

Consultative Committee for Space Data Systems

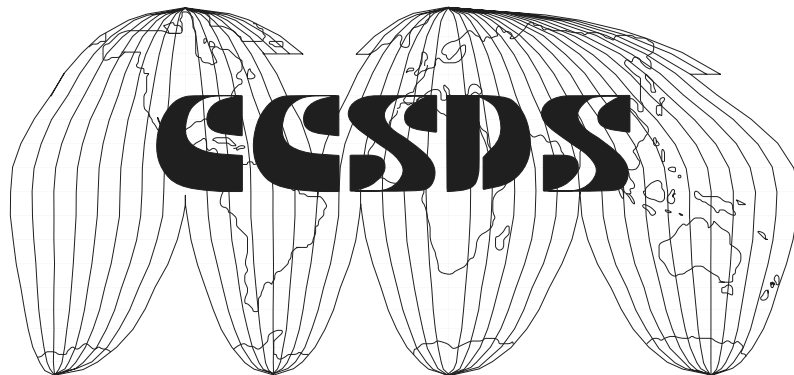
**REPORT CONCERNING SPACE
DATA SYSTEM STANDARDS**

**CROSS SUPPORT CONCEPT—
PART 1
SPACE LINK EXTENSION SERVICES**

CCSDS 910.3-G-2

GREEN BOOK

April 2002



AUTHORITY

Issue:	Green Book, Issue 2
Date:	April 2002
Location:	Oberpfaffenhofen, Germany

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS Recommendations is detailed in reference [1], and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

FOREWORD

This document is a technical Report to assist readers in understanding the Space Link Extension (SLE) service documentation. It has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The Cross Support concept described herein is the baseline concept for ground data communication within missions that are cross-supported between Agencies of the CCSDS.

This Report describes a common framework and provides a common basis for understanding the SLE services. It is intended to assist implementing organizations within each Agency to proceed coherently with the development of compatible derived Standards for the ground systems that are within their cognizance.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This Report is therefore subject to CCSDS document management and change control procedures that are defined in reference [1]. Current versions of CCSDS documents are maintained at the CCSDS Web site:

<http://www.ccsds.org/>

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d’Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Centre (CRC)/Canada.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 910.3-G-1	Report Concerning Space Data System Standards: Cross Support Concept—Part 1: Space Link Extension Services	May 1995	Original Issue (Superseded)
CCSDS 910.3-G-2	Report Concerning Space Data System Standards: Cross Support Concept—Part 1: Space Link Extension Services	April 2002	Current Issue

CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION.....	1-1
1.1 PURPOSE.....	1-1
1.2 SCOPE.....	1-1
1.3 APPLICABILITY.....	1-1
1.4 RATIONALE.....	1-2
1.5 DOCUMENT STRUCTURE.....	1-3
1.6 DEFINITIONS.....	1-3
1.7 REFERENCES.....	1-4
2 BACKGROUND TO SLE SERVICES.....	2-1
2.1 WHAT ARE SLE SERVICES?.....	2-1
2.2 WHY DO WE NEED SLE SERVICES?.....	2-2
2.3 WHAT ARE THE BENEFITS OF SLE SERVICES?.....	2-2
3 CROSS SUPPORT ENVIRONMENT.....	3-1
3.1 CONTEXT.....	3-1
3.2 SPACE DATA SYSTEMS.....	3-2
3.3 SPACE LINK EXTENSION SYSTEM ENVIRONMENT.....	3-2
3.4 SPACE LINK EXTENSION SERVICE CONCEPT.....	3-3
4 SLE DATA TRANSFER.....	4-1
4.1 OVERVIEW OF SLE DATA TRANSFER SERVICES.....	4-1
4.2 RETURN SLE SERVICES.....	4-1
4.3 FORWARD TELECOMMAND SLE SERVICES.....	4-5
4.4 FORWARD AOS SLE SERVICES.....	4-8
4.5 SLE COMPLEXES.....	4-9
4.6 SLE DATA TRANSFER SERVICE OPERATIONS.....	4-11
4.7 THE SLE APPLICATION PROGRAM INTERFACE (API).....	4-15
5 SLE SERVICE MANAGEMENT.....	5-1
5.1 OVERVIEW OF SLE SERVICE MANAGEMENT.....	5-1
5.2 SCOPE OF SLE SERVICE MANAGEMENT.....	5-2
5.3 FEATURES OF SLE SERVICE MANAGEMENT.....	5-3
5.4 SLE SERVICE MANAGEMENT ENVIRONMENT.....	5-4
5.5 TIME SPAN OF SLE AGREEMENTS AND PACKAGES.....	5-6
5.6 OBJECT-ORIENTED APPROACH TO SERVICE MANAGEMENT.....	5-13
ANNEX A ACRONYMS.....	A-1

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
ANNEX B GLOSSARY	B-1
ANNEX C CROSS SUPPORT SCENARIOS	C-1
ANNEX D SLE SERVICE MANAGEMENT: ANTENNA POINTING SCENARIOS	D-1
ANNEX E STORAGE FOR SERVICES	E-1
ANNEX F SLE SECURITY MECHANISMS	F-1

Figure

1-1 SLE Services Documentation Tree.....	1-6
2-1 Ground Stations Provide SLE Services to Users.....	2-1
3-1 Domains of Space Link and Space Link Extension Services.....	3-1
3-2 Space Data System.....	3-2
3-3 Ground Element of a Space Data System.....	3-3
3-4 Example of Space Link Extension (Distance).....	3-5
3-5 Example of Space Link Extension (Information).....	3-6
3-6 Example of Space Link Extension (Distance, Information and Time).....	3-7
4-1 Return SLE Services are Provided in Three Stages.....	4-2
4-10 Return Frame Data Structure.....	4-3
4-11 Return Packet Data Structure.....	4-3
4-12 Return SLE Functional Groups and Services.....	4-4
4-13 Three-Stage Approach to Conventional Forward SLE Services.....	4-5
4-14 Telecommand Packet Format.....	4-6
4-15 Forward Telecommand Frame Format.....	4-6
4-16 Components of the CLTU.....	4-6
4-17 Forward Telecommand SLE Functional Groups and Services.....	4-7
4-18 AOS Forward SLE Services are provided in Two Stages.....	4-8
4-19 Forward AOS SLE Functional Groups and Services.....	4-9
4-20 SLE Complexes.....	4-10
4-21 Example of Connected SLE Complexes.....	4-11
4-22 Simple State Diagram for SLE Services.....	4-13
4-23 Example of SLE Service Operation Table (RCF-START).....	4-14
4-24 Example of ASN.1 Notation for RCF-START.....	4-14
4-25 Layers of the SLE API.....	4-15
5-1 SLE Service Management in Context.....	5-1
5-2 SLE Complex Management Internal Interfaces.....	5-2
5-3 SLE Service Management Architecture.....	5-4
5-4 Example of Functional Group Configuration within a Complex.....	5-5
5-5 SLE Complex/MDOS Interfaces.....	5-6
5-6 Example of a Service Package Lifetime Phases.....	5-7

CONTENTS (continued)

<u>Figure</u>	<u>Page</u>
5-7 Example of Service Agreement Period.....	5-8
5-8 Example of Service Provision Period.....	5-9
5-9 Online and Offline Services in One SLE Service Package	5-9
5-10 Example of Two Service Packages at Two SLE Complexes	5-11
5-11 Managed Objects – Two Examples	5-13
5-12 Managed Object Class	5-14
5-13 Managed Object Containment	5-15
5-14 Reference Managed Object.....	5-17
5-15 Abstract and Concrete Managed Object Classes	5-18

1 INTRODUCTION

1.1 PURPOSE

Cross Support occurs when one organization provides part of its Space Data System resources to service the space data transfer requirements of another organization. This Report explains how CCSDS Space Link Extension (SLE) services are designed to facilitate Cross Support and to maximize the performance and cost benefits to be gained by using standard protocols.

This Report presents the concept of Cross Support using SLE services; describing the background to the services and presenting the functional components of the ground element of a Space Data System that provide the services. It describes the SLE Data Transfer and SLE Service Management interfaces where Cross Support may occur and provides example scenarios to show how the concept may be applied in practice.

1.2 SCOPE

This Report summarizes the technical content of the SLE Recommendations including the *Cross Support Reference Model* (reference [9]), the *SLE Service Management Specification* (reference [10]), and the *SLE Transfer Service Specifications* (references [12] through [14]).

In presenting the SLE Cross Support concept, the following assumptions are made:

- a) the context is that of a single space mission;
- b) within this space mission a single spacecraft is considered;
- c) this spacecraft's telecommand and telemetry is compliant with CCSDS Telecommand (TC) Recommendations (references [5] through [7]) and either Packet Telemetry (TM) or Advanced Orbiting Systems (AOS);
- d) all end-users (i.e., sources or sinks of space data on the ground) are affiliated to a single mission manager.

1.3 APPLICABILITY

The term 'Cross Support services' encompasses all data handling services that one organization can provide to support another organization's spacecraft operations. On the ground, Cross Support services are of three kinds:

- SLE services, extending CCSDS Space Link services as defined in CCSDS Space Link Recommendations (references [10] through [14]).
- Ground Communications services, providing ground communications support, e.g., to relay operational data.
- Ground Domain services, covering all services that handle data related to spacecraft operations, but that do not use the Space Link data structures defined in CCSDS

Space Link Recommendations. Examples of ground domain services are spacecraft tracking, exchanging spacecraft databases, and mission planning.

The Cross Support documentation is divided into three reports:

- Part 1 of the Cross Support Concept (this document) addresses SLE services. It is applicable to any organization that wishes to extend Space Link services across the ground using SLE services.
- Part 2 of the Cross Support Concept (to be developed in the future) will establish the Cross Support concept for the use of communication services underlying the ground element.
- Part 3 of the Cross Support Concept (to be developed in the future) will address Ground Domain services.

1.4 RATIONALE

The primary goal of CCSDS is to increase the level of interoperability between space organizations. CCSDS promotes interoperability by developing data handling techniques for spacecraft operations, space research, and space science applications.

There are substantial costs associated with implementing and operating Space Data Systems. A significant portion of these costs is associated with frequent design and development of customized systems. As the demands placed upon Space Data Systems increase, system cost-reduction measures assume greater importance. The development of standard methods of performing routine tasks will reduce the costs.

CCSDS performs end-to-end systems analyses and adopts or develops standard approaches for solving routine problems relating to the development of Space Data Systems. After these approaches are officially accepted by the CCSDS member agencies, they are issued as CCSDS Recommendations. The Recommendations are subsequently adopted by the International Organization for Standardization (ISO) for use by any space organization.

As space organizations adopt and implement CCSDS Recommendations / ISO Standards, they develop compatible cross support capabilities. It is then easier and more cost-effective for space organizations to:

- share the resources established to acquire data from remote sensors;
- transport this information to designated destinations;
- process their data for various purposes;
- store the data for future reuse.

SLE services further the goal of interoperability by establishing a standard for services to be used in the area where most Cross Support activity occurs—between the tracking stations or ground data handling systems of various organizations and the mission-specific components of a mission ground system. The SLE services are applicable to routine, contingency, and

emergency operations.

1.5 DOCUMENT STRUCTURE

This Report is organized as follows:

- a) Section 1 outlines the purpose, scope, applicability, and rationale of this Report, lists the definitions and references used throughout the Report, and presents the Cross Support documentation structure.
- b) Section 2 provides a high level overview of the SLE services and an explanation of why they are needed and what benefits they offer.
- c) Section 3 identifies the environment in which Cross Support occurs and the concept of the SLE services as extending the availability of space data in distance, information, and time.
- d) Section 4 presents the Data Transfer aspects of the SLE services, relating them to the Space Link services and introducing the concept of SLE Complexes.
- e) Section 5 presents the SLE Service Management concept and the time span of SLE Agreements and Service Packages.
- f) Annex A contains a list of terminology to be used in describing Cross Support.
- g) Annex B defines the acronyms used in the text of this Report.
- h) Annex C identifies and provides examples of Cross Support scenarios.
- i) Annex D provides SLE service management scenarios.
- j) Annex E provides storage scenarios.
- k) Annex F provides security scenarios.

1.6 DEFINITIONS

The definitions provided below are those needed for understanding this Report as a whole:

- **Return Data.** For the purposes of this Report, return data is all data that is sent from the space element to the ground element (e.g., telemetry).
- **Forward Data.** For the purposes of this Report, forward data is all data that is sent from the ground element to the space element (e.g., telecommand).

Many other terms are defined throughout this Report. In addition, several terms are used that have been defined in the *Cross Support Reference Model* (reference [9]).

1.7 REFERENCES

The following documents are referenced in the text of this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommendations.

- [1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-7.4. Yellow Book. Issue 7.4 (interim update). Washington, D.C.: CCSDS, January 2001.
- [2] *Packet Telemetry*. Recommendation for Space Data Systems Standards, CCSDS 102.0-B-5. Blue Book. Issue 5. Washington, D.C.: CCSDS, November 2000.
- [3] *Telemetry Channel Coding*. Recommendation for Space Data Systems Standards, CCSDS 101.0-B-5. Blue Book. Issue 5. Washington, D.C.: CCSDS, June 2001.
- [4] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data Systems Standards, CCSDS 701.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2001.
- [5] *Telecommand Part 1—Channel Service*. Recommendation for Space Data Systems Standards, CCSDS 201.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2000.
- [6] *Telecommand Part 2—Data Routing Service*. Recommendation for Space Data Systems Standards, CCSDS 202.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2001.
- [7] *Telecommand Part 3—Data Management Service*. Recommendation for Space Data Systems Standards, CCSDS 203.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, June 2001.
- [8] *Standard Terminology, Conventions, and Methodology (TCM) for Defining Data Services*. Report Concerning Space Data Systems Standards, CCSDS 910.2-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, November 1994.
- [9] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Recommendation for Space Data Systems Standards, CCSDS 910.4-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1996.
- [10] *Space Link Extension—Service Management Specification*. Draft Recommendation for Space Data Systems Standards, CCSDS 910.5-R-2. Red Book. Issue 2. Washington, D.C.: CCSDS, September 2001.

- [11] *Space Link Extension—Service Management—Space Link Physical Layer Managed Object Specification*. Draft Recommendation for Space Data Systems Standards, CCSDS 910.7-R-1. Red Book. Issue 1. Washington, D.C.: CCSDS, October 2001.
- [12] *Space Link Extension—Return All Frames Service Specification*. Recommendation for Space Data System Standards, CCSDS 911.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, April 2002.
- [13] *Space Link Extension—Forward CLTU Service*. Recommendation for Space Data System Standards, CCSDS 912.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, April 2002.
- [14] *Space Link Extension—Forward Space Packet Service Specification*. Draft Recommendation for Space Data System Standards, CCSDS 912.3-R-2. Red Book. Issue 2. Washington, D.C.: CCSDS, July 2001.
- [15] *Information Technology—Text Communication—Message-Oriented Text Interchange Systems (MOTIS)—Part 3: Abstract Service Definition Conventions*. International Standard, ISO/IEC 10021-3:1990 (E). 1st ed. Geneva: ISO, 1990. As updated by Technical Corrigendum 1, ISO/IEC 10021-3:1990/Cor. 1:1992 (E). Geneva: ISO, 1992.
- [16] *Information Technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)*. International Standard, ISO 8824. 2nd ed. Geneva: ISO, 1990.
- [17] *Information Technology—Open Systems Interconnection—Basic Reference Model Part 4: Management Framework*. International Standard, ISO/IEC 7498-4. Geneva: ISO, 1989.
- [18] *Information technology—Open Systems Interconnection—The Directory—Part 8: Authentication Framework*. ISO/IEC 9594-8. Geneva: ISO, 1993.
- [19] *Object-Oriented Analysis*. Peter Coad and Edward Yourdon (Contributor). New York: Yourdon Press Computing Series, 1991.

The relationship of this document to the other SLE documents is shown in figure 1-1. The suite of Cross Support documents includes:

- *Standard Terminology Conventions and Methodology (TCM)* (reference [8]) contains the methodologies agreed upon for representing abstract services. It also contains a set of agreed-upon references.
- *Space Link Extension—Cross Support Reference Model*: a Recommendation that introduces the *Cross Support Reference Model* (reference [9]);
- *Space Link Extension—Service Management Specifications*: a suite of Recommendations that define the management interface required for the provision of one or more SLE services (references [10] and [11]).
- *SLE Transfer Service Specifications*: Recommendations that define the data transfer services (references [12] through [14]).

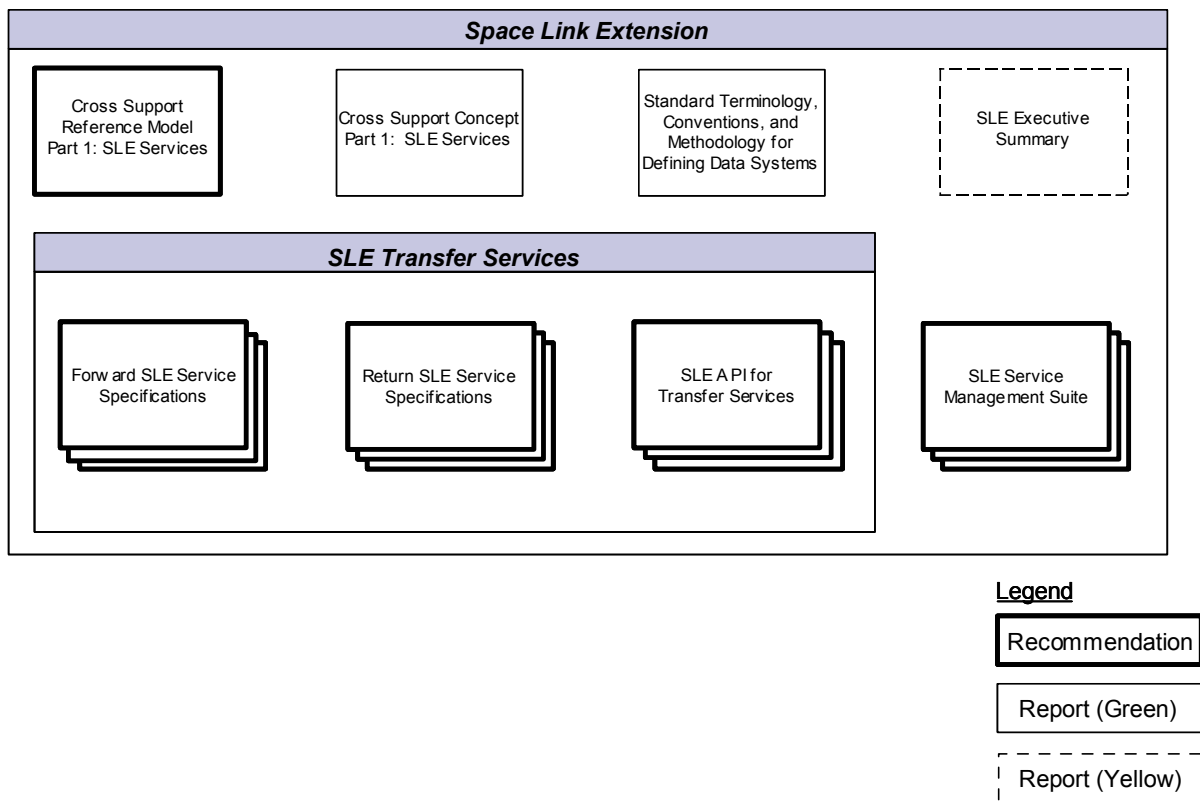


Figure 1-1: SLE Services Documentation Tree

2 BACKGROUND TO SLE SERVICES

2.1 WHAT ARE SLE SERVICES?

The SLE Services extend the return Telemetry (TM), forward Telecommand (TC), and forward Advanced Orbiting Systems (AOS) services defined by CCSDS that are used by many spacecraft operators on the space link between ground stations and spacecraft (see figure 2-1).

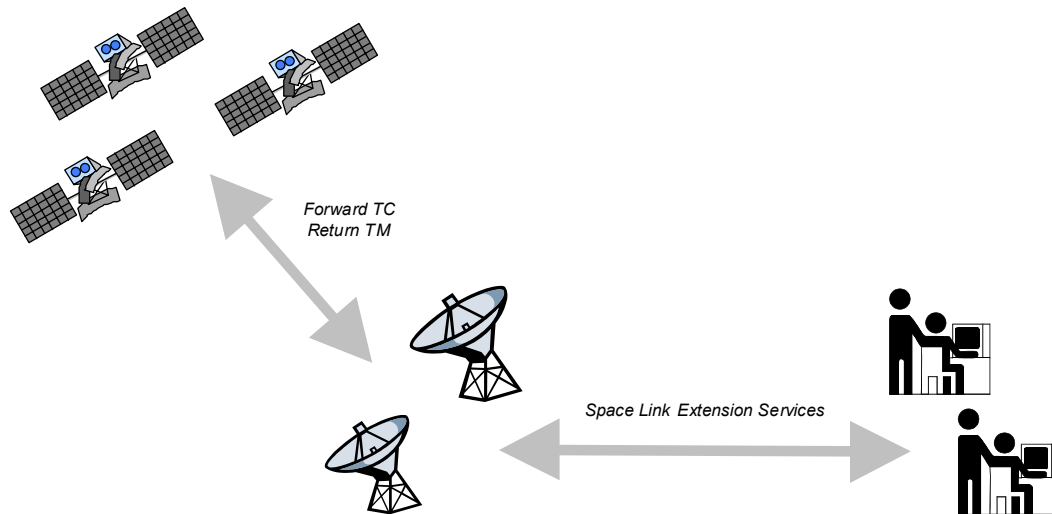


Figure 2-1: Ground Stations Provide SLE Services to Users

The SLE Services include two major elements:

- Data transfer services that move space link data units between ground stations, control centers, and end-user facilities.
- Management services that control the scheduling and provisioning of the data transfer services.

The SLE services operate in two phases:

- The definition phase, when most of the management activities take place.
- The utilization phase, when the data transfer takes place. This can be either in real-time or off-line with respect to the contact time with the spacecraft.

The information carried by the SLE services can be anything from spacecraft commands in the forward direction to science data in the return direction. In addition, the service will convey information such as TM data reception times and ground station configuration information.

2.2 WHY DO WE NEED SLE SERVICES?

The need for SLE Services arises from the desire of spacecraft operations organizations to standardize the interfaces for the transport and management of space data on the ground so that the technical, management and operational costs of providing Cross Support between the organizations can be greatly reduced.

The SLE Services that have been defined by CCSDS at the time of writing cover on-line ‘conventional’ TM and TC services. These are the services that are used by the majority of missions.

Other SLE services may be defined in the future, as the need for them arises.

2.3 WHAT ARE THE BENEFITS OF SLE SERVICES?

SLE Services enable the ground segment assets of space agencies, ground station operators, and space data users to interoperate without the need for ad hoc and complicated gateways specifically designed for each new mission.

By standardizing on the SLE Services, different organizations will be able to link discrete elements of their ground segments to suit a given mission’s needs without having to re-create the interfaces for each new mission.

Since the SLE protocols run over existing communications infrastructure, they help integrate Space Data Systems into the global communications network.

The advantages of SLE services are that:

- space organizations will be able to provide Cross Support to one another more efficiently;
- ground station owners will be able to provide standard services to operators of CCSDS-compliant spacecraft;
- users of spacecraft data will be able to command their payloads and access their data through a familiar interface, using widely available underlying telecommunications technology such as the internet or ISDN lines;
- the standardization of ground station, control center and end user interfaces will permit re-use of systems for successive missions and eliminate the costs and risks associated with mission-specific implementations;
- a truly global market for standard Telemetry, Tracking and Command (TT&C) Commercial Off The Shelf (COTS) products will be created, driving down the cost of these systems;
- SLE services are scalable so that only the actual services required by a service user or a service provider need to be implemented.

3 CROSS SUPPORT ENVIRONMENT

3.1 CONTEXT

A space mission is an undertaking to explore fields of interest by using one or more spacecraft. Each space mission involves significant functionality, both in space and on the ground. This functionality may be implemented by a single space organization, or it may be divided among multiple space organizations. Cross Support is when multiple space organizations provide space mission functionality.

The data services provided between the spacecraft and the ground are services built upon the CCSDS Space Link Recommendations (references [10] through [14]) for conventional telecommand systems, packet telemetry systems, and advanced orbiting systems.

The CCSDS Space Link Recommendations define services that transfer data across the Space Link. However, in order to use the Space Link services to support mission requirements, Space Data Systems generally require additional features on the ground to extend the Space Link services beyond the ground termination of the Space Link. SLE services provide these additional features by extending the Space Link from systems on board the spacecraft, attached to onboard local area networks, to systems on the ground, attached to terrestrial wide area networks. This extension is illustrated in figure 3-1. SLE services also provide the ability to hold data at one or more intermediate points.

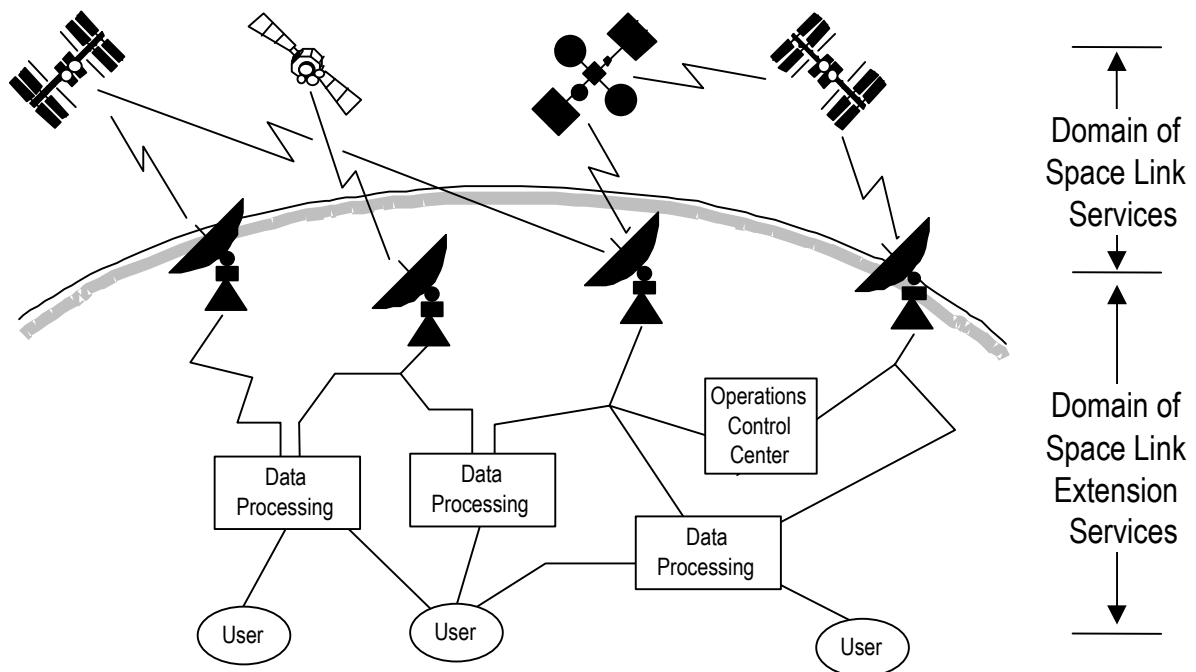


Figure 3-1: Domains of Space Link and Space Link Extension Services

3.2 SPACE DATA SYSTEMS

A Space Data System has two primary elements: a space element and a ground element. The space element and ground element are linked by Space Link communications services. The Space Link services are extended on the ground using the Space Link Extension services. The SLE services extend the Space Link services as follows:

- over distance;
- in time; and/or
- by adding information (i.e., annotation).

The Cross Support environment includes some services normally provided in ground stations, operation control centers, and data processing facilities, as illustrated in figure 3-1.

Figure 3-2 illustrates the two main components of a Space Data System: the space element and the ground element. The space element consists of the payloads, the astronauts/cosmonauts (including electronic systems that they use), and the spacecraft operations subsystems that control and monitor the spacecraft. The ground element consists of the ground data systems. The space element and ground element exchange return and forward data using a Space Link.

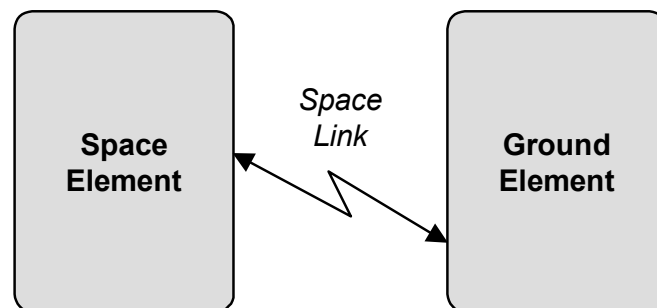


Figure 3-2: Space Data System

The space element acts as a source of return data and as a sink for forward data. In this Report, the space element represents a single spacecraft and comprises the platform and all payloads/instruments of this spacecraft. The space vehicles in a multi-spacecraft mission are treated as individual space elements. Platform and payloads/instruments are considered here only for their capability of generating or receiving data. Any processing that they perform on this data is outside the scope of the SLE services.

3.3 SPACE LINK EXTENSION SYSTEM ENVIRONMENT

The ground element of a Space Data System includes an SLE System and a Mission Data Operation System (MDOS). It may also contain other components, but these are outside the scope of this Report. Figure 3-3 illustrates these components.

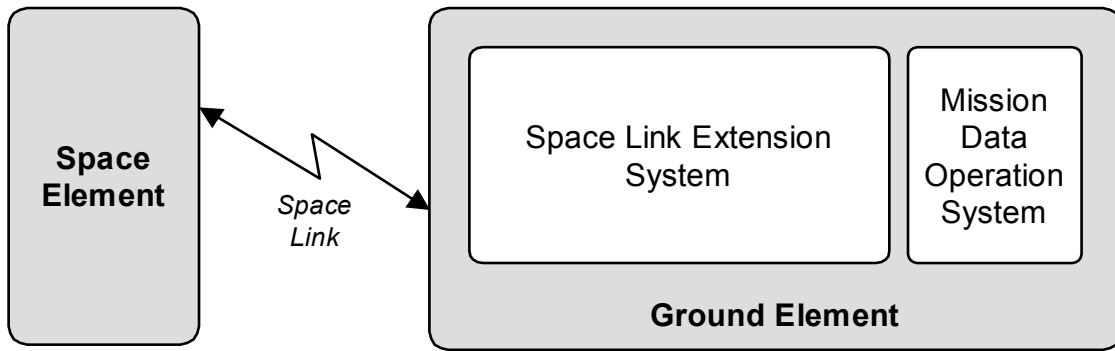


Figure 3-3: Ground Element of a Space Data System

The SLE System extends the transfer and delivery of forward and return data between a Space Link ground termination point and the MDOS. In the context of a mission, the SLE System is managed by the MDOS.

