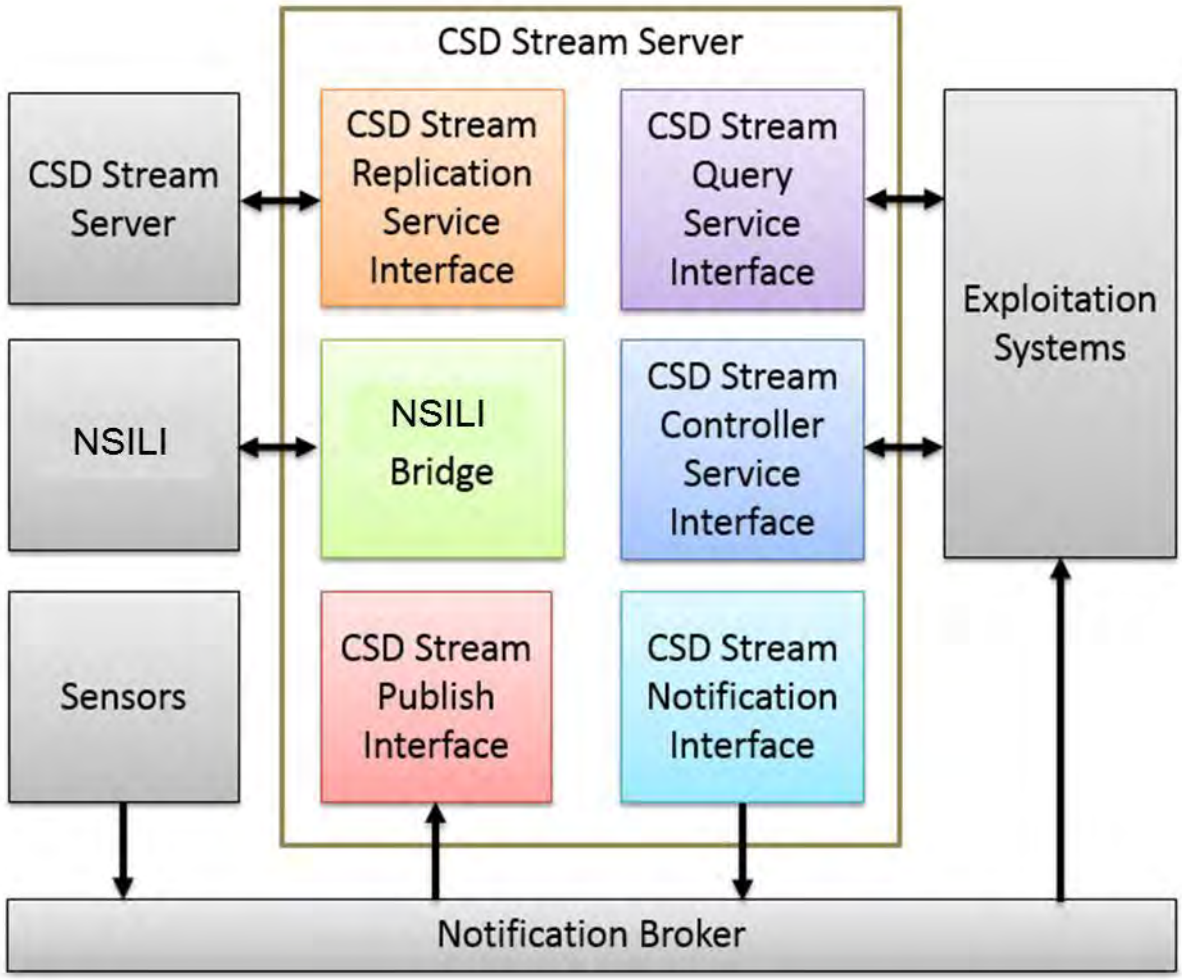

Figure A-1 CSD Stream Server Interfaces

A-2 CSD STREAM PUBLISH INTERFACE

1. Each CSD Stream Server MUST provide a CSD Stream Publish Interface. This interface MUST be in accordance with [AEDP-19 Annex A-2 PUB-SUB SERVICE (WS-Notification)].

2. Each Stream sensor MUST emit notifications describing their Live Stream(s) to their local notification broker. Each CSD Stream Server MUST subscribe for notifications from their local notification broker. Multiple CSD Stream Server instances SHOULD not be connected to the same notification broker, as deconfliction of this is outside the scope of this specification.

A-2.1 Notification Topic

1. The notification topic, dialect and namespace MUST be in accordance with AEDP-19 Annex H WORKFLOW SERVICES BUSINESS RULES and AEDP-19 Annex A-2 PUB-SUB SERVICE (WS-Notification).

2. The notification topic, dialect and namespace and broker endpoint MUST be provided by the notification.topic.sensor and notification.broker.endpoint values returned by the getCapabilities method of the CSD Stream Controller Service Interface (see A-4).

A-2.2 Notification Payload

The notification payload MUST be a single CSD Stream Metadata entry serialized as a Catalogue Entry in accordance with [AEDP-17 Annex E-5 CATALOGUE ENTRIES] using the EntityContainerType. This EntityContainerType MUST be contained within the MAJIICNotificationBody of a MAJIICNotificationEnvelope. CSD Stream Metadata MUST be as described in CHAPTER 4 for this interface.

A-2.3 Dynamic

1. Sensors SHOULD provide regular updates so that the CSD Stream Server can discover the Live Stream if the initial notification was missed, and to maintain an accurate representation of the present state of the Live Stream.

2. A CSD Stream Server MAY also augment the CSD Stream Metadata with embedded Stream Metadata (e.g. KLV for motion imagery) in order to create a finer-grained metadata index for subsequent search operations.

3. If a Stream sensor changes any of the Static CSD Stream Metadata (see Chapter 4.2), a notification indicating that the Live Stream has ended MUST be sent, followed by a notification describing the new Live Stream which is starting. The Stream identifier MUST be changed.

A-2.4 Feeders

In order to accommodate legacy Sensors which do not support the CSD Stream Publish Interface, it is RECOMMENDED that a CSD Stream Server implementation provides Feeders for one or more stream formats. A Feeder receives Stream data, extracts the corresponding metadata and creates suitable notifications, either direct to the CSD Stream Server or via the notification broker. The provision, configuration and management of Feeders is outside the scope of this specification.

A-3 CSD STREAM QUERY INTERFACE

1. Each CSD Stream Server MUST implement a CSD Query Service Interface in accordance with [AEDP-17 Edition A Version 1, Annex E-3 QUERY SERVICE SPECIFICATION], with the additions detailed below, which together are known as the CSD Stream Query Service Interface.

2. The purpose of this interface is to discover Live Streams, and search for Live and Recorded Streams.

A-3.1 Behaviour

1. The CSD Stream Query Service Interface returns a set of CSD Stream Metadata describing the Streams matching the input Boolean Query Syntax (BQS) query in accordance with in accordance with [AEDP-17 Annex I BQS SYNTAX AND SEMANTICS].

2. A CSD Stream Server MUST provide the ability to relay/replay streams via the CSD Stream Controller Service (see A-4).

3. A CSD Stream server MAY provide the ability to download streams via the CSD Stream Controller Service Interface (see 4.3.3).

4. An Exploitation System MAY provide the ability to replay streams via the CSD Stream Controller Service Interface (see 4.3.3)

5. An Exploitation System MAY provide the ability to download streams via the CSD Stream Controller Service Interface (see A-4).

6. The subset of CSD Stream Metadata used by the CSD Stream Server is detailed in CHAPTER 4.

7. See Chapters 5.1.2, 6.1.2 and 7.1.2 for sample CSD Stream Metadata returned by this interface for motion imagery, GMTI and Link 16, respectively.

A-3.2 Additional Capabilities

In addition to the getCapabilities capabilities defined in AEDP-17 Annex D NSIL WEB SERVICE CORE INTERFACE and AEDP-17 Annex E-2 QUERY SERVICE SPECIFICATION the following additional capabilities MAY be declared. If either value is omitted, its value MUST be assumed to be false.

Table A-2 Additional capabilities reported by the getCapabilities method of the CSD stream query service interface

Name	GenericValue class	Usage
query.advanced	BooleanValue	This value MUST indicate whether Advanced Query operation is available. See A-3.3.
query.federated	BooleanValue	As Federated Query is not supported by this version of AEDP-18, this value must always be set to false.

A-3.3 Advanced Query Capability (OPTIONAL)

1. The original NSILI paradigm has a one-to-one mapping between products and query results i.e. each product is represented by at most one result. This means that for a given query, the user may be presented with a single search result indicating that a certain Stream satisfies their query term(s), but without any indication of the portion of that Stream which is relevant e.g. a five minute segment within a twelve hour mission.

2. If the CSD Stream Query Service Interface is used with the “AdvancedSearch” GenericValue set to true, the CSD Stream Server MAY return multiple results (substreams) pointing to the same Stream.

3. These substreams are distinguished by different start and end times as provided in the CSD Stream Metadata.

4. It is RECOMMENDED that the versionDescription part of the URI returned by the getQueryResultSegment is used to encode the time range of the substream for subsequent call to getMetadata.

5. When using the Advanced Query Capability, a CSD Stream Server implementation may need to cope with fusing multiple metadata values to create a single composite value. For example, for a motion imagery stream, the server may form a geospatial bounding box from the union of all the individual sensor footprints within the substream. Metadata fusion is further described in CHAPTER 4.

