442-TP-001-001

External Data Provider Options

Technical Paper--Not intended for formal review or government approval.

August 1996

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

Lynne M. Case /s/ 8/22/96

Lynne Case Date
Interoperability and Data Management Lead
EOSDIS Core System Project

SUBMITTED BY

Rick Kochhar /s/ 8/22/96

Rick Kochhar, Release B Manager Date
EOSDIS Core System Project

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

ECS has requirements to provide core services for data management, data processing, and data archive and distribution of EOSDIS products, as well as numerous requirements for interoperability to other data centers outside the ECS Distributed Active Archive Centers (DAACs). There are also requirements for ECS to enable the separate use of these components within the larger Global Change Data Information System (GCDIS). This paper addresses some of the options available for external providers to become part of ECS. In particular, it addresses how external providers can make their data and services available to ECS users by reusing ECS software at their sites. To a lesser degree the paper also considers how an external provider can provide ECS data to its users. The paper is written from the perspective the Release B design. It does not discuss in detail the direction of future releases, except for a brief discussion in the last section of the document.

Keywords: GCDIS, external provider, API, gateway, LDAS, DIM, LIM, Advertising, evolutionary, federation.

This page intentionally left blank.

Contents

5. EOSDIS One-Stop Shopping Connection Scenarios

6. Summary

7. Future Directions

Figures

Tables

This page intentionally left blank.

1. Introduction

1.1 Purpose

NASA's Earth Observing System (EOS) is a long-term, multi-disciplinary research mission to study the processes leading to global change and to develop the capability to predict the future evolution of the Earth system on time scales of decades to centuries. The EOS Data Information System (EOSDIS) provides computing and network facilities to support the EOS research activities, including data interpretation and modeling; processing, distribution, and archiving of EOS data; and command and control of the spacecraft and instruments.

Although EOSDIS will eventually contain an enormous amount of valuable Earth science data, there are other sources of information that are essential to the study of climate change. Of critical importance are holdings of other Global Change agency organizations, such as NOAA and USGS, and other international organizations. The Global Change Data Management Working Group (GCDMWG) is currently in the process of defining the Global Change Data and Information System (GCDIS) intended to provide linkages between data services through a common set of interoperability services. In the international arena, the Committee on Earth Observation Satellites (CEOS) is developing a Catalog Interoperability Protocol (CIP) designed to support interoperability of data systems. NASA is actively participating in these efforts.

In addition, there is also a growing interest by earth scientists in the possibility of developing information systems for earth science data which not only encompass the major data repositories but also enable users to take an active part in the information system, by providing data/services to the system. This approach, referred to as a User Data and Information System (UserDIS), seeks to encourage the scientific return from the investment in data and information systems by ensuring that the scientists are an integral part of the system.

Although NASA does not have the responsibility for developing either GCDIS or UserDIS it wants to make sure that its development of the EOSDIS Core System (ECS) can support both of these evolutionary paths. This implies taking an architectural direction which opens ECS so that it can be included within wider data systems and identifying architectural components which ECS might contribute to these systems.

The purpose of this document is to describe how the ECS Release B design can enable interoperability with external data providers that are part of the broader GCDIS or UserDIS community. It summarizes the results of the External data provider Design Issue Team that was chartered to examine and propose changes to the ECS design to better support these types of data providers.

The scope of this document is limited to a discussion of ECS interfaces and components that can be reused to enable external data providers to make data and services available to the EOSDIS community (and vice versa). It does not focus on ECS interfaces and components that might be reused to support the ingest and processing of data products.

1.2 Organization

This paper is organized as follows:

- Section 2 provides an overview of the evolutionary requirements placed on ECS and describes how they are addressed by the ECS architecture.
- Section 3 summarizes the system options available for supporting external data providers.
- Sections 4 and 5 present representative scenarios of how these options may be used to incorporate extended providers into the system.
- Section 6 summarizes the options, the capabilities provided through each option, and the Commercial Off The Shelf (COTS) products needed for implementation.
- Section 7 describes the future directions of external providers in the post-Release B development.

