

Report Concerning Space Data System Standards

|  |
| --- |
| OPEN ARCHIVAL  INFORMATION SYSTEM  INTEROPERABILITY  FRAMEWORK (OAIS-IF) Rationale, Scenarios, and Requirements |

Draft Informational Report

CCSDS 000.0-G-0

Draft Green Book

October 2021

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: | Draft Green Book, Issue 0 |  |
|  | Date: | July 2021 |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS INFORMATIONAL REPORT IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)**

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

CCSDS Secretariat

Space Communications and Navigation Office, 7L70

Space Operations Mission Directorate

NASA Headquarters

Washington, DC 20546-0001, USA

FOREWORD

[Foreword text specific to this document goes here. The text below is boilerplate.]

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This document is therefore subject to CCSDS document management and change control procedures which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). 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.
* Canadian Space Agency (CSA)/Canada.
* Centre National d’Etudes Spatiales (CNES)/France.
* China National Space Administration (CNSA)/People’s Republic of China.
* Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
* European Space Agency (ESA)/Europe.
* Federal Space Agency (FSA)/Russian Federation.
* Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
* Japan Aerospace Exploration Agency (JAXA)/Japan.
* National Aeronautics and Space Administration (NASA)/USA.
* UK Space Agency/United Kingdom.

Observer Agencies

* Austrian Space Agency (ASA)/Austria.
* Belgian Federal Science Policy Office (BFSPO)/Belgium.
* Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
* China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
* Chinese Academy of Sciences (CAS)/China.
* Chinese Academy of Space Technology (CAST)/China.
* Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
* Danish National Space Center (DNSC)/Denmark.
* Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
* European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
* European Telecommunications Satellite Organization (EUTELSAT)/Europe.
* Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
* Hellenic National Space Committee (HNSC)/Greece.
* Indian Space Research Organization (ISRO)/India.
* Institute of Space Research (IKI)/Russian Federation.
* KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
* Korea Aerospace Research Institute (KARI)/Korea.
* Ministry of Communications (MOC)/Israel.
* National Institute of Information and Communications Technology (NICT)/Japan.
* National Oceanic and Atmospheric Administration (NOAA)/USA.
* National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
* National Space Organization (NSPO)/Chinese Taipei.
* Naval Center for Space Technology (NCST)/USA.
* Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
* South African National Space Agency (SANSA)/Republic of South Africa.
* Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
* Swedish Space Corporation (SSC)/Sweden.
* Swiss Space Office (SSO)/Switzerland.
* United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Document** | | **Title and Issue** | | **Date** | | **Status** | |
| CCSDS 000.0-G-0 | | OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0 | | January 2021 | | First draft | |
| CCSDS 000.0-G-0 | | OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0 | | May 2021 | | Updated for consistency | |
| CCSDS 000.0-G-0 | | OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0 | | August 2021 | | Updated for consistency with revised concepts | |
| CCSDS 000.0-G-0 | | OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0 | | October 2021 | | Reorganised and added ideas about protocols | |
| CCSDS 000.0-G-0 | | OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0 | | June 2022 | | Reorganised and added ideas about protocols | |

CONTENTS

Contents

[DOCUMENT CONTROL iv](#_Toc106715928)

[CONTENTS v](#_Toc106715929)

[1 Introduction 1](#_Toc106715930)

[1.1 Purpose 1](#_Toc106715931)

[1.2 Scope 1](#_Toc106715932)

[1.3 Applicability 1](#_Toc106715933)

[1.4 Rationale 2](#_Toc106715934)

[1.5 Document Structure 3](#_Toc106715935)

[1.6 Definitions 3](#_Toc106715936)

[1.6.1 Acronyms and abbreviations 3](#_Toc106715937)

[1.6.2 Terminology 5](#_Toc106715938)

[1.6.2.1 Glossary 5](#_Toc106715939)

[1.7 References 8](#_Toc106715940)

[2 OVERVIEW 9](#_Toc106715941)

[2.1 Current situation 9](#_Toc106715942)

[2.2 What OAIS-IF is designed to make possible 10](#_Toc106715943)

[2.3 Example implementation of automatically usable Representation Information for multidisciplinary work 13](#_Toc106715944)

[2.3.1 Example of a Simple Specific Adapter for file-type servers 16](#_Toc106715945)

[3 OAIS-IF Scenarios 17](#_Toc106715946)

[3.1 Information Producer sends Information to an OAIS ConformaNt Archive 19](#_Toc106715947)

[3.2 Information PRODUCER sends Information to a Non-OAIS conformant archive 20](#_Toc106715948)

[3.3 OAIS conformant archive sends information to an Information USER 20](#_Toc106715949)

[3.3.1 An INFORMATION USER wants to obtain information from an OAIS conformant ARCHIVE 21](#_Toc106715950)

[3.3.2 An INFORMATION USER wants to obtain information from a NON-OAIS conformant ARCHIVE 21](#_Toc106715951)

[3.3.3 AN INFORMATION User wishes to get an AIP 21](#_Toc106715952)

[3.3.4 AN INFORMATION User wishes to get information derived from an AIP 21](#_Toc106715953)

[3.3.5 If required, INFORMATION users are authenticated and authorized 21](#_Toc106715954)

[3.3.6 Information is transferred as one or more Information Packages 22](#_Toc106715955)

[3.4 Non-OAIS conformant archive sends information to an information USER 22](#_Toc106715956)

[3.5 OAIS conformant archive exchanges information WITH another OAIS CONFORMANT Archive 22](#_Toc106715957)

[3.6 OAIS conformant archive exchanges information with a non-OAIS Archive 22](#_Toc106715958)

[3.7 Non-OAIS conformant archive exchanges information with another non-OAIS Archive 22](#_Toc106715959)

[3.8 Information Creator sends Information TO INFORMATION Producer 22](#_Toc106715960)

[3.9 Possible RepInfo Use Cases 23](#_Toc106715961)

[3.10 Possible Provenance Use Cases 24](#_Toc106715962)

[3.11 Search 24](#_Toc106715963)

[4 OAIS-IF Requirements 25](#_Toc106715964)

[5 More detailed Considerations 26](#_Toc106715965)

[5.1 Basic OAIS Interfaces 26](#_Toc106715966)

[5.2 supplements needed to OAIS IM interfaces 27](#_Toc106715967)

[5.3 Functionality for Interoperability 27](#_Toc106715968)

[5.4 Communication protocols 28](#_Toc106715969)

[5.4.1 REST communications 28](#_Toc106715970)

[5.4.2 Possible JSON examples 29](#_Toc106715971)

[6 OAIS-IF document Structure 30](#_Toc106715972)

[6.1 OAIS Document Tree 30](#_Toc106715973)

[6.2 OAIS-IF Rationale, Scenarios, and Requirements Document [GREEN]    (This document) 32](#_Toc106715974)

[6.3 Architecture Description Document (ADD) [MAGENTA] 32](#_Toc106715975)

[6.4 OAIS-IF GENERIC ADAPTER [BLUE] 32](#_Toc106715976)

[6.5 OAIS-IF ARCHIVE Adapters [MAGENTA] 33](#_Toc106715977)

[6.6 OAIS-IF Java binding [Blue Book] 33](#_Toc106715978)

[6.7 OAIS-IF XXX binding [Blue Book] 33](#_Toc106715979)

[6.8 OAIS-IF Communication Protocol binding [Blue Book] 33](#_Toc106715980)

[6.9 OAIS-IF SPECIFIC Adapter for XXX [Blue Book] 33](#_Toc106715981)

[6.10 Producer-Archive Interoperability Protocol (PAIP) [Blue Book] 33](#_Toc106715982)

[6.11 CAIS/ PRODUCER-ARCHIVE INTEROPERABILITY PROTOCOL (CAIP) [Blue Book] 34](#_Toc106715983)

[7 ANNEX DESIGN CONSIDERATIONS \_ TO BE DELETED WHEN GENERIC ADAPTER BB IS COMPLETED 35](#_Toc106715984)

[7.1 Assumptions 35](#_Toc106715985)

[7.2 Negotiation for Usability 36](#_Toc106715986)

[7.2.1 Getting enough RepInfo 36](#_Toc106715987)

[7.3 Design ideas 37](#_Toc106715988)

[7.4 (Arbitrary) Decisions 37](#_Toc106715989)

[8 UML diagrams 40](#_Toc106715990)

[8.1 USE CASES 40](#_Toc106715991)

[8.2 SEQUENCE DIAGRAM 41](#_Toc106715993)

[8.2.1 info.oais.interfaces.infomodel.repinfo 43](#_Toc106715994)

[8.2.2 Info.oais.interfaces.infomodel.repinfo. possRepInfoNetwork 44](#_Toc106715995)

[8.2.3 info.oais.interfaces.oaisif 45](#_Toc106715996)

[8.3 COMPONENT DIAGRAM 47](#_Toc106715997)

[8.4 DEPLOYMENT DIAGRAM 47](#_Toc106715998)

[9 References 49](#_Toc106716098)

**Table of Figures**

[Figure 2‑1 Connections between clients and archives currently 9](#_Toc106716099)

[Figure 2‑2 General use of OAIS-IF 10](#_Toc106716100)

[Figure 2‑3 Connections between clients and archives using OAIS-IF 11](#_Toc106716101)

[Figure 2‑4 Expanded view of use of OAIS-IF 12](#_Toc106716102)

[Figure 2‑5 Multi-archive example 14](#_Toc106716103)

[Figure 2‑6 Indication of functionality in adapters 15](#_Toc106716104)

[Figure 3‑1 Roles of entities exchanging information 17](#_Toc106716105)

[Figure 3‑2 OAIS Information Package 18](#_Toc106716106)

[Figure 3‑3 Detailed view of OAIS Archival Information Package (AIP) 19](#_Toc106716107)

[Figure 3‑4 Fundamental types of RepInfo 23](#_Toc106716108)

[Figure 6‑1: NOTIONAL OAIS/OAIS-IF Documentation Tree 31](#_Toc106716109)

[Figure 8‑2 Sequence diagram 41](#_Toc106716110)

[Figure 8‑6 RepInfo grouping interfaces 43](#_Toc106716111)

[Figure 8‑7 Directed Graph Interfaces 44](#_Toc106716112)

[Figure 8‑8 OAIS-IF specific interfaces 46](#_Toc106716113)

[Figure 8‑9 Component diagram 47](#_Toc106716114)

[Figure 8‑10 Deployment diagram 48](#_Toc106716115)

# Introduction

**THIS DOCUMENT :**

* **REFERS TO, AND USES, TERMS AND FIGURES FROM, THE UPDATED VERSION OF THE OAIS REFERENCE MODEL (currently CCSDS 650.0-P-2.1)**
* **CONTAINS LINKS TO OTHER DOCUMENTS ON THE WEB, IN ORDER TO KEEP THE INFORMATION AVAILABLE. EVENTUALLY THESE LINKS WILL BE DELETED.**

## Purpose

The purpose of this document is to describe the rationale, with motivating scenarios, requirements and the document tree for the CCSDS and International Organization for Standardization (ISO) Open Archival Information System Interoperability Framework (OAIS-IF). It is a supplement to the overarching OAIS [1] Reference Model standard.

The purpose of the set of standards which make up OAIS-IF is to define interfaces and services which go beyond the exchange of data (as in communications standards) and instead will enable the exchange of information.

The difference may be understood as follows. Following OAIS, data, and in particular digital data, which OAIS calls a Digital Object, is simply an object composed of a set of bit sequences. Bit sequences by themselves could mean anything. On the other hand, Information is *any type of knowledge that can be exchanged. In an exchange, it is represented by data*. It is made up of the Data Object plus what OAIS refers to as Representation Information.

The aim is to improve interoperability between users and archives (the lower case on “archives” denotes both OAIS conformant Archives and other archives) by transferring Information rather than just Data, and to enable the increased usability of archived information of all types across all domains.

## Scope

The OAIS-IF adds (beyond the OAIS Reference Model) capabilities for system interoperability between more general Users (including Consumers and Producers) and archives, whether or not the archives are fully OAIS Reference Model conformant. The OAIS-IF is designed to support the OAIS Reference Model as the first priority, but does not preclude support non-OAIS archives for which the OAIS-IF capabilities are a good fit.

OAIS-IF does not define what should happen within an archive or user systems except as needed for an interoperable interface. The scope is to describe how information is transferred between users and archives in such a way that the information is understandable, as far as possible, also, if required, can be accompanied by evidence about its Authenticity, and everything needed for preservation (per guidance in the OAIS Reference Model).

## Applicability

OAIS-IF should be applicable to any archive and any user. For an OAIS compliant archive, OAIS-IF should enable the archive to make not only complete Archive Information Packages (AIPs) available but also all of the individual components of its AIPs separately available. A non-OAIS compliant archive may not have all the components required for an AIP available but OAIS-IF can still be used in a way limited by what the archive can provide.

We should describe what we mean by interoperability.

Interoperability is defined in various ways. A useful one is:

* the ability of computer systems or software to exchange and make use of information.
* "interoperability between devices made by different manufacturers“ (from Oxford Languages) and
* the degree to which two products, programs, etc.. can be used together, or the quality of being able to be used together: (Cambridge dictionary).

We can distinguish between:

1. Rendered objects i.e. things we normally display (documents, images) or play (videos, sounds) can only be combined in a human’s head – unless they are treated as data (see next).
2. Data i.e. things which are NOT normally rendered, can be combined in a computer

We will be focussing on DATA here because we are largely interested in computer processing, but recognising that human intervention is needed.

The simplest goal statement for OAIS-IF interoperability is to provide a standardized way for archives so that users (designated community or public) can access and use information from many archives with the same software. Archive-to-archive exchanges and transfers will be enhanced by using the same methods.

## Rationale

The rationale and vision for OAIS-IF is that it will enable at least the following.

* A common mechanism for users (including Producers and Consumers) of OAIS Archives when accessing many diverse kinds of archives through the OAIS-IF.
* An efficient standardized way for archives to exchange information between themselves using the same standardized OAIS-IF interfaces.
* Given broad acceptance of OAIS-IF in the OAIS community, a better chance that long-term preservation will work because future generations can easily find the interfacing resources (plug-ins, etc.) that can be used to access legacy archives.
* Enhanced capabilities for cross-discipline research when many different disciplines use the same interface. OAIS-IF will enable users to access and interpret preserved data which is normally outside of their discipline and Designated Community, if the Archive or other sources make the needed Representation Information available via OAIS-IF channels.

Digitally encoded information is very important for individuals, organisations and societies. The OAIS Reference Model standard tells us how to preserve that information so that it will continue to be usable in the long term by a Designated Community. However, each Archive will have different Designated Communities. Many users of such information will not be members of any Designated Community. In that case, while their ability to understand the information is not guaranteed, OAIS and OAIS-IF still support their ability to access it, with as much Representation Information as is available either from the archive itself or from other sources.

Moreover, not all archives are OAIS conformant (as defined in the OAIS Reference Model section 1.4), yet they too contain important information. While archives that are not OAIS Reference Model conformant were not the drivers of OAIS-IF development, OAIS-IF should still be able to provide significant interoperability for most of them.

Those users within one domain, such as Astronomy, tend to have common software tools and common terminology, algorithms etc., and so are likely to be able to understand and use data from various sources, and in particular from other Astronomical archives. A person outside that community will have much greater difficulty in using such data.

The ability of such other non-OAIS conformant sources to provide additional Representation Information, (and Preservation Description Information) via OAIS-IF technology may provide such users with the possibility of greater understanding of, and confidence in, the preserved data. That, of course, depends on such other sources providing adequate RepInfo. This is envisioned to serve both users in the public, and researchers that are studying cross-domain topics.