6. If a URL is provided for a client to download substream data, that URL MUST point to the same subset of the Stream as indicated in the substream metadata. i.e. if the URL is accessed, it MUST return the portion (e.g. a motion imagery clip) as described by the substream metadata. The URL syntax is up to the CSD Stream Server implementation.

A-3.4 Example of Stream Queries (INFORMATIVE)

In this section the term “LocalServerName” refers to the library name of the CSD Stream Server to which the client is connected.

A-3.4.1 Discovering Local Live Streams

1. Example of query:

(NSIL_CARD.sourceLibrary = ‘LocalServerName’) AND
not (NSIL_COVERAGE.temporalEnd exists)

2. The first term limits results to those Streams whose data is actually stored in the local CSD Stream Server. This may be omitted if the client wants to discover all Live Stream across the coalition.

3. The second term limits results to Live (rather than Recorded) Streams.

A-3.4.2 Searching for Recorded Streams by Time and Location

1. Search for Recorded Streams between date/time T1 and T2 and inside area A.

2. Example of query:

(NSIL_COVERAGE.temporalStart > T1 AND
NSIL_COVERAGE.temporalEnd < T2) AND
(NSIL_COVERAGE.spatialGeographicReferenceBox INTERSECT A)

3. The first two terms limits results to those Streams within the time range (T1, T2) while the third term limits results to those Streams inside geographic area A. If this query is executed using Advanced Query Capability, multiple substreams may be returned if a Stream revisited the same geographic location within the specified time range.

A-4 CSD STREAM CONTROLLER SERVICE INTERFACE

1. Each CSD Stream Server MUST provide a CSD Stream Controller Service Interface in accordance with Annex B STREAM CONTROLLER SERVICE INTERFACE. The purpose of this interface is to relay Live Streams and replay Recorded Streams.

2. This interface does not currently provide more advanced functions such as fast-forwarding or rewind. Exploitation systems MAY provide some of these functions on the client side.

A-4.1 Remote Stream Relay/Replay

1. Using CSD Stream Server Metadata Replication, an exploitation system MAY receive metadata corresponding to a Stream which is stored at a remote CSD Stream Server. Given the principle that a local exploitation system must only connect to its local CSD Stream Server, the local CSD Stream Server must then be able to:

a. Identify the CSD Stream Server through which the Stream data is accessible. Note that this is not necessarily the originating CSD Stream Server at which the Stream data is stored.

b. Request a relay/replay of the Stream from that CSD Stream Server to the local CSD Stream Server.

c. Forward the Stream data to the local exploitation system using the parameters specified by the exploitation system.

This is achieved as follows:

2. Each CSD Stream Server MUST maintain a mapping from library identifier to the CSD Stream Controller Service Interface endpoint through which Streams with that source library identifier value may be accessed. Note that this is not necessarily the same as the originating CSD Stream Server at which the Stream data is stored.

3. If the requested Stream is stored locally, the local CSD Stream Server MUST relay/replay the stream using the parameters specified by the client.

4. If the requested Stream is stored remotely, the local CSD Stream Server MUST use the source library identifier to determine the CSD Stream Controller Service Interface endpoint through which the Stream may be accessed. The local CSD Stream Server MUST request the Stream from that endpoint, and relay/replay it to the client using the specified parameters.

5. This process is illustrated in Figure A-2.

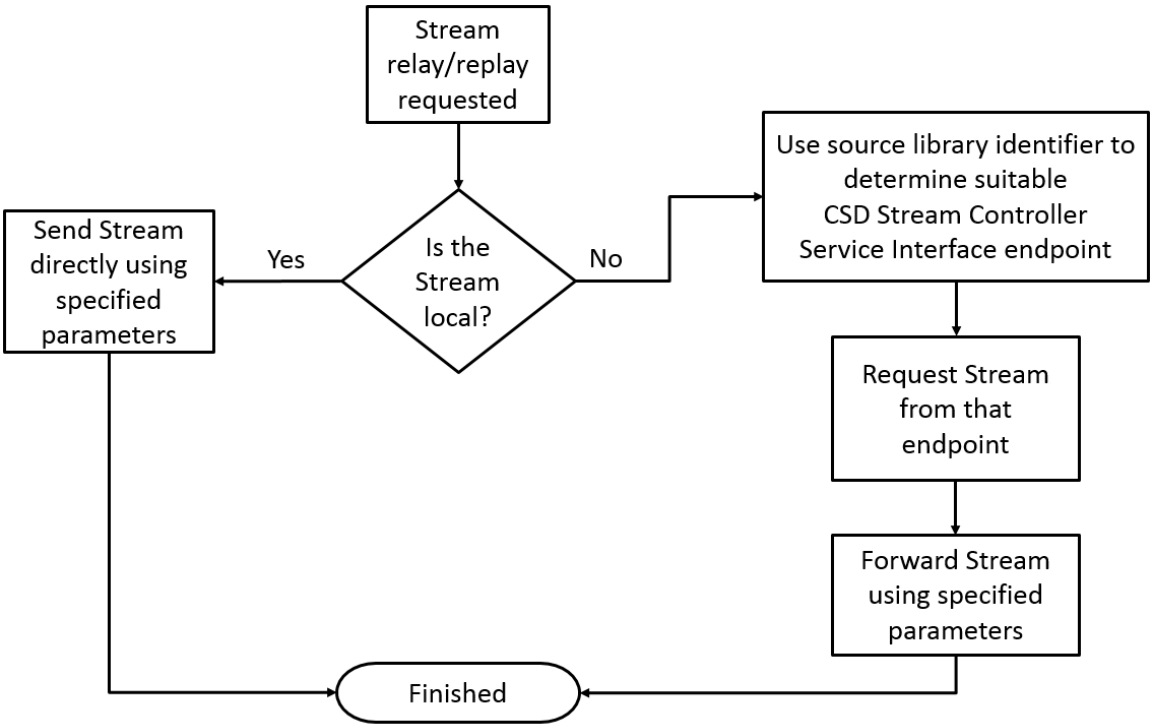

Figure A-2 Local and Remote Stream Relay/Replay

6. If the client has requested stream transcoding parameters (for example, changing the bitrate of a motion imagery stream), these parameters MAY be passed through the chain of requesting CSD Stream Servers to the originating CSD Stream Server, OR they MAY only be implemented when finally forwarding the Stream data to the client.

7. If a CSD Stream Server is aware of a WAN limitation (for example, bandwidth or maximum packet size constraints) on a particular link to a remote CSD Stream Server, it MAY use the CSD Stream Controller Service Interface transcoding capabilities to adapt the relayed/replayed Stream accordingly.

8. A CSD Stream Server MAY cache relayed/replayed Stream data if another request to the same portion of that Stream is expected in the future.

9. A CSD Stream Server SHOULD attempt to ensure that remote relayed streams are only sent once over the WAN, and then distributed to local clients as required.

A-4.2 Example (INFORMATIVE)

Figure A-3 shows an example of remote stream replay.

a. The exploitation system (IP address 1.a.b.2) requests replay of the Stream with identifier ESP_1 from its local CSD Stream Server (GBR, IP address 1.a.b.1).

b. CSD Stream Server (GBR) uses replicated Metadata to determine the source library identifier for ESP_1, in this case ESP. Its mapping indicates that streams with this source library identifier may be accessed through the CSD Stream Controller Service Interface endpoint http://fra/... Therefore CSD Stream Server (GBR) requests replay of the Stream from CSD Stream Server (FRA. IP address 2.a.b.c).

c. Similarly, CSD Stream Server (FRA) uses replicated Metadata to determine the source library identifier for ESP_1, again ESP. Its mapping indicates that streams with this source library identifier may be accessed through the CSD Stream Controller Service Interface endpoint http://esp/... Therefore CSD Stream Server (FRA) requests replay of the Stream from CSD Stream Server (ESP, IP address 3.a.b.c).

d. CSD Stream Server (ESP) stores the stream data for ESP_1 locally, therefore sends it to the specified IP address, corresponding to CSD Stream Server (FRA).

e. CSD Stream Server (FRA) forwards the stream data for ESP_1 received from CSD Stream Server (ESP) to the specified IP address, corresponding to CSD Stream Server (GBR).

f. Finally, CSD Stream Server (GBR) forwards the stream data for ESP_1 received from CSD Stream Server (FRA) to 1.a.b.2:1234, the IP address and port specified by the exploitation system.