The complete set of SLE services is identified later in this Report. A particular mission may use all or a subset of the SLE services. In providing SLE services, the SLE System performs

- RF modulation/demodulation at the ground termination of the Space Link;
- ground termination of the Space Link protocols (that is, the protocols for the CCSDS Space Link services) used by the mission;
- adding of value-added information (called annotation) to Space Link service data;
- terrestrial networking among the ground elements that host the ground applications;
- interface to ground-side Space Link protocol processing and ground-side RF modulation/demodulation functions.

The MDOS is dedicated to a particular space mission. It acts as a source of forward data and as a sink of return data. In practice, the MDOS may not be the ultimate source of forward data or sink of return data on the ground element, but from the perspective of this Report it acts as if it were these sources and sinks.

3.4 SPACE LINK EXTENSION SERVICE CONCEPT

3.4.1 OVERVIEW

The Space Link protocols are unique to the space mission environment and provide the customized services and data communications protocols necessary to perform space/ground and space/space communications.

A key feature of the Space Link protocols is the capability to divide each Space Link physical channel into several separate logical data channels, known as Space Data Channels. Each Space Data Channel can carry data units, such as frames or packets. Different

combinations of data units can be nested in the Space Data Channels but not all combinations are valid - the Space Link Recommendations define which combinations can be used. Different missions may select different combinations.

The SLE services provide the capability to transfer the data units between the Space Data Channels and the users on the ground, by providing users with access to the ground termination of the Space Link. This ‘extension’ of Space Link services has three aspects:

- **Extension over Distance.** Space Link protocol processing may be performed in multiple locations that are geographically separated from the place where the RF signal is received.
- **Extension in Information.** Information, indicating the conditions at the time of receipt, is added to the Space Link data. This is referred to as ‘annotation’.
- **Extension in Time.** The SLE System also extends Space Link services in time, by allowing Space Link data to be transferred through the ground element and/or the Space Link at different times.

The following subsections elaborate these themes.

3.4.2 EXTENSION OVER DISTANCE

The Cross Support environment allows SLE services to be implemented by multiple service provider facilities, each performing a portion of the necessary processing. Each SLE service provider facility interfaces either to the MDOS, the Space Link or to another service provider facility. In the latter case, the facilities may be separated geographically.

In order to reduce processing and communications overhead, Space Link protocols frequently are not layered as rigorously as ground protocols. For example, a single data field may be reused for several purposes, or the value of a data field located in one layer may be needed for processing at higher layers. Such information might be lost if processing is performed at different locations. For example, the CCSDS Application Identifier (APID), located in the CCSDS Packet, is unique within a spacecraft but is not necessarily unique across missions. The Spacecraft Identification (SCID) field, located in the frame or Virtual Channel Data Unit (VCDU), serves to qualify the APID. This qualifying information would be lost if VCDUs were processed at a different location from the location of packet processing, unless provisions were made to send the SCID information. Therefore, each facility must send sufficient information to the next facility to allow the processing to be completed.

In addition to information associated with Space Link data, information is also needed that is not associated with an individual data unit. An example is ‘loss of frame synchronization.’ Such information about each SLE service must be defined as part of its service specification.

Figure 3-4 shows a situation that might arise when all SLE processing is not done at the ground station where the Space Link is terminated. In this case, the ground station provides

the stream of frames extracted from the Space Link signal (shown as ‘all frames’ in the figure), along with associated annotation, to a geographically separate data processing facility that produces Space Packets and sends them directly to the user facility.

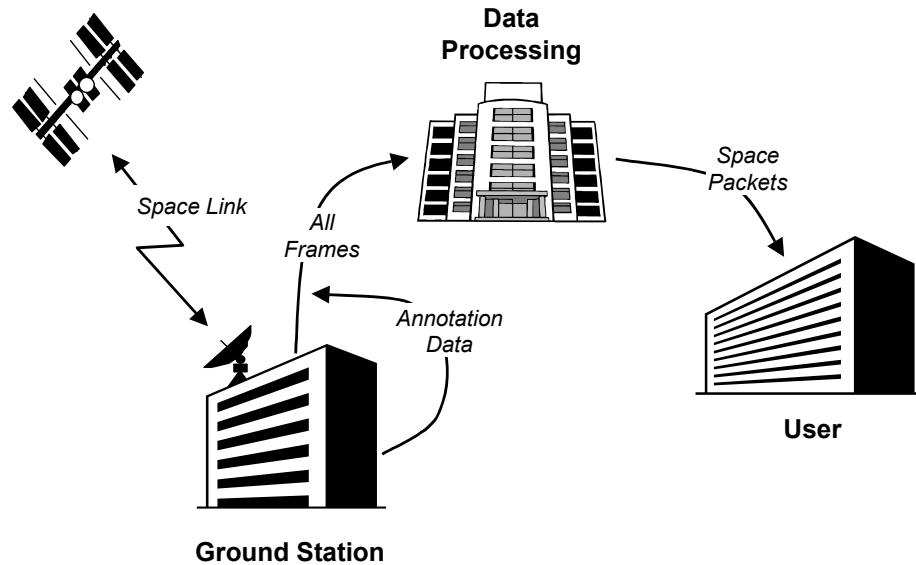


Figure 3-4: Example of Space Link Extension (Distance)

3.4.3 EXTENSION IN INFORMATION

Some information is not known prior to receipt of the Space Link data and must be added by the service facility receiving the Space Link data unit. For example, information that might be added to return link data by an SLE service may include:

- **time** when the Space Link data was received at the RF service facility;
- **identification** of the ground location where the data was received;
- **quality** of the data unit, indicating whether the data unit is correct and complete according to the criteria available;
- **sequence quality**, indicating whether the data unit is a direct successor to the previous instance of the data unit (i.e., an indication of missing data).

CCSDS Space Link protocols use out-of-band signaling for delivery of management information in order to save communications bandwidth and to minimize processing. Examples of additional information that may be added by the SLE service to supplement Space Link data are:

- **data type**, indicating whether the data was stored temporarily on board the spacecraft;

- **operations mode**, indicating the source of the data (e.g., spacecraft, simulator, test, etc.) and delivery option (online, offline/rate buffered, etc.);
- **signaling of data format parameters**, such as frame size or presence or absence of an insert.

Extension in information is illustrated in figure 3-5. This figure shows a simple case in which the Space Link is terminated at a ground station and annotation data is attached to the data stream. In the example, all SLE processing has been done at the ground station and fully processed space packets and annotation data are delivered directly to the user.

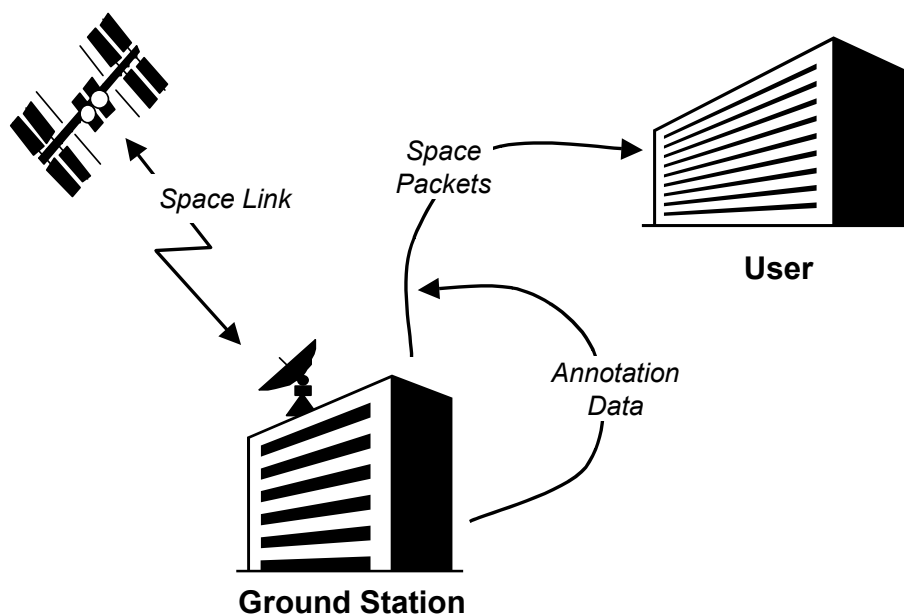


Figure 3-5: Example of Space Link Extension (Information)

3.4.4 EXTENSION IN TIME

The SLE System also extends Space Link services in time; making Space Link data available at a time after the data units are first received on the ground by adding information to ensure that the data is useful at this later time and by temporarily storing the data until it can be transferred to the user. The difference in data delivery time can vary widely. Delivery may be delayed very briefly, such as by a reduced communication rate on the ground, or it may be delayed greatly, such as by temporary storage and later forwarding.

3.4.5 EXTENSION IN DISTANCE, INFORMATION AND TIME

Figure 3-6 shows all three aspects of Space Link service extension. The signal is terminated at the ground station, where annotation data is attached to the data stream. This ground station provides a stream of frames extracted from the Space Link signal (shown as 'all frames' in the figure) to a remote data processing facility. This facility performs the next

layer of SLE processing and provides a stream of Virtual Channel (VC) Frames to yet another data processing facility. The path is shown as a broken line to indicate great geographic distance, perhaps to a different continent. At the second facility the data might be stored temporarily. At a later date, the SLE processing is completed and space packets are produced and sent directly to the user facility.

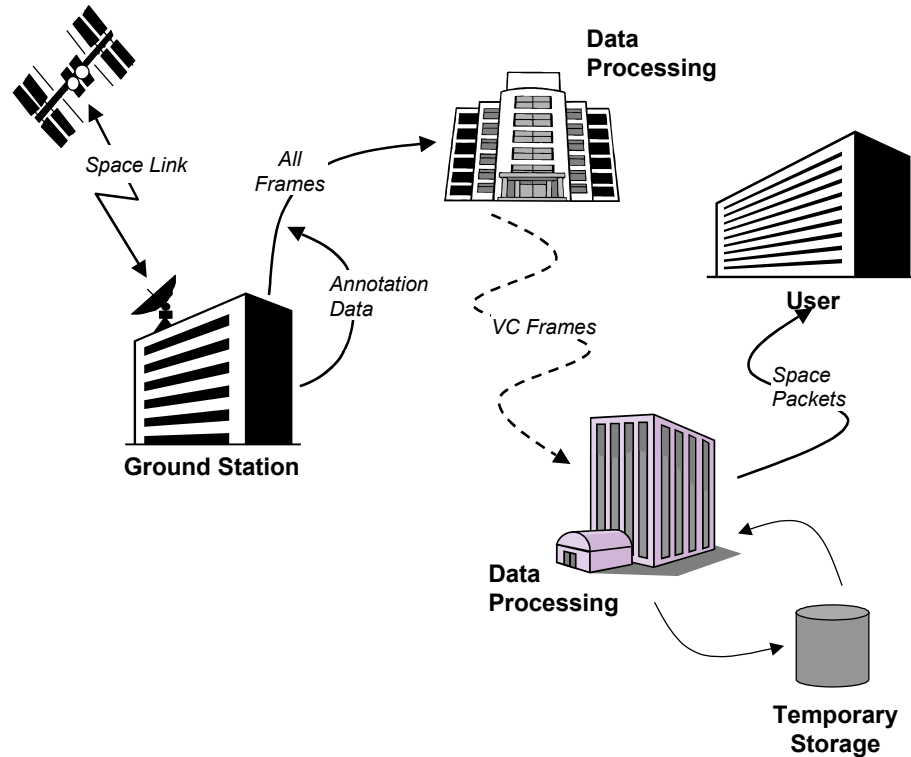


Figure 3-6: Example of Space Link Extension (Distance, Information and Time)

4 SLE DATA TRANSFER

4.1 OVERVIEW OF SLE DATA TRANSFER SERVICES

SLE data transfer services can be considered from both a functional and a physical viewpoint.

From the functional point of view, the processing required to provide SLE services is accomplished by a set of fundamental building blocks called Functional Groups. Functional Groups are derived from the layered functionality of frames, channels, and packets specified in the Space Link protocols (references [2] through [7]). As it is a fundamental building block, a Functional Group cannot be decomposed further and a single Functional Group represents the minimum functionality that can be implemented in a real system. The *Cross Support Reference Model* (reference [9]) provides more detail concerning the functionality of Functional Groups.

From the physical point of view, these Functional Groups can be implemented in various facilities, such as those described in section 3.4. The physical facilities that implement a set of Functional Groups are called SLE Complexes.

SLE data transfer services are separated into three categories:

- return SLE services;
- forward telecommand SLE services;
- forward AOS SLE services.

The following sections describe each of these sets of SLE services and the SLE Functional Groups that are needed to provide the services.

The final sections describe the SLE Complexes that implement the services, SLE data transfer service operations, and the SLE Application Program Interface.

4.2 RETURN SLE SERVICES

4.2.1 LIST OF RETURN SLE SERVICES AND THEIR INTERFACES

The return SLE services associated with TM include:

- **Return All Frames (R-AF).** Provides a complete set of TM frames from a single space link symbol stream to spacecraft operators and other users who might need all the frames.
- **Return Channel Frames (R-CF).** Provides Master Channel (MC) or specific VCs extracted from a particular R-AF channel, as specified by each R-CF service user.

- **Return Frame Secondary Header (R-FSH).** Provides MC or VC Frame Secondary Headers (FSHs) extracted from an R-AF channel, as specified by each R-FSH service user.
- **Return Operational Control Field (R-OCF).** Provides MC or VC Operational Control Fields (OCFs) extracted from an R-AF channel, as specified by each R-OCF service user.
- **Return Space Packet (R-SP).** Enables single users to receive packets with selected Application Process Identifiers (APIDs) from one spacecraft VC.
- **Return Insert (R-Insert) and Return Bitstream (R-Bitstream).** Are AOS services, not yet defined by CCSDS.

Figure 4-1 shows the data transfer interfaces for these services.

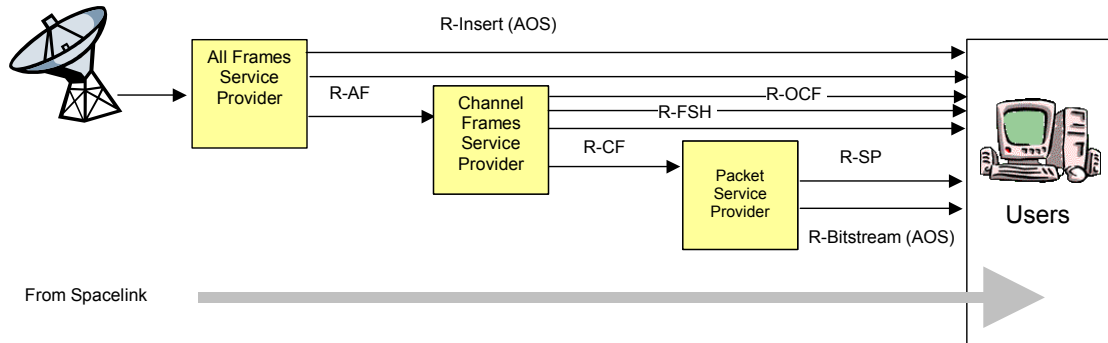


Figure 4-1: Return SLE Services are Provided in Three Stages

Figures 4-2 and 4-3 show the Space Link data structures of the return frame and the return packet, which are transported by the return SLE services. The R-AF service transfers all the frames shown in figure 4-2, whilst the R-CF service transfers a selection of the frames, chosen on the basis of the Virtual Channel ID contained in the Transfer Frame Primary Header. The R-FSH and R-OCF services provide the FSH and OCF extracted from the frame. The R-SP service transfers the packets shown in figure 4-3, selected on the basis of the Application Process ID.

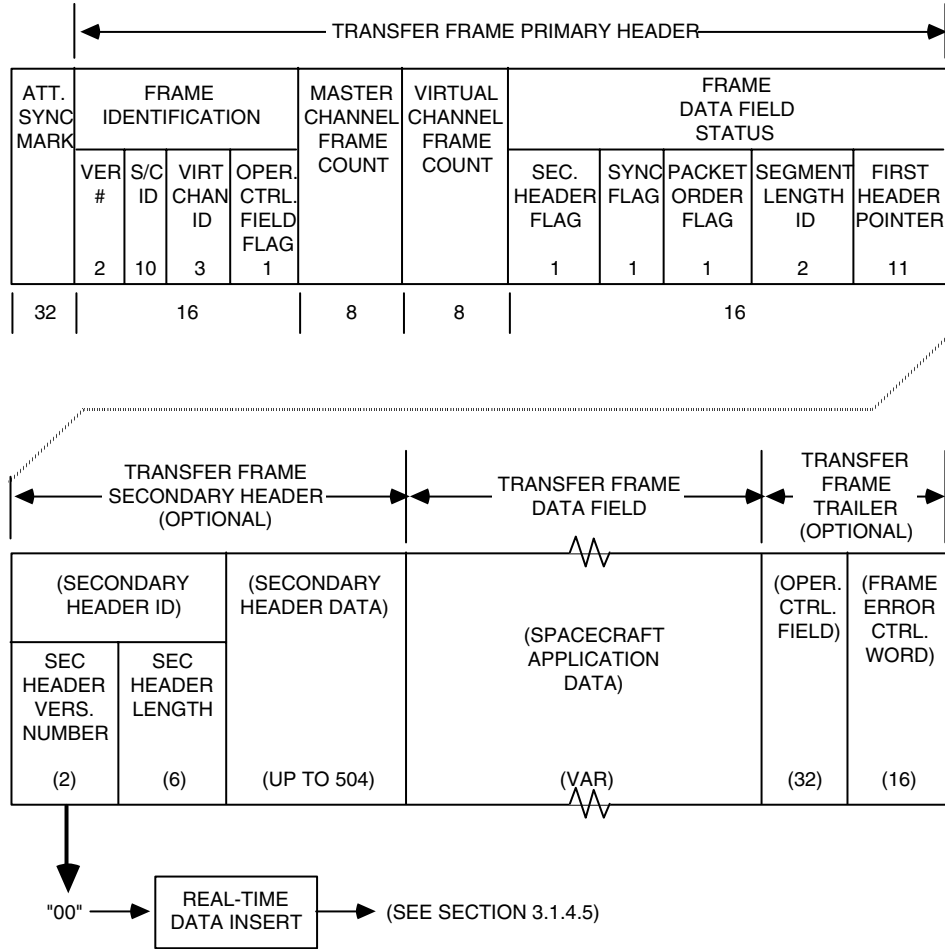


Figure 4-2: Return Frame Data Structure

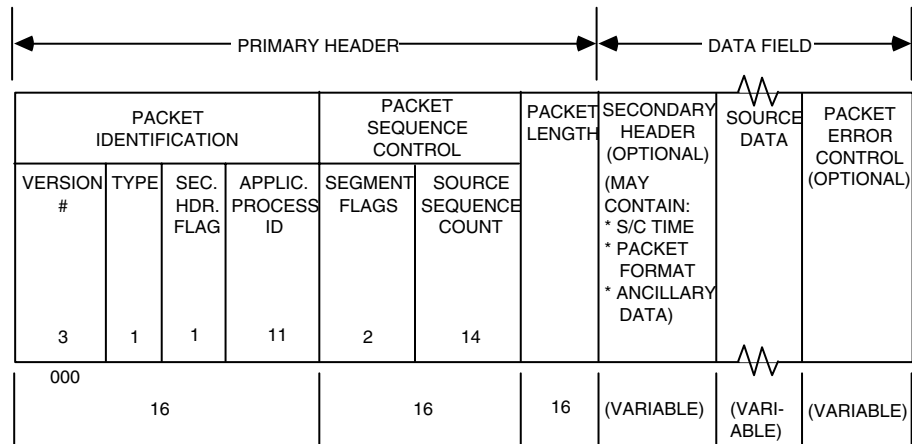


Figure 4-3: Return Packet Data Structure

4.2.2 MAPPING OF RETURN SLE SERVICES TO FUNCTIONAL GROUPS

Return SLE services are implemented as a modular architecture consisting of three ‘SLE Functional Groups’ (SLE-FGs) corresponding to the ‘Service Provider’ boxes shown in figure 4-1. Each SLE-FG presents an interface to a user of a particular SLE service and one or more SLE-FGs need to be implemented to provide that service. For example, organizations wishing to act only as an ‘All Frames Service Provider’ can do so by implementing just a ‘Return Space Link Processing’ SLE-FG. However, to provide a return packet service requires all three SLE-FGs, which may all be implemented by one organization or may be distributed between different organizations.

To be a ...	The following SLE-FGs need to be implemented ...
All Frames Service Provider	Return Space Link Processing SLE-FG
Channel Frames Service Provider	Return Frame Processing SLE-FG Return Space Link Processing SLE-FG
Packet Service Provider	Return Frame Data Processing SLE-FG Return Frame Processing SLE-FG Return Space Link Processing SLE-FG

Figure 4-12 shows the Functional Groups involved in performing return SLE services and identifies the services that are provided by each. All of the services shown are available for Cross Support, including those between the Functional Groups.

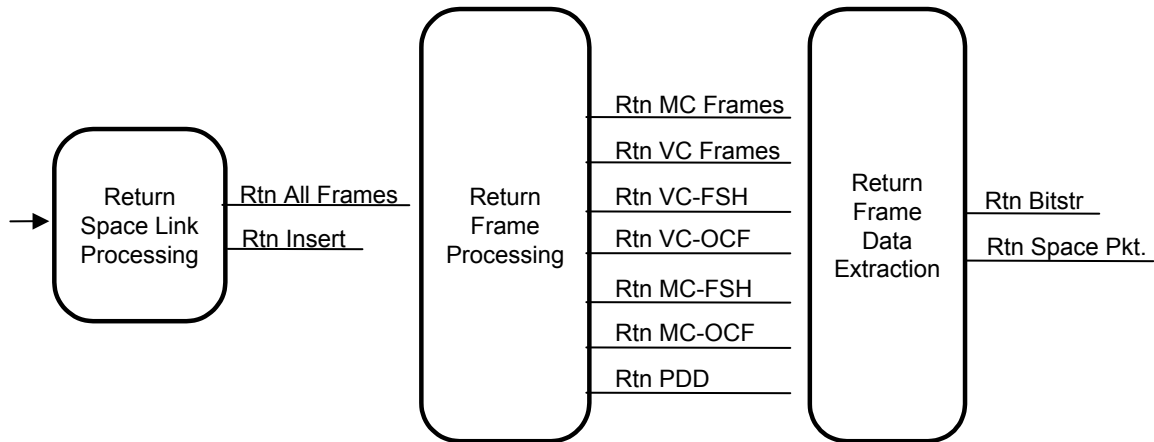


Figure 4-12: Return SLE Functional Groups and Services

Some SLE services are related in ways that make it impossible to distribute them across different Functional Groups. For example, processing for the Return Insert service must be merged in the same Functional Group with processing for the Return All Frames service

since the insert data is included in each and every frame.

4.3 FORWARD TELECOMMAND SLE SERVICES

4.3.1 LIST OF FORWARD TELECOMMAND SLE SERVICES AND THEIR INTERFACES

The forward SLE services associated with conventional TC include:

- **Forward Space Packets (F-SP).** Enables single users to provide packets for uplink to a spacecraft without needing to co-ordinate with other users of the spacecraft.
- **Forward Telecommand Virtual Channel Access (F-TCVCA).** Enables users to provide complete VCs for uplink.
- **Forward Telecommand Frames (F-TCF).** Enables users to supply TC frames to be transformed to Command Link Transmission Units (CLTUs) ready for uplink.
- **Forward Command Link Transmission Unit (F-CLTU).** Enables users to provide CLTUs for uplink to spacecraft.

Figure 4-13 shows the data transfer interfaces for these services.

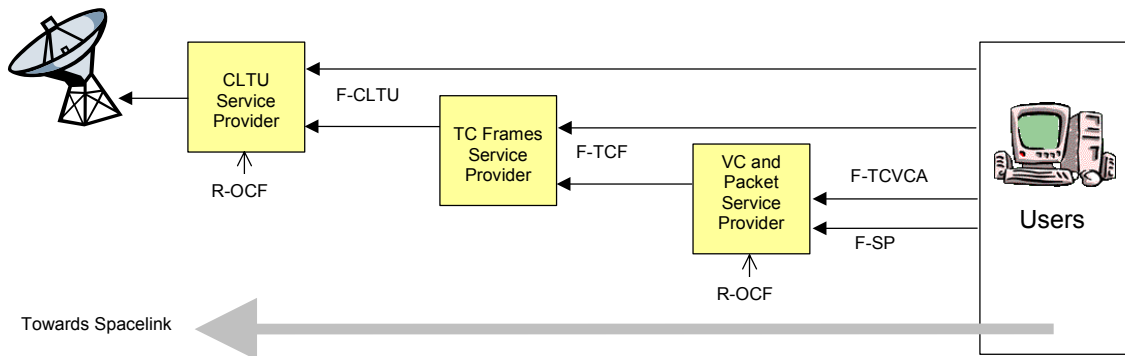


Figure 4-13: Three-Stage Approach to Conventional Forward SLE Services

Figure 4-5 shows the R-OCF service as an input to the CLTU Service Provider and the VC and Packet Service Provider. The R-OCF service provides the Command Link Control Word (CLCW), which is required by the CLTU Service Provider to determine the availability of the physical space link channel and by the VC and Packet Service Provider to determine if TC frames need to be retransmitted when the Command Operation Procedure 1 (COP-1) is in effect.

Figures 4-14 through 4-16 show the Space Link data structures of the telecommand packet, telecommand frame, and CLTU. These are the data structures carried by the F-SP, F-TCF,

and F-CLTU services, respectively.

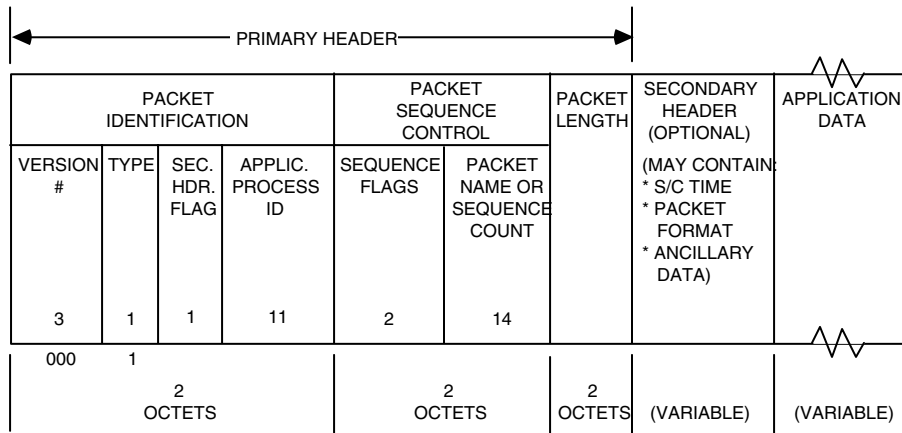


Figure 4-14: Telecommand Packet Format

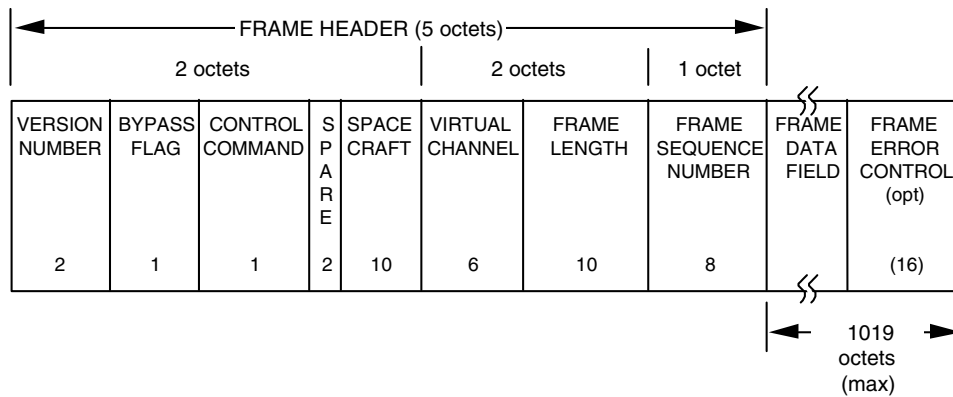


Figure 4-15: Forward Telecommand Frame Format

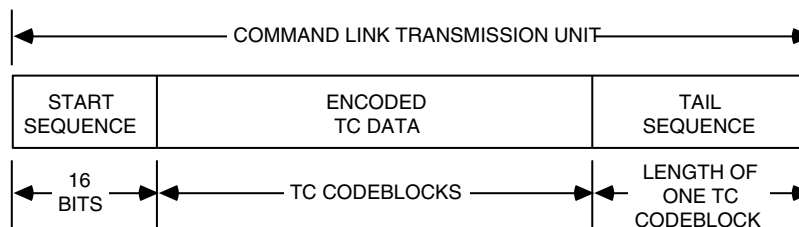


Figure 4-16: Components of the CLTU

4.3.2 MAPPING OF FORWARD TELECOMMAND SLE SERVICES TO FUNCTIONAL GROUPS

The Forward Telecommand SLE services are implemented as a modular architecture consisting of three SLE-FGs corresponding to the ‘Service Provider’ boxes shown in figure 4-5. Each SLE-FG presents an interface to a user of a particular SLE service and one or more SLE-FGs need to be implemented to provide that service. For example, organizations wishing to act only as a ‘CLTU Service Provider’ can do so by implementing just a ‘Forward Telecommand Space Link Processing’ SLE-FG. However, to provide a forward packet service requires all three SLE-FGs, which may all be implemented by one organization or may be distributed between different organizations.

To be a ...	the following SLE-FGs need to be implemented ...
CLTU Service Provider	Forward Telecommand Space Link Processing SLE-FG
TC Frames Service Provider	Forward CLTU Generation SLE-FG Forward Telecommand Space Link Processing SLE-FG
VC and Packet Service Provider	Forward TC VC Data Insertion SLE-FG Forward CLTU Generation SLE-FG Forward Telecommand Space Link Processing SLE-FG

Figure 4-17 shows the Functional Groups involved in Forward Link Telecommand SLE services and identifies the services that are provided by each. All of the services shown are available for Cross Support, including those between the Functional Groups.