OAIS-IF seeks to make the information from different domains more easily usable for a broader range of customers. It will be of general benefit if all can exchange, understand and use information from all domains; it will also be of benefit to archives by increasing the use of their holdings. The OAIS-IF set of documents are designed for use by archives, information creators and users.

OAIS-IF does not seek to specify how an archive preserves information, nor how information should be created and encoded, nor what software users must employ when using information. Instead OAIS-IF defines how information should be exchanged to ensure that information will be usable. In addition, OAIS-IF allows Preservation Description Information (PDI) to be exchanged, and in particular Provenance Information, with which a user can judge Authenticity.

## Document Structure

Section 2 provides a brief overview of OAIS-IF and its uses. A number of scenarios are described in section 3; in each scenario issues, avoiding repetition, are identified. Based on these scenarios, and other considerations, a number of requirements are listed in section 4, with cross-references to the scenarios, where appropriate. More detailed considerations are provided in section 5, looking at the basic design of the OAIS-IF interface methods, what functionality is needed for interoperability, and a default way in which information can be exchanged over the network. Further details of the set of books which define the OAIS-IF are provided in section 6 onwards.

## Definitions

### Acronyms and abbreviations

TO BE COMPLETED

**AIP** Archival Information Package

**CCSDS** Consultative Committee for Space Data Systems

**DEDSL** Data Entity Specification Language

**DIP** Dissemination Information Package

**FITS** Flexible Image Transport System

**GIS** Geographic Information System

**ISO** International Organization for Standardization

**OAIS** Open Archival Information System

**PDI** Preservation Description Information

**SIP** Submission Information Package

**TEI** Text Encoding Initiative

**UML** Unified Modeling Language

**XML** Extensible Markup Language

AIC

AIP

API

CAIP

CAIS

CCSDS

CMS

CSV

DAI

DIP

DO

DTN

EO

FITS

HTML

HPPT

HTTP

ID

IM

ISBN

ISO

JSON

MIME

OAIS

OAIS-IF

OAIS-PF

OAIS RM

OAL

PAIP

PAIS

PDI

PDS

PDU

PF

REQ

REST

RI

RIN

RM

SIP

SOAP

S/W

SW

TCP/IP

UML

URI

UTF

WG

XML

### Terminology

Digital preservation interests a range of different communities, each with a distinct vocabulary and local definitions for key terms. A glossary is included in this document, but it is important to draw attention to the usage of several key terms.

In general, key terms in this document have been adopted from the OAIS Reference Model (add document number or link here?). One of the great strengths of the OAIS Reference Model has been to provide a common terminology made up of terms ‘not already overloaded with meaning so as to reduce conveying unintended meanings’ (reference [1]). Because the OAIS has become a foundational document for digital preservation, the common terms are well understood and are therefore used within this document.

The OAIS Reference Model uses ‘Archive’ to mean the organization responsible for digital preservation. In this document, the term ‘repository’ or phrase ‘digital repository’ is used to convey the same concept in all instances except when quoting from the OAIS Reference Model (reference [1]). It is important to understand that in all instances in this document, ‘repository’ and ‘digital repository’ are used to convey digital repositories and archives that have, or contribute to, long-term preservation responsibilities and functionality. This document uses the OAIS concept of the ‘Designated Community’. A repository may have a single, generalized ‘Designated Community’ (e.g., every citizen of a country), while other repositories may have several, distinct user communities with highly specialized needs, each requiring different functionalities or support from the repository; this document uses the term Designated Community to cover this second case also.

#### Glossary

Unless otherwise indicated, other definitions are taken from the OAIS Reference Model (reference [1]) and are provided here for convenience.

**Access Policy:** Documented statement, authorized by the repository management, that describes the approach to be taken by the repository for providing access to objects accessioned into the repository. The Access Policy may distinguish between different types of access rights, for example between system administrators, members of the Designated Community, and general users.

**Archival Information Package (AIP)**: An Information Package, consisting of the Content Information and the associated Preservation Description Information (PDI), which is preserved within an OAIS.

**Authenticity**: The degree to which a person (or system) regards an object as what it is purported to be. Authenticity is judged on the basis of evidence.

**Consumer**:The role played by those persons, or client systems, who interact with OAIS services to find preserved information of interest and to access that information in whatever level of detail is allowed. In addition to the normally expected entities outside the OAIS, this can also include other OAISes, as well as internal OAIS persons or systems.

**Data Object**:Either a Physical Object or a Digital Object.

**Digital Object**:An objectcomposed of a set of bit sequences.

**Dissemination Information Package (DIP)**:An Information Package, derived from one or more AIPs, and sent by Archives to the Consumer in response to a request to the OAIS.

**Fixity Information**:The information which documents the mechanisms that ensure that the Content Data Object has not been altered in an undocumented manner.

**Identifier**: An identifier is a name that identifies (that is, labels the identity of) either a unique object or a unique class of objects, where the "object" or class may be an idea, physical countable object (or class thereof), or physical noncountable substance (or class thereof).

**Information**:Any type of knowledge that can be exchanged. In an exchange, it is represented by data.

NOTE – An example of Information is a string of bits (the data) accompanied by a description of how to interpret the string of bits as numbers representing temperature observations measured in degrees Celsius (the Representation Information).

**Information Object**:A Data Object together with its Representation Information.

**Information Package**:A logical container composed of optional Information Object(s). Associated with this Information Package is Packaging Information used to delimit and identify the Information Object and optional Package Description information used to facilitate searches for the Information Object.

**Other Representation Information:** A type of Representation Information which cannot easily be classified as Structure Representation Information or Semantic Representation Information. It is a type of Information Object.

NOTE – For example, software, algorithms, encryption, written instructions and many other things may be needed to understand the Content Data Object in ways exemplified by the Preservation Objectives, all of which therefore would be, by definition, Representation Information, yet would not obviously be either Structure Representation Information or Semantic Representation. Information defining how the Structure Representation Information and the Semantic Representation Information relate to each other, or software needed to process a database file would also be regarded as Other Representation Information.

**Packaging Information**:The information that describes how the components of an Information Package are logically or physically bound together and how to identify and extract the components. It is a type of Information Object.

**Physical Object**:An object (such as a moon rock, bio-specimen, microscope slide) with physically observable properties that represent information that is considered suitable for being adequately documented for preservation, distribution, and independent usage.

**Preservation Description Information (PDI**): Theinformation, which along with Representation Information, is necessary for adequate preservation of the Content Data Object and which can be categorized as Provenance Information, Context Information, Reference Information, Fixity Information, and Access Rights Information. It is a type of Information Object.

NOTE – Defining PDI (as well as its components: Provenance Information, Context Information, Reference Information, Fixity Information, and Access Rights Information) as relevant to the Content Data Object does not mean that those concerns are any less important for other data objects or at other levels, for example, it is important to apply reference, fixity, provenance, context and access rights to Representation Information, or to any other information the Archive is preserving. Definition of these terms as relevant to the Content Data Object is simply to ease discussion of these concepts at the Content Data Object level.

**Producer**: The role played by those persons or client systems that provide the information to be preserved. This can include internal or external OAIS persons or systems.

**Provenance Information**:The information that documents the history of the Content Data Object. This information tells the origin or source of the Content Data Object, any changes that may have taken place since it was originated, and who has had custody of it since it was originated. The Archive is responsible for creating and preserving Provenance Information from the point of Ingest; however, earlier Provenance Information should be provided by the Producer. Provenance Information adds to the evidence to support Authenticity.

**Reference Information**:The information that is used as an identifier for the Content Data Object. It also includes identifiers that allow outside systems to refer unambiguously to a particular Content Data Object.

NOTE – An example of Reference Information is an ISBN.

**Representation Information**:The information that maps a Data Object into more meaningful concepts so that the Data Object may be understood in ways exemplified by Preservation Objectives. It is a type of Information Object.

NOTE – An example of Representation Information for a bit sequence which is a FITS file might consist of the FITS standard which defines the format plus a dictionary which defines the meaning in the file of keywords which are not part of the standard. This would then allow the information in the FITS file to be used by a computer program to display the image which may be contained in the FITS file, together with the associated coordinate system so that a human can identify objects of interest, for example stars or galaxies. Alternatively, the computer program may identify such objects automatically.

**Representation Information Network**:The set of Representation Information that fully describes the meaning of a Data Object. Representation Information in digital forms needs additional Representation Information so its digital forms can be understood over the Long Term. It is a type of Information Object.

**Semantic Representation Information:** The Representation Information that further describes the meaning of the Data Object, and its parts or elements, beyond that provided by the Structure Representation Information.

NOTE – For example, Semantic Representation Information may describe the meaning of columns, and perhaps particular values seen in the columns of a spreadsheet.

**Structure Representation Information**:The Representation Information that imparts information about the arrangement of and the organization of the parts or elements of the Data Object.

NOTE – For example, Structure Representation Information maps bit streams to common computer types such as characters, numbers, and pixels and aggregations of those types such as character strings and arrays.

**Submission** **Agreement**:The agreement reached between an OAIS and the Producer that specifies the intended formats and content descriptions, and any other arrangements needed (such as delivery or transmission protocols), for the Data Submission Sessions. Such specifications establish the intended deliverables from the Producer and how they are represented on each delivery through physical media or in a telecommunication dialogue.

**Submission Information Package (SIP)**:An Information Package that is delivered by the Producer to the OAIS for use in the construction or update of one or more AIPs and/or the associated Descriptive Information.

## References

The following documents are referenced in 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 documents.

[A list of documents referenced in the Report goes here. See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014) for reference list format.]

1. *Reference Model for an Open Archival Information System (OAIS)*. Recommendation for Space Data System Practices, CCSDS 650.0-M-2. Blue Book. Issue 1. Washington, D.C.: CCSDS, June 2012. [Equivalent to ISO 14721:2012.], or later Available from: <https://public.ccsds.org/Pubs/650x0m2.pdf> . The definitions are taken from the draft update of OAIS.

For the diagrams used in this document, see the link <https://www.dropbox.com/s/c3ribtzpyz6ynh9/OAIS-IF-diagrams.pptx?dl=1>

# OVERVIEW

While section 3 explores specific user scenarios, this section (2) provides the broadest overview of OAIS-IF scenarios.

## Current situation

An example of exchanging information between entities is a client-server system, for example a web browser and a web server, where what is being transferred between the two is encoded in HTML, with perhaps MIME encoded objects such as images, using for communication the HTTP protocol via TCP/IP. The Representation Information required to deal with the HTML and a limited number of MIME types is embedded in the web browser software, as if part of what is needed to deal with HTTP. The other parts of HTTP and TCP-IP are dealt with by the operating system and internet infrastructure.

A web browser would not be able to deal with scientific data generally except to allow it to be downloaded as a collection of bits, and as such would not be usable without a great deal of effort on the part of the user, for example to identify then read documentation and download and install software. The user may not even know where to get the documentation and software.

The following diagram illustrates the current situation when users want to combine information from multiple archives. Each archive sends whatever it wishes to send and the user must install appropriate software to deal with each, as indicated by the coloured shapes in Figure 2‑1.

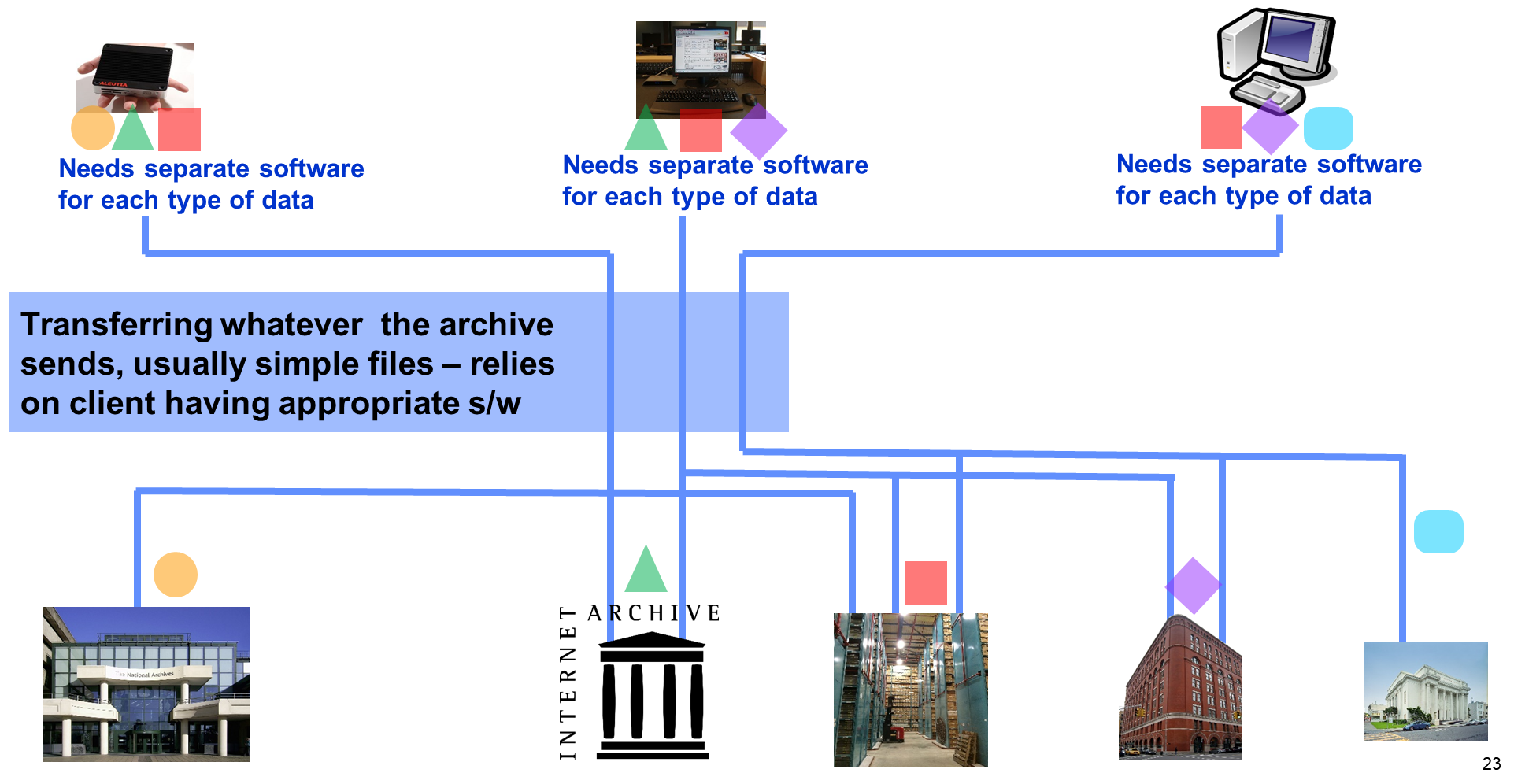


Figure ‑ Connections between clients and archives currently

In order to then combine data from many domains the user may need to go through the same process for each archive, potentially including downloading unique access software (or setting up unique configurations) for each archive. However, even then it may be difficult to combine the data without transforming each set of data to a common format - if such a suitable format exists and is usable by the users’ software.

Each step outlined above is a significant hurdle to the use and integration of information from multiple sources.

## What OAIS-IF is designed to make possible

OAIS-IF defines a set of common Application Programming Interfaces (APIs), protocols and functionalities which will allow developers to create software to provide users with usable information from an archive or set of archives. The software developers may or may not be associated with the archive, but could, for example, be supporting efforts in user groups. Using that software, consumers will be able access data and associated Representation Information (if available) from archives across disparate types of archives or domains of study.

Some of the interfaces will be generally and broadly applicable, and reference implementations should be available. The implementation of other interfaces will be specific to the software being used by the archive or information creator or information user.