Questions regarding technical information contained within this Paper should be addressed to the following ECS and/or GSFC contacts:

- ECS Contacts
  - Lynne Case, Interoperability and Data Management Subsystem Lead, (301) 925-0359, lcase@eos.hitc.com - for technical questions regarding the contents of this paper.
  - Paul Fingerman, System Evolution/Release C Manager, (301) 925-0502, pfingerm@eos.hitc.com - for further information on the prototype plans discussed in Future Directions section.
- GSFC Contacts
  - Ken McDonald, ESDIS IMS Manager, (301) 286-8766, ken.mcdonald@gsfc.nasa.gov

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Information Technology Systems
1616 McCormick Drive
Upper Marlboro, MD 20774-5372

2. ECS Architecture Overview

In order to ensure that ECS is capable of enabling component reuse and interoperation within the GCDIS and UserDIS environments, NASA has specified a set of requirements that address these areas. This section summarizes these requirements and describes how the ECS architecture supports them.

2.1 Evolvability Requirements for External Data Providers

The requirements for ECS to enable support of external data providers are defined in the Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (423-41-02). They are summarized here in Table 2.1-1. These requirements mandate that ECS: be capable of supporting access to heterogeneous services; provide ECS client access to non-ECS services; permit users to invoke combinations of ECS and non-ECS services; enable ECS to operate as a federated unit within the broader GCDIS environment; enable interoperation with International data systems; and provide configuration controlled Application Program Interfaces (APIs) that enable ECS components to be reused within the GCDIS environment.

Table 2.1-1. ECS Requirements for External Data Providers

Number	Requirement Description
EOSD5000	ECS shall enable the addition of other data providers, e.g., DAACs, SCFs, ADCs, ODCs, which may: - provide heterogeneous services, i.e., services in support of EOS which may be less than or different than ECS services - be connected with varying topologies - have variable levels of reliability or operational availability.
EOSD5010	ECS shall enable extended provider support, i.e., client access of data and services at SCFs and DAACs, as authorized, without distrinction to the client.
EOSD5040	ECS shall enable the combination of services from ECS and other data providers in arbitrary, i.e., non-predefined, ways as needed by users to conduct EOS science.
EOSD5060	ECS shall enable interoperability with equivalent International systems, e.g., European and Japanese systems, to support the following: a. Browse services b. Data retrieval services.
EOSD5100	ECS shall enable evolution of ECS to be a federated unit within GCDIS, e.g., GCDIS data centers should not have to negotiate different interfaces with each DAAC.
EOSD5110	ECS shall enable the separate use of ECS data management, data processing, or data archive and distribution software components by a GCDIS data center. The GCDIS data centers will have full responsibility for integration of those components within their environment. Interfaces between the components must be developed to serve the mission of EOSDIS, but be made available for a GCDIS data center.

2.2 ECS Architecture Support for External Data Providers

ECS provides a robust distributed computing infrastructure that enables the EOSDIS Distributed Active Archive Centers (DAACs) to operate with a high degree of site autonomy yet provide consistent quality of service and system-wide services to EOSDIS users. Examples of these system-wide services include:

- Security - users need only login once to gain access to any authorized services at all DAACs. Each DAAC controls which users are authorized to access its services;
- Distributed Search - users can pose a single query that searches data at one or more DAACs. Data location is transparent to the user;
- Request Tracking - users can track the status of a request, even if that request involves multiple DAACs; and
- Accounting and Billing - users are provided with a uniform mechanism for accounting and billing for the Cost of Filling a User Request (COFUR).