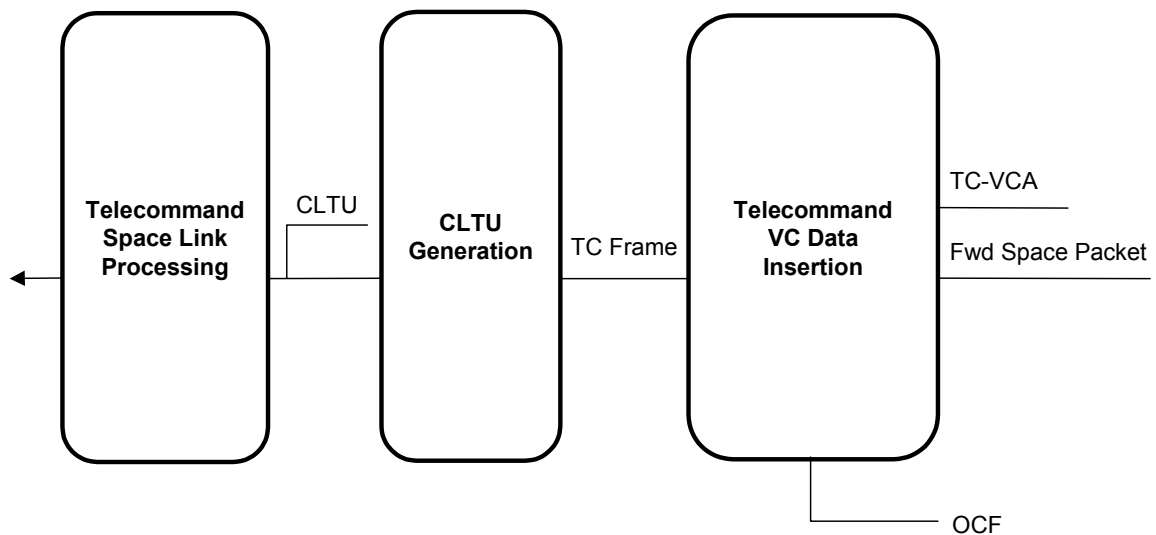


Figure 4-17: Forward Telecommand SLE Functional Groups and Services

4.4 FORWARD AOS SLE SERVICES

4.4.1 LIST OF FORWARD AOS SLE SERVICES AND THEIR INTERFACES

The forward SLE services associated with the Advanced Orbiting Systems (AOS) protocols intended for use in applications such as space station operations are not yet defined by CCSDS but will include:

- Forward AOS Space Packets (F-AOSSP);
- Forward AOS Virtual Channel Access (F-AOSVCA);
- Forward Bitstream (F-Bitstream);
- Forward Proto Virtual Channel Data Unit (F-pVCDU);
- Forward Coded Virtual Channel Data Unit (F-cVCDU);
- Forward Virtual Channel Data Unit (F-VCDU);
- Forward Insert (F-Insert).

The services will be implemented in two SLE-FGs as shown in figure 4-18.

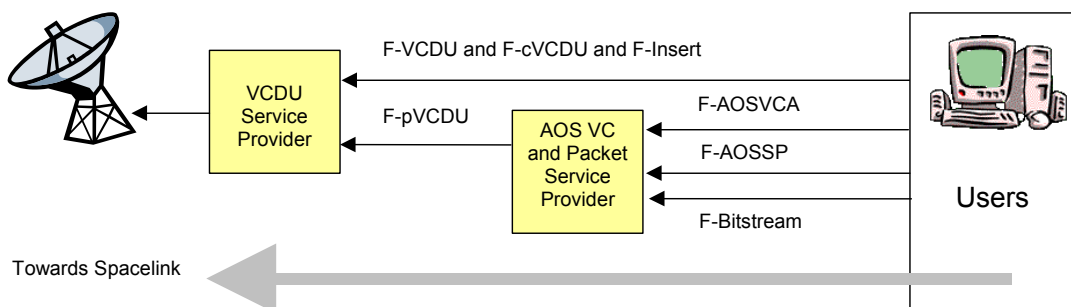


Figure 4-18: AOS Forward SLE Services are Provided in Two Stages

4.4.2 MAPPING OF FORWARD AOS SLE SERVICES TO FUNCTIONAL GROUPS

The Forward AOS SLE services are implemented as a modular architecture consisting of two SLE-FGs corresponding to the ‘Service Provider’ boxes shown in figure 4-19.

To be a ...	the following software needs to be implemented ...
VCDU Service Provider	Forward AOS Space Link Processing SLE-FG
AOS VC and Packet Service Provider	Forward AOS VC Data Insertion SLE-FG Forward AOS Space Link Processing SLE-FG

Figure 4-11 shows the Functional Groups involved in performing Forward Link AOS SLE services and identifies the services that are provided by each. All of the services shown are available for Cross Support, including those between the Functional Groups.

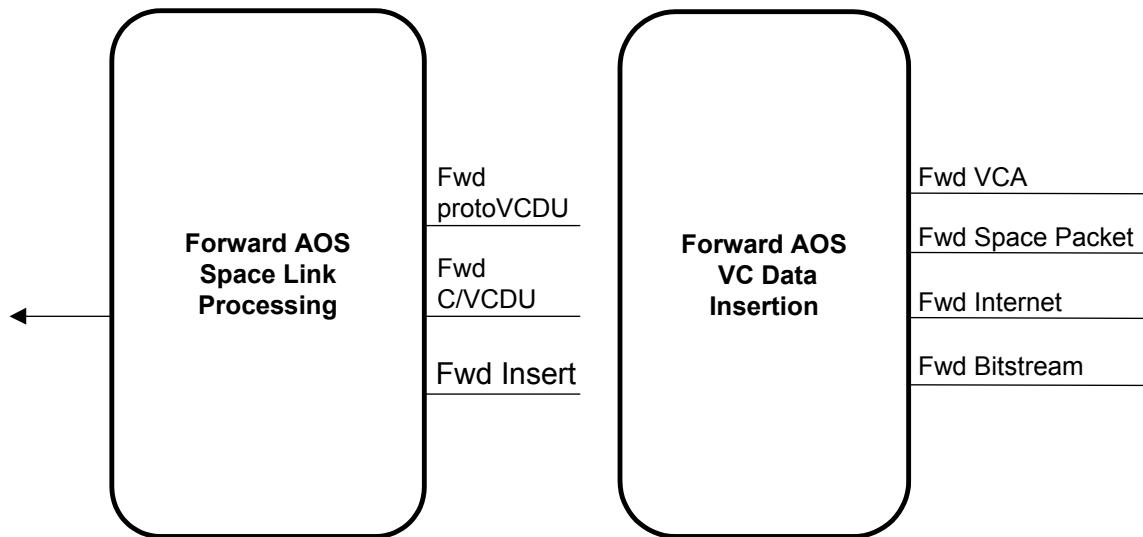


Figure 4-19: Forward AOS SLE Functional Groups and Services

4.5 SLE COMPLEXES

The SLE Complex is a logical grouping of Functional Groups that collectively perform a subset of the entire SLE service functionality. One or more Functional Groups can be implemented in one or more SLE Complexes by different organizations. Individual functions of a Functional Group cannot be distributed across multiple SLE Complexes - all functions of a Functional Group must be implemented within a single SLE Complex.

SLE Systems are often made up of several SLE Complexes that interoperate to provide a space mission with SLE services. These physical systems may be operated independently by different agencies. Each Complex must implement all the functions required to provide the particular SLE services that it provides. However, not all functions need necessarily to be exposed outside the Complex - only those that provide the SLE Service required by the user.

From a Cross Support view, the SLE System is composed of one or more SLE Complexes as illustrated in figure 4-20. The interfaces between SLE Complexes must conform to the interfaces defined for SLE Services in the SLE Service Management and SLE Transfer Service suite of Recommendations.

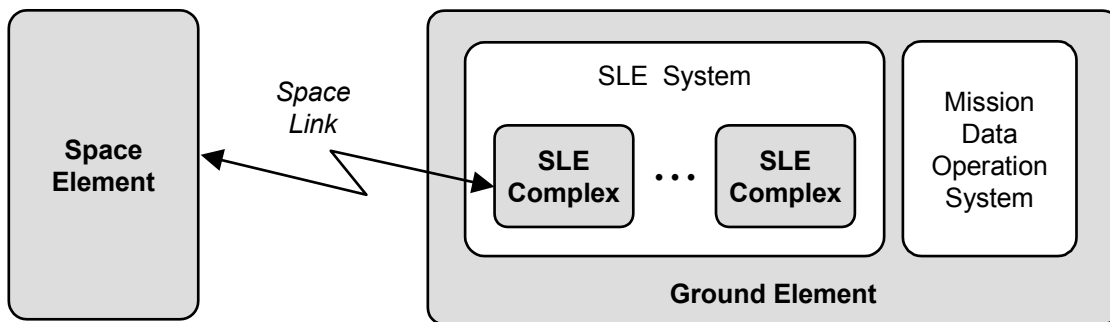


Figure 4-20: SLE Complexes

Figure 4-21 shows an example of how three SLE Complexes could be combined to provide the R-SP and F-SP services to users in an MDOS.

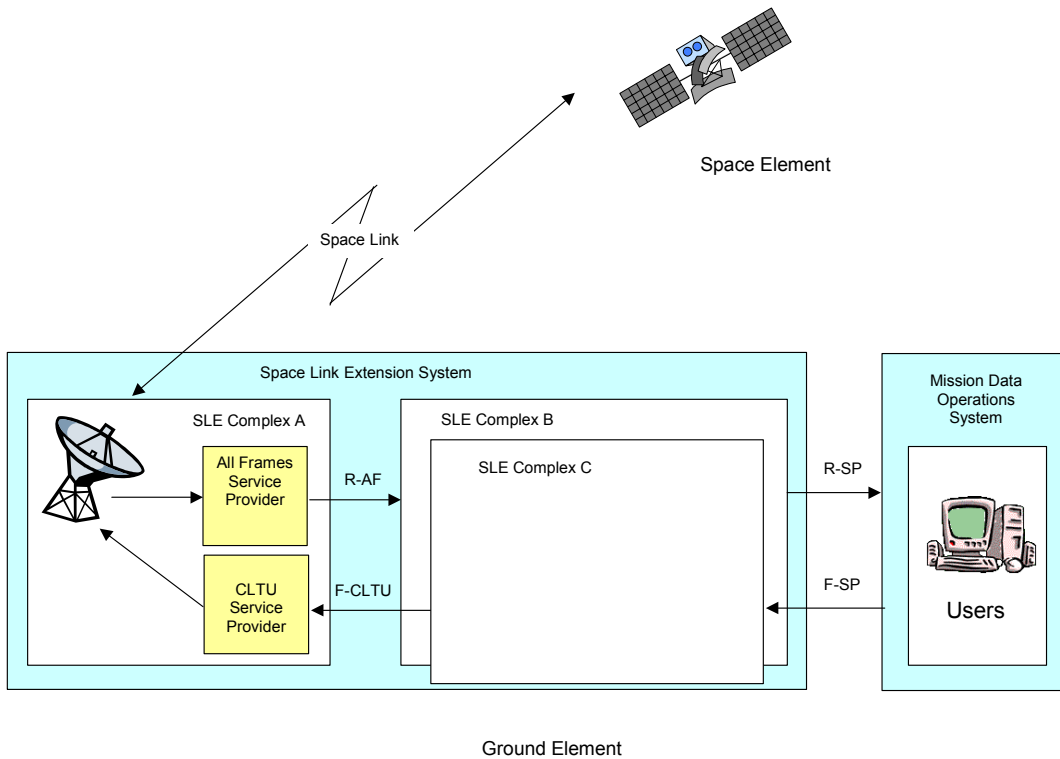


Figure 4-21: Example of Connected SLE Complexes

4.6 SLE DATA TRANSFER SERVICE OPERATIONS

4.6.1 DESCRIPTION OF OPERATIONS

The following operations are common to all the SLE Data Transfer services:

- **BIND** and **UNBIND**. These operations make and (cleanly) break the connection between user and provider.
- **PEER-ABORT** and **PROTOCOL-ABORT**. These operations break the connection between user and provider in the event of a SLE protocol problem (PEER-ABORT) or a problem with an underlying communication service (PROTOCOL-ABORT).
- **START** and **STOP**. These operations change the state of the user and provider to and from the active state in which SLE data units may be transferred from one to the other.
- **TRANSFER-DATA**. This operation transfers data units from the user to provider in the forward case, and from the provider to user in the return case.
- **SYNC-NOTIFY** and/or **ASYNCH-NOTIFY**. These operations either synchronously or asynchronously to the TRANSFER-DATA operation, enable notifications about the state of the service to be sent to the user from the provider.
- **SCHEDULE-STATUS-REPORT** and **STATUS-REPORT**. These operations

allow a user to schedule status reporting from the provider and allow a provider to provide the requested status reports.

- **GET-PARAMETER.** This operation allows a user to get the value of a parameter from a provider, which is relevant to the current service instance.

Some services include these operations too:

- **SYNC-SET-PARAMETER** and/or **ASYNC-SET-PARAMETER.** These operations either synchronously or asynchronously to the TRANSFER-DATA operation, enable the user to set the values of a limited number of parameters that are relevant to the current service instance.
- **THROW-EVENT.** This operation allows a forward direction service user to send an event that requires management action.
- **INVOKE-DIRECTIVE.** This operation allows a ‘privileged’ FSP or TCVCA user to invoke TC directives related to, for example, the operation of a Command Operation Procedure (COP).

4.6.2 STATE TRANSITIONS

Figure 4-22 shows the effect of the basic operations on the states of the SLE service provider.

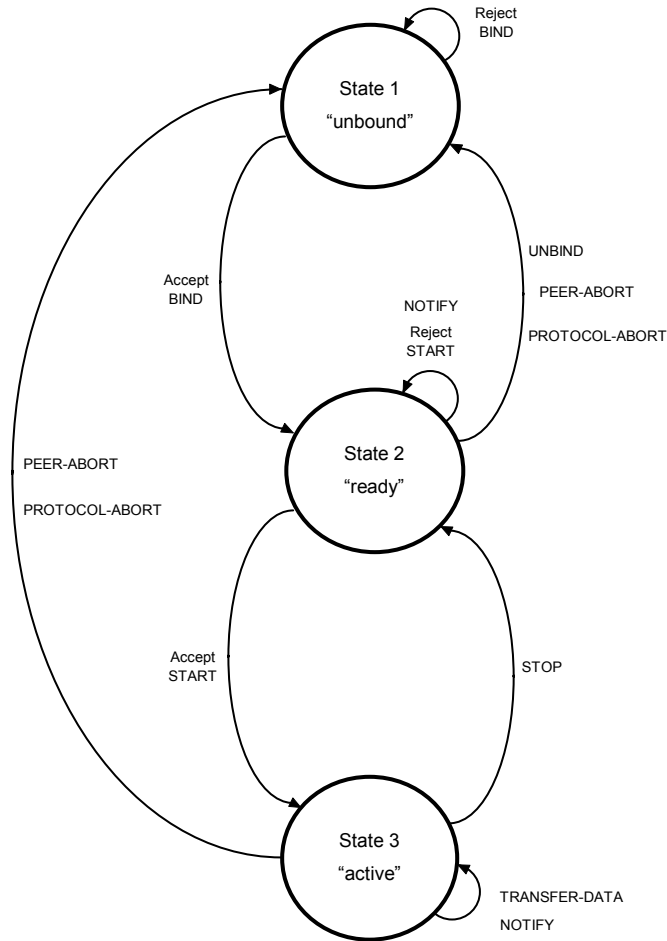


Figure 4-22: Simple State Diagram for SLE Services

4.6.3 OPERATION EXAMPLE

Each operation in an SLE service specification has a table of parameters such as the one shown in figure 4-23. Each parameter in the table is described in the specification, which also lists the values that the parameter may have.

In the table, ‘M’ refers to a mandatory parameter and ‘C’ to a conditional parameter. All parameters are mandatory except the diagnostic parameter, which is only returned by the provider if the result parameter has a value of ‘negative’.

Figure 4-24 provides an example of ASN.1 notation for RCF-START.

Parameter	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invocation-id	M	M
start-time	M	
stop-time	M	
global-VCID	M	
MCID	M	
latency-limit	M	
online-frame-buffer-size	M	
result		M
diagnostic		C

Figure 4-23: Example of SLE Service Operation Table (RCF-START)

Each operation has a corresponding ASN.1 definition in an annex to the service specification. Figure 4-16 shows the ASN.1 data structure of the RCF-START operation.

```

RcfStartInvocation ::= SEQUENCE
{
    invokerCredentials    Credentials
  , invocationId         InvocationId
  , startTime            Time
  , stopTime             Time
  , globalVcId          GvcId
  , masterChannelId     MCId
  , latencyLimit        IntPosShort
  , onlineFrameBufferSize IntPosLong
}

```

Figure 4-24: Example of ASN.1 Notation for RCF-START

4.7 THE SLE APPLICATION PROGRAM INTERFACE (API)

The SLE API for transfer services provides an interface to SLE application programs for exchange of SLE operation invocations and returns between a SLE service user and a SLE service provider. The API consists of two distinct layers, the API Proxy and the API Service Element, as shown in figure 4-25. The SLE Application is not part of the API, but it must provide some interfaces for use by the API, which will be defined in an SLE API Specification (planned for future publication).

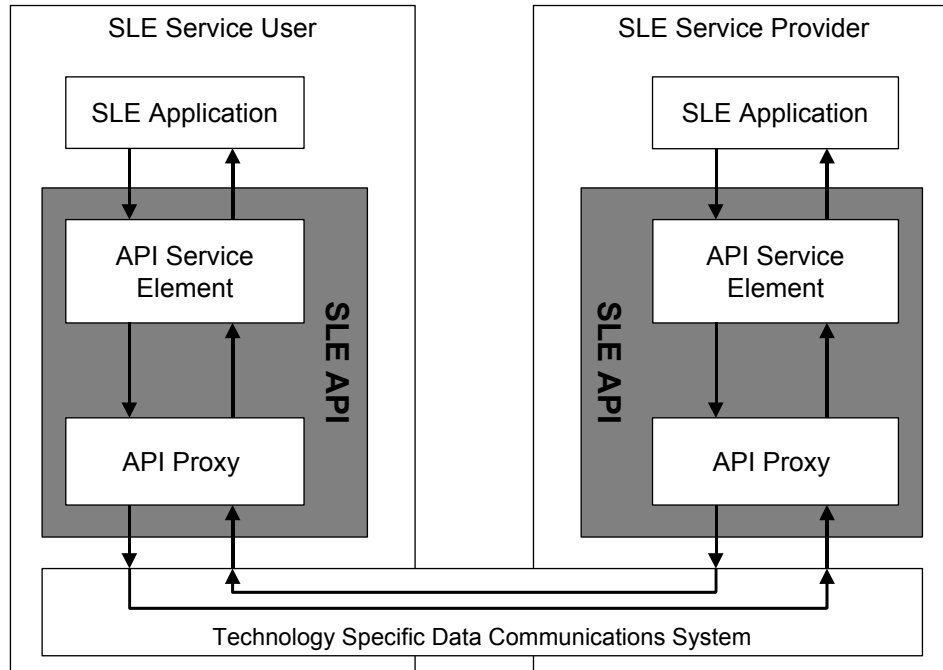


Figure 4-25: Layers of the SLE API

The purpose of the API Proxy is to provide a common, technology independent interface to the data communications system used for transmission of SLE protocol data units. It makes higher layers of the API and SLE applications independent of the specific communications technology and shields applications from technology changes.

The API Service Element implements those aspects of SLE transfer service provisioning that can be clearly separated from service production and service use, offloading applications from lower level aspects of the SLE transfer service protocol. The API Service Element enforces conformance to the state tables defined by the CCSDS Recommendations and checks parameters of SLE operations on completeness, consistency and range. These features support early detection of any potential problems in SLE applications and protect the application software from errors induced by a peer system.

5 SLE SERVICE MANAGEMENT

5.1 OVERVIEW OF SLE SERVICE MANAGEMENT

The purpose of SLE Service Management is to standardize and automate, as far as practicable, those interactions between users and providers of SLE services that are required to agree and schedule the services. In addition, the service management provides the means to monitor and control the resources needed in the provider complex(es) and the MDOS to execute the service. In essence, SLE Service Management is responsible for:

- agreeing the values of the parameters involved in an SLE service;
- allocating the SLE Complex resources needed for the execution of SLE services;
- configuring, monitoring and controlling the SLE Complex resources during the execution of the service.

The entities involved in SLE Service Management are shown in figure 5-1 and are:

- ‘Complex Management’ in one or more SLE Complexes;
- ‘Utilization Management’ in a Mission Data Operations System.

Figure 5-1 shows how SLE Service Management fits in the context of the SLE System, the MDOS, and the SLE Data Transfer services that were discussed in section 4.

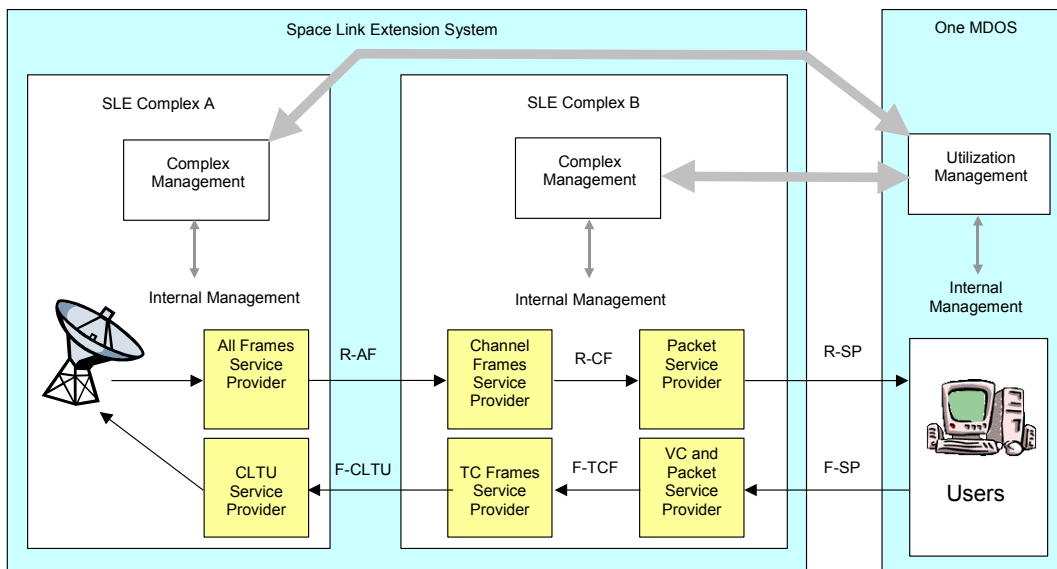


Figure 5-1: SLE Service Management in Context

SLE Complex Management represents the functions performed within the SLE Complex in a standard way, as defined in the SLE Service Management Recommendation. The way in which the Complex Manager interfaces with the equipment used to provide the SLE services is not currently defined by CCSDS. This means that the users in the MDOS do not need to be concerned about the internal workings of the SLE Complex - the SLE Complex Management provides the interface that hides the complexity of the SLE Complex. This is depicted in figure 5-2.

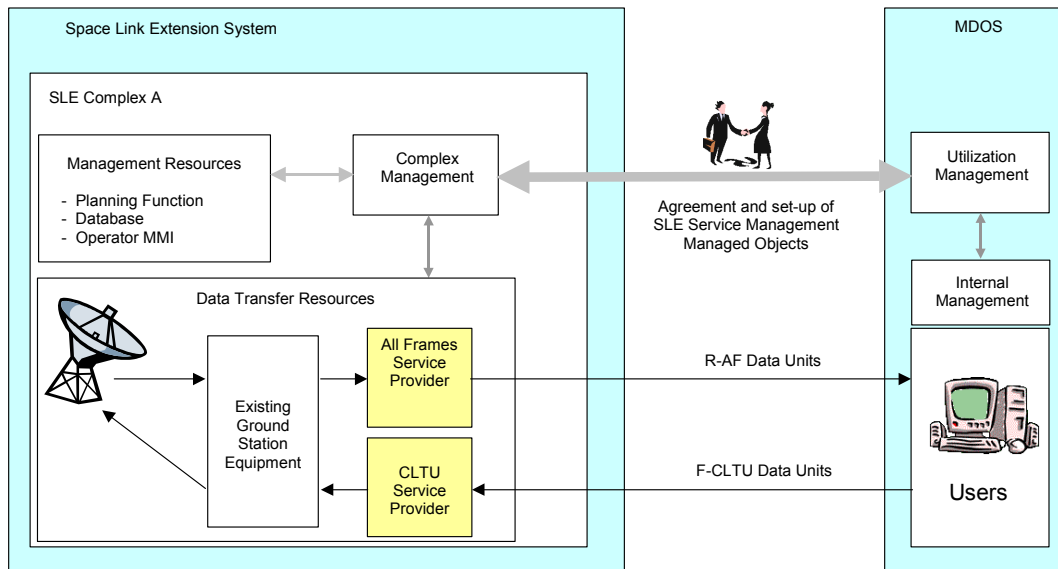


Figure 5-2: SLE Complex Management Internal Interfaces

5.2 SCOPE OF SLE SERVICE MANAGEMENT

The Open Systems Interconnection *Basic Reference Model* (reference 17) identifies five ‘management functional areas’: configuration management, accounting management, fault management, security management, and performance management. SLE Service Management includes aspects of the first four functional areas.

- **Configuration management** is the most extensive aspect of SLE Service Management. The configuration management capabilities enable SLE Utilization Management to interact with SLE Complex Management to define collections of SLE *transfer service instances* grouped into SLE *service packages*. Each service package specifies the input and output data channels, processing parameters, and storage requirements to be provided by the SLE Complex to a space mission. SLE Utilization Management defines these service packages by performing operations that create *managed objects* at the supporting SLE Complex.
- The **accounting management** aspect of SLE service management enables SLE Complex Management to provide SLE Utilization Management with reports, for each service package, of the data that is received by, the data that is sent from, and the data that is processed by that particular SLE Complex.

- The **fault management** aspect of SLE service management provides SLE Utilization Management with notification of faulty conditions within the SLE Complex that affect services being provided. Actual fault detection, isolation, and correction within the SLE Complex are an internal activity that is not exposed to SLE Utilization Management.
- The **security management** aspect of SLE service management protects access to Cross Support services through administration of security credentials, control of access to Cross Support interfaces and authentication of requests for data.
- SLE service management does not include **performance management**, because monitoring and controlling an SLE Complex’s performance is an internal matter for the organization that operates the complex.

5.3 FEATURES OF SLE SERVICE MANAGEMENT

The SLE service management concept includes the following features:

- The user is isolated from the details of the individual organization’s SLE System. The user interacts with SLE Utilization Management to provide service characteristics. The user does not need to know how the service is provided, but only how to interface with the service provider. SLE Utilization Management negotiates with the involved organizations to establish the parameters of the service interface.
- A common management representation of services is provided to users, regardless of the number and combination of services offered by any particular organization. It supports flexibility of an organization to organize internal management domains as it sees fit, while presenting a consistent service view to the user, and it facilitates the ability of organizations to hide system-specific details from users.
- Adequate instrumentation and reporting within the organization’s SLE System is provided to the user. The purpose is to provide sufficient accountability for all services performed by the system. The management concept does not specify SLE Complex-internal management requirements or concepts for management data collection and reporting. It only deals with what information needs to be provided to the mission.
- As far as possible, the management process is designed with automation in mind.
- The user’s needs to support the mission are the drivers of the management concept. The management concepts have not been established for the sake of standardization. Rather, concepts have been adopted to support the real needs of the mission activity.
- The cost of implementation has been kept in mind when establishing management concepts. The management concept has been formulated to minimize costs.

5.4 SLE SERVICE MANAGEMENT ENVIRONMENT

Figure 5-3 shows the SLE architecture model defined in the *Cross Support Reference Model* (reference [9]). This model is defined according to the Abstract Service Definition Conventions (ASDC), as summarized in *TCM* (reference [8]).

The focus of *Service Management* (reference [10]) is the interface indicated by the bold dashed line near the top of the figure. SLE service management is defined from the perspective of SLE Utilization Management.

SLE service management takes place when SLE Utilization Management invokes management operations and SLE Complex Management performs them. *Service Management* (reference [10]) defines the dialog between the two entities, but not the manner in which the operations are performed. This is internal to the SLE Complex.

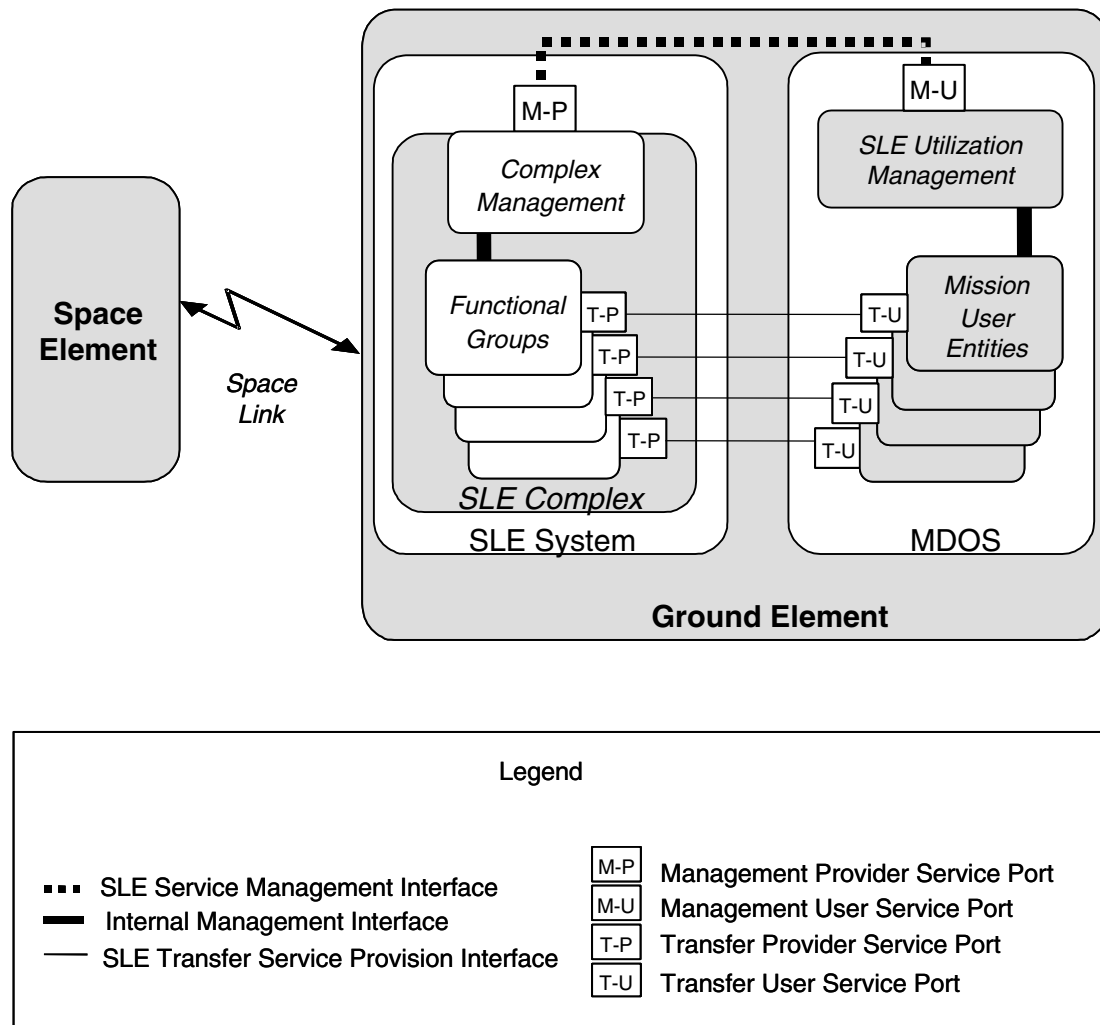


Figure 5-3: SLE Service Management Architecture

All Functional Groups in an SLE Complex are under the authority of a single SLE Complex Management. Figure 5-4 shows an example of an SLE Complex consisting of four Functional Groups managed by a single SLE Complex Management.