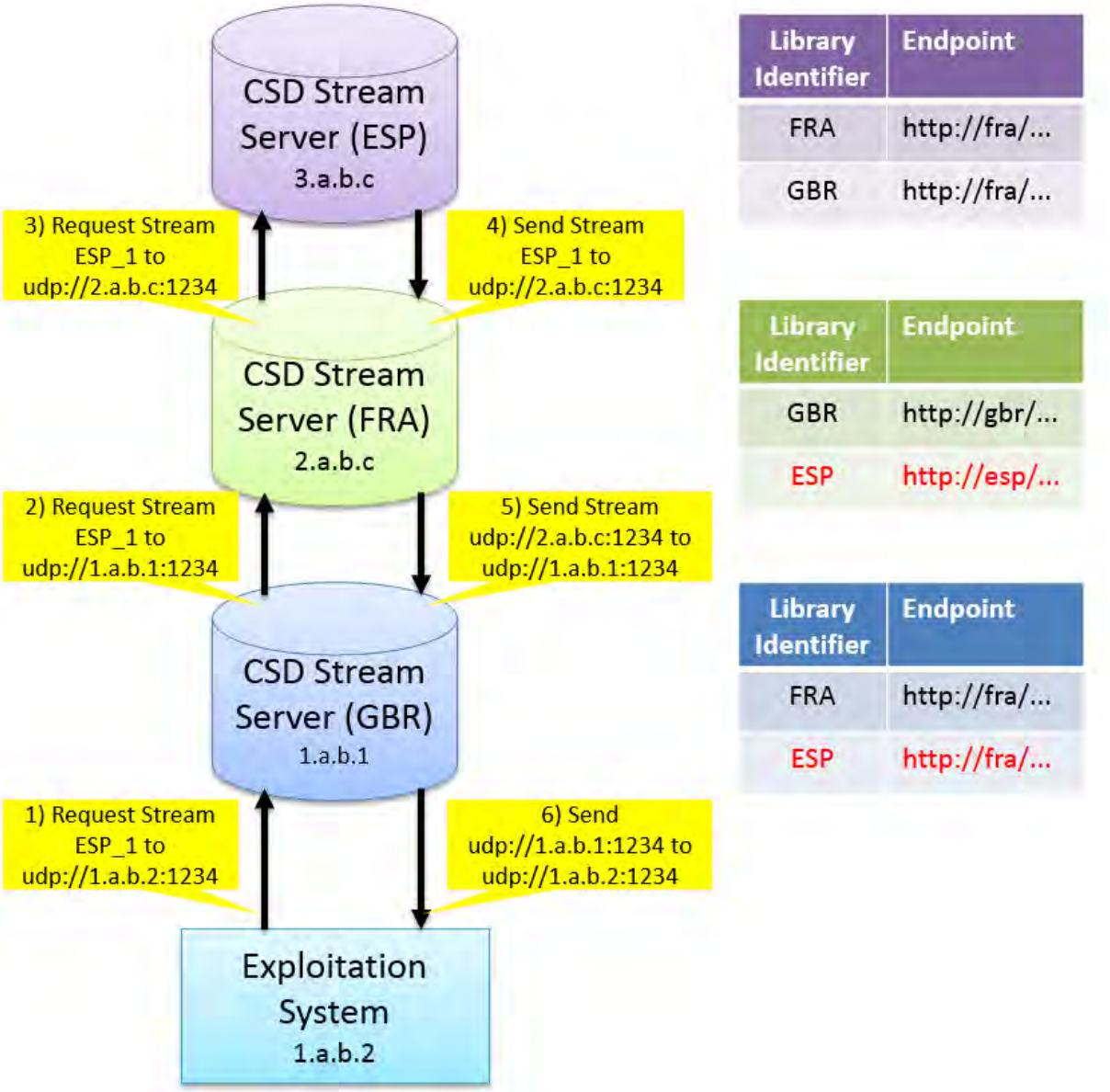

Figure A-3 Example of remote stream replay

A-5 NSILI BRIDGE

1. The purpose of this interface is to create “proxy” NSILI products to allow links or associations between Streams and other intelligence products. For example, a Task in the AEDP-19 Tasking Service may include a link to a NSILI proxy stream product (which is itself linked to a CSD Stream Server Stream by unique identifier) in order to indicate which Stream an exploitation system should exploit. Similarly, a NSILI association between a NSILI proxy stream product and a report indicates the Stream from which data was exploited in order to create the report.

2. The NSILI product MUST be created once when the Stream starts, and MUST be updated when the Stream ends. There SHOULD NOT be any intervening updates.

3. The NSILI product SHOULD contain only the metadata specified in Chapter 4.6.

A-5.1 Interface

1. The CSD Stream Server MUST use the CreationMgr interface in accordance with [AEDP-17 Annex C NSIL CORBA INTERFACE], or the CSD Publish Service Interface according to [AEDP-17 Annex E-4 PUBLISH SERVICE SPECIFICATION] (if implemented), of the NSILI server to create the NSILI proxy product.

2. The CSD Stream Server MUST use the UpdateMgr interface in accordance with [AEDP-17 Annex C NSIL CORBA INTERFACE], or the CSD Publish Service Interface according to [AEDP-17 Annex E-4 PUBLISH SERVICE SPECIFICATION] (if implemented), of the NSILI server to update the NSILI proxy product.

A-6 CSD STREAM NOTIFICATION INTERFACE (OPTIONAL)

1. Each CSD Stream Server MAY provide a CSD Stream Notification Interface. This interface MUST be in accordance with [AEDP-19 Annex A-2 PUB-SUB SERVICE (WS-Notification)]. As such, the requirements in this chapter only apply if the CSD Stream Notification Interface is implemented.

2. Each CSD Stream Server MUST emit notifications describing any relayed or replayed Stream which multiple local exploitation systems may be able to access. For example, this will never include any unicast Streams, but depending on local network constraints may include all broadcast or multicast Streams.

3. Exploitation systems MAY subscribe to their local notification broker, indicated by the getCapabilities response of the CSD Stream Controller service Interface, for notifications regarding Streams which may be of interest to the operator.

A-6.1 Notification Topic

1. The notification topic, dialect and namespace MUST be a full topic in accordance with [AEDP-19 Annex H WORKFLOW SERVICES BUSINESS RULES] and [AEDP-19 Annex A-2 PUB-SUB SERVICE (WS-Notification).

2. The notification topic, dialect, namespace and broker endpoint MUST be provided by the notification.topic.server and notification.broker.endpoint values returned by the getCapabilities method of the CSD Stream Controller Service Interface (see B-4.3.1).

A-6.2 Notification Payload

The notification payload MUST be a single CSD Stream Metadata entry serialized as a Catalogue Entry in accordance with [AEDP-17 Annex E-5 CATALOGUE ENTRIES] using the EntityContainerType. This EntityContainerType MUST be contained within the MAJIICNotificationBody of the MAJIICNotificationEnvelope. CSD Stream Metadata MUST be as described in Chapter 4.6 for this interface.

A-6.3 Dynamic

CSD Stream Servers SHOULD provide regular updates so that late joining Exploitation Systems can also discover Streams.

A-7 CSD STREAM REPLICATION SERVICE INTERFACE

1. Each CSD Stream Server MUST implement the CSD Stream Replication Service Interface.

2. The CSD Stream Metadata Replication Service Interface uses a similar approach to the Simple Persistence as a Service (SPS++) Replication Service Interface, see [AEDP-17 Annex C-3 SIMPLE PERSISTANCE AS A SERVICE (SPS++) SERVICE], namely the use of the “put” operation from WS Transfer [W3C, Web Services Transfer (WS-Transfer), 2006]. In the following text, the replicating CSD Stream Servers are known as “peers”.

A-7.1 Contract First Development

All of the web service specific specification is captured in a Replication Web Service Description Language (WSDL) file and the WSDLs and XSDs that it references.

A-7.2 Behaviour

1. Whenever the persisted state of a replicable Stream changes as a result of local interactions, that change MUST be reflected in an outgoing replication message sent to all peers. In addition, depending on the replication topology, see A-7.8, an incoming replication message MAY be reflected in an outgoing replication message also sent to all peers.

2. Either an individual replication message MUST be generated in response to every change of persisted state/incoming replication message, or multiple changes of persisted state/incoming replication messages for the same Stream MAY be fused into a single outgoing replication message in order to reduce the amount of bandwidth required for replication. This decision, and the time range over which to perform fusion (if any), SHOULD be made on a per-deployment basis depending on the acceptable bandwidth usage and the acceptable loss of search precision for peers. The method of CSD Stream Metadata fusion is detailed in Chapter 4.4.

3. All replication messages pertaining to the same Stream (i.e. the same Catalog Entry identifier) MUST be sent in increasing temporal order. The ordering of replication messages between different Streams (i.e. different Catalog Entry identifiers) is not defined.

4. Replication messages MUST represent the change (delta) in metadata since the last replication message for that Stream.

A-7.3 Replicable Streams

Motion imagery and GMTI streams are defined as being replicable. Link 16 (TDL) streams are not defined as being replicable. In a CSD Stream Server deployment, it is assumed that an external system (e.g. NIRIS) is responsible for Link 16 (TDL) forwarding between LANs, and therefore a CSD Stream Server can listen to its local LAN to record and index all live Link 16 (TDL) traffic.

A-7.4 Functional Requirements

Nearly all of the CSD Stream Server replication requirements fall in to the WS-Transfer [W3C, Web Services Transfer (WS-Transfer), 2006] and interface specifications found below. However a few others remain:

a. CSD Stream Server implementations MUST be able to dispatch messages to zero, one or more other remote CSD Stream Server instances.

b. CSD Stream Server implementations MUST provide runtime dynamic configurability of the remote CSD Stream Server endpoints receiving outgoing messages.

c. CSD Stream Server implementations MUST be able to receive messages from zero, one or more other remote CSD Stream Server instances.

A-7.5 Non Functional Requirements

1. With respect to replication, a number of Non-Functional Requirements (NFR) are considered to be mandatory and MUST be implemented by CSD Stream Server providers:

a. CSD Stream Server implementations MUST be able to provide a throughput of up to 475,200 messages per day. See A-7.9 for calculation of this value.

b. CSD Stream Server implementations MUST use an outgoing asynchronous pattern i.e. loosely coupled dispatch of the replication messages outgoing to other CSD Stream Server instances;

c. CSD Stream Server implementations MUST use an incoming asynchronous pattern i.e. loosely coupled ingestion of the replication messages incoming from other CSD Stream Server instances.