The following illustrates the way in which OAIS-IF will enable increased usage between two generic pieces of software, shown here as S1 and S2. The “OAIS-IF interfaces and objects” is based on the OAIS Information Model, which itself is designed to be applicable to any type of Information. The adapters allow S1 and S2 to use the OAIS-IF interfaces and objects.

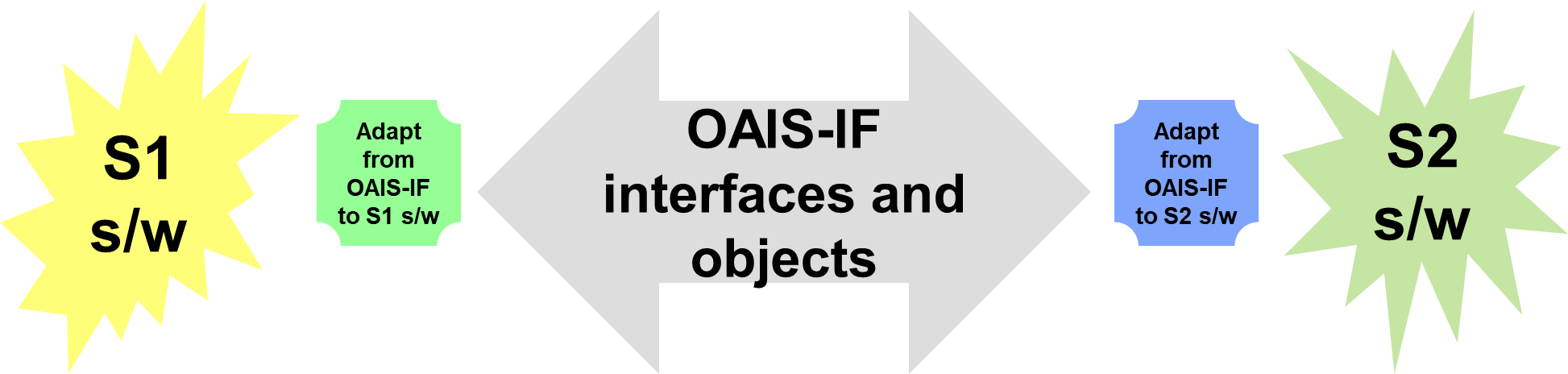


Figure ‑ General use of OAIS-IF

The advantage of using OAIS-related ideas is that the OAIS Information Model was designed to accommodate any type of digitally encoded Information in a way which allows that information to be usable, as well as accommodating to the processes of long-term digital preservation.

This approach can then be generalized to show the way in which users can use their software to access and use information from multiple different archives for which their software is not originally designed.

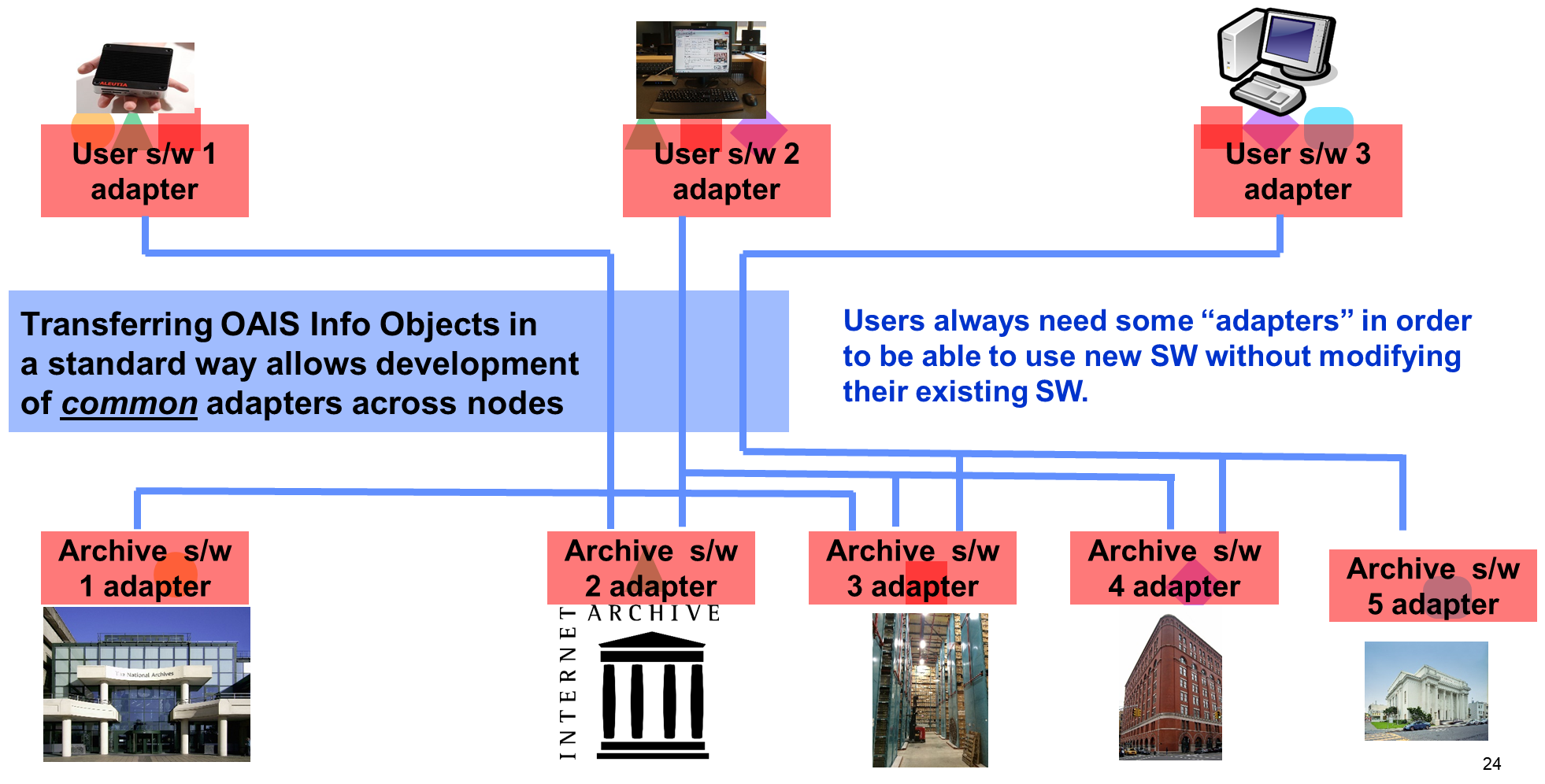


Figure ‑ Connections between clients and archives using OAIS-IF

***NOTE: any of the objects defined in the OAIS Information Model may be requested. In general one would expect to receive either the object itself or an Identifier which allows the object to be obtained at a later time.***

This has the advantage that each client only needs one adapter, which may or may not be specific to the client software being used. This allows that client to accept and use the OAIS-IF Information Objects, independent of the archive with which it is communicating, as long as each of those archives, with their adapters, use OAIS-IF Information Objects.

Of course, the various Information Objects will differ in detail, but they will all contain a Data Object plus the related Representation Information, or more specifically the first part of the Representation Information Network (RIN). Note that OAIS uses the term Representation Information both for a specific item such as a dictionary or piece of software, as well a RIN.

Therefore, the client with its single adapter can get, and understand as far as possible (given the required amount, quality or fidelity of the Representation Information), information from multiple archives simultaneously. As discussed in more detail below, use of OAIS-IF will make combining information from multiple sources easier, but cannot guarantee that it will be automatic or effortless.

An alternative way one might think to implement this would be to find a common format into which all other data could be transformed. However;

1. Format by itself is not adequate. OAIS uses the term Structure Representation Information as a general way of talking about “format”, but it recognises that one also needs semantics (Semantic Representation Information”) and “Other Representation Information” such as software.
2. Those formats which are extremely flexible such as HDF5[[1]](#footnote-1) or HDS[[2]](#footnote-2) require that the software which uses it is specifically tailored to the way in which the structure of these files is arranged.
3. A common format has often been sought, and indeed many formats have the word “Common” or even “Universal” in their name. Moreover, experience shows that such transformations always lose information, and formats named “Common” or “Universal” fail to live up to their names.
4. The sheer size of the data may preclude such transformations in any practical way.

Therefore OAIS-IF is designed to follow the more general OAIS Information Model, so that OAIS Information Objects are transferred, hence explaining why this is called **OAIS**-IF. The advantage of using Information Objects is that they can contain any digitally encoded information but are required to have associated Representation Information, which allows the information encoded in the digital object to be understood – this is discussed in more detail below.

In order to provide a little more detail Figure 2‑2 can be expanded as follows.

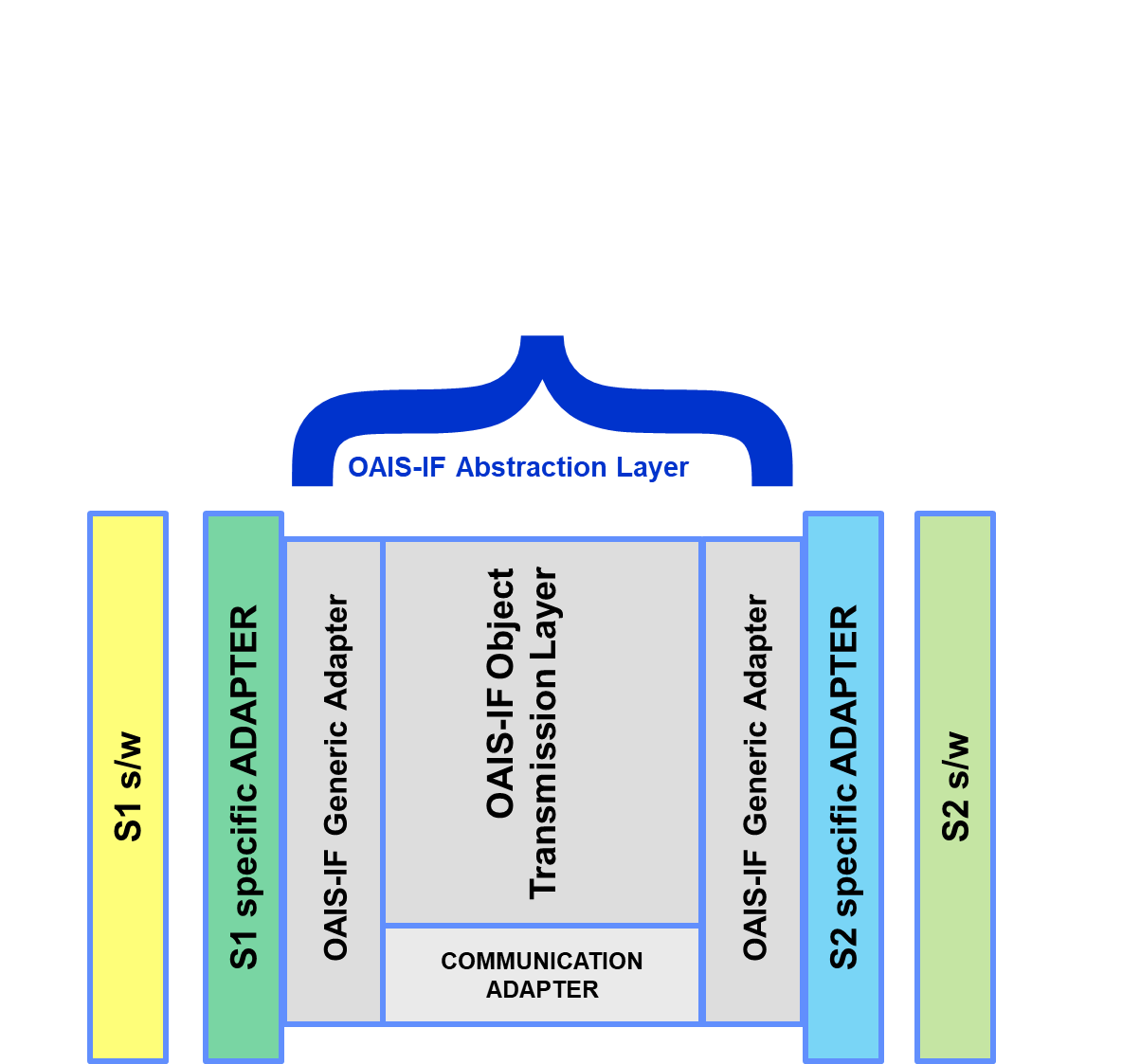


Figure ‑ Expanded view of use of OAIS-IF

The figures shows that the OAIS-IF Abstraction Layer can be broken down into a “generic adapter” at each side of the OAIS-IF Object Transmission Layer. The latter, which uses a communication adapter to deal with network details, arranges largely for the transmission of OAIS Packaged Information Objects.

The S1 s/w is, for example, existing software which probably was created without knowledge of OAIS-IF.

In order to integrate with OAIS-IF one needs an adapter, called here the “S1 specific adapter” i.e. the adapter is specific to the S1 software, for convenience.

The “S1 specific adapter” must be software tailored to the S1 software and must also be software which implements the OAIS-IF interfaces (shown here as the OAIS-IF Abstraction Layer) and required functionality. It may be that the same “specific adapter” is usable for more than one archive.

The adapter by itself may not be able to use the data and Representation Information it initially receives, and therefore it requires additional Representation Information either from the same source or from different sources. It is also possible that there simply is not enough Representation Information available to make the information received usable by the S1 software. For example, the S1 software may only be able to display images and so will never be able to deal with scientific tables, unless the Representation Information essentially does something additional, such as displaying an additional window, independent of the S1 software’s image display, which can deal with such tables.

At the very least, the Representation Information may simply be pointers to documentation for the user to read, or pointers to software which is tailored to the data and which needs to be installed separately.

The generic adapter software includes parts which implement the communications between S1 and S2, either embedded within OAIS-IF or as a separate set of interfaces used by the S1 adapter (shown here as the Communications Adapter). Communications software already exists, and various frameworks provide convenient ways to support multiple communications stacks without changing the adapters.

In a symmetrical way the “S2 specific adapter” (or S2 specific adapter for short) must be software tailored to the S1 software and must also be software which implements the OAIS-IF interfaces and required functionality.

Of course, if S1 (or S2) already implements OAIS-IF then the associated adapter is not needed, but this special case does not cause any problems.

Note that arrangement of this diagram in layers is to show that each layer only communicates to its adjacent layers, through well-defined interfaces.

The layers do not specify **where** those layers are placed. Other diagrams below will show potential placements of the software – see Figure 8‑6.

## Example implementation of automatically usable Representation Information for multidisciplinary work

To provide a more concrete example which also shows the use of the single adapters and the Representation Information, the following diagram illustrates a user’s software S1 which gets information from two archives which use software S2 and S3.