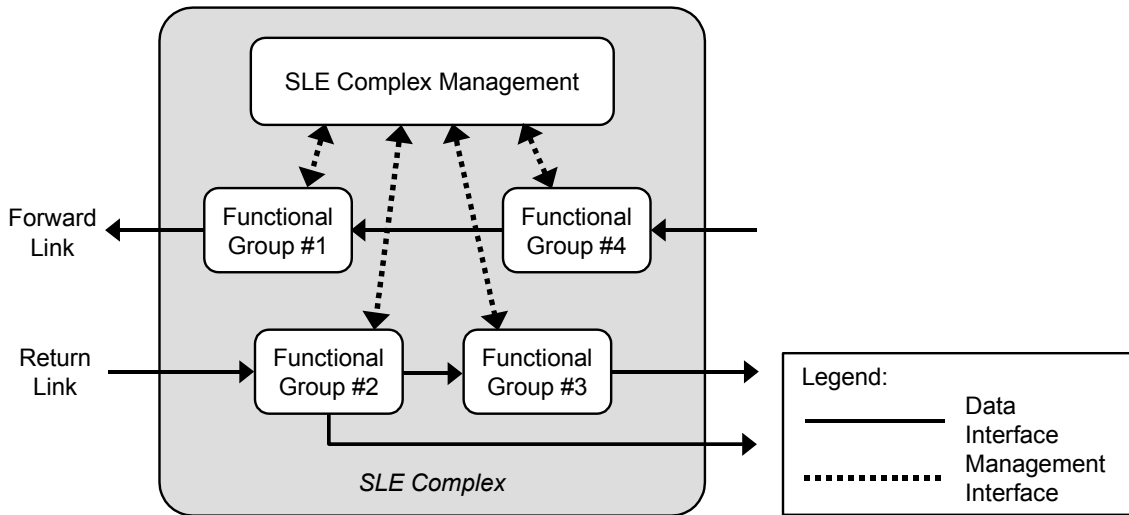


Figure 5-4: Example of Functional Group Configuration within a Complex

SLE Service Management is accomplished through the management of the functions performed by the individual SLE Complexes that provide the SLE services.

If more than one SLE Complex provides a service, SLE Utilization Management coordinates with the SLE Complex Management in each SLE Complex to provide the services the mission requires. SLE Utilization Management must also coordinate and resolve conflicts between multiple service users. These concepts are illustrated in figure 5-5.

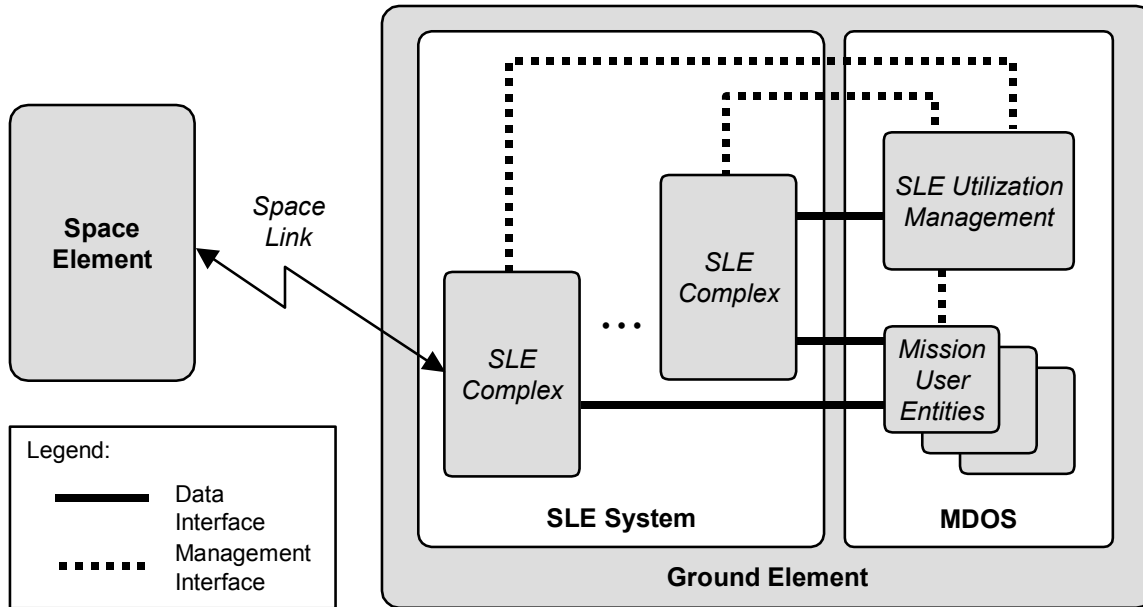


Figure 5-5: SLE Complex/MDOS Interfaces

5.5 TIME SPAN OF SLE AGREEMENTS AND PACKAGES

5.5.1 GENERAL

This section presents the terminology used to describe the lifetime of an SLE service. As noted in section 1.6, several terms that were defined in the *Cross Support Reference Model* (reference [9]) are used in this Report. These include *Space Link session*, *service instance*, *service package*, and *service agreement*.

The section ends by addressing the SLE service package definition phase and utilization phase, including the role of some of the service management *managed objects* introduced earlier in this document.

5.5.2 PHASES OF SLE SERVICE PACKAGE LIFETIME

The SLE service package lifetime consists of two phases: a definition phase and a utilization phase. Both of these phases occur for each SLE service package. An example of the SLE service package lifetime phases is given in figure 5-6.

The arrangement made between a mission and an SLE Complex is formalized in an SLE Service Agreement that sets out the overall arrangements for cross support between the two parties. The details of the cross support are elaborated in the definition phase, within the capabilities and constraints drawn up in the Service Agreement.

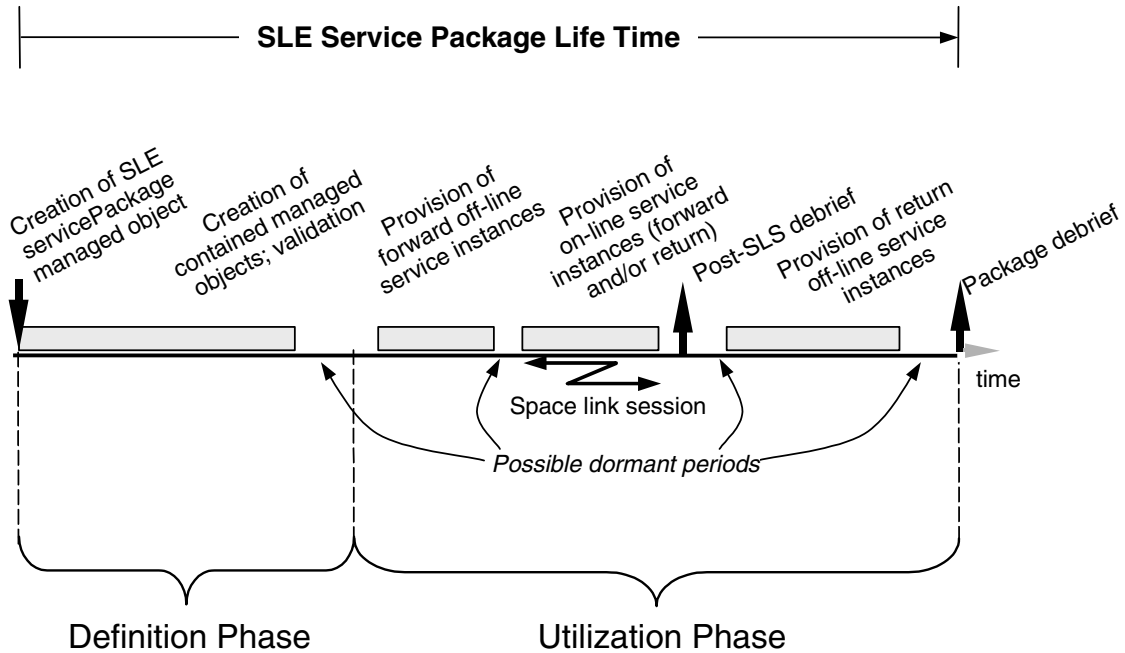


Figure 5-6: Example of Service Package Lifetime Phases

5.5.3 SLE SERVICE AGREEMENT PERIOD

The SLE Service Agreement period is the time during which an SLE Complex provides a space mission's SLE Utilization Management with the capability to create SLE service packages. It is also the time during which SLE Complex Management provides the SLE services defined by those packages, within the constraints of the SLE Service Agreement.

As illustrated by figure 5-7, many service packages may be provided over the course of an SLE Service Agreement period. Some of these may be related to a single Space Link session, while others may provide offline SLE transfer services related to two or more Space Link sessions.

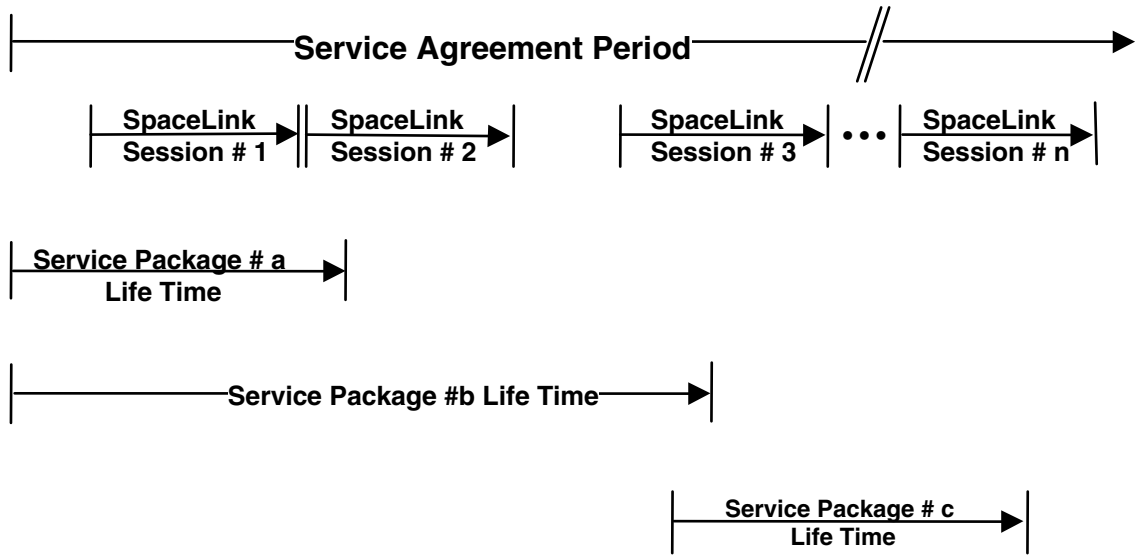


Figure 5-7: Example of Service Agreement Period

The SLE Service Agreement period often encompasses the entire operational lifetime of the supported spacecraft. This is the case when a mission is designed to use ongoing support from an SLE Complex, with a long-term SLE Service Agreement established before launch. A situation such as this may involve several service packages per day and hundreds (or thousands) over the agreement period. Other SLE Service Agreements, however, may be much shorter, as when a mission uses the services of a particular SLE Complex only for launch support, during an orbit maneuver, or for emergency operations. Typically, the SLE Service Agreement period spans multiple Space Link sessions.

5.5.4 SLE SERVICE PACKAGE LIFETIME

The SLE service package lifetime spans the time during which the SLE Complex provides the mutually agreed-upon services defined in an SLE service package and the time during which its contained SLE transfer service instances exist.

5.5.5 SERVICE INSTANCE PROVISION PERIOD

The service instance provision period is the time during which a service instance (i.e., the capability to transfer one or more SLE data channels of a given type) is provided. Figure 5-8 shows examples of three instances of the Return Space Packet (RSP) transfer service.

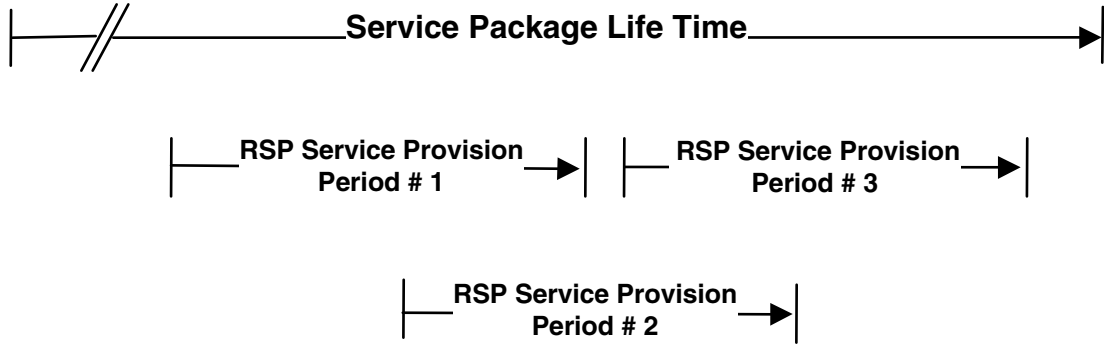


Figure 5-8: Example of Service Provision Period

5.5.6 ONLINE AND OFFLINE SERVICES IN SLE SERVICE PACKAGES

The online SLE service instances of a given SLE service package are related to one single Space Link session, though they may begin before the start time and extend beyond the end time of the Space Link session.

Offline SLE service instances may exist at any time during the SLE Service Agreement period, subject to the conditions agreed upon in the SLE Service Agreement and in the SLE service package definition. Offline SLE service instances of a given package may be related to a Space Link session provided as part of that package or to Space Link sessions provided under one or more other SLE service packages.

Figure 5-9 shows an example of an SLE service package that provides both online and offline services. In this example, the SLE service package #k is a set of one forward online service instance and two return service instances (one online and one offline) associated with Space Link session #n.

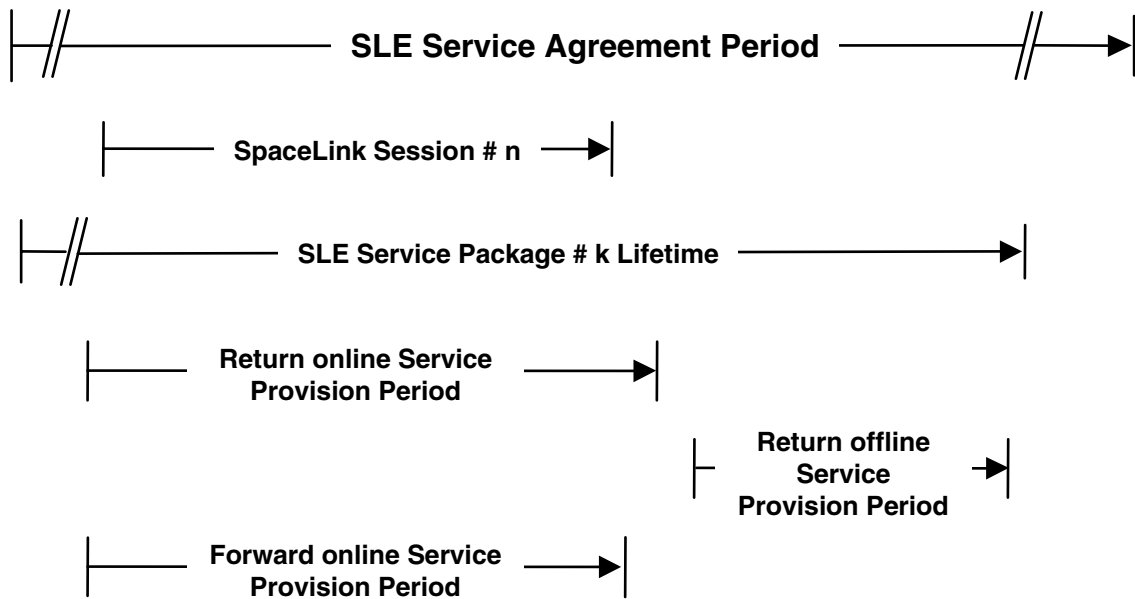


Figure 5-9: Online and Offline Services in One SLE Service Package

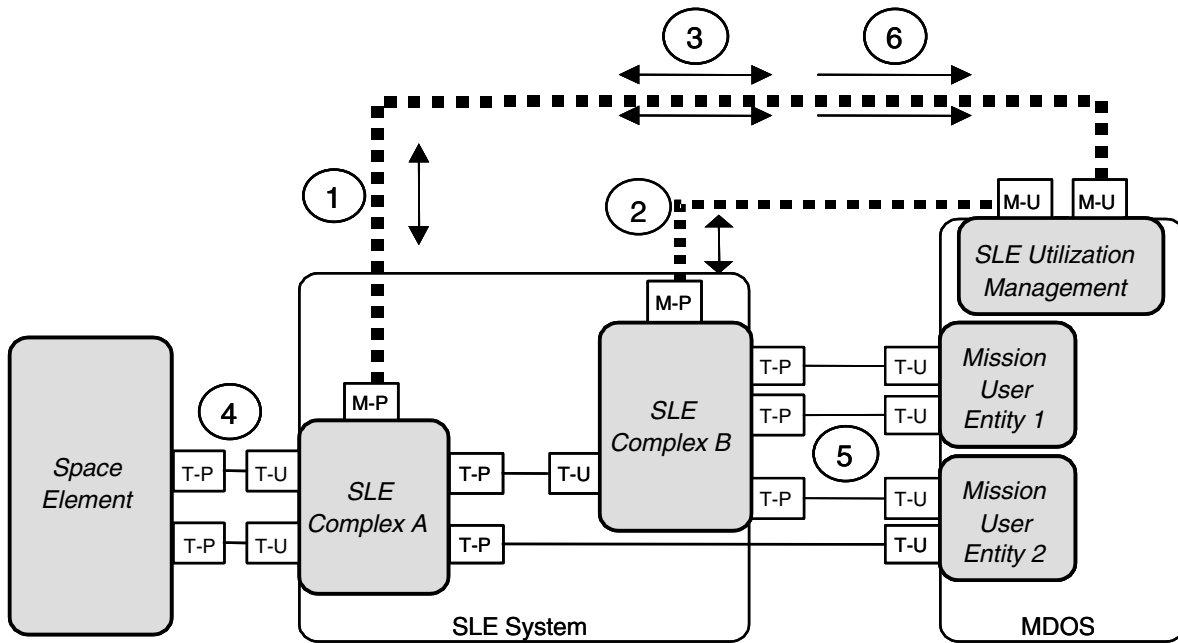
5.5.7 SLE SERVICE PACKAGE DEFINITION PHASE

During the definition phase, SLE Utilization Management creates a *servicePackage* managed object at an SLE Complex. The managed objects contained in the *servicePackage* are created along with the *servicePackage* managed object itself. Parameter values, including schedule parameters, for SLE data transfer service instances are selected within the bounds specified in the SLE Service Agreement. SLE Utilization Management and SLE Complex Management interact during the definition phase to create the SLE managed objects that define and control the SLE service package. The attributes of the managed objects provide:

- a) selection of the SLE data transfer services to be provided by the SLE Complex (within the framework of the SLE Service Agreement);
- b) identification of the interfaces at which these SLE data transfer services are to be made available;
- c) identification of the SLE data transfer services that must be provided to this SLE Complex by the Space Element or by other SLE Complexes;
- d) specification of mission-specific information needed by SLE Complex Management to configure SLE service production within the SLE Complex;
- e) identification of the eventual consumers of SLE transfer services (i.e., Mission User Entities or other SLE Complexes);
- f) specification of the schedule for provision of
 - 1) online instances for a Space Link session, and
 - 2) offline instances of a given SLE data transfer service;
- g) specification of reporting by SLE Complex Management to SLE Utilization Management.

A dormant period may occur after creation is complete and before provision of SLE service package instances begins.

Figure 5-10 presents an example in which each of two SLE Complexes is requested to provide an SLE service package for a mission. The management operations that create these packages are shown as steps 1 and 2 in the figure. The remaining steps in this figure are addressed in the key to circled numbers.



Key to Circled Numbers	
1	ServicePackage created at Complex A through management operations
2	A second servicePackage is created at Complex B
3	Management control of SLE transfer services at both Complexes
4	Physical RF channel provided during space link session (this applies to all data transfers between the space element and the Complex)
5	SLE transfer services provided to mission user entities within the MDOS (this applies to all data transfers between the SLE system and the MDOS)
6	Debriefing reports from each Complex to SLE Utilization Management

Figure 5-10: Example of Two Service Packages at Two SLE Complexes

5.5.8 SLE UTILIZATION PHASE

The utilization phase of a SLE service package is the period during which the SLE Complex provides the mutually agreed-upon SLE data transfer services. The utilization phase consists of *service instance provision periods*, possibly separated by dormant periods during which no service instances are provided. SLE Complex Management provides SLE Utilization Management with debrief reports following service provision periods. Debriefs may occur at any time following a service provision period.

SLE service provision periods occur as scheduled in the SLE service package. During the service provision period, the SLE Complex transfers data to or from the user, according to

the particular SLE service being provided. In the case of an SLE service package that includes one or more offline SLE transfer service instances, there may be dormant periods between these offline provision periods, or between an online SLE service provision period provided at the beginning of the utilization phase and the first offline SLE service provision period of the SLE service package. For a given SLE service package there is only one period (i.e., one Space Link session) during which online SLE services are provided.

The provision of the SLE transfer service instances to the Mission User Entities is controlled in two ways:

- a) through control operations in the data transfer service, invoked by the Mission User Entity;
- b) through management operations invoked by SLE Utilization Management.

In the example shown in figure 5-10 above, the management operations that SLE Utilization Management uses to monitor and control these packages are shown as step 3. The physical channel (radio frequency or RF) transfer is shown as step 4, and the provision of SLE transfer services to the mission user entities within the MDOS is shown as step 5.

The utilization phase ends with a debrief to allow the SLE Complex and SLE Utilization Management to exchange information about the SLE Services that were provided throughout the SLE service package lifetime. In the example shown in figure 5-10 above, the management operations that provide debrief reports on the two SLE service packages that have been provided for the mission are shown as step 6.

5.6 OBJECT-ORIENTED APPROACH TO SERVICE MANAGEMENT

5.6.1 MANAGED OBJECTS

SLE service management is defined in terms of managed objects, and makes use of the concepts of containment and inheritance commonly used in object-oriented analysis and design (*Object-Oriented Analysis*, reference [19]).

SLE managed objects (sometimes abbreviated as MOs) represent the properties of a resource within an SLE system, as seen by SLE Utilization Management. SLE managed objects define SLE services, resources, timing, and conditions.

A managed object is defined in terms of attributes it possesses, operations that may be performed on it, notifications that it may issue and its relationships with other managed objects. This is distinct from, but related to, any definition or specification of the actual resource that the managed object represents as an element of an SLE system.

Figure 5-11 shows two examples of SLE managed objects. In (a), a managed object represents an SLE Service Agreement, which specifies the SLE transfer services that may be provided by the SLE Complex over the lifetime of the SLE Service Agreement. In (b), a managed object represents the provision of a return space packet transfer service. In both cases, the presence of attributes of the managed object is illustrated.

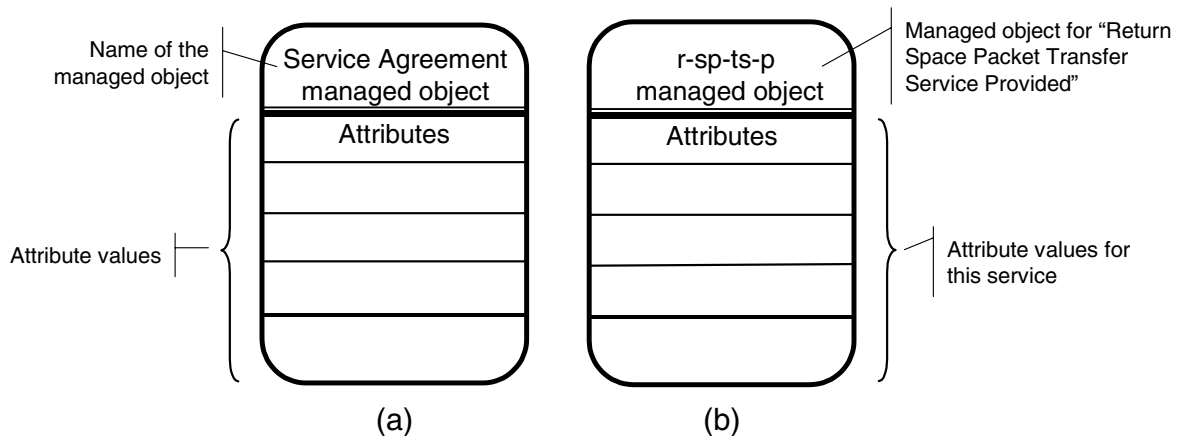


Figure 5-11: Managed Objects – Two Examples

5.6.2 MANAGED OBJECT CLASSES

Figure 5-12 shows an object class. An object class defines a type of managed object - a named set of managed objects that have the same naming scheme, attributes, and other characteristics. Thus an object class specifies the characteristics that each object in the class will possess. It provides a template for creation of objects. Each object created according to the template is called an instance of the class.

The specification for a managed object class defines the attributes of each instance but not necessarily their values. It may define states that the object can assume, actions that an object instance may perform, when requested to do so by management, and notification messages that the object instance may issue to management under certain specified circumstances. The object class also specifies whether other objects may (or must) be contained in an instance of its class and any limitations on that containment.

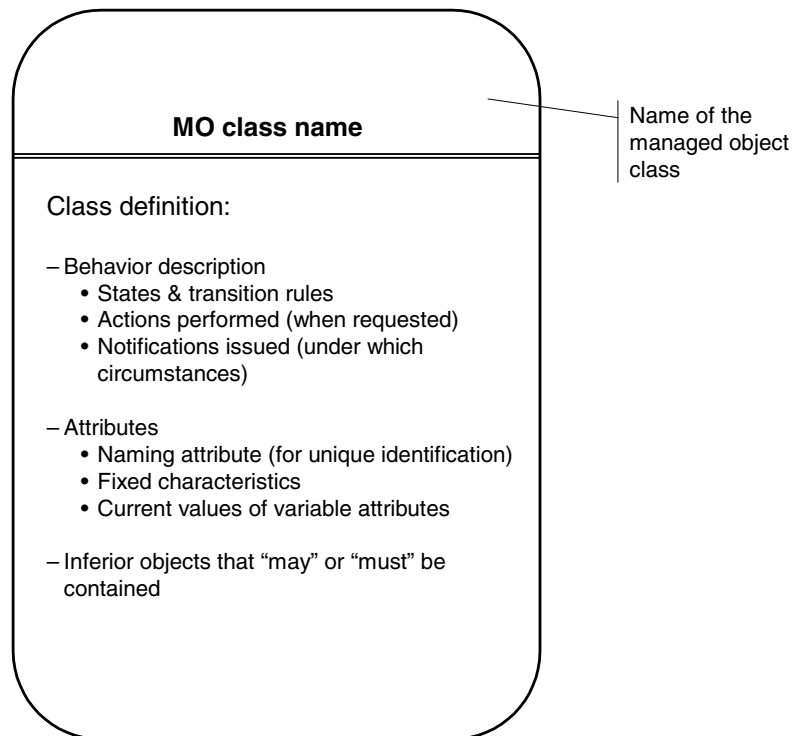


Figure 5-12: Managed Object Class

5.6.3 MANAGED OBJECT CONTAINMENT

As noted in subsection 5.6.1, a managed object may contain other managed objects. Containment provides the flexibility to construct complex managed objects, in order to represent complex cross support requirements, but allows the construction of simpler managed objects to represent simpler requirements. The containment concept is illustrated in figure 5-13.

The containment rules provided by the managed object specifications in this Report enable SLE Utilization Management to model the specific services needed for mission operations, as well as the resources that are required to provide those services.

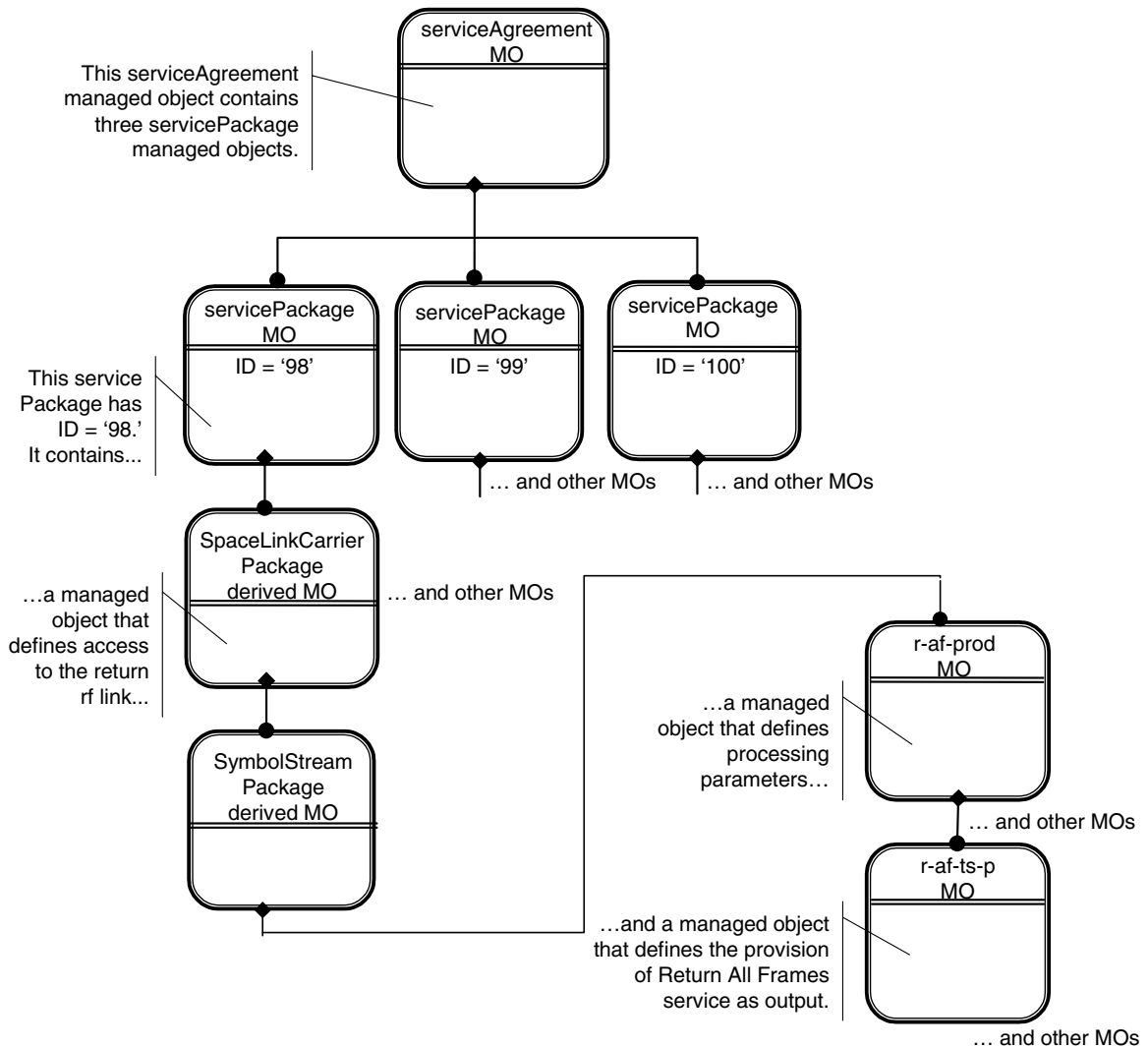


Figure 5-13: Managed Object Containment

5.6.4 MANAGED OBJECT CREATION

5.6.4.1 Service Agreement Managed Object Creation

Managed objects are created to define services, resources, timing and conditions.