2. The latter two requirements lead to the concept of queues. The inbound replication queue (i.e. that which receives replication messages from peers) MAY be limited in size. If it exceeds capacity, a WriteFailed exception MUST be thrown. The outbound replication queue (i.e. that which sends replication messages to peers) MUST be unbounded. Both queues MUST be durable e.g. persist across restarts of the local CSD Stream Server.

A-7.6 Interface Definition

The CSD Stream Replication Service Interface MUST be compliant with the following stack:

- MAJIIC Replication Entity Payload
- WS-Transfer.

A-7.6.1 MAJIIC Replication Entity Payload

The following referenced types are defined in the namespace "http://srvc/int/nato/majiic/services/CSDStreamServerReplication/CSDStreamServer-ReplicationMessages/"

a. The payload of the replication message MUST be captured in the content of a single XML entity element of type "Entity".

b. The content of this entity MUST be plain text and MUST be a single CSD Stream Metadata entity of type "EntityContainerType" encoded using [AEDP-17 Edition A Version 1, Annex E-5 CATALOGUE ENTRIES].

c. For encoding purposes, the content MUST be wrapped in a CDATA section, or it MUST be XML escaped text.

d. CSD Stream Server implementations MUST place the Entity element in the SOAP body.

A-7.6.2 Payload Metadata

1. There are two additional metadata attributes for each payload:

a. EntityID: the identifier of the Catalog Entry of this replication message

b. SecurityType: the security information of the replication message.

2. The VersionID and ReplacedVersionID attributes used by the SPS++ are not required by the CSD Stream Server, and hence are omitted.

3. The following referenced types are defined in the namespace "http://srvc/int/nato/majiic/services/CSDStreamServerReplication/CSDStreamServer-ReplicationMessages/"

a. CSD Stream Server implementations MUST place the Catalog Entry identifier in the SOAP header of type "EntityID";

b. CSD Stream Server implementations MUST place appropriate security metadata in the SOAP header of type "SecurityType"

A-7.6.3 WS-Transfer Wrapper

1. WS-Transfer [W3C, Web Services Transfer (WS-Transfer), 2006] is a minimal (compared to some) specification with a number of requirements in scope here:

- the MEP (Message Exchange Pattern) – request-response pattern; The WS-Addressing requirements outlined below are consistent with the usage of the request-response MEP.
- the use of WS-Addressing in the PUT request message.
- the use of WS-Addressing in the PUT response message.
- the use of optional return message types in the SOAP body. These are NOT REQUIRED at this time for the CSD Stream Server replication PUT.

2. All of this is captured in the WSDL for the replication interface. Whilst WS-Transfer [W3C, Web Services Transfer (WS-Transfer), 2006] defines other operations (GET and DELETE), at this time the CSD Stream Replication Service Interface does NOT REQUIRE these operations.

A-7.6.4 Other WS-* Standards - WS-Security

1. WS-Security is used to specify the communication of Username and Passwords as WS-Security compliant SOAP Headers.

2. CSD Stream Server implementations MUST implement this security policy and MUST provide a username and password in the SOAP header in accordance with [AEDP-19 Annex A-5 SOA SERVICES PROFILE].

A-7.6.5 Exceptional Behaviour

In addition to the WS-Addressing faults utilised by the WS-Transfer [W3C, Web Services Transfer (WS-Transfer), 2006] specification, this specification reuses the “WriteFailed” Exception from the “urn:nato:majiic:common:isrcommon” namespace. If a CSD Stream Server implementation is unable to persist the incoming replication message, it MUST respond with the WriteFailedException fault.

A-7.7 Joiners

The CSD Stream Replication Service Interface defined above covers the steady-state in which a number of CSD Stream Servers are exchanging metadata such that each holds a representation of the metadata catalogue for all Streams in an enterprise coalition. In any operational deployment, it is likely that new CSD Stream Servers will join the network and wish to catch up with historic Stream metadata. As with the CSD Stream Replication Service Interface, the approach taken is similar to the SPS++ and the requirements are as follows. Note that the mechanism by which the Synchronisation File is transferred between CSD Stream Servers is not defined.

A-7.7.1 Synchronisation File Creation

1. Each CSD Stream Server MUST be able to represent its Stream metadata contents as a single Synchronisation File.

2. The Synchronisation File contents MUST represent all Stream metadata persisted by the CSD Stream Server at the point of creation.

    a. Metadata fusion MAY be used to limit the size of the Synchronisation File.

3. The Synchronisation File MUST adhere to the Sync.xsd schema defined in the namespace http://srvc/int/nato/majiic/services/ CSDStreamServerReplication/

4. Each entry in the Synchronisation File MUST contain a SecurityLabel with appropriate security markings.

5. Each entry in the Synchronisation File MUST contain an EntityID containing the Catalog Entry identifier for that Stream.

6. Each entry in the Synchronisation File MUST contain an Entity.

   a. Each Entity MUST be a single CSD Stream Metadata entity of type "EntityContainerType" encoded using [AEDP-17 Edition A Version 1, Annex E-5 CATALOGUE ENTRIES].

   b. Each entity MUST be encoded as plain text i.e. either wrapped in a CDATA section, or as XML-escaped text.

7. All Synchronisation File entries pertaining to the same Stream (i.e. the same Catalog Entry identifier) MUST be ordered in increasing temporal order.

8. The ordering of Synchronisation File entries between different Streams (i.e. different Catalog Entry identifiers) is not defined.

9. The mechanism by which the Synchronisation File is generated is not defined.

10. Creation of the Synchronisation File MUST have no impact on achieving the other functional and non-functional requirements of the CSD Stream Server.

A-7.7.2 Synchronisation File Ingestion

1. Each CSD Stream Server MUST be able to ingest a Synchronisation File as defined above.

2. Ingestion of a Synchronisation File MUST persist each entry in the Synchronisation File into the CSD Stream Server's Stream metadata store where it can be queried by the CSD Stream Query Service Interface.

3. During ingestion of the Synchronisation File, the CSD Stream Server MUST NOT generate outgoing CSD Stream Replication Service Interface messages.

4. The mechanism by which the Synchronisation File is ingested is not defined.

5. Ingestion of the Synchronisation File MUST have no impact on achieving the other functional and non-functional requirements of the CSD Stream Server.

A-7.7.3 Process

The process by which a new CSD Stream Server joins an existing network SHOULD be as follows:

   a. Start the CSD Stream Server. Disable all client connections via the CSD Stream Server interfaces.

   b. Configure outbound replication from an existing CSD Stream Server to the new CSD Stream Server.

c. Generate a Synchronisation File from an existing CSD Stream Server. Note that this process should take place as soon as possible following the previous step to minimise duplicates.

d. Transfer the Synchronisation File from the existing CSD Stream Server to the new CSD Stream Server.

e. Ingest the Synchronisation File into the new CSD Stream Server.

f. Enable incoming replication from the existing CSD Stream Server to the new CSD Stream Server.

g. Enable outgoing replication to necessary CSD Stream Server instances.

h. Enable client connections via the CSD Stream Server interfaces.

A-7.8 Routings and Replication Modes

1. This section describes how CSD stream server instances can be deployed into a network of different CSD stream server nodes.

2. The CSD stream server MUST support Bridgehead Routing on all links between nodes:

a. With Bridgehead Routing, a CSD stream server MUST route ALL changes to persisted local state of replicable streams to the WS-transfer PUT on all configured external replication partners (with OPTIONAL replication message fusion);

b. With Bridgehead Routing, a CSD Stream Server MUST replicate ALL messages received through a PUT on its local Replication Interface to all configured replication peers (with OPTIONAL replication message fusion).

3. The CSD stream server MUST support three possible replication modes:

a. Standalone Mode: a CSD stream server MUST support running in isolation with no outbound replication routing enabled;

b. Bridgehead Mode: a CSD stream server MUST support running in a full or partial mesh using Bridgehead Routing between all nodes;

c. Read Only Mode: a CSD stream server MUST support running with no outbound replication and only inbound replication configured.

4. Figure A-4 shows a sample deployment in which each CSD Server is operating in one of these three modes. (INFORMATIVE)

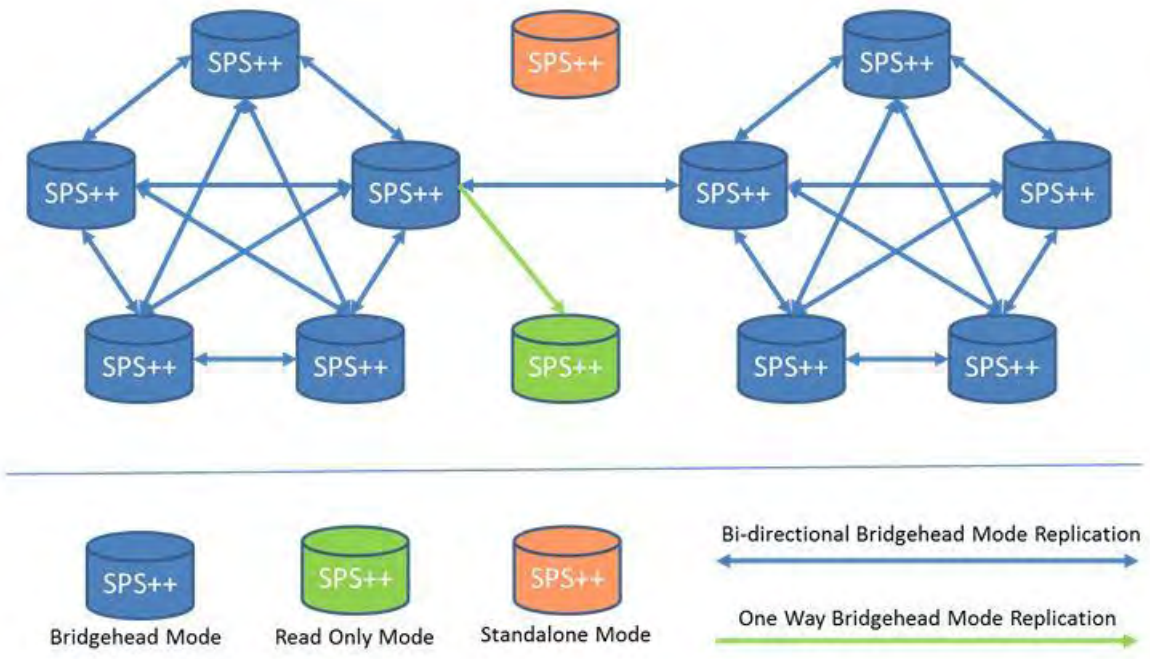

Figure A-4 Example CSD stream server replication topology

A-7.9 Example Scaling Calculation (INFORMATIVE)

The non-functional requirements relating to the volume of replication messages and the required bandwidth for replication traffic is clearly highly dependent on the particular deployment scenario. Here, one particular scenario is outlined.