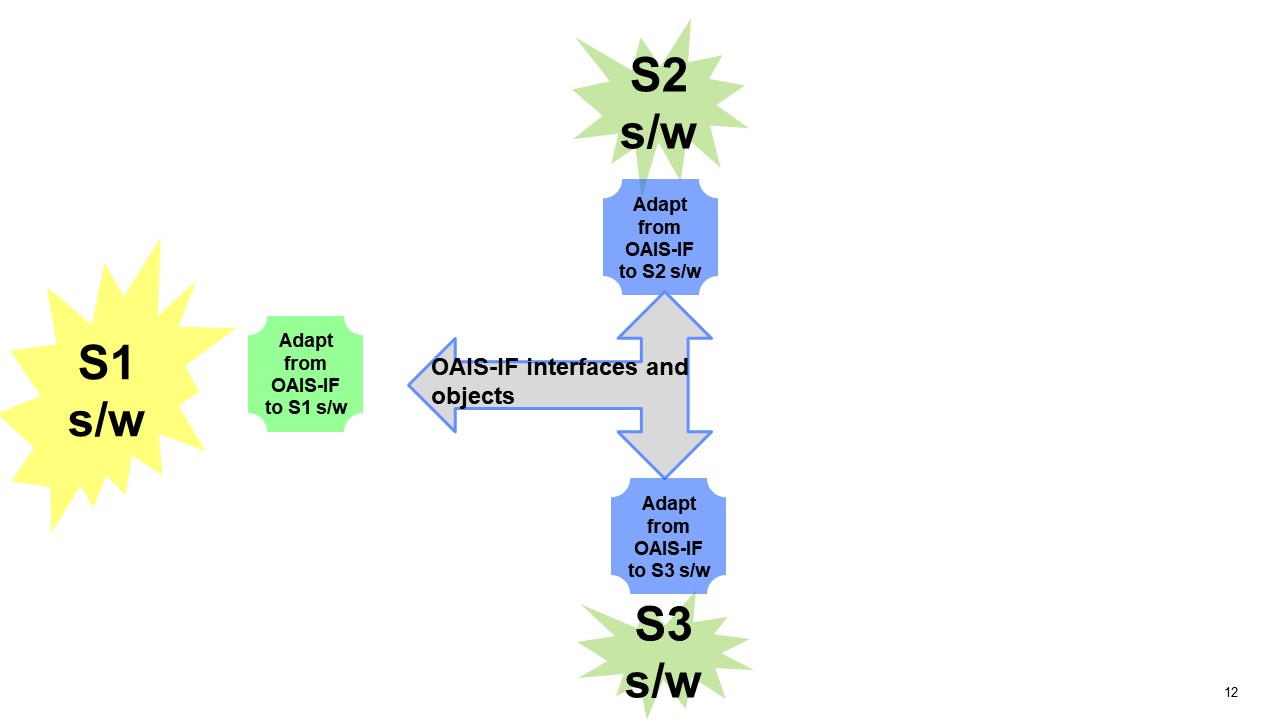


Figure ‑ Multi-archive example

One type of Representation Information which can be used quite easily is as serialized Java objects in a Java implementation. Such an object can be de-serialised to create a Java Object which can be examined through the Java Reflection API to discover which interfaces this object implements. This type of Representation Information is used in the example below.

For example, if one is using client software (S1) which can use JTable interfaces and which, via the adapter, receives information from an archive containing FITS files (S2). The information is in the form of an Information Object as defined by OAIS-IF.

Information may also be obtained from S3, with its own adapter. In this case the archive contains tables in the form of CSV files.

The adapter associated with S2 constructs and sends the Information Object. The adapter associated with S1 can separate the Data Object, which in this example is a FITS file which contains a FITS table, and Representation Information. The Representation Information in this example is a serialized Java object which, when de-serialised in the adapter, becomes a Java object. This Java object, as the adapter is able to discover, implements an interface which can read FITS tables and also implements the Java JTable interface. The adapter is tailored to the client software and therefore is built specifically to be able to provide the JTable objects to S1.

In a similar way the adapter for S3 constructs Information Objects consisting of a Data Object (the CSV file) and the associated Representation Information (the Structure RepInfo which described the UTF-8 and CSV encoding, and the Semantic RepInfo which describes the names, meanings and units of the columns plus Other RepInfo which in this case is again a serialised Java Object). As with the Information Objects from S2, the S1 adapter separates the Data Object from the Representation Information. De-serialising the Other RepInfo reveals that the Java Object also implements the [Java JTable interface](https://docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html), using the Data Object, Structure RepInfo and Semantic RepInfo.

In exactly the same way as with the Information Objects from S2, the S1 adapter provides JTable Objects to S1.

Having JTables both from archives S2 and S3 the Information User (see section 3) of S1 can then combine, compare, etc., the information from these two sources.

In this way each of S1, S2 and S3 have their own unique adapters while the Representation Information allows the information to be combined for multidisciplinary work.

It can be seen that the same will apply whether S1 obtains Information Objects from any number of archives.

The separation of the adapters into generic and specific parts is shown in Figure 2‑6, which includes an initial indication of functionality.



Figure ‑ Indication of functionality in adapters

The red squiggly lines show where interfaces must be defined. It may be possible to define interfaces elsewhere.

The Switchboard is an optional/supplemental capability that has been introduced to possibly provide information needed for communication between, in this case, users and archives. While the Registry may provide additional Representation Information if required.

### Example of a Simple Specific Adapter for file-type servers

If one of the common information sources is a file server, for example a web server, which provides the Data Object containing, in the particular case of a web server, HTML, HTTP, encoded images as well as scientific Data Objects etc, then the adapter would have to construct an Information Object consisting of this Data Object with appropriate Representation Information. The Representation Information may be kept in a separate part of the web server and a simple look-up table could be used to identify which pieces of Representation Information to include.

This Information Object would then be sent in an Information Package to the receiver (S1 for example) using some agreed communications protocol, for example TCP/IP.

Other components of the Information Model may be handled in the same way. For example, the Data Object which encodes Provenance Information may be made up of a standard file, for example saying that all the web pages were constructed by a certain team of people, with additional text identified using a similar look-up table, including for example a script which extracts the version history of each page from the Content Management System (CMS).

A set of interfaces may be defined for such a simple, specific adapter which could be used quite widely and easily for a large number of archives. Configurable software will be created as part of the CCSDS approval process; this software could then be made available to these kinds of archives. Such software would allow archives to get something up and running quickly which the archive could later be improved in terms of efficiency and speed of operation by providing custom built/bespoke software.

# OAIS-IF Scenarios

The following provides a limited set of user scenarios which OAIS-IF will support.

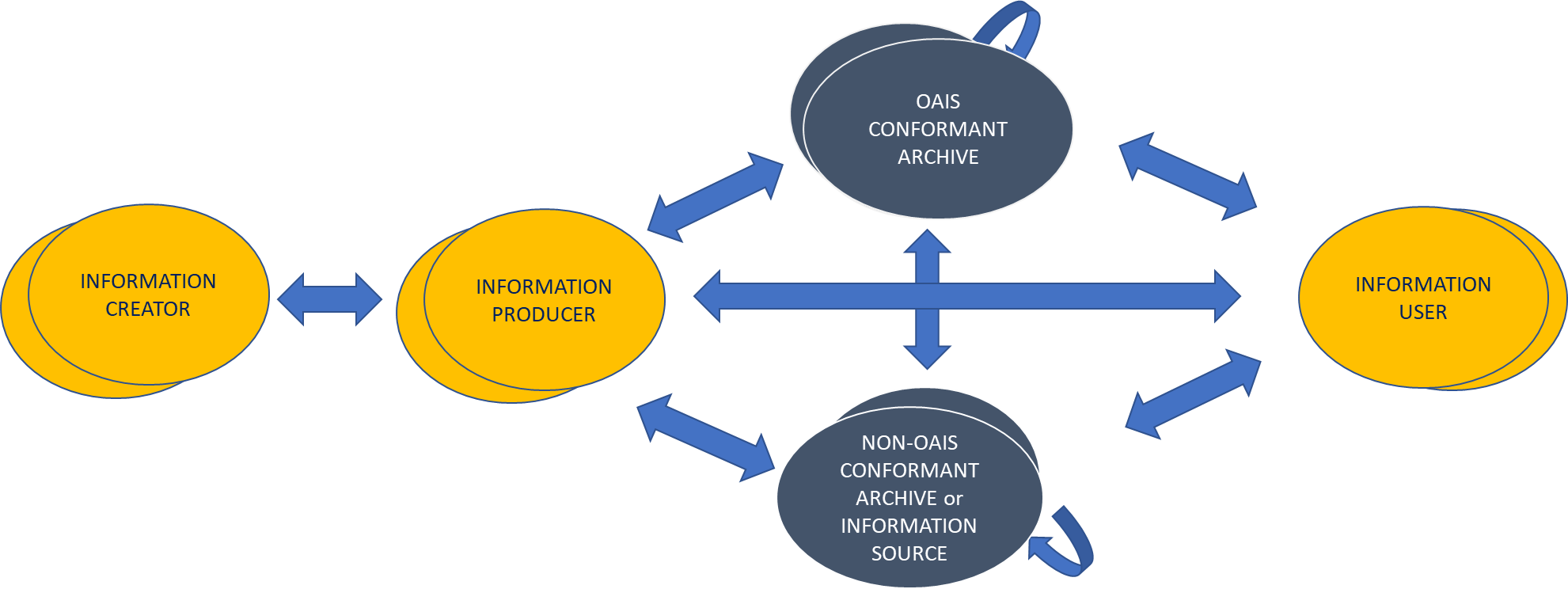


Figure ‑ Roles of entities exchanging information

Information of various types is exchanged between individuals or organisations or systems which may be fulfil one or more of the following roles:

* Information Creators
* Information Producers
* Information Users, some of whom may be members of one or more Designated Communities of one or more OAIS conformant archives.

NOTE: OAIS uses the term “Consumers” to refer to the role played by those persons, or client systems, who interact with OAIS services. Therefore, Consumers are a subset of Information Users because the latter may interact with services which are not connected with an OAIS.

* OAIS conformant Archives (with capitalised “A”)

NOTE: Archive-to-archive interfaces (for both OAIS-conformant archives and non-OAIS-conformant archives) may be specified in this document (and accomplished by implementers) by using “Producer” and “Consumer” interfaces rather than by archive-specific interfaces for transactions between archives.

* Archives, or other Information Sources, which are not OAIS conformant, normally denoted by a lower case “a”. The lower case is also used to indicate that the archive may be either OAIS conformant or not.

In the OAIS-IF set of documents all these exchanges of information are assumed to be encoded in one or more Data Objects, with associated Representation Information. It is reasonable to propose that these are packaged together into OAIS Information Packages (Figure 3‑2).

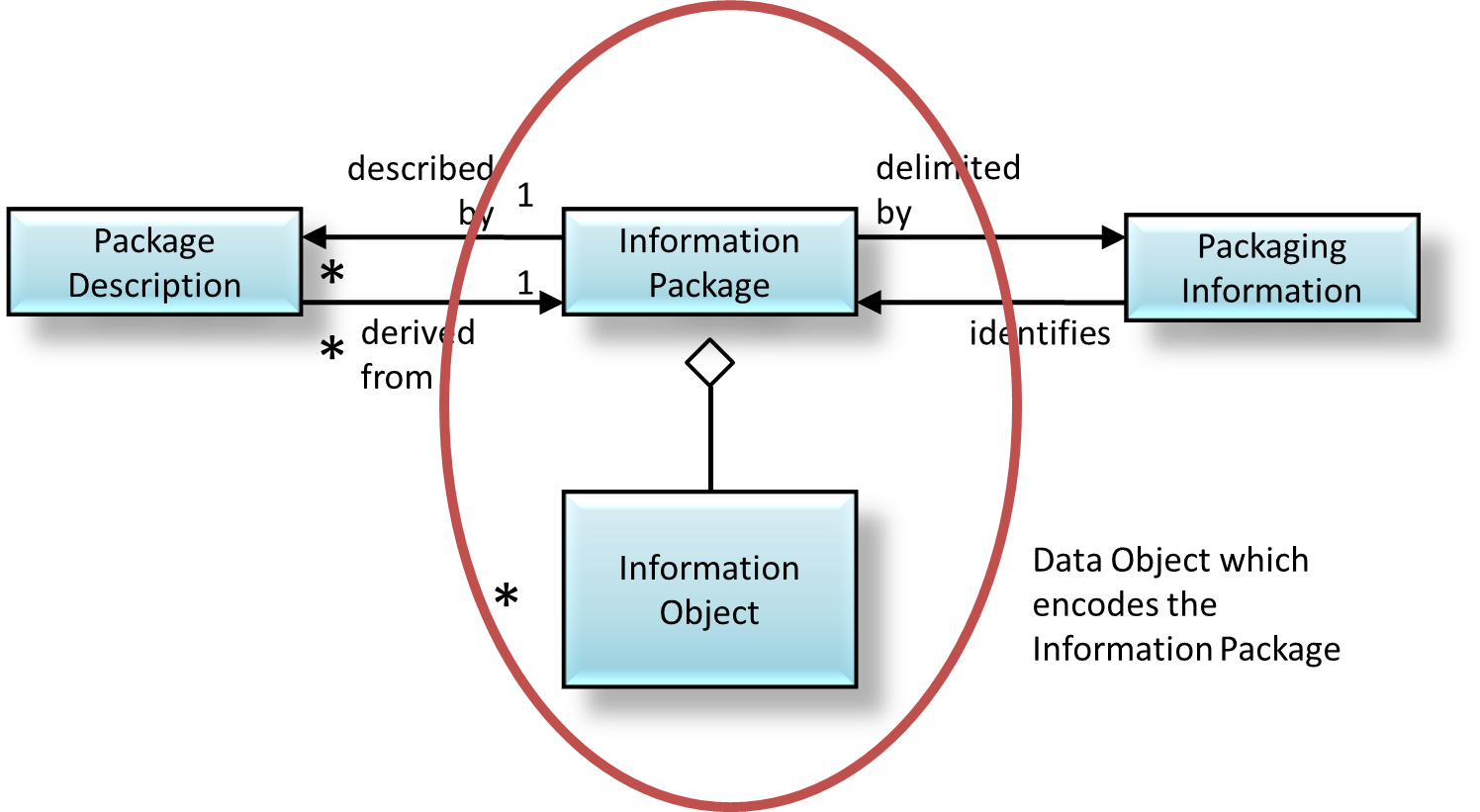


Figure ‑ OAIS Information Package

The Information Packages are themselves made up of sets of bits, in other words Data Objects. These must be associated with their own Representation Information, of which, for the purposes of this document, Packaging Information is a part. The Information Packages will be transmitted between the entities by communication protocols such as TCP/IP, CCSDS DTN etc.

The full Archival Information Package (AIP) identifies all the pieces of information required to preserve the Content Information is shown in Figure 3‑3. OAIS-IF should allow an implementation to deal with all these components of the AIP. However, this does not imply that all information sources, nor all transfers, use AIPs. The only requirement is that the design of OAIS-IF should support all the component objects required by an AIP.

For an OAIS Archive, the transferred Information Packages should have adequate content to allow an OAIS archive to build AIPs. For non-OAIS archives, other transactions (submission agreements, etc.) between the users and the archive will have to establish the needed content of the transferred Information Packages.



Figure ‑ Detailed view of OAIS Archival Information Package (AIP)

Specific scenarios are listed next. In the following, for generality the term “User” can refer to a user, an archive (whether OAIS conformant or not), an Information Creator, an Information Producer, or a Consumer. The term “identifier” is used as a general term for something used to obtain the item required.

## Information Producer sends Information to an OAIS ConformaNt Archive

Information is created by an Information Creator. It is sent to an OAIS Archive, via an Information Producer, which may be different from the Information Creator. OAIS defines, in general terms, a Submission Agreement that specifies the intended formats and content descriptions, and any other arrangements needed (such as delivery or transmission protocols), for the Data Submission Sessions. Such specifications establish the intended deliverables from the Producer and how they are represented on each delivery through physical media or in a telecommunication dialogue.

The Submission Agreement may be a document to be interpreted by a human. However, the extent to which these things can be computer interpretable is of interest for the OAIS-IF requirements.

The Information Producer may use the PAIS standard to encode the SIPs, in which case OAIS-IF may be used to define how the SIPs are transferred to the OAIS.