SLE Utilization Management may invoke the CREATE operation. Refer to *Service Management* (reference [10]) for a discussion of this operation. A managed object's attribute values are determined when it is created. The values are given by one of the following:

- SLE Utilization Management, as parameters of the CREATE invocation;
- SLE Complex Management, as information returned in response to the CREATE invocation;
- The definition of the managed object class (as specified in *Service Management*).

Before managed objects can be created, we must first have a framework for creating managed objects pertaining to the space mission. A serviceAgreement managed object provides this framework.

ServiceAgreement managed object creation is a special case. SLE Complex Management creates it using attribute values agreed upon between the SLE Complex and the MDOS of the space mission to be supported. Putting a serviceAgreement managed object into place within a SLE Complex sets the stage for creation of all of the other managed objects that will be needed to manage cross support for the space mission.

5.6.4.2 Creation of Service Packages and Their Contained Managed Objects

Once a serviceAgreement managed object is in place, a space mission's MDOS can request that SLE Complex Management create servicePackage managed objects. Each servicePackage managed object, in turn, serves as the starting point for creation of the lower-level managed objects that define the specific SLE transfer services to be provided.

5.6.4.3 Creation Using a Reference Managed Object

Many space missions perform similar or identical operations repeatedly. To avoid repetitive creation of a managed object for each Space Link session, with little change from the last instance, reference managed objects may be used as shown in figure 5-14. Any or all attribute values of a reference managed object may be copied into the new object instance, with the exception of the unique managed object identifier. Typically, attribute values such as data channel identifiers, frequencies, and service users are copied from the reference managed object, while others, such as start and stop times, tracking data or passwords, are given new values.

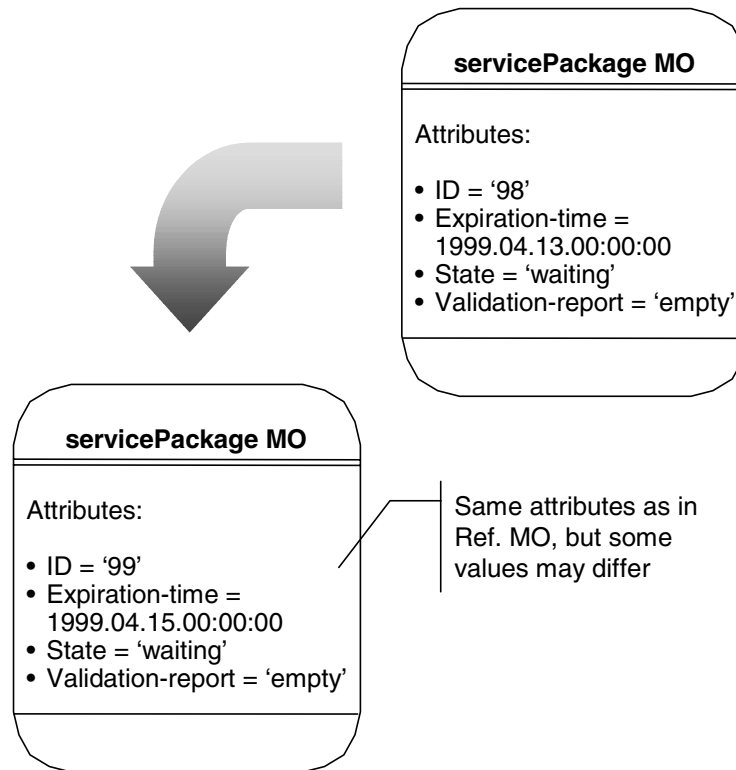


Figure 5-14: Reference Managed Object

5.6.5 MANAGEMENT OPERATIONS

SLE Utilization Management and SLE Complex Management interact by invoking operations at their management ports. SLE Utilization Management interacts with SLE Complex Management to define, schedule, monitor, and control SLE service packages at an SLE Complex; specify the SLE service instances to be provided; and identify authorized mission user entities to SLE Complex Management.

SLE Utilization Management and SLE Complex Management are able to specify or change values of managed object attributes, obtain the attribute values, request or provide reports, invoke actions (immediately or at a specified time in the future), and inform each other of exceptional conditions.

5.6.6 ABSTRACT AND CONCRETE CLASSES

An abstract object class provides partial definition of a class. An abstract object class can have no instances since it cannot be realized. An abstract class is established only to provide the basis for the definition of derived classes that can have instances. To emphasize that a particular class can have instances, i.e., that it is not an abstract class, it is sometimes called a concrete class. The use of the Abstract classes makes use of the powerful concept of inheritance. A concrete class inherits the characteristics of its parent abstract class. This approach clearly shows which characteristics are common to two or more derived classes of an abstract class and which attributes are different.

Figure 5-15 shows several examples of abstract and concrete classes in a managed object inheritance hierarchy. At the top of the hierarchy, an abstract class defines the most basic characteristics of a transfer service. At the next level, another abstract class restricts that definition to return transfer services. At the lowest level are concrete classes that define several types of actual return transfer services; these classes can be instantiated and thus are concrete classes.

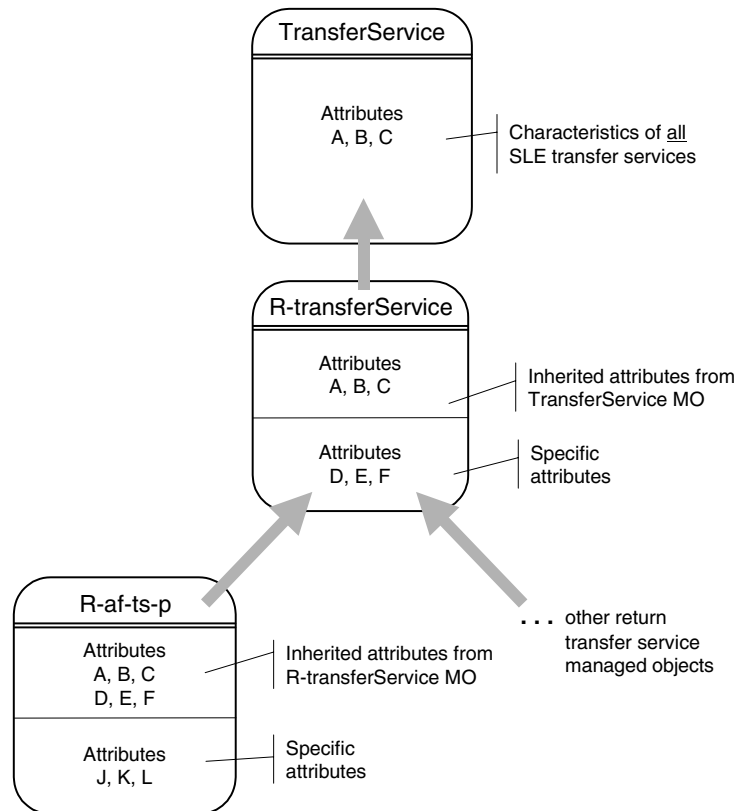


Figure 5-15: Abstract and Concrete Managed Object Classes

Each managed object class provides a template for the managed objects of that class. Components of these templates are:

- a) Objects contained by the managed object;
- b) States of the managed object;
- c) Attributes of the managed object;
- d) Actions performed by the managed object;
- e) State-related behavior of the managed object;
- f) Notifications provided by the managed object;
- g) Validation aspects of the managed object;
- h) Contribution to the servicePackage *report* notification.

ANNEX A

ACRONYMS

This annex identifies and defines the acronyms that have been adopted in this Report.

AOS	Advanced Orbiting System
APID	Application Identifier
CLTU	Command Link Transmission Unit
FSH	Frame Secondary Header
MC	Master Channel
MCID	Master Channel Identification
MDOS	Mission Data Operation System
OCF	Operation Control Field
SCID	Spacecraft Identification
SLE	Space Link Extension
TC	Telecommand
VC	Virtual Channel
VCA	Virtual Channel Access
VCDU	Virtual Channel Data Unit

ANNEX B

GLOSSARY

Annotation	The extension of the Space Link services requires that information be added to the Space Link data units. The process of adding information to the Space Link data is called annotation.
Cross Support.....	The situation when one organization uses part of another organization's data-system resources to complement its own system.
Forward Data	All data sent from the ground element to the space element (e.g., telecommand).
Functional Group	The fundamental building block of an SLE service; it is not decomposed further. Functional groups are derived from the layered functionality specified in the Space Link protocols.
Ground Element.....	The collection of systems and organizations based on the ground that provide SLE services used by a specified mission.
Mission.....	For the purposes of this Report, an undertaking to explore and/or utilize fields of interest using one or more spacecraft.
Online.....	For the purposes of this Report, the transfer of SLE service data through all or part of the SLE System during the time that the associated Space Link session is active.
Offline	For the purposes of this Report, offline refers to the transfer of SLE service data through all or part of the SLE System at a time other than that during which the associated Space Link session is active.
Operations Phase.....	In the Operations phase, the user defines support requests that are sent to the Provider Organization, which in turn responds by establishing the support in the form of a schedule. If the support cannot be scheduled, the fact is communicated to the user, who can in turn formulate new Support Requests.

REPORT CONCERNING CROSS SUPPORT CONCEPT—PART 1: SLE SERVICES

Payload.....	The on-board equipment that directly relates to the purpose of the spacecraft's flight.
Return Data.....	For the purposes of this Report, all data sent from the space element to the ground element (e.g., telemetry).
Schedule.....	For the purposes of this Report, a Schedule is a chronological list or timeline of spacecraft activities and allocated resources constituting the daily operations plan for mission support.
Scheduling.....	For the purposes of this Report, the task of placing activities onto a timeline and allocating resources.
SLE Complex.....	The functions that perform SLE System services can be distributed across multiple systems. This distribution is aligned with the layering of the Space Link services. The systems performing individual functions of a service may belong to different organizations and have varying size and structure. The systems performing SLE service are grouped into SLE Complexes by the organizations that implement them. Each SLE Complex has two components, a Service Provision component and a Management component.
SLE Complex Management.....	The component of an SLE Complex that manages space data transfer.
SLE Service.....	The set of services that extend one of the CCSDS Space Link Subnetwork services, providing access to the ground termination of that service from a remote ground-based system. An SLE service supplies or consumes one or more channels of the same Space Data Channel type.
SLE Service Agreement.....	The agreement between the agencies that are engaged in Cross Support. The SLE Service Agreement defines the set of Service Packages that are to be supported over the lifetime of the SLE Service Agreement. The resources that will be accessible and the privileges that will be extended are identified.
SLE Service Instance.....	The provision by an SLE Complex of the capability to transfer one or more SLE data channels of a given type, all of which are related to the same Space Link session.

REPORT CONCERNING CROSS SUPPORT CONCEPT—PART 1: SLE SERVICES

- SLE Service PackageAn SLE Service Package is the set of service instances, together with the specification of the characteristics of the production of those service instances, that are provided by one SLE Complex to one or more SLE Transfer Service users, with respect to one Space Link session.
- SLE System.....The global collection of systems and organizations that provide SLE services.
- SLE Utilization ManagementThe entity within an MDOS that interfaces with the Complex for use of SLE services.
- Space Data ChannelFor the purposes of this Report, a virtual stream of space link data units of the same type, with a single unique identification.
- Space Element.....The systems and organizations on board the spacecraft, which provide SLE services used by a specified mission.
- Space Link ServiceThe Space Link service is the service provided over the Space Link Subnetwork to the user during a contact with the spacecraft.
- UserFor the purposes of this Report, an entity receiving services.

ANNEX C

CROSS SUPPORT SCENARIOS

C1 CROSS SUPPORT ORGANIZATION

Cross support may occur between the SLE System and an application in the spacecraft or mission domain. The SLE System builds on the Space Link standards by standardizing the SLE and Service Management protocols and services. Such standardization allows a mission to interface with the SLE System, concatenating SLE services by interconnecting SLE Complexes within the SLE System, no matter how or where the services are implemented.

Implementation of Cross Support is each organization's responsibility. In other words, the groupings of Functional Groups into SLE Complexes are at the discretion of each organization. However, it should be noted that there is a limited number of rational groupings that may be selected by an organization.

The following subsections illustrate the SLE service concept for providing Cross Support by presenting a set of scenarios that first illustrate how the functional distribution of SLE Complexes may differ between organizations and then show some examples based on previous implementations of (non-SLE) cross support.

C2 FUNCTIONAL DISTRIBUTION OF COMPLEXES

When discussing Cross Support, a functional rather than a physical view is needed. There are cases where different organizations use the same word for a facility but mean different functions. For example, in figure C-1, 'ground station' contains an extensive processing capability; the control center distributes files of user data as well as performing forward link functions. In figure C-2, 'ground station' has a limited capability; return link processing, offline processing, and command control are performed at different facilities.

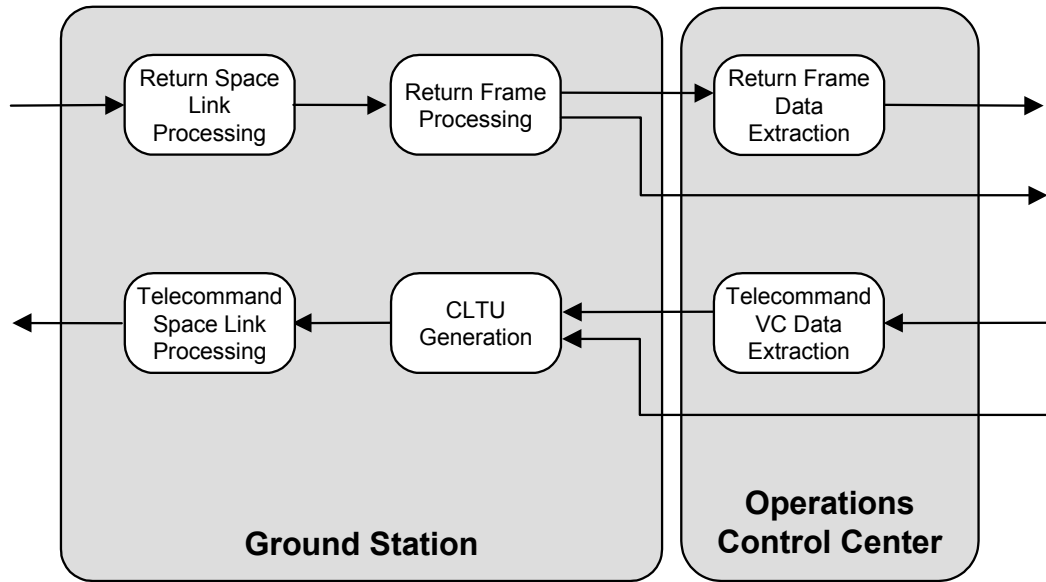


Figure C-1: Example of Ground Station with Extensive Capability

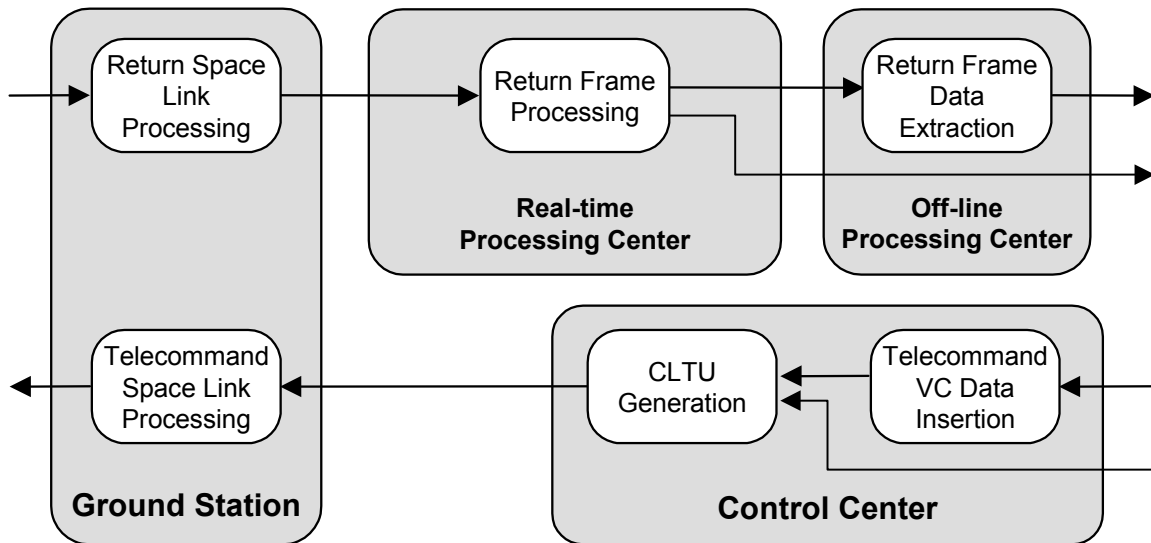


Figure C-2: Example of Ground Station with Limited Capability

An SLE Complex may contain the complete set of functions required by a mission (in which case the SLE Complex and the SLE System are the same), or a subset of functions. Figure C-3 is an example of an SLE System comprising two SLE Complexes. In this example, the Ground Terminal SLE Complex implements the Return Space Link Processing and Forward Telecommand Space Link Processing. The Remote Processing SLE Complex implements the Return Frame Processing, Return Data Extraction, CLTU Generation, and Telecommand VC Data Insertion. In this example, the mission uses only the Space Packet and Insert services.

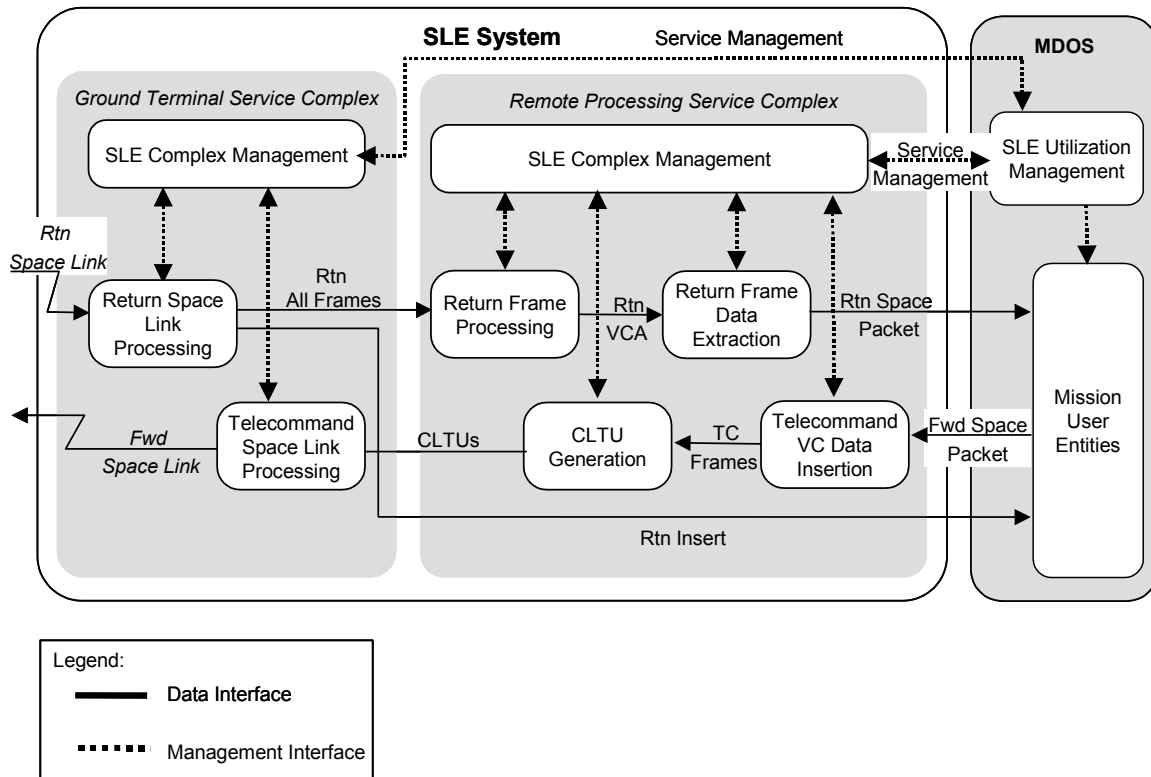


Figure C-3: SLE System with Two SLE Complexes

C3 CROSS SUPPORT SCENARIO ILLUSTRATIONS

This section illustrates examples of cross support drawn from the experiences of the space agencies that contribute to CCSDS. Although the agencies have implemented their systems differently, they must all perform common functions. This allows us to illustrate current agency implementations according to the Functional Groups described earlier in this document.

This subsection provides five scenarios of Cross Support that are generalizations and extensions of actual cross support implementations. Between them, these scenarios illustrate different functional allocations and different aspects of Cross Support that have been performed between agencies.

Cross Support Scenario #1, shown in figure C-4, shows the return link for a situation in which Agency B, which owns the spacecraft, may receive the return data either directly from its own SLE Complex, or from the Complex of another organization. The return data is processed and individual data units or files of sorted packets are distributed to users and to other agencies. In the figure, Agency A owns the payload and receives data from it directly. It feeds in processed data to Agency B, which also receives data from the spacecraft directly. The Agency B complex in the center of the figure sends products to the MDOS, as represented by the three boxes on the right of the figure.

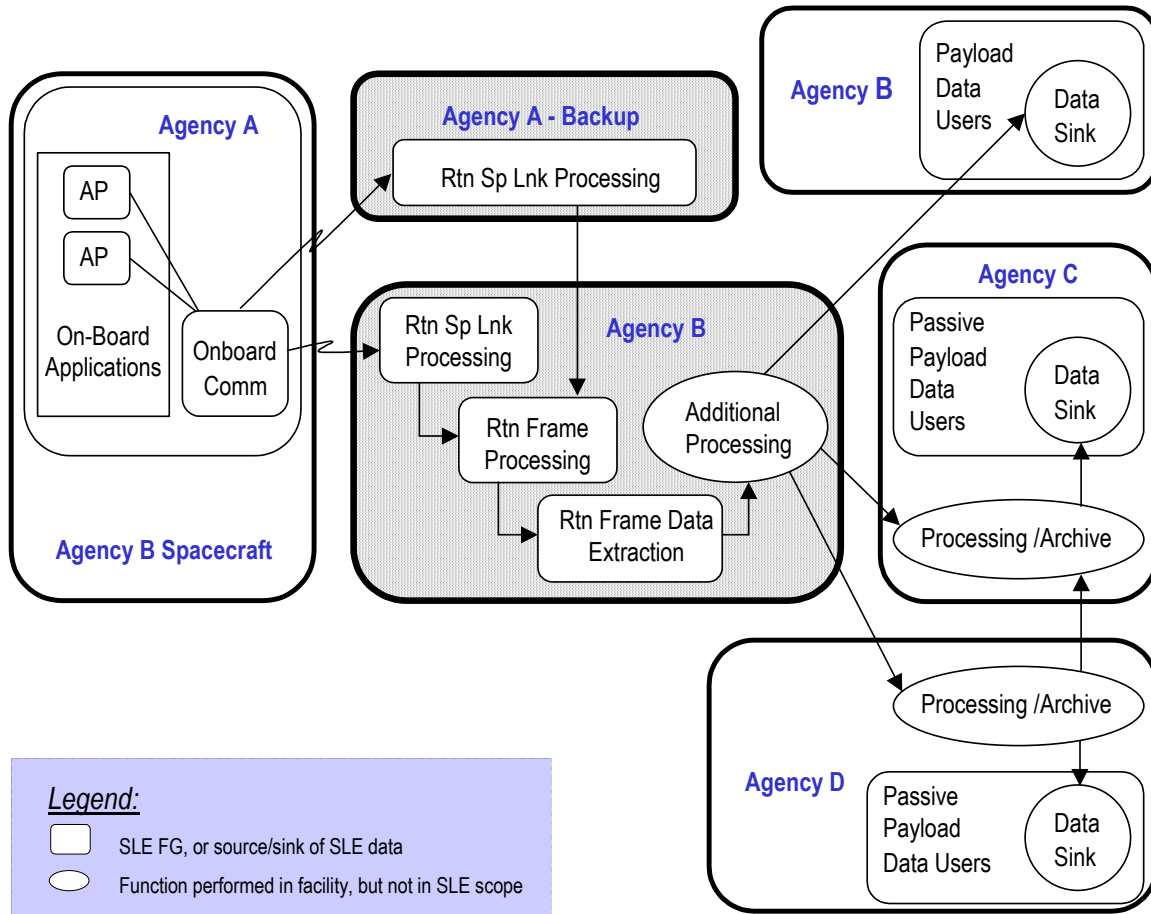


Figure C-4: Cross Support Scenario #1: Multiple User Agencies

Cross Support Scenario #2, shown in figure C-5, illustrates the case in which multiple minimal ‘ground stations’ each send all frames received during a pass to a single Complex. This Complex performs all the remaining return processing and distributes the data to users. Similarly, this Complex accepts forward data from users, processes it, and transmits it to the ground stations for transmission to the mission spacecraft.

In the figure, the spacecraft is shown on the left, the Complexes are in gray, and the MDOS is on the right.

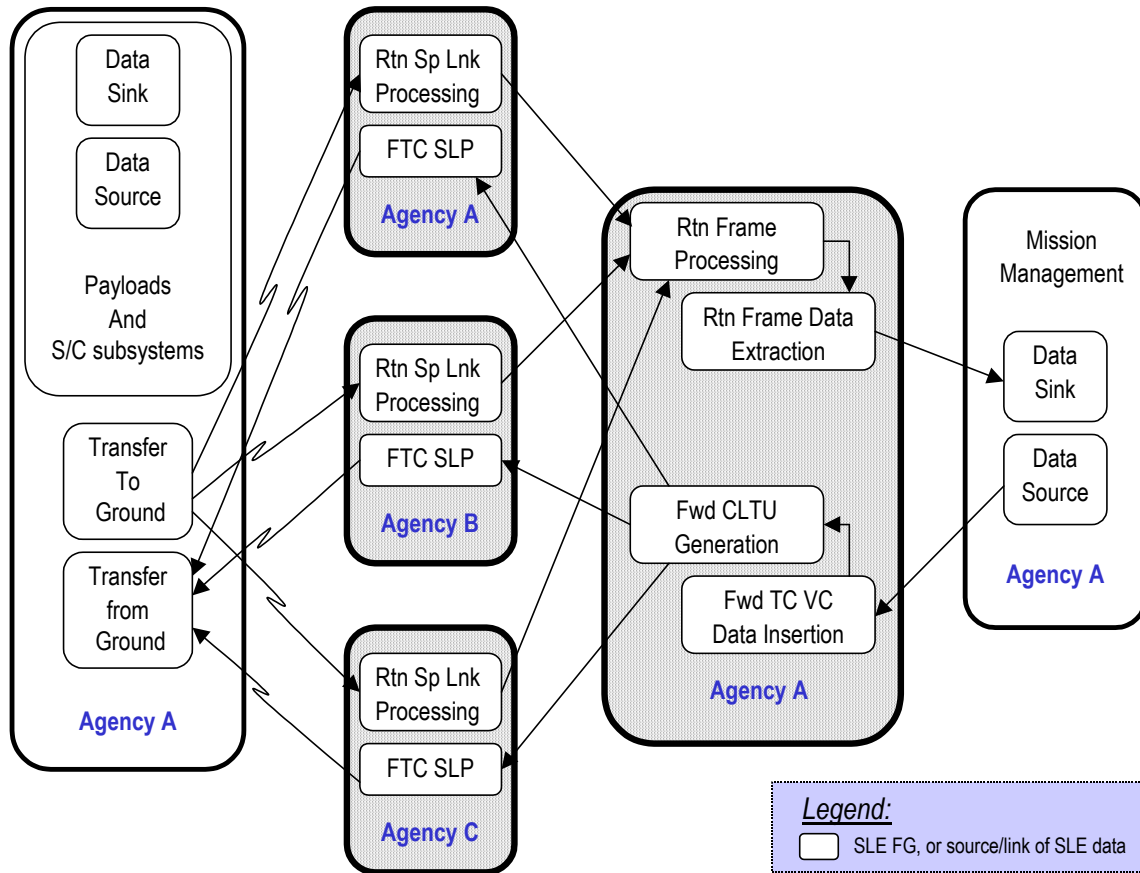


Figure C-5: Cross Support Scenario #2: Multiple ‘Limited Capability’ Ground Stations

In Cross Support Scenario #3, shown in figure C-6, Agency A typically sends data from its spacecraft to a network of ground stations, each of which performs all data processing functions through Return Frame Data Extraction. However, Agency B must use two facilities to process the data to this level. While this does not affect the service interface, it may affect the management interfaces between the agencies.