A-7.9.1 Assumptions (INFORMATIVE)

1. There are five motion imagery collectors, providing metadata updates at a rate of 1 Hz.

2. There are five GMTI collectors, providing metadata updates at a rate of 0.1 Hz.

3. Each metadata update is 1.5 kBytes.

4. Therefore the resulting number of replication messages per day is:

   ((5 [motion imagery streams] * 1 [update / second])

   + (5 [GMTI streams] * 0.1 [updates / second]))

* 60 [seconds / minute] * 60 [minutes / hour] * 24 [hours / day] = **475200 [updates / day]**

5. And the required bandwidth is:

((5 [motion imagery streams] * 1 [update / second])

+ (5 [GMTI streams] * 0.1 [updates / second]))

* 1.5 [kB / update] * 8 [bits / byte] = **66 [kbits / second]**

A-8 SEQUENCE DIAGRAMS

A-8.1 Live Stream Publishing and Discovery

1. The following sequence diagram (Figure A-5) describes how sensors publish their Streams through a dedicated topic on the WS-Notification.

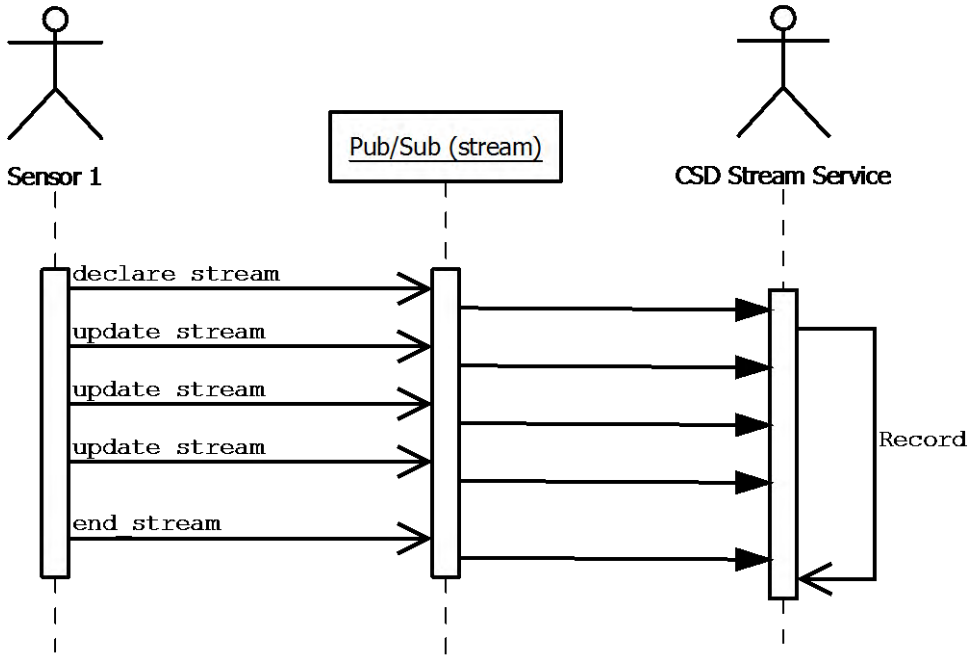

Figure A-5 Live Stream management

2. First, the sensor declares the Source Stream by sending a notification with status “New”.

3. Then the sensor regularly updates the Stream entry (e.g. changing temporal/spatial coverage) by sending notifications with status “Changed” (e.g. every minute).

4. At the end of the Stream, the sensor informs the CSD Stream Server that the Stream is terminated by sending a final notification with status “Obsolete”.

5. In response to the events sent by the sensor, the CSD Stream Server records the Live Stream and makes it discoverable to exploitation operators through the CSD Stream Query Service Interface.

6. The specifics of Stream recording are not described in this document and are left to the design of the implementation.

7. Exploitation operators search for Streams of operational interest by querying the CSD Stream Server via the CSD Stream Query Service Interface. The query returns a set of metadata results and the exploitation operators can start exploitation by requesting replay through the CSD Stream Controller Service Interface, or by downloading the stream contents using the appropriate attribute (if present).

A-8.2 Stream Replay and Exploitation

1. Figure A-6 describes how an exploitation operator searches for Recorded Streams and requests a replay. Replay is performed by the CSD Stream Server.

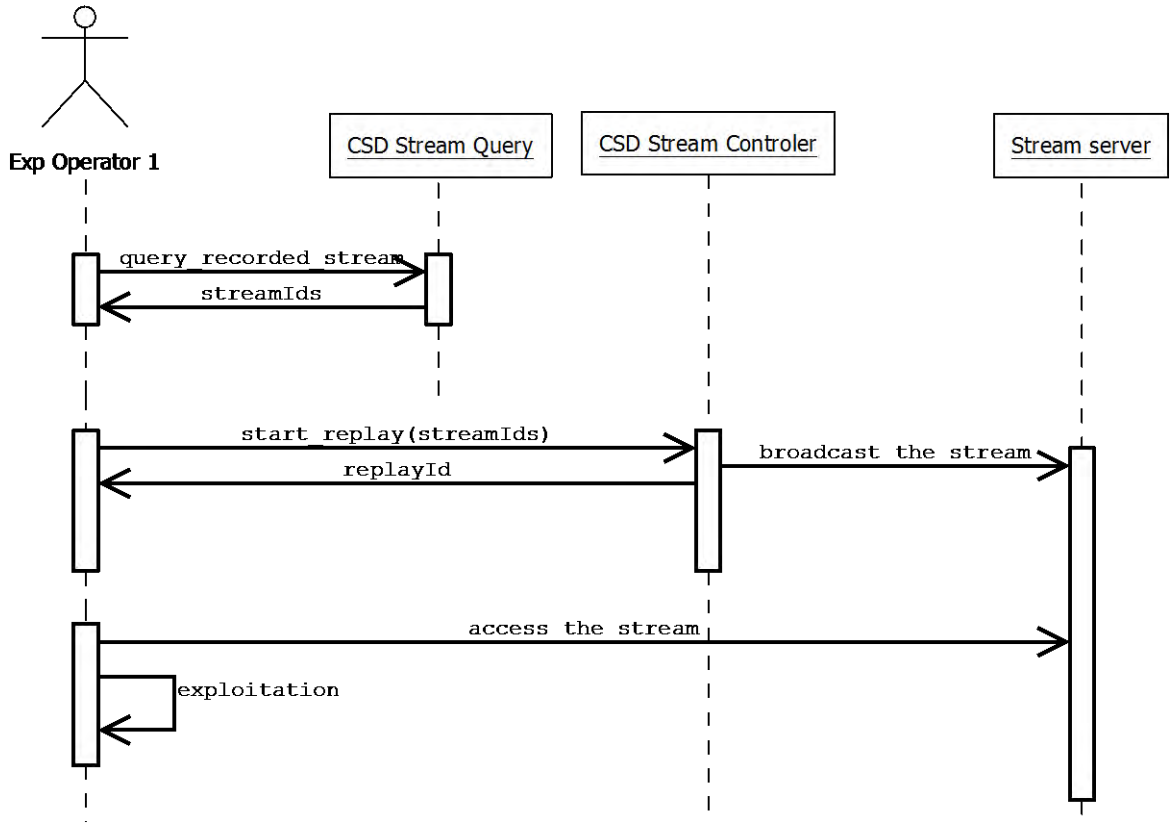

Figure A-6 Replay recorded Streams

2. The exploitation operator searches Recorded Streams through the CSD Stream Query Service Interface that returns a set of metadata entries.

3. The exploitation operator requests a replay and provides the technical parameters required for replay and a time frame. In this scenario, the replay is a broadcast. A replayId is returned. The operator can stop the replay using this ID.

4. The Stream server replays the Stream with the given technical parameters.

5. The exploitation operator receives the Stream(s) and can begin exploitation.

A-9 WIDE AREA NETWORK OPERATIONS

A-9.1 Principle

1. In an enterprise coalition network, the CSD Stream Server user experience should be identical no matter where the user is located, subject to bandwidth, latency and reliability constraints of the Wide Area Network (WAN).