However, the Information Producer may use a different encoding for the SIPs, in which case OAIS-IF may be used to ensure that the SIPs may be understood by the OAIS – just as happens with any Data Object.

Issues to be addressed include several which are mentioned as part of the Submission Agreement, but here the aim is to be more specific:

* How to arrange and carry out the transmission
* How to ensure that the Representation Information (Packaging Information) of the package is provided to the Archive.
  + If the Producer uses PAIS then the PAIS standard itself can be referred to, but if in the future there may be updates of PAIS then the correct version must be identified.
* How to ensure that the Archive can extract components from the Information Package Data Object.
* The OAIS Archive will expect to receive, perhaps over the course of several packages, the information required to create an AIP.

If a sequence of transmissions is to be arranged, with authentication mechanisms and fixity checks, a context may need to be maintained for the Information Packages, for example to ensure packages are kept in sequence. Alternatively, the information may be provided in documentary form, rather than through an API.

## Information PRODUCER sends Information to a Non-OAIS conformant archive

This scenario is similar to the one with the OAIS Archive. Significant differences are that there might not be a formal Submission Agreement and the archive might not require all the components of an AIP.

## OAIS conformant archive sends information to an Information USER

In this scenario the user searches for the information he/she requires and the OAIS Archive sends back information as a response to the search and then sends the information requested in the form of one or more DIPs.

The following steps are involved:

1. The user inputs search criteria and identifies repositories which contain required information.
2. The user chooses one (or more repositories) and queries what is available.
   1. The user may need to log in to see what she/he is allowed to access.
3. The user then obtains the information from the repository.
   1. The user may simply get a copy of the whole AIP or
   2. The user may get a piece of information created by the repository which is create by some processing its holdings.

OAIS-IF treats queries and responses as messages in Information Packages. The DIPs are also Information Packages.

Since the Archive is OAIS conformant then according to one of the Mandatory Responsibilities a user who is a member of the Designated Community must be able to:

1. obtain enough Representation Information to understand/use the Data Object in the DIP and
2. obtain all the information related to Authenticity, and so the OAIS-IF must support this.

If the user is not a member of the Designated Community, then it does no harm to allow him/her to have the same functionality but in this case the Archive will not guarantee to be able to supply enough Representation Information to allow that user to understand/use the Data Object. This would be the case if the provided technical lexicon or jargon is not familiar to a Consumer from outside the Designated Community.

### An INFORMATION USER wants to obtain information from an OAIS conformant ARCHIVE

1. User uses an identifier for a repository to search for and select specific information to obtain an object identifier.
2. The user requests the object or part of an object.
3. The user receives the object requested.

### An INFORMATION USER wants to obtain information from a NON-OAIS conformant ARCHIVE

1. User uses an “identifier” for repository to search and select specific information to obtain an object identifier.
2. The user requests the object or part of an object.
3. The user receives the object requested.

### AN INFORMATION User wishes to get an AIP

1. The object identifier is confirmed as pointing to an AIP rather than any other object.
2. The Data Object of the AIP is retrieved.
3. The Representation Information, of which Packaging Information is a part, associated with the Data Object is retrieved.
4. The Package Description Information is retrieved if required.

### AN INFORMATION User wishes to get information derived from an AIP

1. The user uses the identifiers for one or more AIPs to obtain object identifiers for the components of the AIPs.
2. The user may request some operation to be performed on the components to create new pieces of Information.
3. The user uses that identifier to obtain identifiers for any components of that component for example if the original AIP is an AIC then identifiers for the component.

AIPs can be obtained and then identifiers for the components of those AIPs can be obtained, and so on.

### If required, INFORMATION users are authenticated and authorized

1. User requests authentication.
2. The user provides the appropriate username/password or private key etc.

### Information is transferred as one or more Information Packages

1. The user requests some information using the object identifier that can be used to obtain local identifiers for the various components of the AIP, or other pieces of information.
2. The information source constructs the object to be transferred e.g.:
   1. Extracting the components from internal storage such as a database or filestore.
   2. Some components may have sub-components such as individual events relevant to Provenance.

## Non-OAIS conformant archive sends information to an information USER

This scenario is similar to the previous one except that the archive may not be able to guarantee to be able to supply Representation Information or evidence about Authenticity.

## OAIS conformant archive exchanges information WITH another OAIS CONFORMANT Archive

The previous scenarios involving an OAIS could apply here, with one of the OAISes acting as either an Information Producer or Information User. The special case is where the first OAIS should be able to send a copy of a complete AIP to the second OAIS, either as a single Information Package or as several related packages.

## OAIS conformant archive exchanges information with a non-OAIS Archive

In this case either archive could play the role of Information Producer or Information User. The difference from the previous scenario is that complete copies of AIPs are unlikely to be sent from the non-OAIS conformant archive.

## Non-OAIS conformant archive exchanges information with another non-OAIS Archive

This scenario is similar to the previous one but, since non-OAIS archives may not actually have AIPs, copies of complete AIPs are not likely to be exchanged.

## Information Creator sends Information TO INFORMATION Producer

Information is created by an Information Creator. This may involve obtaining Information from other sources, where the Information Creator acts as an Information User.

It is sent to an Information Producer, which may be different from the Information Creator.

## Possible RepInfo Use Cases

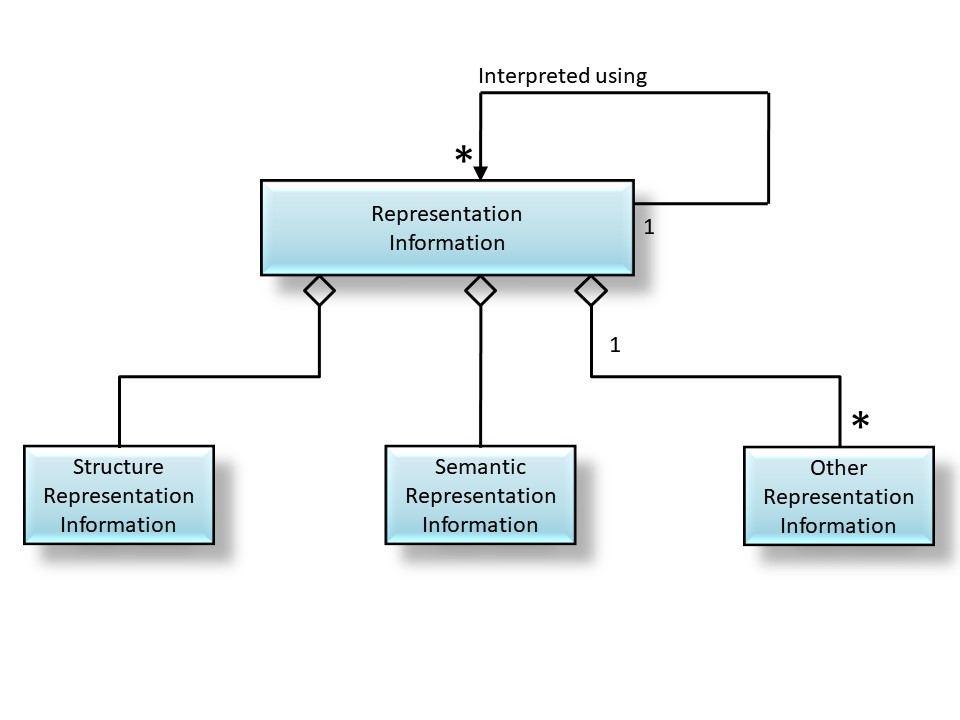


Figure ‑ Fundamental types of RepInfo

1. Use Representation Information (RI) to do something with a Data Object (DO) e.g. as specified in the Preservation Objectives:
   1. Identify the RI associated with the DO.
   2. Use Semantic Information to identify an information element required:
      1. Identify the Semantic Info components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN).
      2. List the Semantic Info components of the RI related to the DO.
      3. Are the Semantic Info components things which is understandable/usable by the user?
   3. Use Structure Information to extract the specific information element:
      1. Identify the Structure Info components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN).
      2. List the Structure Info components of the RI related to the DO.
      3. Are the Structure Info components things which is understandable/usable by the user.
   4. Use Other RI:
      1. Identify the Other RI components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN)
      2. List the Other RI components of the RI related to the DO
      3. Are the Other RI components things which is understandable/usable by the user
   5. In addition one can see that there are other considerations:

* There may be equivalent “versions” of an example of Representation Information e.g. software which may be written in C, C++ Java etc - any of these may be used i.e. it may be (C-version OR Java version OR C++ version).
* Some types of Representation Information may be required e.g.:
  + Structure Representation Information e.g a CSV text file AND
  + Semantic Representation Information e.g. a dictionary with the meaning, units etc of the table columns.

## Possible Provenance Use Cases

1. Use Provenance Information to find out what has happened to a DO
   1. Identify the Provenance Information associated with a Data Object (DO)
   2. List provenance events in time order
   3. For each event, obtain (e.g. following PREMIS model): Object, Environment, Rights, Agent, Event - more specifically
      1. Data Object Identification Information (Reference Information??)
      2. Information about systems used
      3. Access Rights Information
      4. Agent Identification Information
      5. Event Description Information

## Search

Very little has been written here about search capabilities, although it is clear that there MUST be such capabilities to all the user to identify the information to request. OAIS has very little guidance on search capabilities. All data/information providers will have their own search capabilities, but each one will have its own search parameters language and search capabilities.

One can identify a number of general search mechanisms which could work for any information source and could also be used to identify data sets which could be combined:

1. Get a simple list with a description of all information available, which can then be used in a simple text search.
2. Full text search of the contents of all the data objects.
3. Treat the search capability as encoded in a Data Object which has Representation Information to enable the user to user that Data Object in order to query the source e.g. an archive.

# OAIS-IF Requirements

The term “archive” is used for a general information repository, whether conformant to OAIS or not; the term “OAIS” is used where the archive is OAIS conformant.

The following (incomplete) list of requirements for APIs have been identified so far. Requirements are not placed on the internal workings of the various entities, however some requirements may imply functionality from the archive, but if this functionality is not available then a NULL response may be made, and will be acceptable.

Table 1 Requirements for OAIS-IF - TO BE COMPLETED

|  |  |  |
| --- | --- | --- |
| **REQ ID** | **REQUIREMENTS** | **SCENARIO ID** |
| 1 | APIs, based on, but not restricted to, the OAIS Information Model will be available. | 3.1, 3.2 3.3, 3.5, 3.3.1, 3.3.6, 3.4, 3.6 |
| 2 | APIs for negotiations to allow additional Representation Information to be provided, if available, will be available. | 3.3, 3.9 |
| 2-1 | API to identify the category of a piece of RI e.g. Structure, Semantic, Other, and ideally more detailed categories. | 3.9 |
| 2-2 | API to obtain the full RIN or the immediately related pieces of RI | 3.9 |
| 3 | An API which allows a negotiation on transformations of the Data Object before transmission, will be available. |  |
| 4 | An API which allows the parties to agree on or discover a communications protocol to use will be available. | 3.1 |
| 5 | An API to allow the archive to be able to verify that a user requesting access to the archive is authorized will be available. | 3.3.5 |
| 6 | An API to allow the archive to be able to accept SIPs will be available, in particular | 3.1 |
| 6-1 | Allow the archive to verify that Producer requesting access to the archive is authorized. | 3.1 |
| 6-2 | Make available the definitions of the types of submission packages that the archive will accept. | 3.1 |
| 6-3 | one or more interfaces which can be used to submit an SIP and reports back on status of the ingest | 3.1 |
| 7 | APIs to support search of the archive to provide an identifier for required information will be available. | 3.3, 3.11 |
| 8 | APIs to retrieve information, as Packaged Information (i.e. Information Packages with Representation Information to allow the package to be unpacked), given one or more identifiers will be available. | 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.6 |
| 9 | APIs to allow the discovery of communication options, e.g., how a client can communicate with an archive, should be available | 3.1 |
| 10 | API to obtain Provenance Information related to a Data Object | 3.10 |

# More detailed Considerations

The approach inherent in the OAIS-IF can be carried out through the following stages.

## Basic OAIS Interfaces

Construct Interfaces for each OAIS Information Model (IM) object, with minimal details. For example:

1. an Information Object must have a Data Object and Representation Information, therefore there should be get/put methods for each of these.
2. Representation Information often comes in groups which include “Structure”, “Semantic” and “Other” categories of RepInfo. Therefore it would make sense to include the following:
   1. Any piece of RepInfo may have a category which at the very least could be a String so that an application could make an informed choice as to which piece of Representation Information to proceed with.
   2. The group of RepInfo could be one in which ALL the individual pieces of RepInfo must be used, alternatively the group may consist of alternatives so that a choice may be made to select the SINGLE piece of use e.g. software may be implemented as JAVA or C; these alternatives would be in a group from which one may be selected.
3. An Information Package will have get/put methods for the following objects, all of which are Information Objects:
   1. Information Object
      1. If this includes PDI then this could be treated as an And Group, and in particular if the package is an AIP then all the components must be there.
   2. Package Description
   3. Packaging Information
4. An Archival Information Package will extend the Information Package by adding that the Information Object MUST include:
   1. Data Object and its
      1. Representation Information
      2. Preservation Description Information, which itself contains the following Information Objects:
         1. Provenance Information
         2. Context Information
         3. Reference Information
         4. Fixity Information
         5. Access Rights Information.

Note that the any Information Package may contain these objects, but there is no guarantee that they are complete from the point of view of the repository. Therefore, it would be sensible to add a method e.g.

*Boolean iscomplete()* which is TRUE when the AIP is believed to be complete and false otherwise.

1. Identifiers for objects must be available. These would allow more generality that simply a URI etc. The Identifier interface might have methods such as:

* String toString() - returns a string.

## supplements needed to OAIS IM interfaces

1. The OAIS IM does not provide any additional detail about object interfaces. All one can say is that any of the Information Objects implement these get/put methods. The implication is that one relies entirely on the Representation Information to be able to use/understand the Data Object in some way (to be defined), for example if the Identifier for the Representation Information is recognised and some existing software may be used to deal with the Data Object for that object. For example, a Web browser can use the Media Type (aka MIME Type), as Structure RepInfo, to determine which pre-configured application to use.
2. Identifiers for objects must be available. These would allow more generality that simply a URI etc. The Identifier interface might have methods such as:

toString() - returns a string

toURI() – return URI if possible

1. The set of Interfaces would not provide constructors for any object – they must be provided by the implementing classes. Constructors could include:
   1. InformationObjectImpl() which creates a trivial class which implements the Information Object interface into which the put methods can insert the Data Object and RepInfo Object.
   2. InformationObjectImpl(DataObject identifier1, RepresentationInformation idendifier2), which essentially performs the “put”s in one step.
2. Authentication and Authorisation must be provided by the implementing classes.
3. The OAIS IM does not provide any guidance on search interfaces. A number of possibilities could be considered.
   1. We may define a simple interface such as
      1. search(String query) which return a list of identifiers. However, the structure of the string must be defined somehow.
   2. The search may be performed using a repository specific method – perhaps using the “Hollywood Principle”.

## Functionality for Interoperability

Some specific functionality is required for interoperability including:

* Negotiate – to agree how to agree a strategy for communication and usability
* Get more RepInfo – request and receive RepInfo if required for usability

This functionality involves behaviour which cannot be described only by interfaces, instead other UML diagrams are needed including Sequence diagrams, Component diagrams and Deployment diagrams (section 7).

## Communication protocols

In order to support interoperability, the communication protocols must be defined.

### REST communications

REST (<https://restfulapi.net/>) is well defined as a way to communicate.

JSON (<https://www.json.org/json-en.html>) is widely used.