Thus, while EOSDIS can be viewed as consisting of a federation of DAACs, there are extensive requirements that mandate consistent quality of service and system-wide capabilities across the DAACs. However, as one looks at the broader GCDIS and UserDIS environments, the requirements for consistent quality of service and system-wide services must be relaxed. Different agencies and individuals will have differing objectives and resources that will dictate their willingness or ability to support system-wide services. The ECS architecture, therefore, must enable support for a wide range of external data providers that may exist in the GCDIS and UserDIS environments. Some of these providers will be willing and capable of supporting ECS system-wide services (e.g., distributed search) while others will not.

The ECS architectural concept is shown in Figure 2.2-1. It can be divided into three layers: the client layer, the service provider layer and the intersite layer. Individual sites, which may host one or more of these layers, are heterogeneous and autonomously managed. The user layer is characterized by client environments, which may be interactive (e.g., workstation graphical interface, World Wide Web interface) or process environments (e.g., analysis algorithm). The intersite layer is characterized by a set of distributed services which assess user needs against service offerings and connect the user with appropriate service providers. Finally, the service provider layer is characterized by organizations who choose to provide a set of services related to data collections or to computer resources that they can offer. This includes the traditional data center concept and also external data providers, whether commercial or government related (e.g., education specialists). Since the service provider layer must allow autonomous management and development, the details given here are limited to those which allow sites to interoperate. The architectural concept then, is in essence the interoperability infrastructure (the Intersite Architecture) and how the user and data provider services interface to this infrastructure.

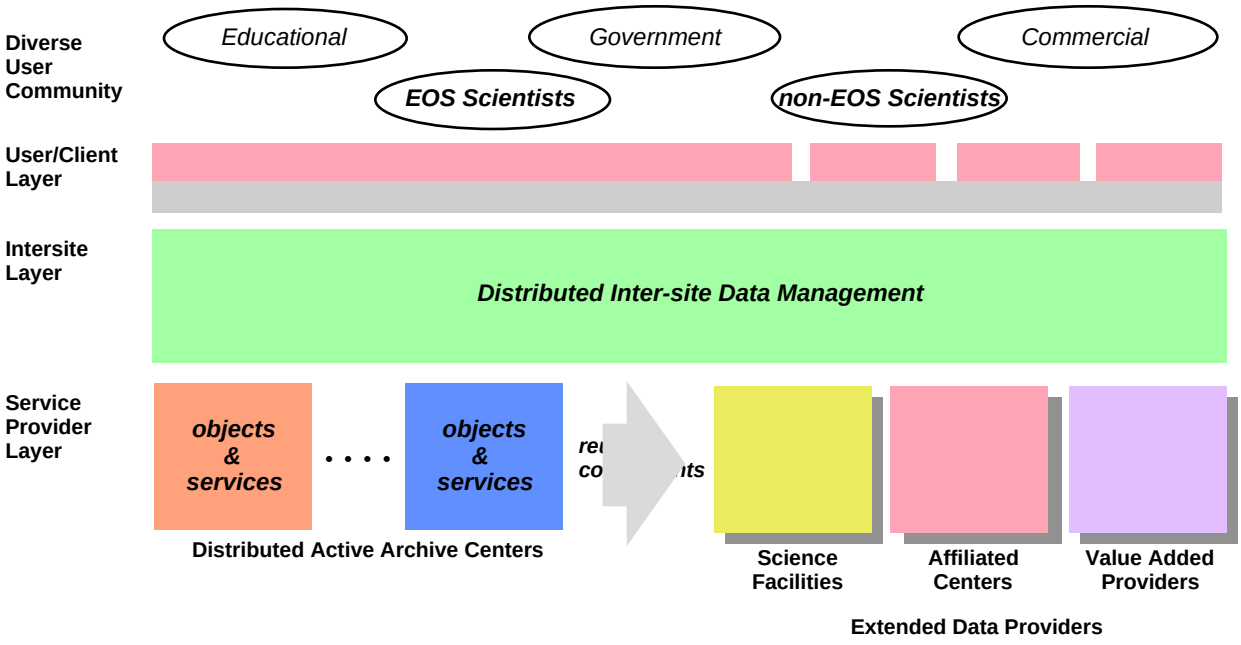

Figure 2.2-1. ECS Conceptual Architecture

2.2.1 Intersite Architecture

The intersite architecture is necessary to connect the services offered by providers to needs expressed by users in terms of requests to the system. Three classes of software component reside in this layer:

Advertising Service: The services offered by providers on the network need to be advertised to users. The advertising service is also used by other components in the intersite architecture to perform their function.