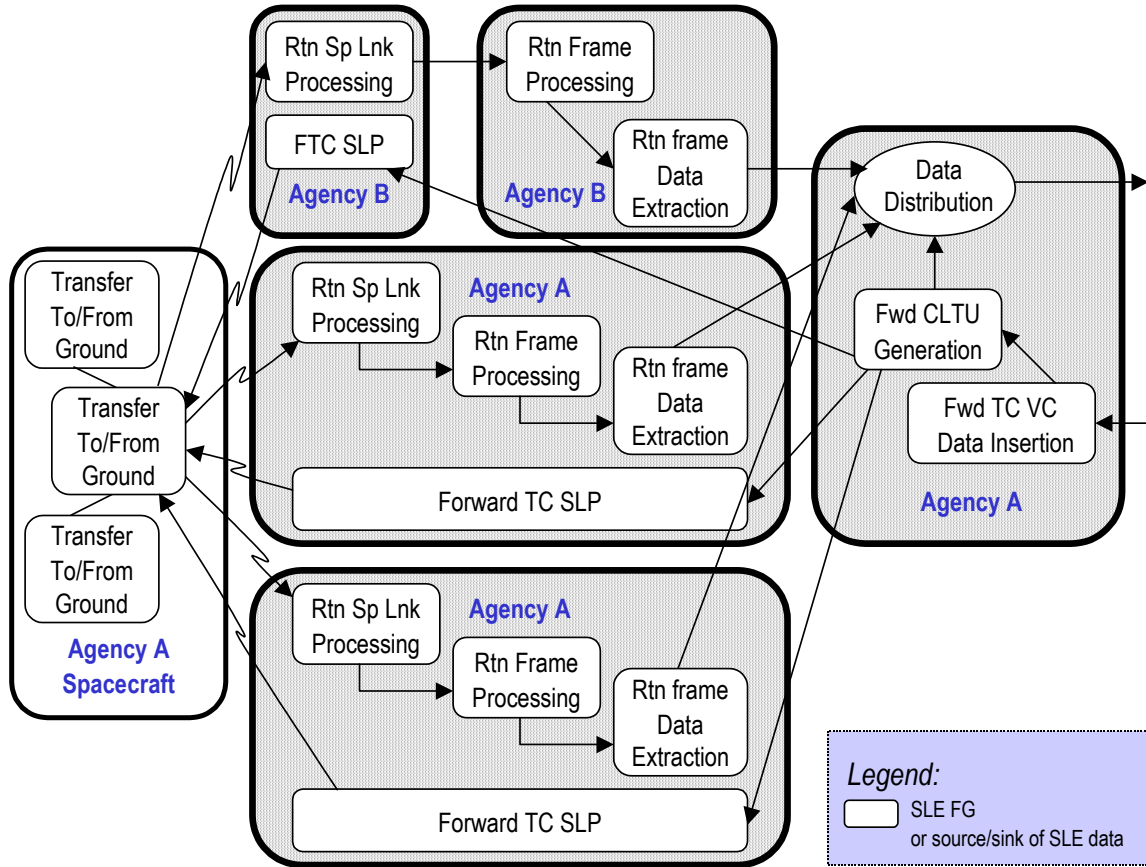


Figure C-6: Cross Support Scenario #3: ‘Extensive’ and ‘Limited’ Capability Ground Stations

In Cross Support Scenario #4, shown in figure C-7, Agency B provides a payload on Agency A's spacecraft. Agency A is responsible for receiving the return data and forwarding processed data for the Agency B payload to Agency B, and also for accepting forward data from Agency B and forwarding it to the spacecraft. Agency B plans to develop its own return data processing capability in the future, and will be able to receive the return data directly in the future. Agency A will remain responsible for the forward link.

The 'mission ground applications' shown in this figure correspond to the SLE MDOS.

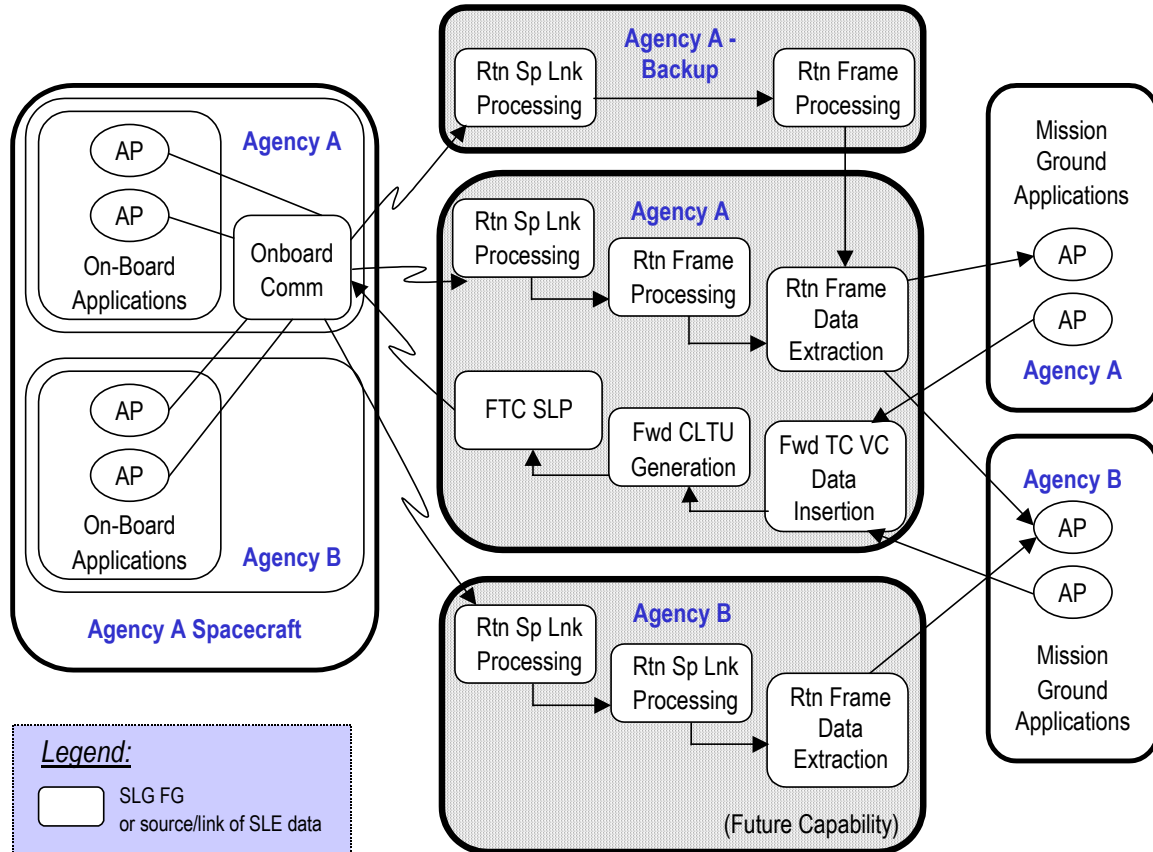


Figure C-7: Cross Support Scenario #4: Multiple 'Extensive' Capability Ground Stations

Cross Support Scenario #5, shown in figure C-8, illustrates Agency B receiving and processing data from the payloads on its spacecraft, including that provided by Agency A. On the ground, Agency B uses Agency A for backup.

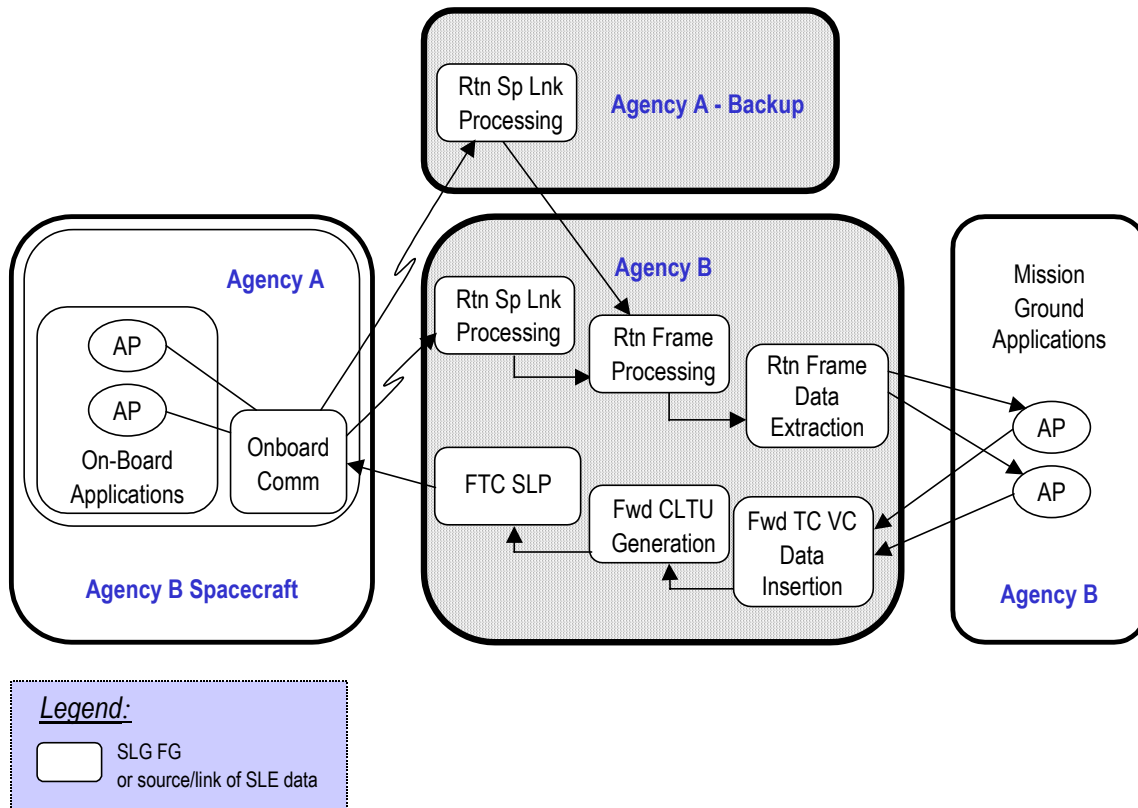


Figure C-8: Cross Support Scenario #5: Simple Cross Support Backup

ANNEX D

SLE SERVICE MANAGEMENT: ANTENNA POINTING SCENARIOS

D1 GENERAL

The basic prerequisite for any space link processing at a ground station is the ability to point an antenna to a location in space where a spacecraft is expected to come into sight.

This, of course, requires that the antenna pointing subsystem in the ground station has timely access to trajectory prediction information for the target spacecraft.

For this purpose, the SLE Service Management MOs include an antennaPointing MO which has been conceived for the general handling of file-based trajectory information.

Before we see step by step what such an antennaPointing MO provides, let us lay out a simple yet typical scenario where handling of trajectory information is vital:

SCENARIO

An SLE Complex is entrusted with acquiring the space link of a spacecraft during launch, early orbit operations, and routine phase.

- The launch is successful but it has slipped for a few minutes.
- Over the next hours, several subsequent orbit adjustment-manoevres establish the target orbit.
- The spacecraft undergoes commissioning and then enters its routine operations phase.

In order to arrive at widely acceptable standards, the authors of the Recommendations have formed an interaction model capturing current practices in several agencies and it is assumed that SLE Utilization Management and SLE Complex Management behave according to the following general pattern:

INTERACTION MODEL

Trajectory information is computed by the spacecraft operations organization, i.e., the MDOS in the abstract terms of the *Cross Support Reference Model* (reference [9]). It predicts the trajectory of the target spacecraft in a ground station independent manner, for a certain time interval, relative to a certain reference system. The computed trajectory information is encoded, packed into a file and transferred in time from the MDOS to the destination SLE Complex Management. There it is deposited in a mission dedicated trajectory file repository. When antenna pointing becomes due, SLE Complex Management decodes the trajectory information, transforms it into some complex-internal reference system and translates it into micro-commands to be executed by the antenna drive.

Now, let us consider the actual antennaPointing managed object, which has been conceived for the generic handling of trajectory information files:

MANAGED OBJECT

AntennaPointing

Attributes:

spacecraft-tracking-mo-id. This attribute holds the name of this particular antenna pointing MO.

trajectory-file-id. By specifying a specific name, SLE Utilization Management selects the operationally applicable trajectory prediction file from the trajectory prediction repository.

time-offset. With this attribute, SLE Utilization Management can specify a time shift (in seconds) which is applied to the trajectory elements when the selected file is used for antenna steering.

tracking-state. This attribute shows the current overall state of the tracking activity within the SLE Complex (e.g. 'ready' for tracking or 'actively' moving the antenna).

tracking-time. This attribute holds the accumulated time intervals during which the antenna was dedicated to the target spacecraft. It may be used for accounting purposes or just for reporting.

Actions:

new trajectory. When called up by SLE Utilization Management, this action forces the antenna to abandon tracking with its current trajectory prediction and switch immediately to a new trajectory prediction file, which has been specified beforehand in the *trajectory-file-id*.

Notifications:

start antenna-pointing activity. This notification is issued when the antenna is positioned to a location in space where the target spacecraft is expected to come into sight.

stop antenna-pointing activity. This notification is issued when the antenna pointing activities for the target spacecraft are terminated.

NOTE – The antennaPointing managed object is not intended to report the actual movement of the antenna; the antennaPointing managed object is strictly confined to handling the trajectory prediction information. Specific ‘ground segment services’ relating to tracking and Doppler measurement will be specified in the future by CCSDS Panel 3.

Now we explore step by step how antenna pointing is performed on the basis of interaction model and the antennaPointing managed object.

D2 STEP 1: ESTABLISH THE INTERACTION FRAMEWORK

The currently available CCSDS Recommendations do not cover all elements of the interaction model and a number of them must be left subject to bilateral agreement between SLE Utilization Management and SLE Complex Management. In a phase, typically one year before the actual launch of the spacecraft, negotiations between mission customer and the service providing SLE Complex take place. Among many other topics, the prerequisite agreements for antenna pointing operations must be settled. These agreements concern the following:

- The specification of a common format for the trajectory prediction files. As long as there are no pertinent CCSDS Recommendations, this format specification covers: the choice of the commonly understood reference system for a trajectory specification, the general layout of the file (header, body) and specification of trajectory information elements (name, data type, engineering unit), and the encoding of trajectory information for transfer. The name which is given to this agreed upon format is laid down in the *trajectory-format* attribute of the serviceAgreement managed object.
- The maximum size of the spacecraft dedicated repository for trajectory information. The agreed maximum size is fixed in the *max-size-trajectory-filestore* attribute of the serviceAgreement managed object.
- The path to the spacecraft dedicated repository for trajectory information. The character string describing the path to the spacecraft dedicated trajectory filestore is fixed in the *trajectory-filestore-id* attribute of the serviceAgreement managed object.
- A protocol for file transfer and file management in the repository (add file, remove file), and security measures for the protection of the trajectory file repository. It is expected that a reference to an existing standard file transfer protocol (e.g., FTP or FTAM) and the agreement about suitable user identifications and passwords is sufficient.

D3 STEP 2: PRODUCE TRAJECTORY PREDICTIONS

Service Management (reference [10]) does not make any assumption about the actual production of trajectory prediction files by the mission organization. In fact, the necessary ground segment services for tracking and ranging have not yet been standardized by CCSDS; hence, ranging and tracking measurements must be acquired in custom formats as done today.

However, the file that will ultimately be shipped to the supporting Complex must adhere to the format conventions as negotiated in step 1 and, of course, SLE Complex Management must be prepared to read and interpret this format correctly.

D4 STEP 3: TRANSFER AND DEPOSIT TRAJECTORY FILES

Using the file transfer protocol, the directory path negotiated, and applying the security measures (step 1), the file is transferred and stored in the directory under a unique name (unique relative to the directory).

If the new file cannot be accommodated in the negotiated space, the transfer is rejected.

Before a retry, directory housekeeping (e.g., remove of outdated files) may be required to free sufficient space. This may be affected via the file transfer (and management) protocol or by other suitable means which are not covered in the SLE Service Management Recommendations.

It shall be noted that several trajectory prediction files may well co-exist in the directory.

D5 STEP 4: SELECT AN OPERATIONALLY APPLICABLE TRAJECTORY FILE FROM THE REPOSITORY

In order to support space link acquisition correctly during the execution of a service package, antenna pointing must be driven by an operationally applicable trajectory prediction. For this, the antennaPointing MO in the relevant servicePackage must be selected and its attribute *trajectory-file-id* must be set to the name of the applicable file.

Whenever SLE Utilization Management sets this attribute, a few tests are performed on the file: is it really deposited?, can it be read?, has it the agreed format and encoding?, does it fit with the time bounds of the service package?, can the antenna move along the desired trajectory arc? If something is discovered wrong with the prediction file the setting is rejected and an indicative diagnostic is returned to SLE Utilization Management. SLE Complex Management has no other choice than using the 'old' file for eventual antenna pointing.

Note that service package validation would fail if the antennaPointing managed object has no—or a bad—trajectory file selected. Therefore, at validation time, the best then-available prediction shall be downloaded and selected.

If the trajectory file is OK, SLE Complex Management will convert it into a Complex-internal format, process it as required, and ensure that this information will be used with the next upcoming antenna pointing activity.

If an antenna pointing activity is going on at the moment when a new trajectory is selected, the new trajectory may be prepared, but it will be actively used only for the next upcoming antenna pointing, i.e., in general for the subsequent spacecraft pass (for the major exception see section D8).

D6 STEP 5: DRIVE THE ANTENNA

So, what does ‘antenna pointing activity’ mean?

For pointing an antenna to a target spacecraft, SLE Complex Management must assign the antenna to that spacecraft and associate the respective antennaPointing managed object with the antenna resource.

Eventually, the antenna begins a timely move into a start position, which has been computed from the trajectory prediction for the time when space link acquisition is supposed to begin. (This time is specified in the attribute *carrier-start-time* of a spaceLinkCarrierPackage managed object and should be taken with a certain safety margin.) When the antenna has reached its starting position, the antennaPointing managed object fires an informative ‘*start antenna pointing activity*’ notification to SLE Utilization Management.

Then, depending on the tracking mode, the antenna moves along the trajectory arc (programmed tracking) or follows the spacecraft, once it is locked on it (auto-tracking).

At *carrier-stop-time* (again in a spaceLinkCarrierPackage MO) or a little later, the antenna stops its movement on behalf of the target spacecraft. At this point in time, the associated antennaPointing MO sends a ‘*stop antenna pointing activity*’ notification to Utilization Management. Thereafter, the antenna may be reassigned to another spacecraft.

D7 STEP 6: ADJUST THE TIME TAG OF A TRAJECTORY PREDICTION

In our scenario, we assume that the launch of the target spacecraft slips for a couple of minutes. This means that the validity time of the launch-trajectory prediction must be adjusted in order to compensate for the delay.

For this, SLE Utilization Management simply sets the attribute *time-offset* in the antennaPointing managed object to the desired corrective value (in seconds). SLE Complex Management must take this correction into account when it eventually converts the selected prediction into antenna specific commands.

If an antenna pointing activity is going on (one can see this from the value of the attribute *tracking-state*), the correction will become effective only at the next pointing exercise (for the big exception see section D8).

D8 STEP 7: ENFORCED SWITCHING BETWEEN TRAJECTORY PREDICTIONS

Following our scenario, we enter the Early Orbit Operations phase. In this phase, one must be prepared for orbit manoeuvres which may fail or perform with certain degradations. Usually, several alternative orbit predictions are prepared for the time after a maneuver. These alternative predictions are transferred into the directory (in fact, that is why this directory can hold several prediction files).

One of the effects of the manoeuvre may have been to move the spacecraft to a new orbit. In order to best track the spacecraft in its new orbit, the antenna must be switched immediately to use the best available file from the prepared trajectory predictions. From the perspective of SLE Utilization Management, i.e., on the level of the antennaPointing managed object, this switching is performed as follows: first, the *trajectory-file-id* attribute is set to the name of the best suited prediction in the repository (of course, the usual checking is performed as in step 4); second, the *time-offset* is adjusted as required; third, the 'new trajectory' action is invoked.

On resource level, this invocation causes the antenna to move immediately (or as fast as possible without breaking apart) from the current position to a new position, which has been calculated on the basis of the new trajectory file and the applicable time-offset correction. Then, the antenna continues the movements with the new trajectory information.

D9 STEP 8: ROUTINE UPDATING OF TRAJECTORY PREDICTIONS

Typically, in the routine phase of spacecraft operations, trajectory predictions are gradually refined.

Such refined trajectory predictions files can be downloaded to the repository as in step 3 and then used for upcoming antenna pointing activities as described in step 4.

D10 STEP 9: DEBRIEFING

As an antenna is a scarce and expensive resource, antenna-time may be the limiting factor for cross support costs; therefore it is important to account for the usage.

For this purpose, the antennaPointing MO accumulates the duration of active periods (i.e., the times of antenna movement on behalf of the target spacecraft) in the attribute *tracking-time*. SLE Utilization Management may inspect this attribute any time during the life of the antennaPointing managed object. It may use the readouts for administrative purposes (e.g., account checking).

ANNEX E

STORAGE FOR SERVICES

E1 GENERAL

Providing offline delivery of SLE data channels requires the reservation of sufficient storage in the SLE Complex that supports a mission. Storage capacity is reserved as part of the SLE Service Agreement through creation of dataStore managed objects (see *Service Management*, reference [10]).

NOTE – The offline delivery mode is used to transfer sets of SLE service data through all or part of the SLE system. This transfer typically occurs outside the associated Space Link session. The online delivery mode is also used to transfer sets of SLE service data through all or part of the SLE system. However, this transfer typically occurs during the associated Space Link session. These delivery modes are discussed in the *Cross Support Reference Model* (reference [9]).

E2 PARTITIONS OF DATA STORES

To provide an instance of offline service, a set of SLE data units from a specific SLE data channel is stored in a partition of a data store. A channelStored managed object identifies a specific partition (e.g., by a file name or path name within the SLE Complex) that is used during an offline service instance. ChannelStored managed objects are contained in all SLE service packages that store or retrieve SLE data channels.

Offline service instances are used either

- a) to prepare for later uplink of a forward SLE data channel to the space element (as an online service instance); or
- b) to retrieve return data that was stored during a Space Link session.

In either case, an offline service instance and its associated online service instance may be provided as part of the same or separate SLE service packages. The storage partition identified by a channelStored managed object links the two service instances.

The partition specified by a channelStored managed object must be consistent with the storage capacity or data rate restrictions at the supporting SLE Complex (see *Service Management*, reference [10]).

E3 FORWARD OFFLINE SLE TRANSFER SERVICE

In the forward direction, an offline service instance is used to transfer SLE data units to an SLE Complex prior to uplinking those data units to the space element. This offline transfer often occurs before the Space Link session begins, but may occur during the Space Link session as well.

Figure E-1, which illustrates an example of forward online service production and provision, provides a point of reference for the discussion of offline service that follows. In the example, an SLE Complex implements the following functions involved in providing forward online service:

- consume the input data channel(s) (from the MDOS or from another SLE Complex);
- perform the processing to produce the resulting data channel; and
- supply that data channel in a continuous flow to the spacecraft or another SLE Complex.

These three functions are common to all online services, but the specific details vary according to the functional group resource performing the functions. For this discussion, the details of the functional group are not necessary.

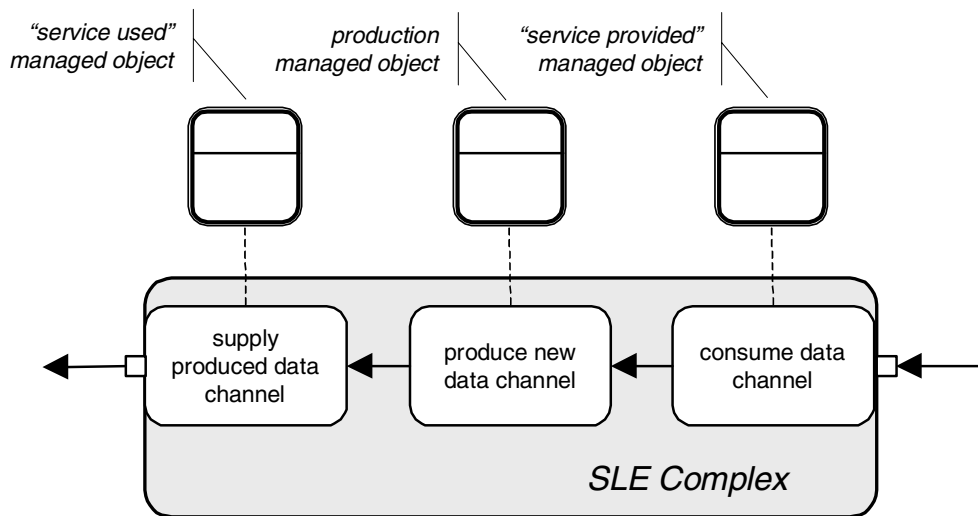


Figure E-1: Online Consumption, Production, and Supply of Forward SLE Data Channels

Figure E-1 also illustrates the management aspect of each of these three functions. For each function, there is a managed object as described in *Service Management*. For the forward SLE services, the management aspects of the functions that consume SLE data channels are encapsulated in ‘service provided’ managed objects, the production functions are represented by production managed objects, and the supply functions are represented by ‘service used’ managed objects. For return SLE services, the management aspects of the functions that consume SLE data channels are encapsulated in ‘service used’ managed objects, the production functions are represented by production managed objects, and the supply functions are represented by ‘service provided’ managed objects.

NOTE – ‘Service used’ and ‘service provided’ are descriptive names used in this discussion. There are no actual managed objects with these names; rather there are a number of managed objects that serve this purpose. Each one is named for its SLE service type.

In contrast with the online case, in offline service production and provision the data flow through the functions terminates in a data store partition. A second data flow exists by which SLE data units are supplied to a recipient after retrieval and in some cases, production processing. Storage of forward SLE data can occur in two ways:

- a) Service production can be performed offline, storing the SLE data units of an SLE data channel in a partition managed through a channelStored managed object, for later uplink to the space element as an online service instance. Figure E-2 is an example of the offline production and storage of forward SLE data units prior to the Space Link session. Figure E-3 illustrates the subsequent retrieval and online supply of stored SLE data units during the Space Link session.

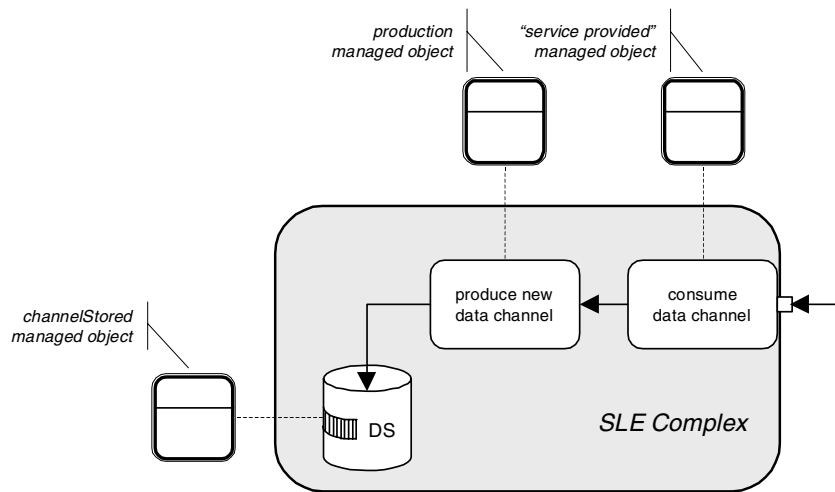


Figure E-2: Consumption, Production, and Storage of Forward Data Units Before Space Link Session

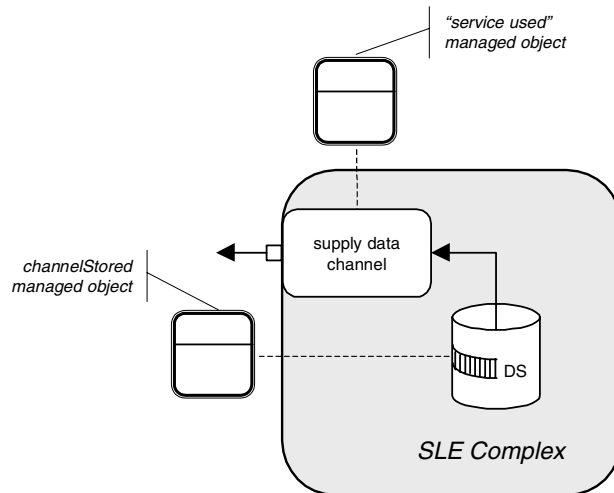


Figure E-3: Retrieval and Supply of Stored Forward Data Units During Space Link Session

- b) Service production can be performed online, using SLE data units that were previously stored to produce an SLE data channel to provide an instance of online service during the Space Link session. Figure E-4 is an example of the offline storage of forward SLE data units for subsequent SLE data channel production. Figure E-5 illustrates the online production and supply of the stored forward SLE data units during the Space Link session.

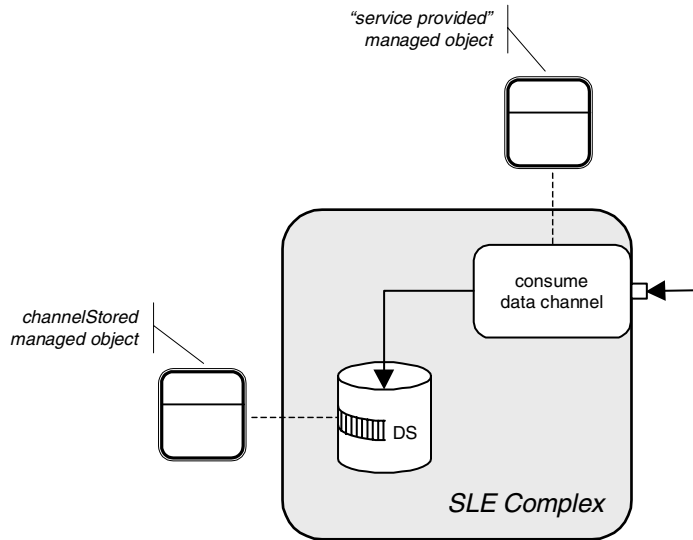


Figure E-4: Storage of Forward Data Units Before Space Link Session

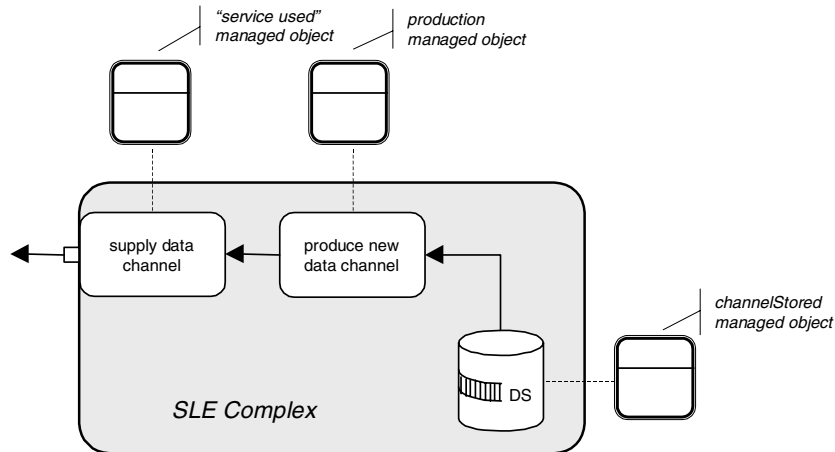


Figure E-5: Retrieval, Production, and Supply of Stored Forward Data Units During Space Link Session

The two stages involved in storage and retrieval of forward data can be combined as shown in figure E-6. Here, SLE data units are appended to a partition in offline mode while additional data units are retrieved from the partition by online service production.