### Possible JSON examples

This is a simple example for a InfoPackage and an InfoObject, which should be understandable. The only non-OAIS native part if what is called the “AndGroup” and “OrGroup” with the example of the OtherRepInfo where there are 2 separate implementations – C# and JAVA – and one can choose which to use depending on which programming language is preferred.

In this example all the separate parts are referred to by URI, but one could instead put in simple text or an encoded binary object with something like binhex.

A General Information Package would be as follows.

{

` "InformationPackage":{

"PackageType":"General",

"PackageDescription": "This is an example IP",

"InformationObject":{

"DataObject":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/do},

"RepInfo":{

"AndGroup":[

{"RICategory":"SemanticsRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/risem},

{"RICategory":"StructureRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/ristr},

{"RICategory":"OtherRI",

"OrGroup": [

{"RICategory":"OtherRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/rioth-java-sw},

{"RICategory":"OtherRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/rioth-csharp-sw}

]

}

]

}

}

}

}

A complete Archival Information Package would be as follows:

{

"InformationPackage":{

"PackageType":"AIP",

"PackageDescription": "This is an example AIP",

"PDI":{

"Provenance":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/prov},

"Reference":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/ref},

"AccessRights":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/ar},

"Context":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/context},

"Fixity":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/fix}

}

"InformationObject":{

"DataObject":{"IdentifierType":"URI", "Identifier":http://myprov.example.com/do},

"RepInfo":{

"AndGroup":[

{"RICategory":"SemanticsRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/risem},

{"RICategory":"StructureRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/ristr},

{"RICategory":"OtherRI",

"OrGroup": [

{"RICategory":"OtherRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/rioth-java-sw},

{"RICategory":"OtherRI", "IdentifierType":"URI", "Identifier":http://myprov.example.com/rioth-csharp-sw}

]

}

]

}

}

}

}

# OAIS-IF document Structure

The following subsections outline documents in the OAIS-IF set, with notes on their contents.

**NOTE: The following diagrams are drafts and will be updated in due course** – see <https://www.dropbox.com/s/c3ribtzpyz6ynh9/OAIS-IF-diagrams.pptx?dl=0> (Shared Dropbox: \CCSDS-DAI-Shared\Draft Documents\OAIS-IF Architecture Description\ OAIS-IF-diagrams.pptx)

Note that there may be name changes to the documents, and there may (temporarily) be inconsistencies in the set of documents defined.

## OAIS Document Tree

The following figure describes the OAIS Documentation Tree. The box colors follow the book color conventions of CCSDS: Magenta, Green and Blue.

See Figure 6‑1 below, note the division between documents in the OAIS Process Framework (OAIS-PF)\_and the OAIS Interoperability Framework (OAIS-IF). This document provides rationale, scenarios and requirements for the OAIS-IF. The OAIS-PF is shown as reference only. Acronyms are listed in section 1.6.1.

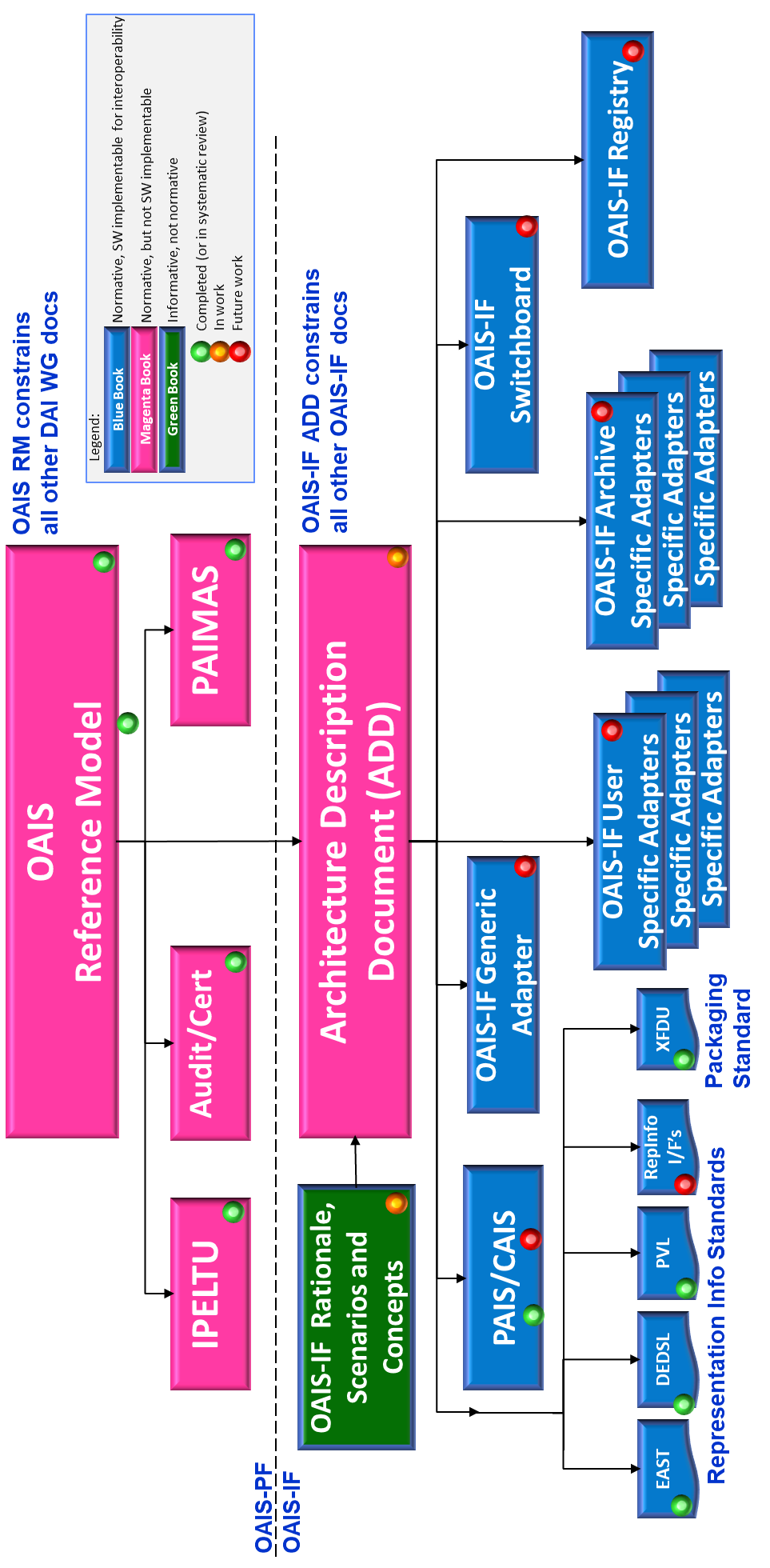


Figure ‑: NOTIONAL OAIS/OAIS-IF Documentation Tree

## OAIS-IF Rationale, Scenarios, and Requirements Document [GREEN]    (This document)

* An overview of the architecture, what issues we believe the set of standards is to address, what it will cover and what it will not cover (we had lots of discussion of that) and how it relates to OAIS and the other DAI Workgroup (and other) standards.
* What people/systems will be able to do which they cannot do now
* Some simple use cases to illustrate what we are addressing.
* Make it clear that we solve the central problem of how to deal with “bags of bits” in a general way applicable throughout the architecture.
* An indication of what Information Producers, Information Users and Archives will need to change to benefit from these standards.

## Architecture Description Document (ADD) [MAGENTA]

The document contains:

* an overview of the OAIS-IF Architecture, the overall diagram, and its components.
* UML diagrams (Class, Sequence, Components, etc.)

See <https://www.dropbox.com/s/my91nnl1o8qplfo/02_White_Book_Recommended_Standard_OAIS-IF_Draft_191216%20v5.3%2006_ComparedCombined_04_05_201222b_Cleaned.docx?dl=0> and [https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit#](https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit)

## OAIS-IF GENERIC ADAPTER [BLUE]

This is a Blue Book because it is implementable and there should be 2 interoperable implementations from NASA and ESA.

|  |
| --- |
| To publish a protocol interface spec, it should define   * actual Protocol Data Units (PDU)s, * info models, * behavior, * peer to peer interactions in some unambiguous form |

* The directly OAIS related Information Model interfaces, adding in methods and additional interfaces such as Identifiers etc.
* The OAIS Unique Selling Points of understandability
  + Negotiation interface and
  + a minimal interface for Registry and Switchboard
* Protocol specification

How we pass information between components e.g., Packaged Information

* + PDU
    - JSON encoding – see section 5.4.2 which shows a complete AIP, but in general some components may be omitted – only the Data Object and Representation Information is needed for an Information Package.
  + Information Model:
    - OAIS Information Model, with natural interfaces implied by the UML
  + Behaviour
    - as outlined in this document the Generic Adapter Interfaces and functionality will be defined, including the Generic Adapter to Generic Adapter and the Generic Adapter to Specific Adapter interfaces.
  + Peer to peer interactions in some unambiguous form
    - REST: HPPT(S), JSON serialisation and HTTP responses e.g. <http://swagger.io>
* In order to have a single book which will allow a usable implementation this book will show the API in JAVA and include a simple specific adapter- see section 2.3.1.

### Proposed BASIC Required Methods

**The methods use DataObjID rather InfoObjID – this is consistent with the changes made to the OAIS Information Model.**

#### Obtain basic Information Model Components other than Representation Information

Given a Data Object Identifier “doid”:

* **DataObjectID doid.getXXX()** where “XXX” is for example “Provenance” or “Context” etc (but see discussion of Representation Information below)

One issue is that a Data Object does not have an intrinsic type. However given a Data Object in an archive the one should be able to obtain its Representation Information.

#### Obtain Representation Information

* **RepInfoNetwork doid.getRIN()**

This will return the full RIN associated with the Data Object – the RIN will only contain pointers plus relationship information.

The following sub-sections discuss some possible ways to deal with RIN.

##### info.oais.interfaces.infomodel.repinfo

The following interfaces are incomplete ideas for navigating a Representation Information Network. The need for these is that one expects that RepInfo may be grouped. For example one might have alternatives e.g. software that is in the form of either Java OR C# - the choice is up to the user/user software. Also one might have pieces of RepInfo that should be used together e.g. one almost certainly needs the Structure PLUS the Semantic RepInfo.

The following Interfaces are essentially marker interfaces.

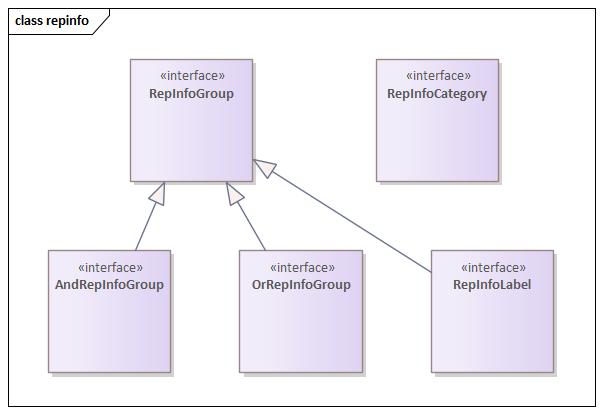


Figure 8‑2 RepInfo grouping interfaces

##### Info.oais.interfaces.infomodel.repinfo. possRepInfoNetwork

A RepInfoNetwork is an Acyclic Directed Graph. There are a number of existing APIs to deal with Directed Graphs, however they are quite complex and so a simple set of interfaces are suggested below.

The existing libraries may be used to implement these interfaces.

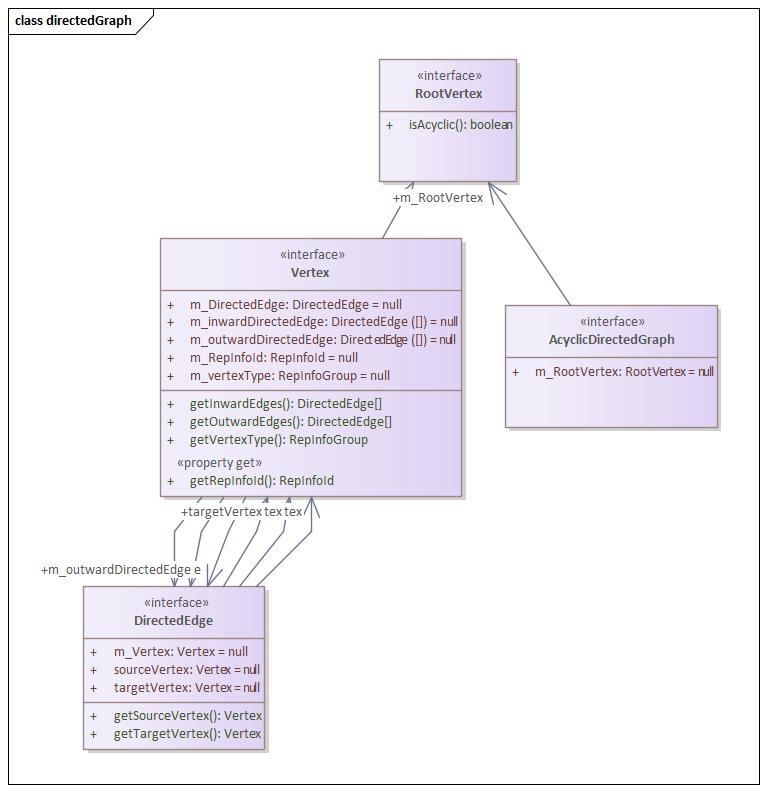


Figure 8‑3 Directed Graph Interfaces

The methods here do not include the set methods.

The aim is to identify Vertexes which have incoming or outgoing directedEdges. The RepInfoNetwork is a Directed Graph of DirectedEdges and Vertexes.

***To be consistent with the previous section we must say that a Vertex has an associated Data Object which is that Data Object which is part of the Representation Information object.***

Each Vertex may have a pointer to an associated piece of RepInfo, for which the outgoing edges provide the RepInfo. The edge may point to a Vertex which may simply be a way to a single piece of RepInfo or a group that may be an “AndGroup” or an “OrGroup”. The former must be taken together, for example Structure plus Semantic plus Other RepInfo, while the latter provides alternatives. Using interfaces derived from these ideas one can navigate the RepInfoNetwork, or serialise it as a serialised Java Object.

### Query

An archive may have any type of query language.

Therefore the suggestions is a two stage process:

1. Request the query language from the archive
2. Use that information to construct a query

In the following the entity being queries is “archID”

#### Request Query Language

* **DataObjectID archId,getQueryLanguage();**

The DataObjectID for the query will have its own RepInfo, which we hope will be understandable to the requestor. We could define some simple interfaces to make this easier.

The requestor would then create the appropriate query object “queryDataObjID”.

* **DataObjectId archId.sendQuery(queryDataObjID)**

Some specific interfaces for Representation Information, Provenance, Fixity, Access Rights, Reference (which may link to Identifiers) if we actually want interoperability. With modern programming languages we do not need to say that these are the only possible interfaces for these specific types of Information, but we recommend these.

See <http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/model/> and <http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/framework/> for an example of interfaces from an old project.

Additional Books may be required as follows:

## OAIS-IF ARCHIVE Adapters [MAGENTA]

* What Archives need to have in order to interact – how to get from what the archives can provide to what the OAL demands?
* Example Specific Adapters e.g. for systems which normally simply provide a Data Object such as a science data file or a web page.
* What users need to have in order to be able to interact. This could be how to deal with encoded information in general terms plus some specific examples such as images, tables and trees.

## OAIS-IF Java binding [Blue Book]

* How to implement the OAL in Java (should be fairly straightforward)

## OAIS-IF XXX binding [Blue Book]

* How to implement the OAL in language XXX (difficulty depends on the language)

## OAIS-IF Communication Protocol binding [Blue Book]

* How to use existing communication protocols, including CCSDS, TCP/IP, SPRING, etc

## OAIS-IF SPECIFIC Adapter for XXX [Blue Book]

* Definition of specific adapter interfaces to interact with software XXX.