Data Dictionary Service: This service provides access to the descriptions of data items and objects as used by one or several of the data access services advertised on the network. This service is invoked by the user or other services to correctly formulate or validate a service request.

Distributed Information Manager: Where multiple sites are needed to resolve a request then an intersite search service is required to manage the process. The service will break up the query if necessary and generate a plan containing sub-requests which will be processed at individual sites. The sub-requests will generally be characterized by queries and operations, where an operation is usually some manipulation of results to provide output in the form or context requested by the user.

2.2.2 Provider Interfaces to the Intersite Architecture

Each provider site chooses, or is mandated by its management, to provide certain services to the system community, or some part of that community. In many cases a service would be related to data set(s) held at the providers site, but it is not mandatory; a service potentially could access data held at another site, or even provide a service for data passed to it from another site as part of a request.

The main issues to be resolved in the interfacing of a provider's services to the entire system are briefly described in this section; they are:

- autonomous internal organization
- advertising services to users
- support for searching
- support for notifications to users when new data and/or services are available - subscription services
- support for incompatibility management
- the role of data servers.

Provider sites should be allowed to *autonomously organize* and manage their internal services and data to permit political and technical flexibility and therefore a definition of how the provider organizes their data and services is not part of the architectural concept. What matters is what the site offers for external access and how it can be accessed externally. The following discussion defines how the provider architecture is represented to the system.

Sites *advertise* their services to the system; services which are not advertised do not exist from the system perspective. The advertisements are managed by the advertising service within the interoperability infrastructure and describe what the service is and how it can be accessed. Since data is essentially equivalent to a service (a user can't access data at a provider site without some sort of "fetch" service), advertisements can refer to data and services. Advertisements must support service representations interpretable by end users and by other, automated clients.

Sites which offer *data searching* must have an external interface for accepting searches (e.g., from the Distributed Information Manager (DIM) or directly from a client), and a service for processing these requests. This is called the Local Information Manager (LIM), and is equivalent to the Distributed Information Manager, in that it resolves inexact search requests into exact queries which can be placed on individual data servers. This might involve some interaction with the user.

The open nature of the architecture described here may result in potential *incompatibilities* between the user tools in the client layer and the services provided. Characteristics of data, tools, services, etc. which are essential for determining incompatibility are captured in an

'interoperability profile' which is stored in the advertising service. This profile is specific to the type of object under consideration (e.g., data format profile). Using these profiles it is possible to warn users of potential incompatibilities, and offer advice on how to mitigate them.

Conceptually data at a site is organized into one or more collections of related items. Each collection forms a data set which will contain both data and meta-data, the discrimination between these being provider and data set specific. To each data set one or more services are attached; the services may operate on all types of data in the data set or only one part of it, e.g., a search service on a relational database management system (DBMS) inventory table. Descriptions of these services and the parts of the data set they operate on are passed to the advertising service, establishing a 'data scope' against which users requests can be evaluated against. These services are called 'type' services in the concept since one type of service may be related to all data of that type (e.g., a text query service could be used for all text data within one data set and across all data sets at one site). Careful design of the type services should mean that they are adaptable by other providers for similar data. It is possible that a particular type of data would have more than one 'type' service associated with it, (e.g., two different text query services to support different access protocols) and that a type service would deal with more than just a single data type (e.g., a search service on an Object-Relational DBMS could deal with inventory data and the associated browse data).