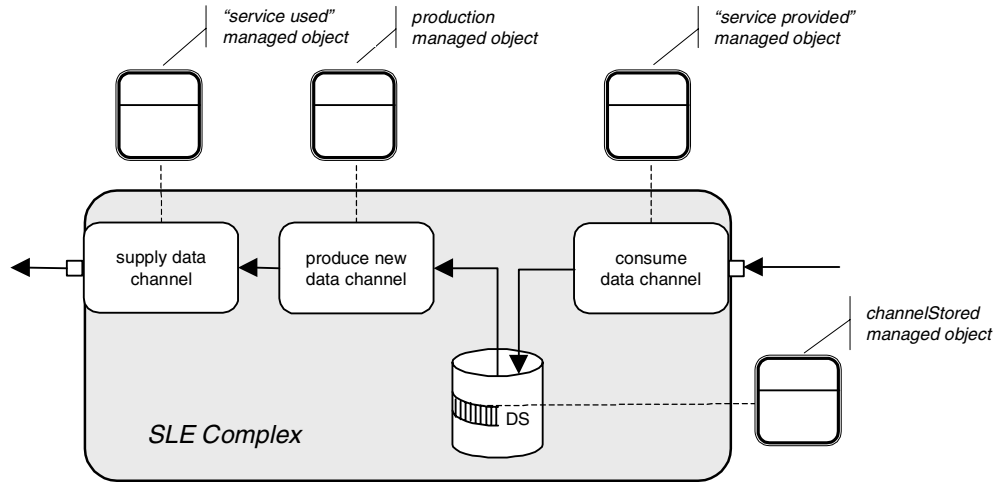


Figure E-6: Combined Storage and Retrieval During Space Link Session.

E4 RETURN OFFLINE SLE TRANSFER SERVICE

In the return direction, an offline service instance is used to transfer SLE data units from an SLE Complex after those data units have been downlinked by the space element and put into the storage resource as part of an online transfer service instance. The data may be retrieved from the storage resource during or after the Space Link session as part of an offline service instance. In other words, data goes into the storage resource as part of an online service instance, but may come out as part of an offline service instance.

Storage of return SLE data can occur in one of two ways:

- a) Service production can be performed online, with the real-time flow of SLE data units of a return SLE data channel as input, and a storage partition as output. This partition is managed through a channelStored managed object and is available to support supply of offline service instances. Figure E-7 is an example of the production and storage of return SLE data units. Figure E-8 illustrates the offline supply of the stored SLE data units. The two stages involved in storage and retrieval of return data can be combined as shown in figure E-9. Here, SLE data units are retrieved from a partition in offline mode while additional data units are appended to the partition by online service production.

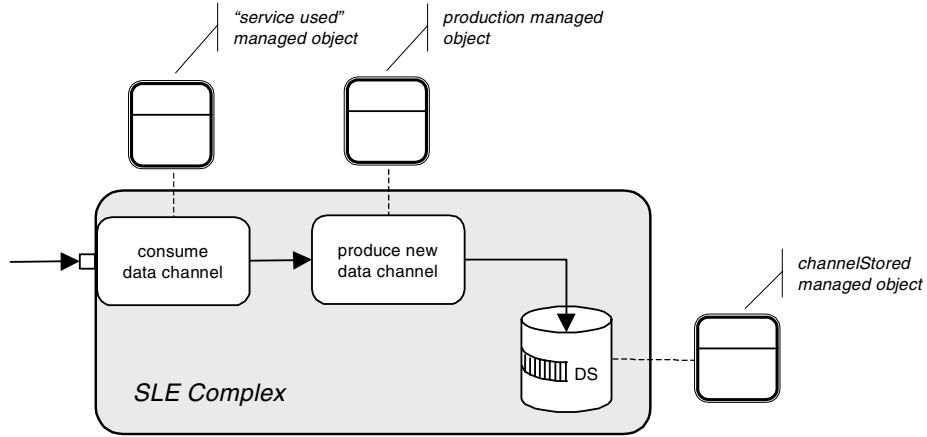


Figure E-7: Production and Storage of Return Data Units During Space Link Session

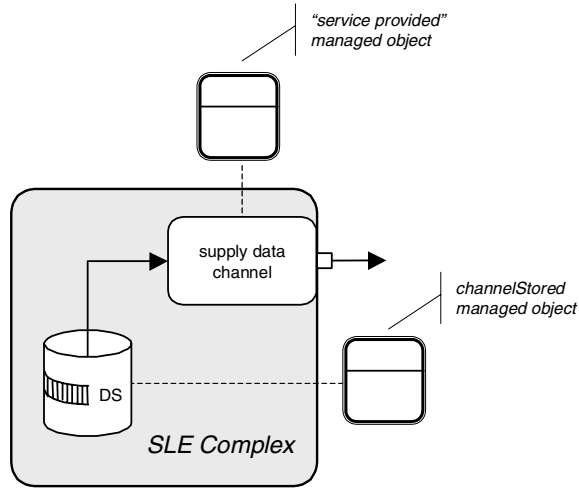


Figure E-8: Retrieval and Supply of Stored Return Data Units After Space Link Session

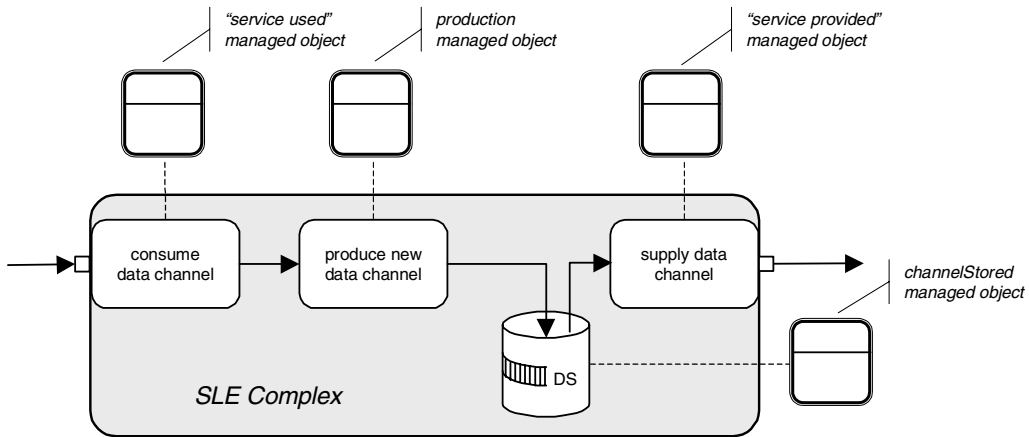


Figure E-9: Combined Retrieval and Appending During Space Link Session.

- b) Service production can be performed offline, using the SLE data units previously stored in a partition as input to produce an SLE data channel to provide one or more instances of offline service. Figure E-10 is an example of storage of return SLE data units provided by another SLE Complex. Figure E-11 illustrates the subsequent retrieval of the stored data units, and production and supply of the derived data channel.

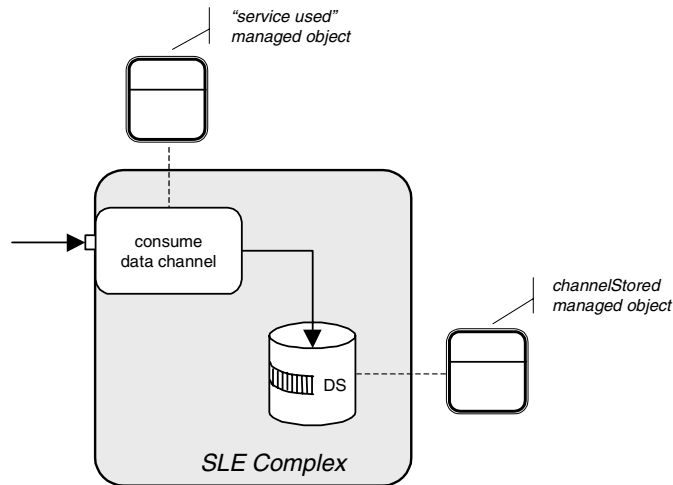


Figure E-10: Storage of Return Data Units During Space Link Session

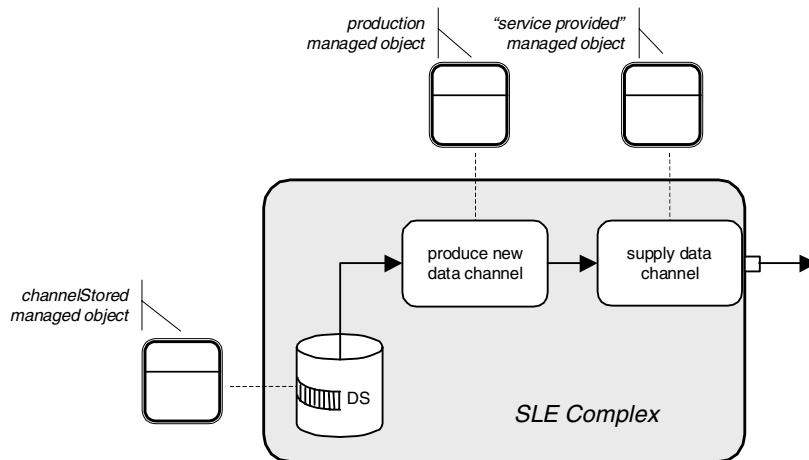


Figure E-11: Retrieval, Production, and Supply of Stored Return Data Units After Space Link Session

E5 EXAMPLE: RETURN OFFLINE SLE TRANSFER SERVICE

This section provides an extended example of the containment relationship among the managed objects associated with two service packages that are part of the same Service Agreement. The first service package configures an SLE Complex to

- consume an RAF data channel from another SLE complex;

- demultiplex VCs 1 and 3;
- produce one or more space packet channels from VC 1;
- supply the space packet channels to an online user;
- store VC 3 for access by a user through the second service package.

The second service package configures the same SLE Complex to

- retrieve VC 3 from the data store;
- supply VC 3 to an offline user.

Figures E-12 and E-13 together provide an example scenario of the production and provision of online and offline return services, and informally introduce the kinds of managed objects that represent the different functions that must be performed.

The service package represented by figure E-12 deals with an RAF data channel acquired via a single service instance from another SLE Complex (not shown). The service package specifies the online provision of VC 1, and the storage of VC 3 for subsequent offline service provision. From the perspective of configuring and managing SLE Complex resources to support the service package, the RAF ‘service used’ managed object represents the functionality that consumes the incoming RAF data channel, and the channel frame production managed object represents the functionality that produces VC 1 and VC 3. VC 1 immediately has its frame data field extracted (as represented by the frame data field production managed object). The frame data fields are further processed to extract the various space packet channels (represented by the space packet production managed object), some or all of which are supplied to a user (the functionality of which is represented by the space packet service provision managed object). VC 3 is stored in a partition of the data store. This partition is represented by a channel frame storage managed object.

The service package represented by figure E-13 provides the user access to the VC 3 data stored in the partition through the operation of the service package of figure E-12. The service package specifies the retrieval of the data from the data store partition (represented by a channel frame storage managed object), and the provision of the VC 3 data (the functionality of which is represented by the channel frame service provision managed object).

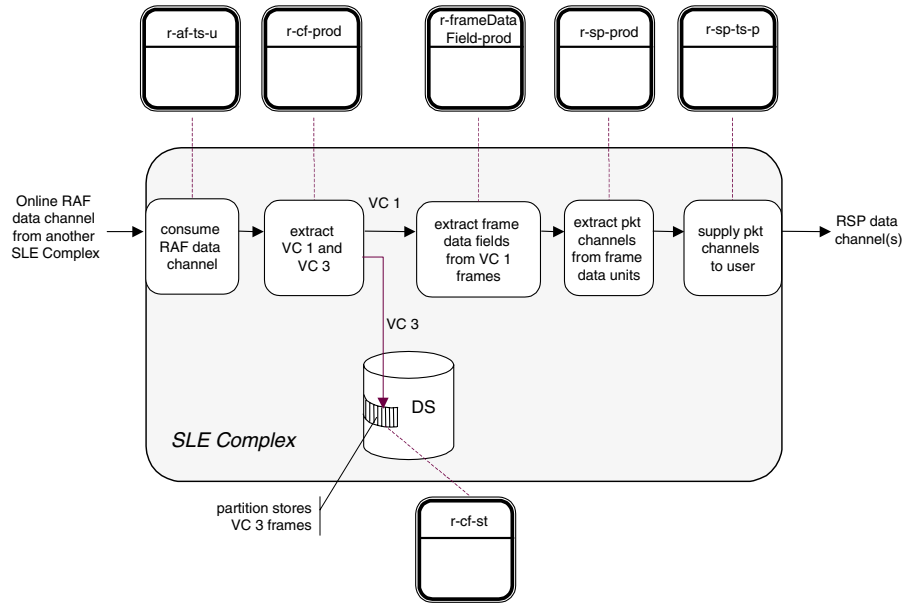


Figure E-12: Example: Online Storage of VC 3 Data Channel and Provision of RSP Data Channels

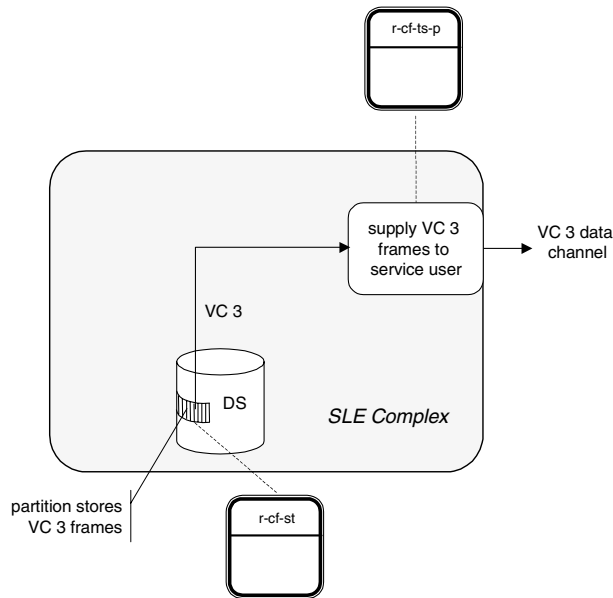


Figure E-13: Example: Retrieval from Storage and Offline Provision of VC 3 Data Channel

Figures E-14 and E-15 depict the actual managed objects associated with the elements of E-12 and E-13, respectively, and show the hierarchical containment relationship among them. The Service Agreement in this example is designated as the x^{th} service agreement between this particular MDOS and SLE Complex. The two service packages in service agreement (x) are designated the n^{th} and the m^{th} service packages. In the following figures, a single 'x' in parentheses indicates that the managed object persists across the lifetime of the service agreement. An 'x' followed by either 'n' or 'm' in parentheses indicates that the managed object exists only during the lifetime of the respective service package.

Figure E-14 illustrates the managed objects involved in the n^{th} service package. The serviceAgreement object contains a dataStore object and multiple servicePackage objects (in the figure, only the servicePackage object associated with the n^{th} service package is shown). The dataStore managed object persists across the lifetime of the service agreement, for it represents the storage capabilities available to the user mission across and between service packages.

The n^{th} servicePackage contains one Return All Frames Transfer Service Used (r-af-ts-u) managed object, which holds information needed by the SLE Complex to manage the interface by which the SLE Complex consumes the input RAF data channel. The r-af-ts-u managed object contains an r-cf-prod managed object.

The r-cf-prod managed object contains a Return Channel Frame Storage (r-cf-st) managed object and a Return Frame Data Field production (r-frameDataField-prod) managed object. The r-cf-prod managed object exists in the n^{th} service package because the production is to be performed online; that is, simultaneously with the consumption of the input RAF channel. The r-cf-st managed object holds information needed to identify the partition of the dataStore that is to receive VC 3, and to manage the act of storing that data.

The r-frameDataField-prod managed object contains the information needed to find the frame data fields within the channel frames and serves as a container for the Return Space Packet Production (r-sp-prod) managed object. The r-sp-prod managed object represents the functionality associated with the extraction of selected space packet channels from VC 1. The r-sp-prod managed object exists in the n^{th} service package because the production is to be performed online; that is, simultaneously with the consumption of the input RAF channel. The r-sp-prod managed object contains a Return Space Packet Transfer Service Provided (r-sp-ts-p) managed object. The r-sp-ts-p managed object holds information needed to provide a selection of space packet channels to the user of the online RSP service instance.

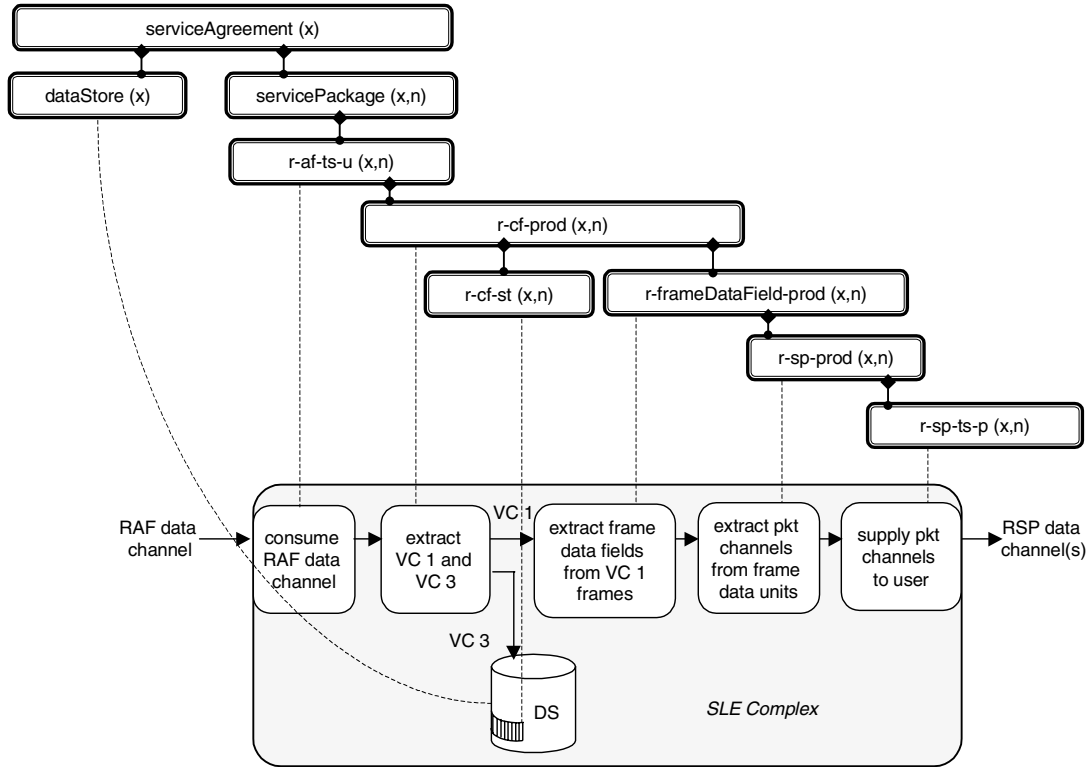


Figure E-14: Containment Example: Online Storage of VC 3 Data Channel and Provision of RSP Data Channels

Figure E-15 illustrates the managed objects involved in the m^{th} service package, in which the user is provided access to the data stored in the partition. In this example, the serviceAgreement and dataStore managed objects are the same, but the servicePackage managed object and all contained managed objects are different because they correspond to a different (i.e., the m^{th}) service package.

The m^{th} servicePackage managed object contains a r-cf-st managed object, which holds information needed to identify the partition of the dataStore that contained the stored VC 3 data units, and to manage the act of retrieving that data.

The r-cf-st MO contains a Return Channel Frame Transfer Service Provided (r-cf-ts-p) managed object. The r-cf-ts-p managed object holds the information necessary to provide the user with the capability to retrieve VC 3 data units from the partition of the dataStore.

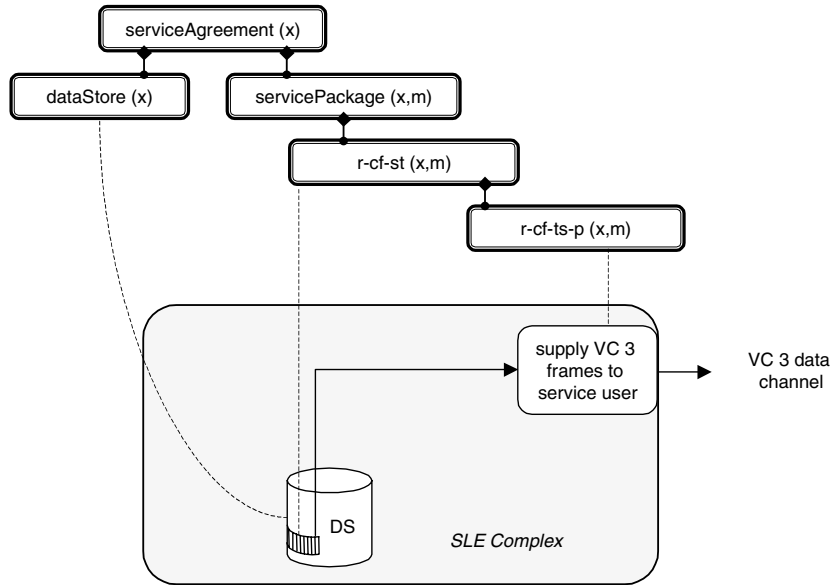


Figure E-15: Containment Example: Retrieval from Storage and Offline Provision of VC 3 Data Channel

ANNEX F

SLE SECURITY MECHANISMS

F1 GENERAL

SLE transfer and management services are provided with two security mechanisms: access control and authentication.

F2 AUTHENTICATION

Authentication is the process of verifying that another SLE entity is indeed the SLE entity that it identifies itself as in operation invocations and returns. SLE transfer and management services provide for rudimentary authentication through the use of credentials. The use of credentials relies on the existence of a ‘secret’ (such as a password) that is shared by two parties (in this case, the two SLE entities), such that the presentation of the secret in a communication from one party to the other authenticates the sending party to the receiver. There are any number of authentication methods (that is, schemes for generating credentials), with the better, more powerful ones offering better protection against the compromise of the secret to a third party.

The ability to completely authenticate a peer SLE entity will most likely involve not only the authentication performed by the SLE entities themselves, but also capabilities provided by the underlying data transfer and management technologies used. Some of these underlying technologies may provide strong authentication capabilities, whereas others may provide little or no such capability. In order to complement the varying degree of authentication provided by underlying technologies, each SLE service instance is specified in the service package to have one of three *authentication levels*: ‘full authentication’, ‘bind authentication’, and ‘no authentication’. When ‘full authentication’ is in force, all SLE operation invocations and returns carry credentials. When ‘bind authentication’ is specified, only the BIND invocation and its return carry credentials. When no authentication is needed at the SLE application level, ‘no authentication’ is specified.

F3 SECURITY-RELATED MANAGED OBJECTS

F3.1 GENERAL

The key data item in the structure of SLE security management information is the SLE entity identifier. The SLE entity identifier has two primary purposes. First and most obvious, it is used to populate the access control list of an SLE entity. Second, the SLE entity identifier is used as the key to management information used to authenticate that identifier. This management information comprises

- a) the authentication level;
- b) the authentication method; and

- c) the values of the security attributes (both secret and non-secret) associated with the authentication method.

SLE entity identifiers exist across multiple service packages associated with a single Service Agreement. The set of SLE entity identifiers is initially defined by SLE Utilization Management in the process of creating the Service Agreement, and identifiers may be added to or deleted from the set during the lifetime of the Service Agreement. The relatively static nature of the SLE entity identifiers allows the definition of security management information that persists across multiple service packages (that is, the security management information does not have to be re-specified for every service package).

Security-relevant information is contained in the Authentication, NoAuthentication, X509SimpleAuthentication, TransferService, and AuthenticationInformationBase (AIB) managed objects. These managed objects are formally defined in *Service Management* (reference [10]), and the following subsections provide a brief summary.

F3.2 Authentication MANAGED OBJECT CLASS

An Authentication managed object contains all information necessary to authenticate, and generate credentials associated with, a given SLE entity identifier. Authentication is itself an abstract class, with derived classes defined for each authentication method. Every SLE entity that is contained by this Complex or that interacts with this Complex has an authentication managed object for it contained by the Security Information Base (SIB). Every derived class of the Authentication class has an authentication-method-unique naming attribute, an *sle-entity-identifier*, the *authentication-mode* associated with the *sle-entity-identifier*, and the set of authentication method attributes associated with the derived class and *sle-entity-identifier*. The number and content of the set of authentication method attributes are specific to the security method, and thus to the particular method-derived class.

SLE entity identifiers exist across multiple service packages associated with a single Service Agreement. The set of *sle-entity-identifiers* is initially defined by SLE Utilization Management in the process of creating the Service Agreement and identifiers may be added to or deleted from the set during the lifetime of the Service Agreement. The relatively static nature of the *sle-entity-identifiers* allows the definition of security management information that persists across multiple service packages (that is, the security management information does not have to be respecified for every service package).

The conceptual similarity between the derived class-unique naming attribute and the *sle-entity-identifier* might lead one to ask whether both are necessary. The naming attribute is necessarily different for each concrete derived class because object naming requires it to be unique. The *sle-entity-identifier* label is necessarily the same across all Authentication managed objects so that searches for an *sle-entity-identifier* can be conducted across all derived class instances of authentication managed objects. It is useful to note that since the value of the *sle-entity-identifier* must be unique across all derived class instances of authentication managed objects contained by an SIB managed object, these values easily meet the less-strict uniqueness requirements for naming attribute values. Thus the same *sle-*

entity-identifier value can be used for the value of the *xxx-mo-id* attribute as well within the same managed object.

F3.3 NoAuthentication MANAGED OBJECT CLASS

NoAuthentication is the concrete derived class of Authentication that is used to identify a legitimate *sle-entity-identifier* for which authentication is not to be performed. More precisely, it identifies an *sle-entity-identifier* for which the credentials are always set to ‘unused’ on the generation side and for which any value for the credentials is accepted on the authenticating (receiving) side. The NoAuthentication class has the *no-authentication-mo-id* naming attribute and the *authentication-level* attribute inherited from Authentication. However, for instances of this derived class the value of *authentication-level* is constrained to ‘no authentication’.

F3.4 X509SimpleAuthentication MANAGED OBJECT CLASS

X509SimpleAuthentication is the concrete derived class of Authentication that is used to specify the management information for an SLE entity identifier that uses the X.509 Simple Authentication Procedure (reference 18). The X509SimpleAuthentication class has the *x509-simple-authentication-mo-id* naming attribute, the *authentication-level* attribute inherited from Authentication, and a *password* attribute. However, for instances of this derived class the value of *authentication-level* is constrained to ‘full authentication’ or ‘bind authentication’.

F3.5 TransferService MANAGED OBJECT CLASS

TransferService (specified in *Service Management*, reference [10]) is the parent class of all ‘Service Used’ and ‘Service Provided’ managed objects for all of the SLE transfer services. The TransferService managed object class has a number of attributes, but the ones that are of interest for security management are the *local-sle-entity-identifier* and the *peer-sle-entity-access-list*. The *local-sle-entity-identifier* is used by the managed object to (1) identify itself to the peer entity and (2) use the appropriate authentication method and attributes. The *peer-sle-entity-access-list* is used by the managed object to verify access privileges to the corresponding peer entity.

F3.6 SecurityInformationBase (SIB) MANAGED OBJECT CLASS

The SIB managed object is a container for various authentication managed objects. It is contained by the serviceAgreement managed object. While the SIB managed object has no attributes of its own, it exists to allow the security information to be grouped under the serviceAgreement managed object in much the same way as spacecraft characteristics, trajectory files, etc.

F4 SECURITY MANAGEMENT INFORMATION IN SLE OPERATION INVOCATIONS AND RETURNS

This subsection describes the use of the security management information in generating and processing SLE operation invocations and returns.

- Section F4.1 presents the process flow for binding when the *authentication-level* is ‘full authentication’ or ‘bind authentication’.
- Section F4.2 presents the process flow for other SLE operations when the *authentication-level* is ‘full authentication’.
- Section F4.3 presents the process flow for binding when the *authentication-level* is ‘no authentication’.
- Section F4.4 presents the process flow for other SLE operations when the *authentication-level* is ‘no authentication’ or ‘bind authentication’.

The processing flows are described in terms of a transfer service instance, but the concepts apply equally to the management interface.

In the following sections, the term ‘xxx-TS-/ MO’ (with a ‘forward slash’) is used to identify an instance, either used or provided, of a particular (but otherwise unspecified) subclass of the TransferService class. The term ‘xxx-TS\ MO’ (with a ‘back slash’) refers to the peer managed object of ‘xxx-TS-/ MO.’

The process flows in sections F4.1 and F4.2 are accompanied by illustrated scenarios.

However, the scenarios depict only the ‘success path’ for the processing flows. That is, the error/exception cases are not addressed.

F4.1 BIND INVOCATION/RETURN WITH FULL OR BIND-ONLY AUTHENTICATION

The following steps describe the process flow for binding when the *authentication-level* is ‘full authentication’ or ‘bind authentication’. Figure F-1 illustrates the BIND invocation portion of this process, and figure F-2 illustrates the return portion. For simplification, figures F-1 and F-2 illustrate the case where the user is the invoker and the provider is the performer of the BIND operation.

F4.1.1 Initiator invokes BIND

- a) Using the *local-sle-entity-identifier* in the associated xxx-TS-/ MO, the Initiator (initiating xxx-TS entity) selects the corresponding authentication managed object.
- b) The Initiator uses the *authentication-level* and set of security attributes in the authentication managed object to generate the *invoker-credentials* for the BIND invocation.

- c) The Initiator sets the *initiator-identifier* in the BIND invocation equal to the *local-sle-entity-identifier* in the xxx-TS-/ MO.
- d) The Initiator sets the *responder-port-identifier* in the BIND invocation equal to the *responder-port-identifier* in the xxx-TS-/ MO.
- e) The Initiator sets the *service-instance-identifier* in the BIND invocation:
 - 1) If the Initiator is the user, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the *service-instance-identifier* attribute of the xxx-TS-/ MO.
 - 2) If the Initiator is the provider, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the naming attribute of the xxx-TS-/ MO.

NOTE 1— Since there is exactly one transfer service provided managed object for each transfer service instance, the name of the service instance is equal to the Distinguished Name (DN) of the managed object that provides that service. This DN is automatically part of the xxx-TS-P managed object and the value is set by management in the *service-instance-identifier* attribute of the xxx-TS-U managed object.

- f) The Initiator invokes the BIND operation on the responder port named by the *responder-port-identifier*.

NOTE 2 – The *responder-port-identifier* allows the Initiator to set up the underlying communication connection with the responder. However, the exact details of how this communication connection is established depend on the communication technology employed. In any case, this underlying connection is assumed to exist unless and until it is terminated as a result of an UNBIND or PEER-ABORT.