## Producer-Archive Interoperability Protocol (PAIP) [Blue Book]

* SIPs and how to transfer them using:
  + PAIS
  + OAIS-IF documents above
* Notes on “out of band” communications to make arrangements.

## CAIS/ PRODUCER-ARCHIVE INTEROPERABILITY PROTOCOL (CAIP) [Blue Book]

* Queries – Use variety of existing standards
* DIPs

illustrates the use of the various components described by the documents described in this section, where the client software (e.g. PDS, Astronomy or EO software tools) uses an Adapter with a specific programming language binding to create objects specified by the OAL. The Archive adapter, with a specific programming language binding converts the OAL objects into ones which the Archive Native Interface can use to obtain the required information from the Archive Internals.

These objects are communicated through the Communications Adapter (shown here as being based on SPRING, REST or TCP, but others are possible). It is shown as a vertical bar to indicate that there are alternative ways of distributing the various pieces of software.

# ANNEX DESIGN CONSIDERATIONS \_ TO BE DELETED WHEN GENERIC ADAPTER BB IS COMPLETED

## Assumptions

1)      We are focussed on information that comes out of and goes into things like OAISes and Applications (Producers/Consumers/Management).

2)      We are concerned with OAIS functions within the archive **only** when they have characteristics that surface at user interfaces (users including other archives).

3)      The information we are specially qualified to facilitate the exchange of is that which is related to OAIS Information Model.

4)      The OAIS Information Model should form the basis of our API/protocol/binding/plugin, but things need to be added e.g. security.

5)      We should not be concerned about optimisation right now.

6)      We want to facilitate the exchange of information between disciplines and OAISes so it can be used and/or preserved.

7)      An Information Package is a Data Object; its Packaging Information is its RepInfo which allows one to unpack the package and identify its components..  Together they make an Information Object.

The way to deal with an Information Object is the following

**How to deal with an IO – fundamental ideas**

|  |
| --- |
| a)       Each Information Object (IO) must have an ID which identifies its RepInfo, an Object RepInfo ID (ORIID)  b)      Given the ORIID the receiver of the IO can do one of the following:  a.   Use the IO because the receiver knows how to deal with things that have that ORIID  OR  b.   Negotiate with the sender to see if it can send an alternative “form” which it does know how to use e.g. send a list of ORIIDs which it knows how to deal with and see if the sender can transform the Data Object to a preferred ORIID.   1. If the negotiation is successful then (a) applies  **OR** 2. Use the ORIID to get as much of the RIN (RepInfo Network) from a registry as it needs to use the DO.    1. The receiver may negotiate with the registry to get RepInfo it can use e.g. is there some Java software that can use the Data Object in the IO?       1. To help decide which piece of the RIN to use, the RIN must have some mechanism for indicating which role the next pieces of RepInfo play e.g. Semantic, Structure or Other. |

This is used throughout the application of the OAIS-IF.

## Negotiation for Usability

One of the key capabilities which an OAIS should be able to provide is Representation Information to accompany the Data Object to make an Information Object which can be understood by its Designated Community, however the OAIS may have more Representation Information, perhaps kept in anticipation of changes in Knowledge Bases. A non-OAIS archive may have some RepInfo so here we focus on OAIS archives but remember that the archive (whether OAIS or not) may return “null” if it cannot supply what is needed, and this may mean that the Data Object cannot be made to be usable by the requestor.

For those outside the Designated Community the OAIS can get Representation Information which may be enough for the Information to be enough. Alternatively, it may be necessary to Transform the Data Object in order for the Information to be usable by someone in this other group of users.

### Getting enough RepInfo

Thinking in general about the one requesting information getting enough RepInfo one can think of a number of steps:

1. Ask if it already has enough RepInfo e.g. checkUsability which would return
   1. “true” - which means that there is enough RepInfo so nothing more is needed
   2. “false” – which means that either more RepInfo [SEE (2)] OR a Transformation is needed [SEE (3)]

This is rather crude and does not provide a route to solve the issue of what extra RepInfo is needed, which is why one needs (2) or (3).

1. Options for getting more RepInfo:
   1. Get everything that the archive has – e.g. the whole RepInfo Network (RIN), at least as a network of pointers BUT
      1. If only using pointers then somehow the requestor must be able to recognise the identifiers, which might be OK if they use the same registry
         1. NOTE: if the Data Object is a FITS file then if the RepInfo is a PDF of the FITS standard then that would be OK because one would see that the RepInfo needed is a PDF reader, which I have.
         2. if the RepInfo is a piece of DRB then what the requestor needs to recognise is NOT that specific example of DRB but rather the requestor would be looking for the DRB processor which is used to extract information from the combination of the DRB file and the Data Object.
   2. As an alternative to getting the full RIN then one could get the nearest pieces of RepInfo i.e. the Structure, Semantic and Other RepInfo for the Data Object (which might itself be the DO for some RepInfo).
   3. If nothing will work then see (3)
2. Negotiate a Transformation by sending a list of acceptable formats for the DO – but would need more details such as how Semantics are dealt with. The archive would need to see if any of these formats are acceptable e.g. in terms of Transformational Information Properties.

## Design ideas

We want to design the system in a way that makes implementations easier and more widely usable.

1. For example if we can design software in a way that we can re-use the same piece of software in lots of different places in the system then that is a simplification because we don’t need to write so much software.
2. On the other hand we may realise that we need to change the software to deal with different circumstances e.g. we might want to use the system with different types of networks, or with different sources of information or different ways of putting information together, or using different programming languages. If we break the design into layers we stand a better chance of isolating these various dependencies so that, for example, we can replace the software that deals with one network (say TCP/IP) with that which deals with another network (such as some CCSDS protocols), without changing anything else in the system.
3. Another simplification is whether we can re-use some existing design/software which itself helps with (1) and/or (2)
4. Existing software, which is specific to an Archive, may be used to access information.

The ideas here are aimed at designing the system to be able to re-use (see (1)) some pieces of software to deal with understandability – which pops up everywhere unless what we send is very rigidly specified.

The CCSDS MAL speaks to (3) above, but there are lots of other possibilities. The use of the JAVA repository and registry API would also be in line with (3).

For point (2) there are lots of possibilities and design choices. The discussion about, say, an API for Provenance was to provide a proposal to address part of this. Different Producers, Consumers and Archives also strongly suggest that we try to define layers where they are obvious now which isolate the specifics of each while keeping the rest of the system unchanged. Other layers may be identified later, perhaps breaking an initial layer into sub-layers.

UML diagrams support the development of designs to support these ideas.

## (Arbitrary) Decisions

Send information in OAIS Information Packages

* + - * This allows us to send any information, so we can re-use software in many places.
      * If a package goes into an OAIS we can call it an SIP
      * If a package comes out of an OAIS we can call it a DIP
      * If the package goes to or comes from a place other than an OAIS, then we just call it a package.
      * If a package has everything needed for preservation of the (content) Information then it can be called an AIP.
      * Whether or not it actually has everything needed for preservation depends upon the Designated Community of the AIP, as defined by the OAIS, hence we have to decide:

1. The Package needs enough information about the Designated Community in order to decide whether there is enough RepInfo etc

OR

2.   The specific OAIS can declare that this Package is an AIP e.g. by setting a flag

- This is probably the easiest way to go

One very convenient type of RepInfo is software that can be used in an application. Here are some proposed standard APIs

For an Information Object, software should support an interface:

* + - * getDataObject (returns a DataObject – perhaps as ID (DOID))
      * getRepInfo (returns RepInfo – perhaps as ORIID)

For an Information Package, software should support an interface:

* + - * - getPackagingInfo (returns a ORIID)

 The PackagingInfo has the following interface:

* + - * getDataObject (returns a DataObject – perhaps as ID (DOID))
      * getRepInfo (returns RepInfo – perhaps as ORIID)
      * getProvenanceInfo (return ProvenanceInfo – perhaps as DOID -  or NULL if not there)
      * getReferenceInfo (return ReferenceInfo – perhaps as DOID - or NULL if not there)
      * getAccessRightsInfo (return AccessRightsInfo – perhaps as DOID - or NULL if not there)
      * getContextInfo (return ContextInfo – perhaps as DOID - or NULL if not there)
      * getFixityInfo (return FixityInfo – perhaps as DOID - or NULL if not there)
      * isAIP(return true or false)

For each of these Info Objects we can provide a default API e.g.

ProvenanceInfo   (see<https://docs.google.com/document/d/1YCyhBZKRP7IWhArdI3MnwahjZY5K93UsDZq-cWApYeI/edit?usp=sharing>)

Contex

- getContext (returns an Information Object which contains text which provides the context)

ReferenceInfo

- getURI()

AccessInfo

              Get????

FixityInfo

- getFixity(????)

RepInfo

# UML diagrams

TO BE UPDATED

UML defines a number of different diagram types, each designed to show particular aspects of the system as any one particular diagram cannot show everything.

The following diagrams are similar, but not identical, to the text above.

## USE CASES

The Use Cases shown in the diagram focus on the interactions between Information Users and Archives. The Information Producers are largely ignored here because

1. The Producer-Archive interactions are more complex
2. There are typically many fewer Producers than Consumers for any specific Archive,
3. Typically Producers would interact with a small number of Archives whereas Consumers would tend to interact with a large number of Archives.

Consumers will interact with familiar Archives, getting Information with which they are familiar (although this may not apply to Provenance Info etc). We show here that such Consumers use the Native Access Methods provided by the Archive. These Consumers will already know how to use the Information held by the Archive, and indeed they may be members of the Designated Community.

What we wish to focus on here are the interactions with other Consumers, those for which the Archive is unfamiliar. In this case we will need to provide extra Representation Information.

## SEQUENCE DIAGRAM

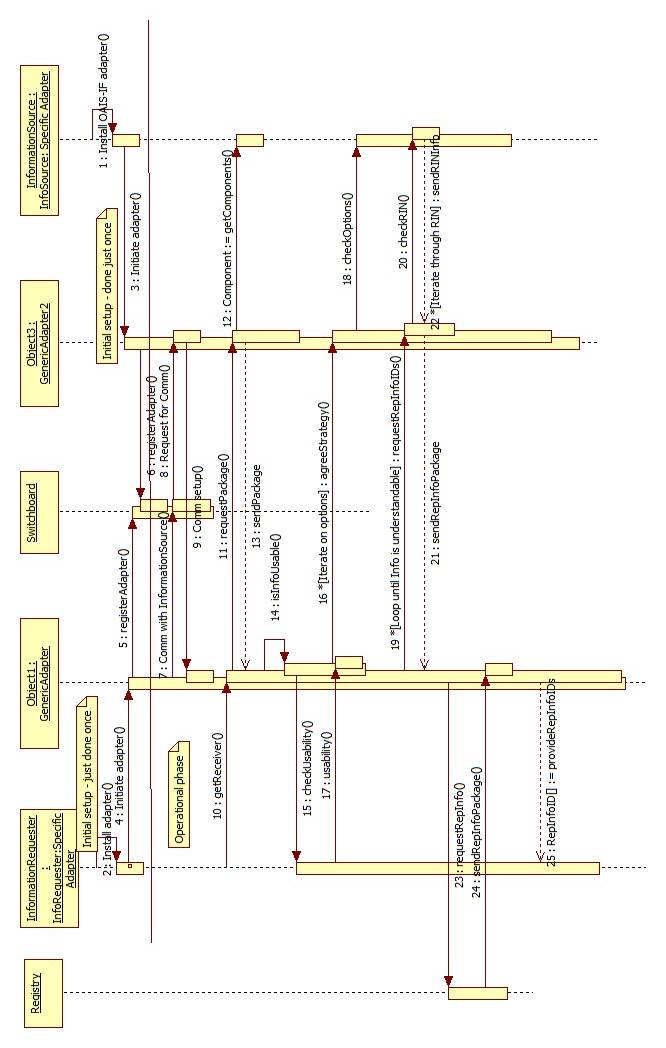


Figure ‑ Sequence diagram

(See [here](https://www.dropbox.com/s/7t9n00b7gu4md6i/SequenceDiagram-with-adapters-v3.jpg?dl=0) for diagram)

The details of the sequence are as follows:

|  |  |
| --- | --- |
| **Activity** | **Arrow number** |
| 1. Each will need to buy/borrow/create their specific portion of the adapter that depends upon its specific software design and capabilities e.g. |  |
| * 1. for an InfoSource - is it a simple file server which we can supplement with a database of RepInfo etc | [1] |
| * 1. for an InfoRequester – can it deal with tables and/or images or is it only for displaying documents etc | [2] |
| 1. They each then install the generic portion of the adapter – I think this can be common to all. | [3], [4] |
| This generic portion does the following things: |  |
| * 1. Find out about InfoSource using interface we define |  |
| * 1. Register itself and the InfoSource with the Switchboard e.g. how others can communicate with it – HTTP/port80 or HTTP/port 2678 or REST or CORBA or RMI etc etc |  |
| This is done just once for each source (or requester) | [5], [6] |
| 1. During operations the requester discovers somehow that it needs some information from an InfoSource – how this is done is not covered here. I think lots of work has been done by others, and OAIS does not really provide any guidance. |  |
| 1. InfoRequester generic adapter finds out from the Switchboard how to talk to the InfoSource in order to fit into the “OAIS-IF world”, and sets up the communication | [7], [8], [9] |
| 1. The specific adapter of InfoRequester passes on the request to the generic adapter which passes on the request to the InformationSource generic adapter | [10], [11] |
| 1. The generic adapter of the InfoSource receives the request and passes it to the specific adapter which gets the DataObject and RepInfo (at least the start of the RepInfoNetwork) and passes that back | [12] |
| 1. The InfoSource generic adapter creates an PackagedInfo object and sends that to the generic adapter of the InfoRequester | [13] |
| 1. The generic adapter of the InfoRequester asks the specific adapter if this is OK. If not then the generic adapter exchanges messages with the InfoSource generic adapter, which may need to talk to the InfoSource specific adapter | [14], [15], [16], [17], [18] |
| 1. PackagedInformation packages are send containing RepInfo until the InfoRequester is satisfied, or else a strategy is decided to use Transformation, if the InfoSource can do that | [19], [20], [21], 22] |
| 1. It may be that a Registry of RepInfo (essentially another InformationSource) is available and known to the generic adapter, and additional RepInfo may be obtained from that. | [23], [24] |

Additional clarifications:

* The Sequence diagram shows the InformationRequester asks for some information for which (by some mechanism not covered here) it has an identifier.
* On receipt of this request the InformationSender creates an InformationPackage, and then creates a PackagedInformation object by combining that with its PackagingInformation.
* On receipt of the PackagedInformation the Requester decides whether or not it has enough RepInfo to understand/use the data.
* If it has enough RepInfo then there is no need for further requests.
* If on the other hand there is not enough RepInfo then
  + The Sender and Receiver agree on a Strategy.
  + The list of identifiers of RepInfo which the Requester “understands” is sent to the Sender.
  + The Sender then sends a PackagedInformation object which contains the appropriate RepInfo.

### info.oais.interfaces.infomodel.repinfo

The following interfaces are incomplete ideas for navigating a Representation Information Network. The need for these is that one expects that RepInfo may be grouped. For example one might have alternatives e.g. software that is in the form of either Java OR C# - the choice is up to the user/user software. Also one might have pieces of RepInfo that should be used together e.g. one almost certainly needs the Structure PLUS the Semantic RepInfo.