Since external data providers may have different data organization concepts and may employ different data management and storage technologies, the service provider layer includes a Gateway component that performs the mapping from the ECS environment to the external data provider's environment. The Gateway component performs low-level prototcol translation, security mapping, application level protocol translation (effectively all 7 layers of the OSI model), and database schema mapping.

2.2.3 User Interfaces to the Services

As for the provider layer, the architecture does not mandate how the user interfaces should work, only that they are compatible with one or more of the protocols that the interoperability infrastructure supports. The GCDIS / UserDIS concept seeks to encourage community development of components, and it is likely that the client layer is one of the areas where this will have the most impact.

Three categories of access to the interoperability layer are considered in the architectural concept:

General Access Interfaces: These interfaces are applicable to a group of similar services across all data in the network (e.g. search and order). This type of interface will be the main access to the services of EOSDIS and many of the large archives. The general interfaces can be customized in their operation by specific information from the service provider (e.g., vocabulary), etc.

Specialized Access Interfaces: The above approach would support dynamic modification of an existing interface, but would not support special interfaces for specific services, e.g. an interface which is particularly oriented towards the coincident location and analysis of sea surface temperature ‘images’ and sub-surface profile measurements of temperature and salinity. In this case the service provider provides a software module which provides a completely specialized interface, configured to the specific service/provider. The software module is stored in the advertising service along with the provider’s service advertisement. The software module can be dynamically downloaded and configured on a client workstation, after which the user can invoke the service.

Object Access Interfaces: Finally, objects resulting from previous queries should be capable of initiating further service requests. For example a search of an image inventory might result in a results object which contains the inventory records matching the query and a reference in the object which would enable a user to automatically initiate a browse service and review the image being referenced by one of the inventory records.

3. External Data Provider Options Summary

3.1 External Data Provider Options

The ECS architecture and design provides for two types of interoperability with the ECS components. The first type is a (extremely) loosely coupled interoperability through the publishing of available services. In this type of interoperability the external data provider supplies to ECS some information to access services located at the external site, but in most cases the interaction between the user and data provider does not require any ECS infrastructure after the initial installation of the service. The ECS Advertising Service provides the data providers with the ability to publish their services and distribute direct interfaces.

The second type of interoperability requires a more tightly coupled interaction between the user, the ECS infrastructure, and the data provider supplied services. In this type of interoperability, the ECS client can issue one search that can access both ECS sites and sites of external data providers. The data provider must supply both a description of the services provided to the Advertising Service and a description of the conceptual schema[1] of its holdings to the Data Dictionary Service. This type of interoperability is known as "one-stop shopping."

The remainder of this section will define the options, while Sections 4 and 5 go into more details of how to implement these options.

3.2 EOSDIS Advertising Network Option

The ECS Advertising Service is composed of a database of services and products and a WWW interface to the database for user searching.[2] The EOSDIS Advertising Network is defined as the list of providers listed in the Advertising Service database. The network might include EOSDIS DAACs as well as international partners (IPs), Affiliated Data Centers (ADCs), and other external data providers.

There are two service advertisement types available: a MIME (Multipurpose Internet Mail Extension) type and a signature type. The MIME type allows for advertising internet services using URLs. The signature type is a service which describes a function call of some kind. One can think of it as the name and arguments of either a C function or C++ member function. The signature type is used primarily by the ECS components such as the Data Server to describe the operations on ESDTs. It can also be used by other data providers as described below.

As stated in section 2, the Advertising Service relates each service to the client component which will execute that service. The client component can be an ECS supplied client or one supplied by a DAAC or other data provider. The Advertising Service contains a description of the client and

---

1 Defined in Section 3.3

2 Programmatic access is also supported via a server API. Refer to the Application Programmer Interface (API) Interaction Definition Document (IDD) 819-RD-001-001 [API IDD] and the Release B CSMS/SDPS Internal Interface Control Document (313-CD-006-002)

how it should be installed on the user's workstation. For MIME type services, it is assumed that the client is a Web browser and the user must have a Web browser if he/she is currently looking at the Advertising Service. For signature type services, the Advertising Service should contain an installer program that would get invoked to download and install client software if necessary.