F4.1.2 Responder processes the BIND invocation:

- a) The Responder ‘listens’ at the responder port identified in the *responder-port-identifier* attribute of the Responder’s xxx-TS-\ managed object for a BIND invocation with the *service-instance-identifier* parameter equal to the *service-instance-identifier* found in the in the xxx-TS-\ managed object:
 - 1) If the Responder is the user, the value of the *service-instance-identifier* is found in the *service-instance-identifier* attribute of the xxx-TS-\ managed object.
 - 2) If the Initiator is the provider, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the naming attribute of the xxx-TS-\ managed object.
- b) Using the *initiator-identifier* in the BIND invocation, the Responder (responding xxx-TS entity) accesses the corresponding authentication managed object.

- c) If a match for the *initiator-identifier* exists in the *sle-entity-identifier* attribute of one of the local authentication managed objects, the Responder uses the *authentication-level* and set of security-attributes in the managed object to authenticate the *invoker-credentials* in the invocation.
 - If a match for the *initiator-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Responder remains unbound and returns a negative result with *diagnostic* value ‘access denied’ and *performer-credentials* set to ‘unused’.
- d) If authentication is successful, the Responder verifies that the *initiator-identifier* is contained in the Responder’s xxx-TS-\ managed object *peer-sle-entity-access-list*.
 - If authentication is not successful, the Responder ignores the invocation and remains unbound.
- e) If the *initiator-identifier* is contained in the Responder’s xxx-TS-\ managed object *peer-sle-entity-access-list*, the Responder (1) stores the *initiator-identifier* as the *peer-entity-identifier* for use in subsequent authentication and/or access privilege verification, (2) executes the BIND, and (3) returns a positive result (if the BIND is successful).
 - If the *initiator-identifier* is not contained in the *peer-sle-entity-access-list*, the Responder remains unbound and returns a negative result with *diagnostic* value ‘service instance not accessible to this initiator’.

F4.1.3 Responder returns a positive result:

- a) Responder uses the *local-sle-entity-identifier* from the Responder’s xxx-TS-\ managed object to select the appropriate authentication managed object.
- b) Responder generates the *performer-credentials* for the return using the *security-attributes* from the authentication managed object and (possibly) the *local-sle-entity-identifier* in the Responder’s xxx-TS-\ managed object.

NOTE 1 – Under the scheme described here, it is technically possible for two paired SLE entities to have used different authentication modes, although it seems that in almost all cases matched modes would be desirable.

- c) The Responder sets the *responder-identifier* in the invocation set equal to the *local-sle-entity-identifier* in the xxx-TS-\ managed object.
- d) Responder dispatches the BIND return.

NOTE 2 – Consistent with the assumption stated in a previous Note, the underlying communication connection ‘knows’ how to get the BIND return to the initiator.

F4.1.4 Initiator processes the BIND return:

NOTE 1 – It is assumed here that the checks for expected data units are performed prior to authentication and access control.

Positive result:

- a) Using the *responder-identifier* in the BIND return, the Initiator accesses the corresponding authentication managed object.
- b) If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of one of the local authentication managed objects, the Initiator uses the *authentication-level* and set of *security-attributes* in the managed object to authenticate the *performer-credentials* in the return.
 - If a match for the *responder-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Initiator invokes a PEER-ABORT with *diagnostic* ‘access denied’.
- c) If authentication is successful, the Initiator verifies that the *responder-identifier* is contained in the Initiator’s xxx-TS-/ managed object's *peer-sle-entity-access-list*.
 - If authentication is not successful, the Initiator ignores the return.

NOTE 2 – If the Initiator does not receive an authenticatable BIND return within a specified time, the Initiator invokes a PEER-ABORT with *diagnostic* ‘time-out’.

- d) If the *responder-identifier* is contained in the Initiator’s xxx-TS-/ managed object *peer-sle-entity-access-list*, the Initiator (1) stores the *responder-identifier* as the *peer-entity-identifier* for use in subsequent authentication and/or access privilege verification, and (2) proceeds with the association established.
 - If the *responder-identifier* is not contained in the *peer-sle-entity-access-list*, the Initiator invokes a PEER-ABORT with *diagnostic* ‘unexpected responder ID’.

Negative result:

- a) If the value of the *diagnostic* is ‘access denied’:

NOTE 3 – In this case, the *performer-credentials* have been set to ‘unused’, so authentication is not attempted

- 1) If the *responder-identifier* is contained in the Initiator's xxx-TS-/ managed object *peer-sle-entity-access-list*, the Initiator remains unbound and ceases waiting for a BIND return. The Initiator may attempt to re-invoke the BIND.
- 2) If the *responder-identifier* is not contained in the Initiator’s xxx-TS-/ managed object's *peer-sle-entity-access-list*, the Initiator invokes a PEER-ABORT with *diagnostic* ‘unexpected responder ID’. The Initiator may attempt to re-invoke the BIND.

- b) If the value of the *diagnostic* is not ‘access denied’:
- 1) Using the *responder-identifier* in the BIND return, the Initiator accesses the corresponding authentication managed object.
 - 2) If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of one of the local authentication managed objects, the Initiator uses the *authentication-level* and set of *security-attributes* in the managed object to authenticate the *performer-credentials* in the return.
 - If a match for the *responder-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Initiator invokes a PEER-ABORT with *diagnostic* ‘access denied’.
 - 3) If authentication is successful, the Initiator verifies that the *responder-identifier* is contained in the Initiator’s xxx-TS-/ managed object's *peer-sle-entity-access-list*.
 - If authentication is not successful, the Initiator ignores the return.

NOTE 4 – If the Initiator does not receive an authenticatable BIND return within a specified time, the Initiator invokes a PEER-ABORT with *diagnostic* ‘time-out’.

- 4) If the *responder-identifier* is contained in the Initiator’s xxx-TS-/ managed object’s *peer-sle-entity-access-list*, the Initiator remains unbound and ceases waiting for a BIND return.
 - If the *responder-identifier* is not contained in the *peer-sle-entity-access-list*, the Initiator invokes a PEER-ABORT with *diagnostic* ‘unexpected responder ID’.

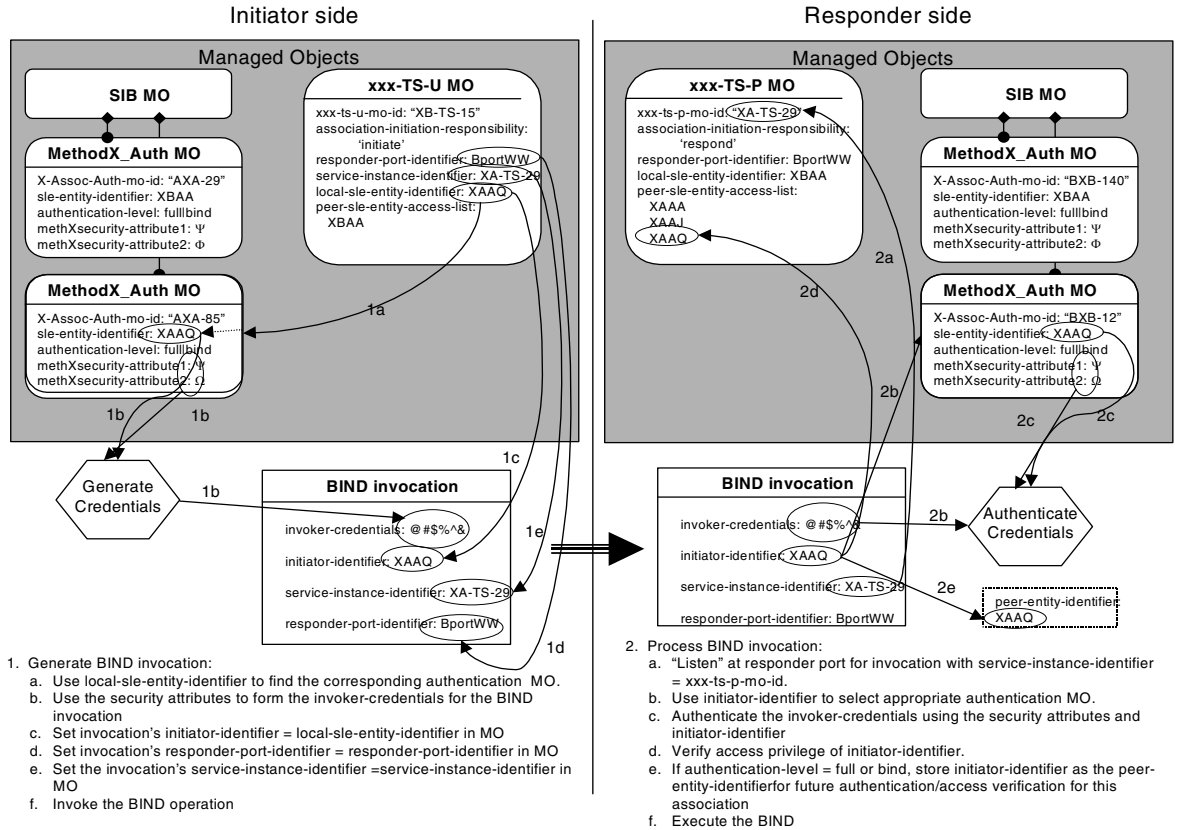


Figure F-1: BIND Invocation with Full or Bind-Only Authentication

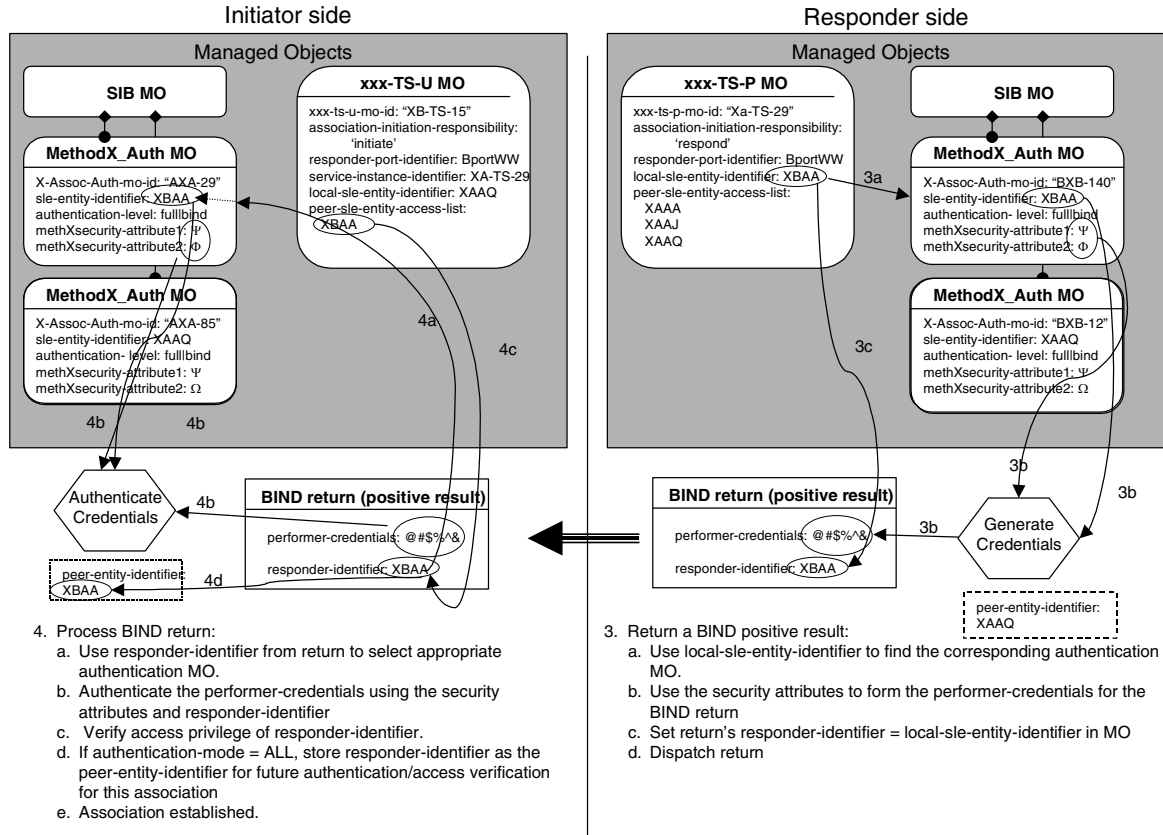


Figure F-2: BIND Return with Full or Bind-Only Authentication

F4.2 XXX-OPERATION INVOCATION/RETURN WITH FULL AUTHENTICATION

The following steps describe the process flow for performing an arbitrary (designated as 'XXX') SLE operation when the *authentication-level* is 'full authentication'. Figure F-3 illustrates the BIND invocation portion of this process, and figure F-4 illustrates the return portion.

F4.2.1 Invoker invokes XXX-operation

- Using the *local-sle-entity-identifier* in the associated xxx-TS-/ managed object, the Invoker (invoking xxx-TS entity) selects the corresponding authentication managed object.
- The Invoker uses the *authentication-mode* and set of *security-attributes* in the authentication managed object to generate the *invoker-credentials* for the XXX invocation.
- The Invoker invokes the XXX operation on the performer.

F4.2.2 Performer processes the XXX invocation:

- a) Using the locally-stored *peer-entity-identifier*, the Performer (performing xxx-TS entity) accesses the corresponding authentication managed object.
- b) The Performer uses the *authentication-mode* and set of *security-attributes* in the authentication managed object to authenticate the *invoker-credentials* in the invocation.
- c) If authentication is successful, the Performer executes the operation and (if successful) returns a positive result:
 - If authentication is not successful, the Performer ignores the invocation.

NOTE – If authentication is successful, access privilege is assumed because of successful access verification during the BIND

F4.2.3 Performer returns a positive result:

- a) Performer uses the *local-sle-entity-identifier* from the Performer's xxx-TS-\ managed object to select the appropriate authentication managed object.
- b) Performer generates the *performer-credentials* for the return using the *security-attributes* from the authentication managed object and (possibly) the *local-sle-entity-identifier* in the Performer's xxx-TS-\ managed object.
- c) Responder dispatches the XXX return.

F4.2.4 Invoker processes the XXX return:

NOTE 1 – It is assumed here that the checks for expected data units are performed prior to authentication and access control.

- a) Using the locally-stored *peer-entity-identifier*, the Initiator accesses the corresponding authentication managed object.
- b) The Invoker uses the *authentication-mode* and set of *security-attributes* in the managed object to authenticate the *performer-credentials* in the return.
- c) If authentication is successful, the Invoker accepts the positive or negative result conveyed by the operation.
 - If authentication is not successful, the Invoker ignores the return.

NOTE 2 - If the Invoker does not receive an authenticatable XXX return within a specified time, the Invoker invokes a PEER-ABORT with *diagnostic* 'time-out'.

NOTE 3 - If authentication is successful, access privilege is assumed because of successful access verification during the BIND.

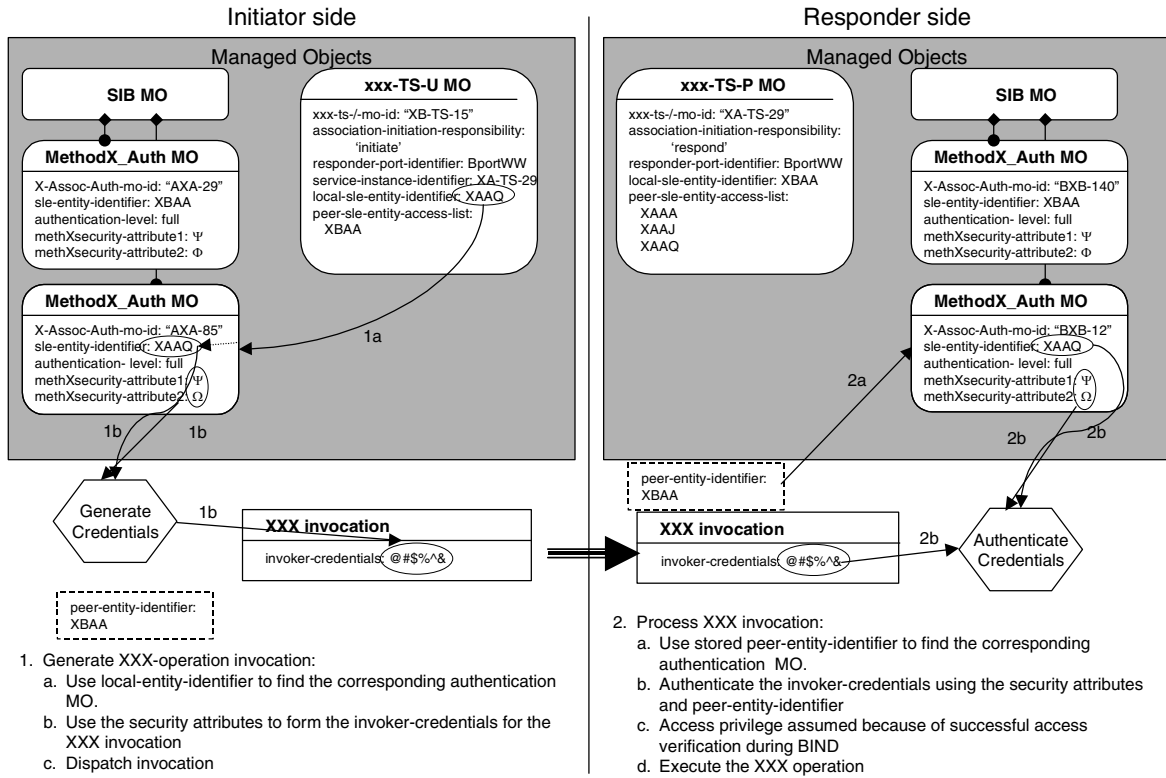


Figure F-3: Subsequent XXX-Operation Invocation with Full Authentication

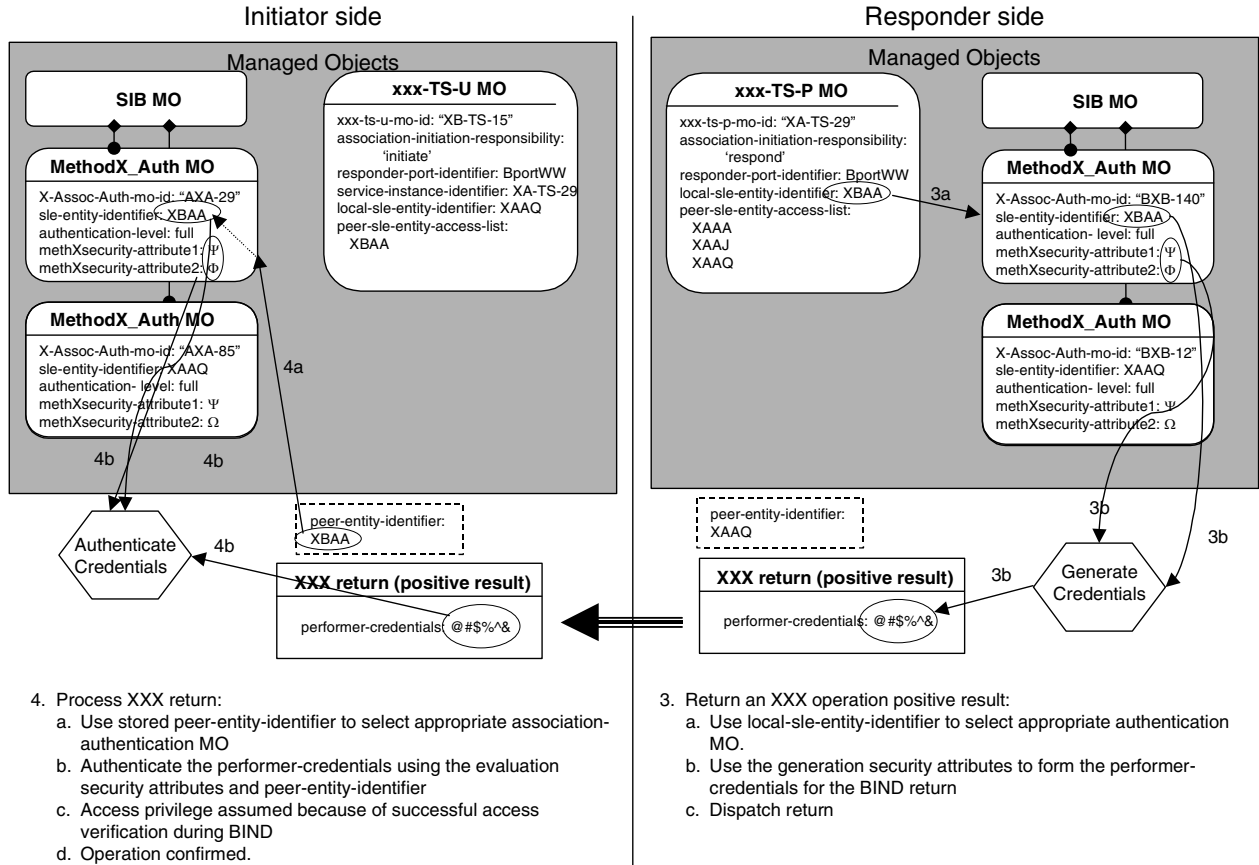


Figure F-4: Subsequent XXX-Operation Return with Full Authentication

F4.3 BIND INVOCATION AND RETURN WITH NO AUTHENTICATION

The following steps describe the process flow for binding when the *authentication-level* is 'no authentication'.

F4.3.1 Initiator invokes BIND

- Using the *local-sle-entity-identifier* in the associated xxx-TS-/ managed object, the Initiator (initiating xxx-TS entity) selects the corresponding authentication managed object, which in this case is a noAuthentication managed object.
- The Initiator uses the *authentication-level* ('no authentication') to set the *invoker-credentials* to 'unused' for the BIND invocation.
- The Initiator sets the *initiator-identifier* in the invocation equal to the *local-sle-entity-identifier* in the xxx-TS-/ managed object.
- The Initiator sets the *responder-port-identifier* in the BIND invocation equal to the *responder-port-identifier* in the xxx-TS-/ managed object.
- The Initiator sets the *service-instance-identifier* in the BIND invocation:

- 1) If the Initiator is the user, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the *service-instance-identifier* attribute of the xxx-TS-/ managed object.
- 2) If the Initiator is the provider, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the naming attribute of the xxx-TS-/ managed object.

NOTE 1 – Since there is exactly one transfer service provided managed object for each transfer service instance, the name of the service instance is equal to the DN of the managed object that provides that service. This DN is automatically part of the xxx-TS-P managed object and the value is set by management in the *service-instance-identifier* attribute of the xxx-TS-U managed object.

- f) The Initiator invokes the BIND operation on the responder port named by the *responder-port-identifier*.

NOTE 2 – The *responder-port-identifier* allows the Initiator to set up the underlying communication connection with the responder. However, the exact details of how this communication connection is established depend on the communication technology employed. In any case, this underlying connection is assumed to exist unless and until it is terminated as a result of an UNBIND or PEER-ABORT.

F4.3.2 Responder processes the BIND invocation:

- a) The Responder ‘listens’ at the responder port identified in the *responder-port-identifier* attribute of the Responder’s xxx-TS-\ managed object for a BIND invocation with the *service-instance-identifier* parameter equal to the *service-instance-identifier* found in the in the xxx-TS-\ managed object:
 - 1) If the Responder is the user, the value of the *service-instance-identifier* is found in the *service-instance-identifier* attribute of the xxx-TS-\ managed object.
 - 2) If the Initiator is the provider, the value of the *service-instance-identifier* parameter in the BIND invocation is set to the value of the naming attribute of the xxx-TS-\ managed object.
- b) Using the *initiator-identifier* in the BIND invocation, the Responder (responding xxx-TS entity) accesses the corresponding authentication managed object, which in this case should be a noAuthentication managed object.
 - If a match for the *initiator-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Responder remains unbound and returns a negative result with *diagnostic* value ‘access denied’ and *performer-credentials* set to ‘unused’.
- c) If a match for the *initiator-identifier* exists in the *sle-entity-identifier* attribute of one of the local noAuthentication managed objects, the Responder uses the

authentication-level ('no authentication') to determine that authentication is not required.

- If a match for the *initiator-identifier* exists in the *sle-entity-identifier* attribute of a local authentication managed object that is not a noAuthentication managed object, the Responder attempts to authenticate the credentials using the *authentication-method* associated with the managed object class. Since the credentials have the value 'unused', the authentication fails and the Responder ignores the invocation and remains unbound. Eventually, the Initiator will time-out waiting for a valid return and PEER-ABORT with *diagnostic* value 'time out'.
- d) If the *initiator-identifier* is contained in the Responder's xxx-TS-\ managed object *peer-sle-entity-access-list*, the Responder (1) stores the *initiator-identifier* as the *peer-entity-identifier* for use in subsequent authentication and/or access privilege verification, (2) executes the BIND, and (3) returns a positive result (if the BIND is successful).
 - If the *initiator-identifier* is not contained in the *peer-sle-entity-access-list*, the Responder remains unbound and returns a negative result with *diagnostic* value 'service instance not accessible to this initiator'.

F4.3.3 Responder returns a positive result:

- a) Responder uses the *local-sle-entity-identifier* from the Responder's xxx-TS-\ managed object to select the appropriate authentication managed object, which in this case is a noAuthentication managed object.
- b) Responder sets the *performer-credentials* to 'unused'.

NOTE – It is technically possible for two paired SLE entities to have used different *authentication-modes*, but the assumption in this scenario (and possibly a good general rule) is that the *authentication-modes* match – in this case, both are 'no authentication'.

- c) The Responder sets the *responder-identifier* in the invocation set equal to the *local-sle-entity-identifier* in the xxx-TS-\ managed object.
- d) Responder dispatches the BIND return.

Initiator processes the BIND return:

NOTE 1 – It is assumed here that the checks for expected data units are performed prior to authentication and access control.

Positive result:

- a) Using the *responder-identifier* in the BIND return, the Initiator accesses the corresponding authentication managed object, which in this case should be a noAuthentication managed object.

- If a match for the *responder-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Initiator invokes a PEER-ABORT with *diagnostic* ‘access denied’.
- b) If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of one of the local noAuthentication managed object, the Initiator uses the *authentication-level* (‘no authentication’) to determine that authentication is not required.
 - If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of a local authentication managed object that is not a noAuthentication managed object, the Initiator attempts to authenticate the credentials using the *authentication-method* associated with the managed object class. Since the credentials have the value ‘unused’, the authentication fails and the Initiator ignores the return. Eventually, the Initiator will time-out waiting for a valid return and PEER-ABORT with *diagnostic* value ‘time out’.
- c) If the *responder-identifier* is contained in the Initiator’s xxx-TS-/ managed object *peer-sle-entity-access-list*, the Initiator (1) stores the *responder-identifier* as the *peer-entity-identifier* for use in subsequent authentication and/or access privilege verification, and (2) proceeds with the association established.
 - If the *responder-identifier* is not contained in the *peer-sle-entity-access-list*, the Initiator invokes a PEER-ABORT with *diagnostic* ‘unexpected responder ID’.

Negative result:

- a) If the value of the diagnostic is ‘access denied’

NOTE 2 – In this case, the *performer-credentials* have been set to ‘unused’, and so authentication is not attempted):

- 1) If the *responder-identifier* is contained in the Initiator's xxx-TS-/ managed object *peer-sle-entity-access-list*, the Initiator remains in the unbound state and ceases waiting for a BIND return. The Initiator may attempt to re-invoke the BIND.
- 2) If the *responder-identifier* is not contained in the Initiator's xxx-TS-/ managed object *peer-sle-entity-access-list*, Initiator invokes a PEER-ABORT with *diagnostic* ‘unexpected responder ID’. The Initiator may attempt to re-invoke the BIND.
- b) If the value of the *diagnostic* is not ‘access denied’:
 - 1) Using the *responder-identifier* in the BIND return, the Initiator accesses the corresponding authentication managed object, which in this case should be a noAuthentication managed object.
 - If a match for the *responder-identifier* does not exist in the *sle-entity-identifier* attribute of any of the local authentication managed objects, the Initiator invokes a PEER-ABORT with *diagnostic* ‘access denied’.

- 2) If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of one of the local authentication managed objects, the Initiator uses the *authentication-level* ('no authentication') to determine that authentication is not required.
 - If a match for the *responder-identifier* exists in the *sle-entity-identifier* attribute of a local authentication managed object that is not a noAuthentication managed object, the Initiator attempts to authenticate the credentials using the *authentication-method* associated with the managed object class. Since the credentials have the value 'unused', the authentication fails and the Initiator ignores the return. Eventually, the Initiator will time-out waiting for a valid return and PEER-ABORT with *diagnostic* value 'time out'.
- 3) If the *responder-identifier* is contained in the Initiator's xxx-TS-/ managed object *peer-sle-entity-access-list*, the Initiator remains unbound and ceases waiting for a BIND return.
 - If the *responder-identifier* is not contained in the *peer-sle-entity-access-list*, the Initiator invokes a PEER-ABORT with *diagnostic* 'unexpected responder ID'.

F4.4 XXX-OPERATION INVOCATION/RETURN WITH NO OR BIND_ONLY AUTHENTICATION

The following steps describe the process flow for performing an arbitrary (designated as 'XXX') SLE operation when the *authentication-level* is 'no authentication' or 'bind authentication'.

Invoker invokes XXX-operation

- a) Using the *local-sle-entity-identifier* in the associated xxx-TS-/ managed object, the Invoker (invoking xxx-TS entity) selects the corresponding authentication managed object.
- b) The Invoker uses the *authentication-level* ('no authentication' or 'bind authentication') in the authentication managed object to set the *invoker-credentials* to 'unused' for the XXX invocation.
- c) The Invoker invokes the XXX operation on the performer.

F4.4.1 Performer processes the XXX invocation:

- a) Using the locally-stored *peer-entity-identifier*, the Performer (performing xxx-TS entity) accesses the corresponding authentication managed object.
- b) The Performer uses the *authentication-mode* ('no authentication' or 'bind authentication') to bypass authentication of the *invoker-credentials* in the invocation.
- c) The Performer executes the operation and (if successful) returns a positive result.

F4.4.2 Performer returns a positive result:

- a) Performer uses the *local-sle-entity-identifier* from the Performer's xxx-TS-\ managed object to select the appropriate authentication managed object.
- b) Performer sets the *performer-credentials* to 'unused' for the return based on the *authentication-level* of the authentication managed object.
- c) Responder dispatches the XXX return.

F4.4.3 Invoker processes the XXX return:

NOTE – It is assumed here that the checks for expected data units are performed prior to authentication and access control.

- a) Using the locally-stored *peer-entity-identifier*, the Initiator accesses the corresponding authentication managed object.
- b) The Invoker uses the *authentication-level* ('no authentication' or 'bind authentication') to bypass authentication of the *performer-credentials* in the return.
- c) The Invoker accepts the positive or negative result conveyed by the operation.