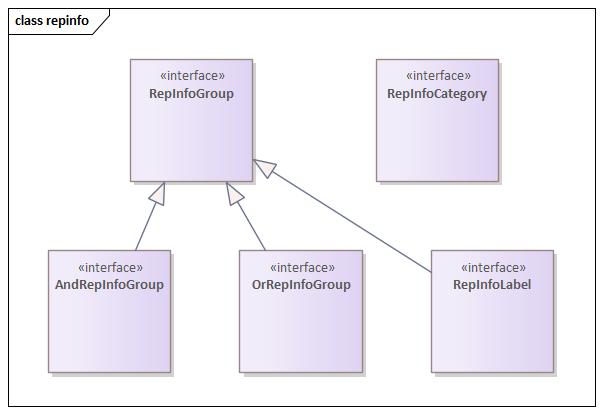


Figure ‑ RepInfo grouping interfaces

### Info.oais.interfaces.infomodel.repinfo. possRepInfoNetwork

A RepInfoNetwork is an Acyclic Directed Graph. There are a number of existing APIs to deal with Directed Graphs, however they are quite complex and so a simple set of interfaces are suggested below.

The existing libraries may be used to implement these interfaces.

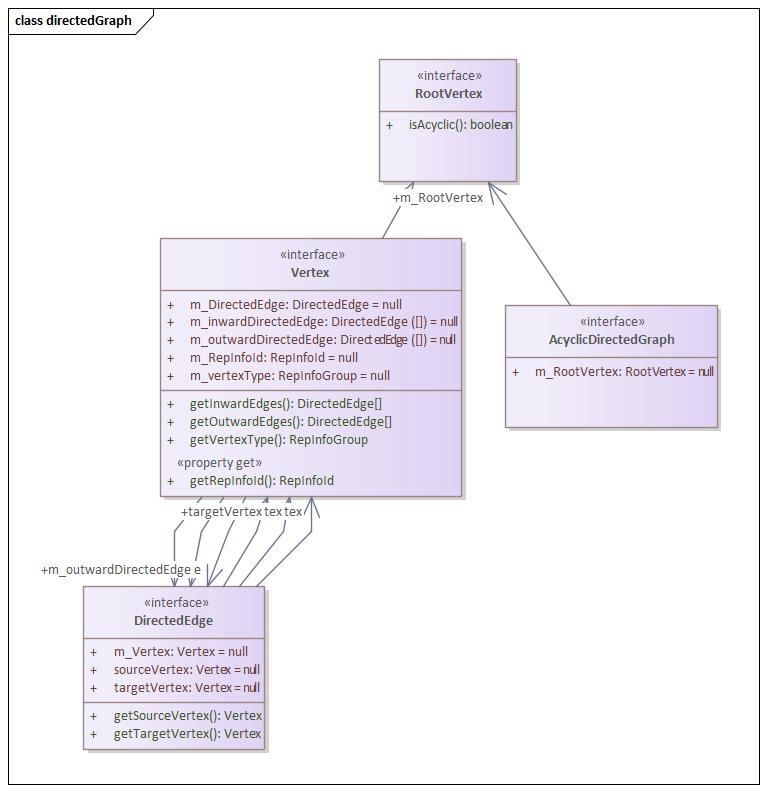


Figure ‑ Directed Graph Interfaces

The methods here do not include the set methods.

The aim is to identify Vertexes which have incoming or outgoing directedEdges. The RepInfoNetwork is a Directed Graph of DirectedEgdes and Vertexes.

Each Vertex may have a pointer to an associated piece of RepInfo, for which the outgoing edges provide the RepInfo. The edge may point to a Vertex which may simply be a way to a single piece of RepInfo or a group that may be an “AndGroup” or an “OrGroup”. The former must be taken together, for example Structure plus Semantic plus Other RepInfo, while the latter provides alternatives. Using interfaces derived from these ideas one can navigate the RepInfoNetwork, or serialise it as a serialised Java Object.

### info.oais.interfaces.oaisif

The following interfaces make an INCOMPLETE start at trying to define interfaces for the adapters in a way consistent with the Sequence Diagram.

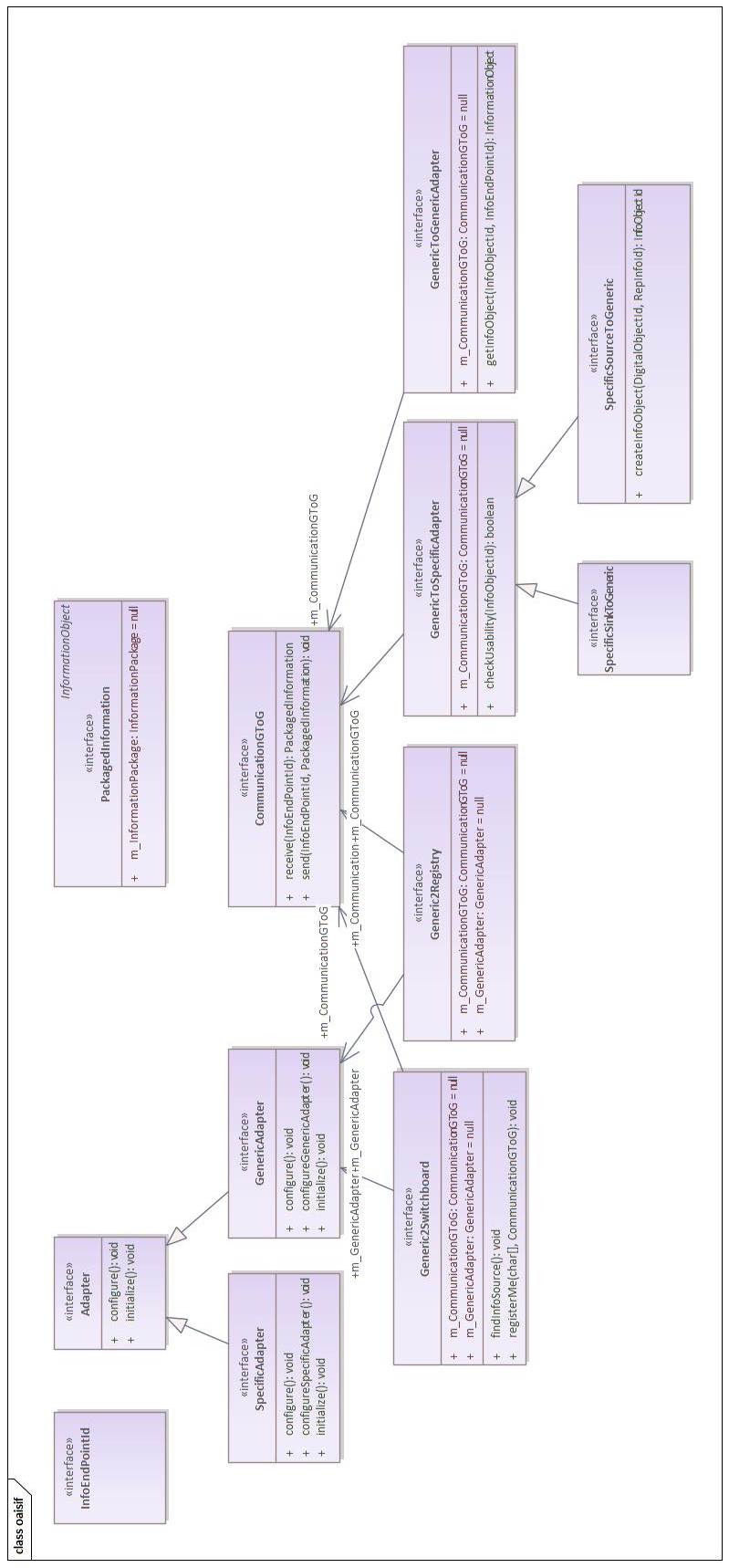


Figure ‑ OAIS-IF specific interfaces

## COMPONENT DIAGRAM

Component diagrams can be used to model the logical components that make up a system. They can be used to model the applications of an organization including their Provided and Required Interfaces and the information that is exchanged between the interfaces.

The diagram below shows the components shows that any system will have, besides the Native System, a Specific Adapter and a copy of the Generic Adapter. The interfaces are indicated. A “System” represents any of the entities which exchange information using OAIS-IF.

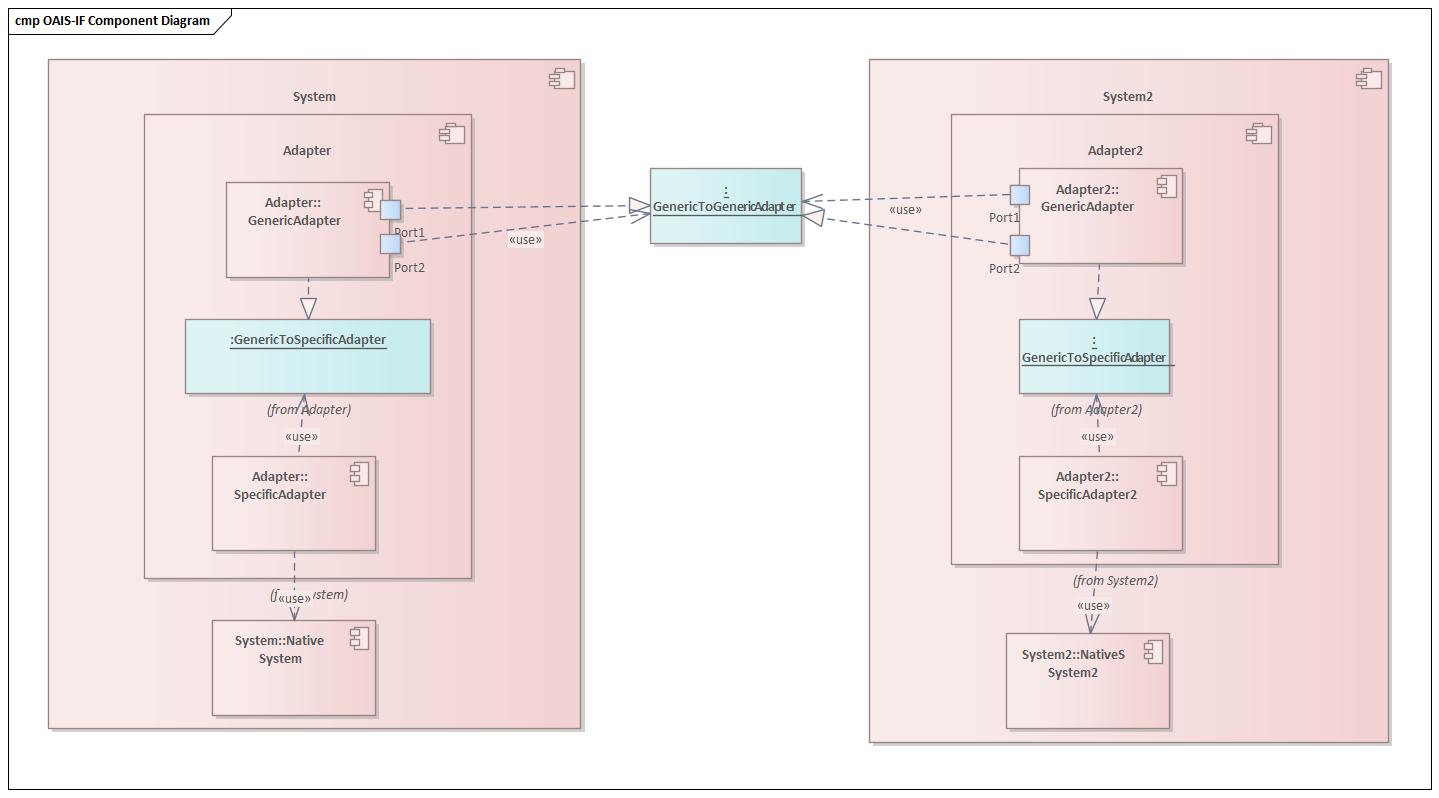


Figure ‑ Component diagram

There is a negotiation, shown in the Sequence Diagram to decide on whether, in order for the Consumer to be able to use the information returned, the source of information can either:

1. Send additional RepInfo to extend the RepInfo Network so that the Consumer will have enough RepInfo. The Producer may use a Registry/Repository of Representation Information to supplement the RepInfo it already has.
2. The Data Object could be Transformed to match what the Consumer can deal with.

To implement this, CCSDS (or one or more agencies) can provide Reference Implementations. A Domain Archive could initially simply use the Reference Implementation. It would only need to create the Interoperability Table, and supply the Applications (if any) to perform the Transformation, which should be chosen based on maintaining the Transformational Information Properties.

## DEPLOYMENT DIAGRAM

The Deployment diagram indicates (some of the) the software which runs on the physical nodes (e.g. computers).

This shows nodes which match the OAIS-IF scenarios discussed in section 3.

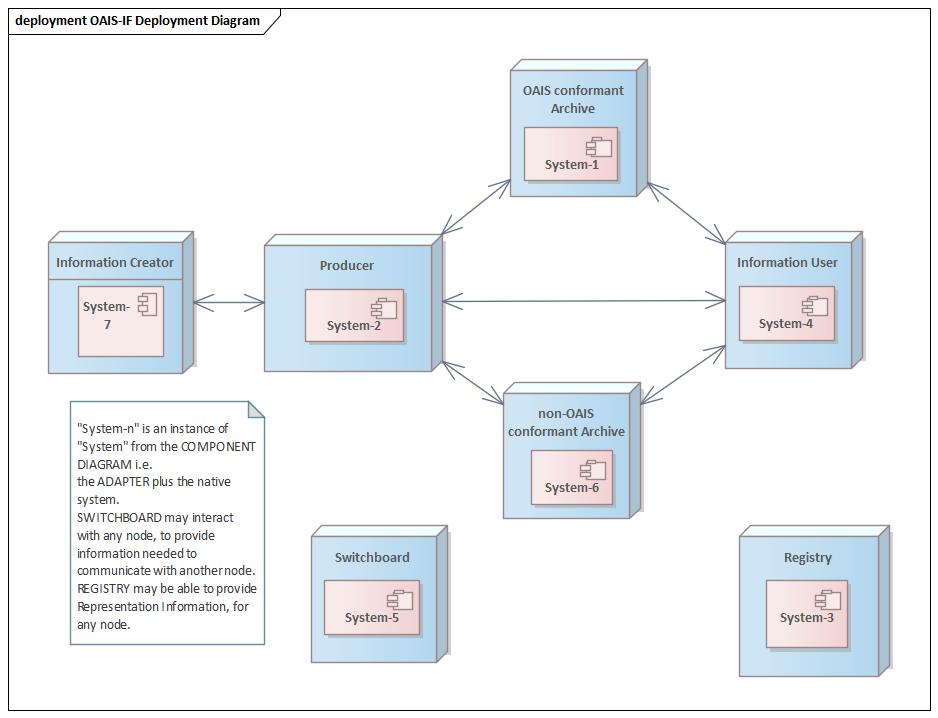


Figure ‑ Deployment diagram

# References

[1]    A concept paper on Provenance – see<https://docs.google.com/document/d/1YCyhBZKRP7IWhArdI3MnwahjZY5K93UsDZq-cWApYeI/edit?usp=sharing>

[2]    A concept paper on Representation Information -<https://docs.google.com/document/d/1TcYIDKc9WMdyK1POSuitQjdWc5VtlL-YG8Qkt8c8V18/edit?usp=sharing>

[3]    A concept paper on OAIS-IF ideas – see [https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit#](https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit)

1. https://www.hdfgroup.org/solutions/hdf5/ [↑](#footnote-ref-1)
2. See http://starlink.eao.hawaii.edu/docs/sun92.htx/sun92.html [↑](#footnote-ref-2)