2. All Exploitation Systems and Sensors MUST only connect to their own local CSD Stream Server. The local CSD Stream Server is therefore responsible for providing this consistent user experience whether Streams are local or remote.

A-9.2 Consistent Search

The CSD Stream Server provides consistent search across the WAN using CSD Stream Metadata Replication (see A-7).

A-9.3 Access to Remote Streams

An Exploitation System MAY request access to a remote stream over the WAN. Three separate use cases are supported:

a. If the WAN path from the originating remote CSD Stream Server to the client (via the local CSD Stream Server) is adequate, the Stream MAY be replayed/relayed using a chain of individual replays/relays between servers.

b. If the WAN path described above is inadequate for the original Stream data (for example, insufficient bandwidth), transcoding MAY be used on certain links in the chain.

c. If the WAN path is still not adequate for replay/relay of transcoded Stream data, the Stream data MAY be downloaded using the appropriate metadata attribute for non-real-time exploitation.

ANNEX B STREAM CONTROLLER SERVICE INTERFACE

TABLE OF CONTENT

Tables

Figures

B-1 PURPOSE

1. This Annex defines the CSD Stream Controller Service Interface which must be provided by every CSD Stream Server instance. This interface is used to replay recorded streams, or relay live streams. Optionally, certain stream types can be transcoded, for example, to change the bitrate of a motion imagery stream.

2. The CSD Stream Controller Service Interface is agnostic to the type of streaming data which it is replaying or relaying. It is the responsibility of the sensor generating the stream data to ensure it is compliant with the appropriate standards.

B-2 NAMESPACES

Table B-1 XML Namespaces used by the CSD Stream Service Controller

Namespace	Version (if applicable)
http://srvc/int/nato/majiic/services/StrmCtrlr/StreamControllerMessages/	4.4
http://srvc/int/nato/majiic/services/StrmCtrlr/StreamControllerService/	4.4
http://srvc/int/nato/majiic/services/StrmCtrlr/	4.4

B-3 SERVICE OVERVIEW

B-3.1 Operational Overview

This interface is used by Exploitation Systems and by other CSD Stream Server instances to start and stop replays of Recorded Streams, and relays of Live Streams.

a. An Exploitation System uses this interface once it has identified one or more Streams of interest using the CSD Stream Query Service Interface, see 4.3.2 and 4.3.3.

b. A CSD Stream Server uses this interface if a replay or relay has been requested by a local Exploitation System for a Stream which is stored at a remote CSD Stream Server.

Figure B-1 shows these actors.

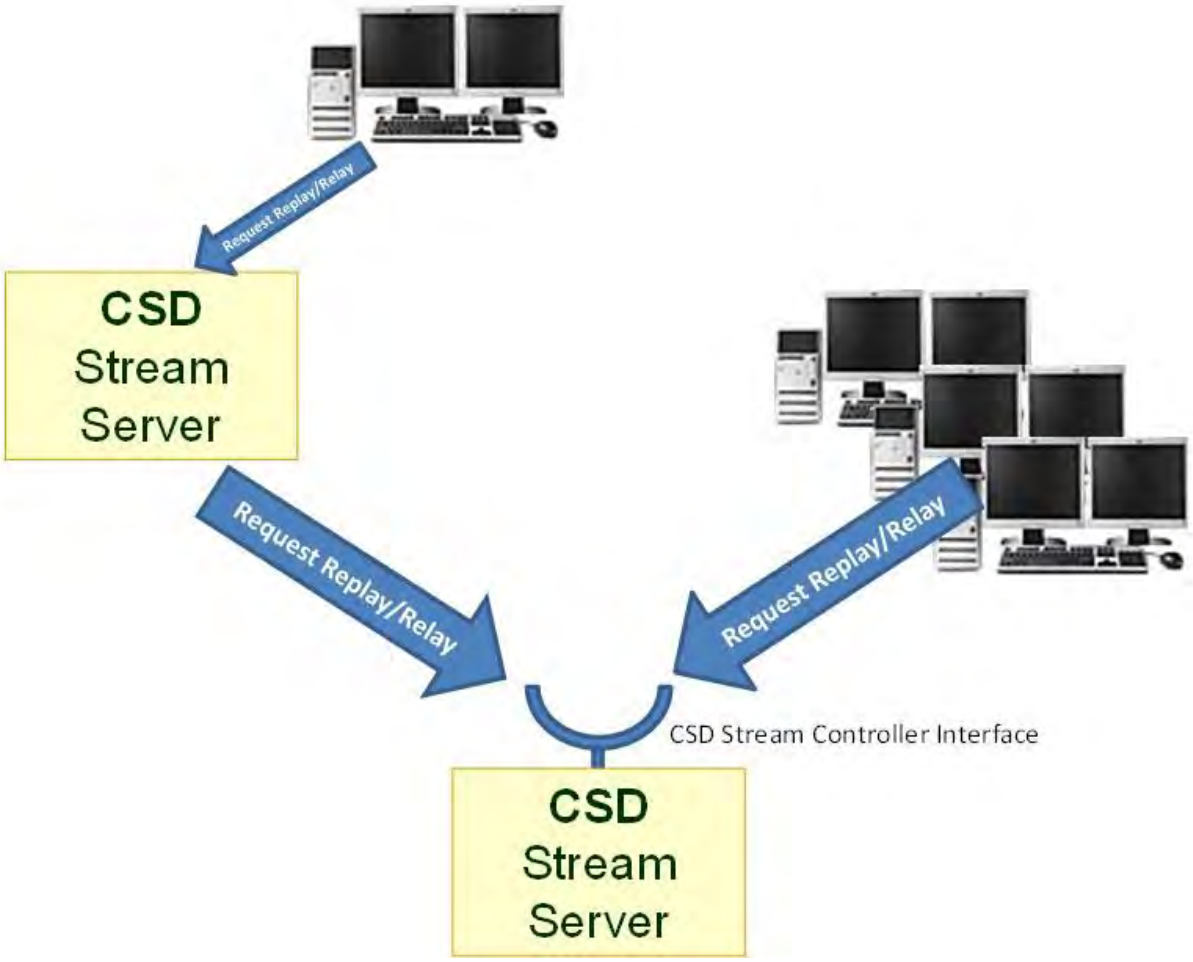

Figure B-1 Actors using the CSD Stream Controller interface

B-3.2 Service Description

This service provides the ability to start and stop a replay/relay of a set of streams from a CSD Stream Server. The interface by itself is stream agnostic. A client may request a replay/relay of multiple motion imagery, GMTI or Link 16 Streams and they will be replayed time synchronously.

B-4 SERVICE CONTRACT

B-4.1 Service Contract Overview

The CSD Stream Controller Service Interface exposes three operations:

- getCapabilities, which provides information regarding the functionality and configuration of the CSD Stream Server. This operation is defined in [AEDP-17 Annex D NSIL WEB SERVICE CORE INTERFACE]. The core interface provides the common operations for all STANAG 4559 services.

- startReplay to start a replay/relay of one or more Streams.
- stopReplay to stop an existing replay/relay.

These operations are illustrated in Figure B-2.

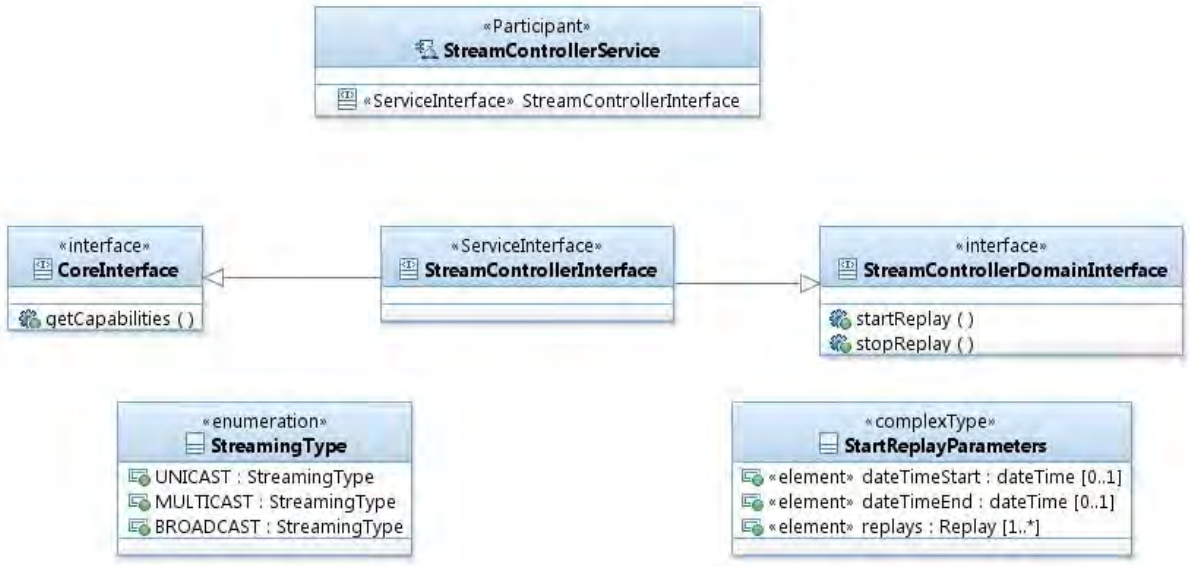

Figure B-2 Overall interface description

B-4.1.1 Service Name

The service name is StreamControllerService.

B-4.1.2 Service Namespace

The service target namespace is http://srvc/int/nato/majiic/services/StrmCtrlr/StreamControllerService/.

B-4.1.3 Service Version

The current version of the service is 4.4.

B-4.1.4 Service Interface

Figure B-3 illustrates more details of the service interface and data types used.

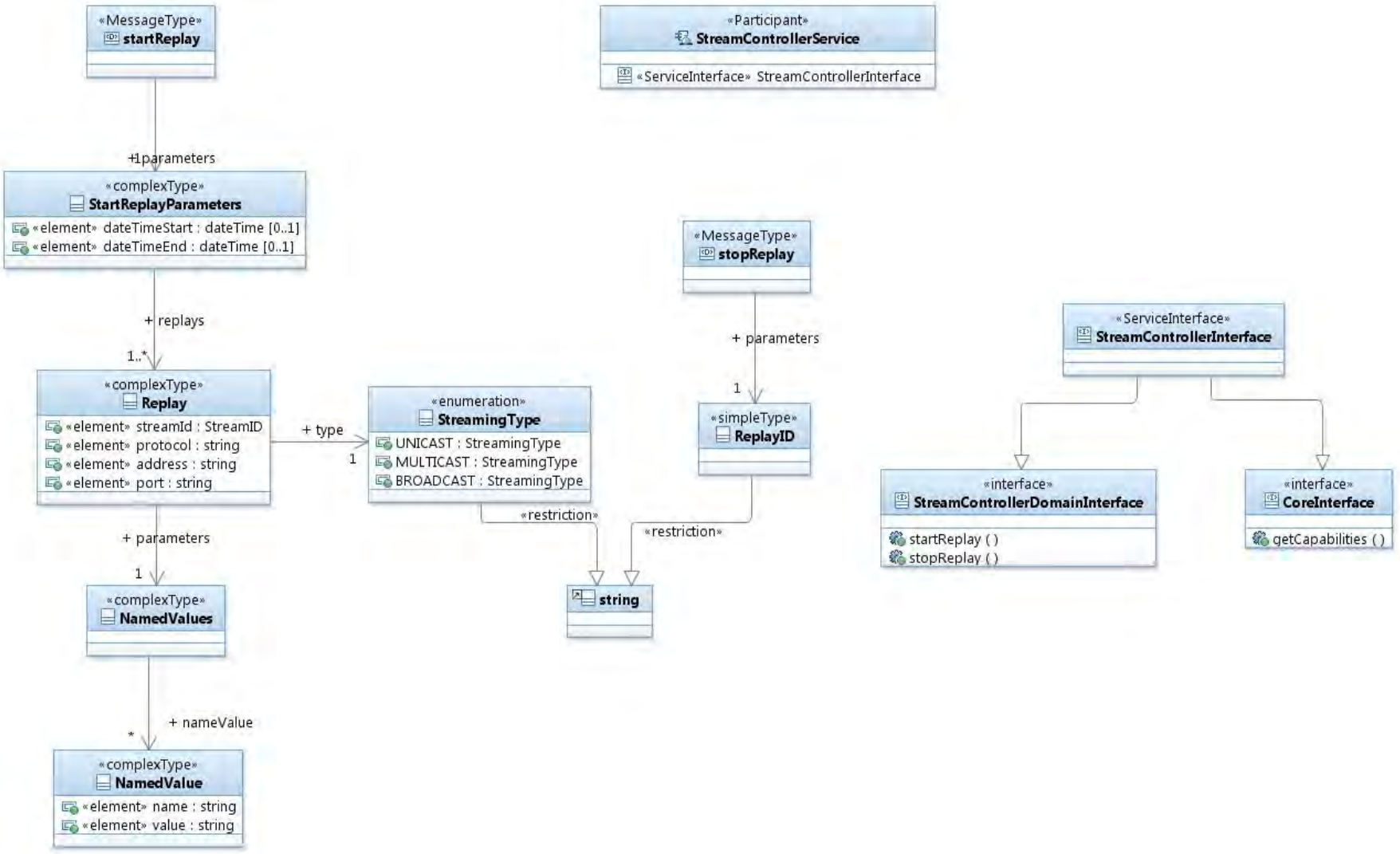

Figure B-3 CSD Stream Controller Service Interface operations and parameters

B-4.1.5 Service Operations

Table B-2 CSD Stream Controller Service Interface Operations

Interface	Operation	Mandatory/Optional	Reference
CSDStreamController	getCapabilities	Mandatory	Chapter B-4.3.1
	startReplay	Mandatory	Chapter B-4.3.2
	stopReplay	Mandatory	Chapter B-4.3.3

B-4.2 Security Considerations

All CSD Stream Controller Service operations MUST include a username and password complying with [AEDP-19 Annex A-5 SOA SERVICE Profile].

B-4.3 CSD Stream Controller Service Interface

B-4.3.1 getCapabilities

This operation is defined in [AEDP-17 Annex D NSIL WEB SERVICE CORE INTERFACE]. The core interface provides the common operations for all STANAG 4559 services. Table B-3 lists the additional capabilities that MUST (mandatory capabilities) or MAY (optional capabilities) be returned.

Table B-3 Additional capabilities which must or may be provided by a CSD Stream Server using the CSD Stream Controller Service Interface

Name	GenericValue class	Usage	Mandatory/ Optional
system.supportedReplayTypes	TextualValue	Comma separated, case-sensitive list of types of replay supported: • BROADCAST • MULTICAST • UNICAST	MANDATORY
system.supportedStreams	TextualValue	Comma separated list of supported Stream types: • GMTI • VIDEO • LINK16	MANDATORY

Name	GenericValue class	Usage	Mandatory/ Optional
notification.topic.sensor	NotificationTopicValue	Returns the notification topic for Live Streams originating at local sensors as a NotificationTopicValue	MANDATORY
notification.topic.server	NotificationTopicValue	Returns the notification topic for replayed/replayed Streams which may be visible locally as a NotificationTopicValue	MANDATORY
notification.broker.endpoint	TextualValue	The URL of the notification broker.	MANDATORY
video.codec	TextualValue	Comma separated list of supported video codecs: • MPEG-2 • H.264	OPTIONAL
video.container	TextualValue	Limited to MPEG-2 TS	OPTIONAL
video.transport	TextualValue	Limited to UDP	OPTIONAL
video.metadatacodec	TextualValue	Comma separated list of supported metadata encodings: LDS or UDS	OPTIONAL
video.bitrate	FloatingValue	If present with any value, indicates bit rate changes are possible. If not present, bit rate changes are not possible.	OPTIONAL
video.maxPacketSize	IntegerValue	If present with any value, indicates video packet size changes are possible. If not present, video packet size changes are not possible.	OPTIONAL
gmti.maxPacketSize	IntegerValue	If present with any value, indicates GMTI packet size changes are possible. If not present, GMTI packet size changes are not possible.	OPTIONAL
l16.maxPacketSize	IntegerValue	If present with any value, indicates Link 16 packet size changes are possible. If not present, Link 16 packet size changes are not possible.	OPTIONAL

B-4.3.2 startReplay

1. The purpose of this operation is to request the replay of one or more Recorded Streams, or the relay of one or more Live Streams.

2. Replay MUST be performed with the same time distribution as it was recorded i.e. multiple Streams are played in parallel, not sequentially. Figure B-4 shows the

packets of three Streams that were recorded and cover nearly the same period of time (for instance GMTI, motion imagery and Link 16).

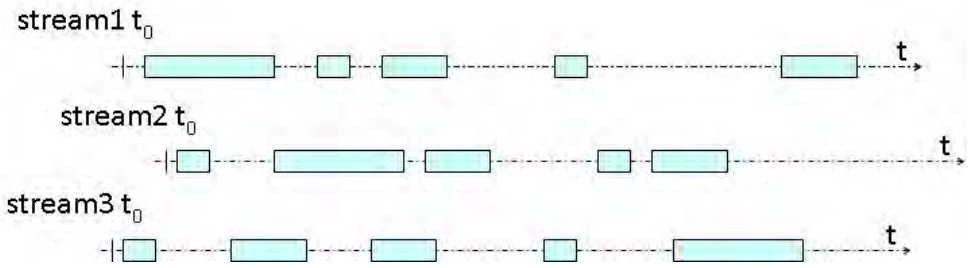

Figure B-4 Real-time recording of packets from three Streams

3. Replay can be requested for several Streams in a given period of time. In Figure B-5 two Streams are requested in the grey time frame. Packets inside this time frame are replayed in a timely manner. Note that some Stream types will require suitable packet manipulation in order to generate a valid Stream for the client.

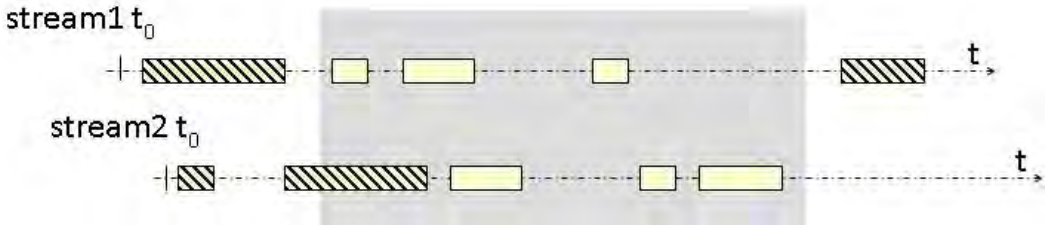

Figure B-5 Replay of two Streams in a given time interval

B-4.3.2.1 Input

The structure of the message is as follows:

B-4.3.2.1.1 Parameters

1. The parameters of this operation are shown in Figure B-6 and defined in Table B-4.

Figure B-6 Illustration of the startReplay parameters

Table B-4 startReplay parameters

Parameter name	Description
dateTimeStart	Date/time within the Stream(s) at which the replay should start.
dateTimeEnd	Date/time within the Stream(s) at which the replay should stop.
replays	The set of Streams to be replayed/relayed with the technical replay/relay parameters.

2. This section defines the following terms which are illustrated in Figure B-7:

- Start = value of the dateTimeStart parameter
- End = value of the dateTimeEnd parameter
- CoverageStart = the earliest time of the Stream
- CoverageEnd = the latest time of the Stream. For a Live Stream, this will be continually increasing.
- CurrentTime = the time of a Live Stream when the startReplay operation is invoked.
- Replay Group = one or more Streams specified for replay within the same startReplay operation.

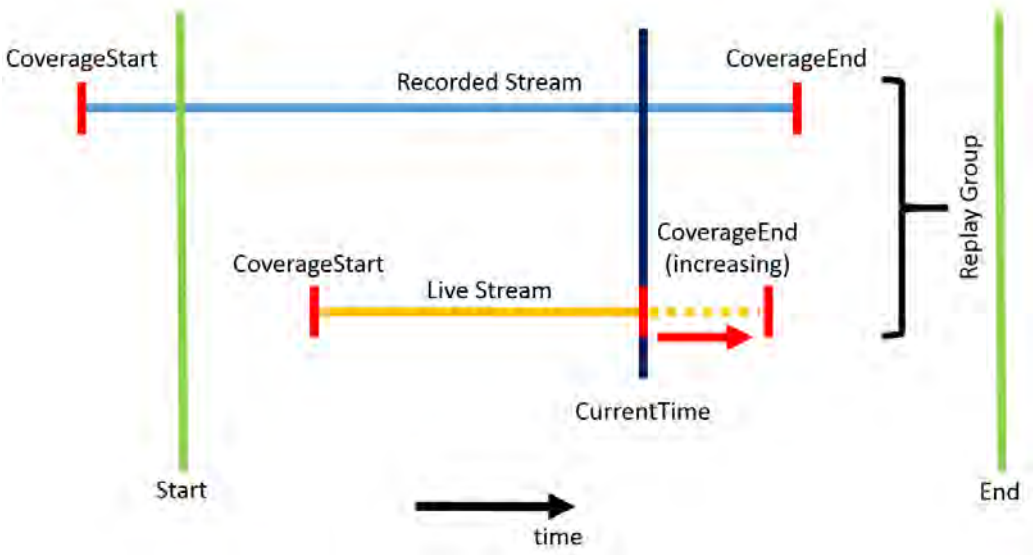

Figure B-7 Terminology definitions for a Replay Group containing one Recorded and one Live Stream

3. The following constraints apply:

   a. The Client MUST provide Start for a Recorded Stream.

   b. The Client MAY provide End for a Recorded Stream

      If not provided, the Recorded Stream MUST keep playing until CoverageEnd

      Note: data from other Streams in the same Replay Group may keep playing beyond this point

   c. The Client MAY provide Start for a Live Stream

      If not provided, the replay MUST start from the CurrentTime of the Live Stream

   d. The Client MAY provide End for a Live Stream

      If not provided, the Live Stream MUST keep playing until CoverageEnd

   e. Start MUST be before End

      Note: there are no constraints on Start & End relative to CoverageStart and CoverageEnd.

      If Start is before CoverageStart, the CSD Stream Server MUST wait for a time period of (CoverageStart – Start) before replaying the Stream from CoverageStart

      Note: this is the case for the Live Stream in Figure B-7

   f. End MAY be after CoverageEnd

      Note: this is the case for both Streams in Figure B-7

There are no other constraints on the values of Start and End, in particular with respect to the alignment with Stream metadata notifications.

4. The Replay type is illustrated in Figure B-8 and defined in Table B-5.

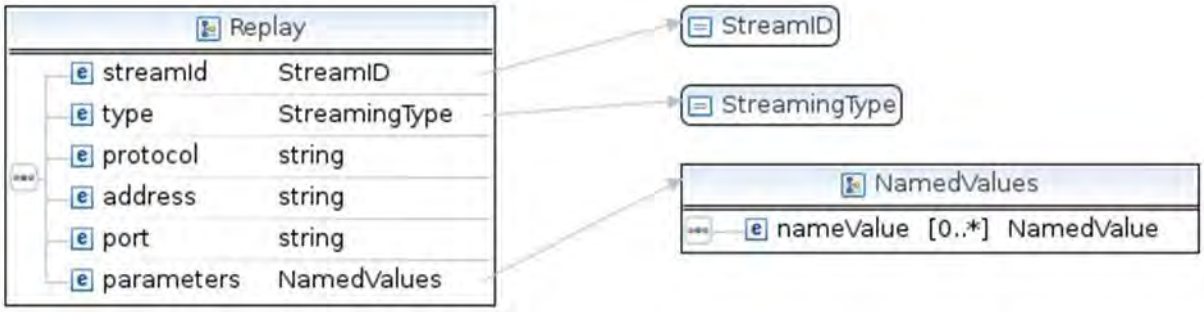

Figure B-8 Illustration of Replay Type

Table B-5 replayInfo values

Value	Description
streamID	The identifier of the Stream to replay/relay.
type	UNICAST, MULTICAST or BROADCAST
protocol	The network protocol to use. Currently only UDP is allowed.
address	The Internet Protocol (IP) address to replay/relay the Stream data.
port	The IP port number to replay/relay the Stream data.
parameters	A list of additional NamedValue pairs describing any applicable Stream transcoding.

5. The client MUST choose replay/relay settings that do not conflict with others. It is RECOMMENDED that a set of multicast and/or broadcast IP addresses and ports be reserved per system.

6. If transcoding is used, the NamedValue pairs contain attributes and values taken from getCapabilities for the appropriate Stream type. For example:

- video.codec: H.264
- video.bitrate: 1000000
- video.container: MPEG-2 TS

indicate that the motion imagery stream should be transcoded in H.264 with a bit rate of 1,000,000 bps (1 Mbps) and encapsulated in an MPEG-2 TS container.

7. Table B-6 defines the available parameters, whose availability is indicated by the getCapabilities operation (see also Chapter B-4.3.1). Note that the value attribute of a NamedValue pair has a xsd:string type, so the client MUST provide the appropriate values converted into a string.

Table B-6 Available Replay parameters

Name	Meaning
video.codec	Motion imagery video compression algorithms. Either MPEG-2 or H.264.
video.container	Only MPEG-2 TS
video.transport	Only UDP
video.metadatacodec	Either LDS (STANAG 4609 Ed 3 Local Data Set) or UDS

	(STANAG 4609 Ed 3 Universal Data Set)
video.bitrate	Average bit rate of the transcoded Stream expressed in bits per second.
video.maxPacketSize	Maximum packet size to be used for transmitting a motion imagery Stream. This value MUST be a multiple of 188.
l16.maxPacketSize	Maximum packet size to be used for transmitting a Link 16 Stream. If necessary, the CSD Stream Server MUST rewrite the Link 16 message.
gmti.maxPacketSize	Maximum packet size to be used for transmitting a GMTI Stream. If necessary, the CSD Stream Server MUST rewrite the GMTI packet.

B-4.3.2.2 Output

B-4.3.2.2.1 startReplayResponse Message Type

The contents of the startReplayResponse type are defined in Table B-7.

Table B-7 startReplayResponse Type

Element or attribute	Comments
returned	A unique identifier for the replay/relay. This value may be used to stop the replay/relay using the stopReplay operation.

B-4.3.2.2.2 Exceptions

Table B-8 lists the exceptions which may be returned by the startReplay operation, and the cases in which they may be returned.

Table B-8 Exceptions which may be returned by the startReplay operation

Exception	Description
AccessDeniedException	The provided credentials are invalid. For example, an unknown user, incorrect password or a valid username and password but the user has insufficient privilege to perform the operation.

Exception	Description
InvalidInputParameterException	The provided parameters are wrong. For example, if the streamID does not exist, or if an invalid value is provided for transcoding.
ReadFailedException	The system was not able to fulfil the request.

B-4.3.3 stopReplay

1. The purpose of this operation is to stop a replay/relay currently in progress. The first stopReplay call MUST immediately stop all ongoing and future data transmission from all Streams in that Replay Group

2. Subsequent stopReplay calls specifying the same ReplayId as the first call MUST not have any effect.

B-4.3.3.1 Input

The structure of the message is as follows:

B-4.3.3.1.1 Parameters

Table B-9 shows the input parameters to this operation.

Table B-9 stopReplay parameters

Parameter name	Description
parameters	The identifier of the replay/relay to be stopped.

B-4.3.3.2 Output

B-4.3.3.2.1 stopReplayResponse Message Type

The stopReplayResponse “returned” parameter MUST NOT be used.

B-4.3.3.2.2 Exceptions

Table B-10 lists the exceptions which may be returned by the stopReplay operation, and the cases in which they may be returned.

Table B-10 Exceptions which may be returned by the stopReplay operation

Exception	**Description**
AccessDeniedException	The provided credentials are invalid. For example, an unknown user, incorrect password or a valid username and password but the user has insufficient privilege to perform the operation.
InvalidInputParameterException	The provided parameters are wrong. For example, if the specified replay identifier does not exist.
ReadFailedException	The system was not able to fulfil the request.

AEDP-18(A)(1)