The three Advertising Service options can be defined as:

1. *Internet Service:* This option allows the service provider to specify a MIME type. Users searching advertisements in the Advertising Service can connect directly to these MIME type services through the Web browser they are already using to look at the advertisements. Installation of a MIME type to the ECS client Desktop will create an object on the Desktop which will spawn the Web browser with the appropriate URL to connect to the service automatically.
2. *Data Provider Supplied Client:* This option allows the data provider to supply a client program that will work with a signature type service. The service may be a service resident at the data provider's site or it may be a service resident within ECS. For a service resident at the data provider's site, the ECS infrastructure may not be involved with the client-server connection, after the initial Advertising Service installation. For a ECS services, data providers can supply clients that use the ECS infrastructure to provide a specialized client interface for an ECS service.
3. *ECS Client with Local Data Access Service (LDAS) at Provider Site:* In this option, the data provider obtains an LDAS from ECS. The LDAS is a portion of the ECS Data Server Subsystem which does metadata ingest, validation, and searching. It is typically used by Science Computing Facilities while testing their algorithms to validate the metadata formats that are output, but could also be used by external providers to advertise metadata holdings.[3] The data provider configures the LDAS, advertises the services, and exports the schema to the Data Dictionary service. The corresponding advertisement would be a signature type similar to those advertised by ECS' Science Data Servers that specifies the ECS client, (Earth Science Search Tool - ESST) as the client software. Users who install this option will get a desktop object which points the ESST directly to the external data provider site.

3.3 EOSDIS One-Stop Shopping

One-Stop shopping is defined as the use of the ECS client to search and access data regardless of the architecture, query language, or database management system at individual sites. ECS provides this capability through a layered architecture that can hide the underlying languages and protocols from the user and the client applications. While the Advertising Service provides the capability for data providers to supply services to users, there is a user burden involved in using multiple user interfaces to access the data. The one-stop shopping options allow the external data provider to appear as if it was one of the ECS DAACs. The user can use one client to seamlessly access both ECS DAACs and other data providers.

---

[3] Though the LDAS only does metadata operations, it also contains the client interfaces to perform other functions such as data order and browsing. Other Data Server pieces would be available to the external providers, but are not discussed in this paper.

3.3.1 Schema Integration and Federation

In order to achieve one-stop shopping there must be some consistency in the representation of attributes available at each site. The Data Dictionary Service database contains the "conceptual" schema of each site and of each search agent (i.e. DIM, LIM, etc.). As stated in the architecture overview section, the data at the Science Data Server (SDSRV) is organized in collections. The conceptual schema in the Data Dictionary Service relates the collections to attributes that are searchable at the SDSRV and other search agents such as the LIM or V0 Gateway. The Data Dictionary Service does not maintain the actual physical representation of the underlying data servers.

The meanings of attributes and terms must be consistent across sites for the one-stop shopping approach to work. This is accomplished by mapping terms at the lowest level in the local data model to the ECS metadata standard (currently documented in 311-CD-002-004). Sites may have their own physical interpretation of the attributes, but the LIM or Gateway must map the conceptual ECS standard term to the physical representation at the lower levels. In other words, each data provider does not have to have the same physical database structure as all other sites, but Data Dictionary entries must map from the ECS metadata standard to the internal structure at the data provider site (and vice-versa).

3.3.1.1 Schema Integration

In the ECS System Design Specification (SDS 194-207-SE1-001) the term schema integration was used in relationship to the LIM. The primary purpose of the LIM is to present an ECS metadata and protocol standard view of the underlying data center. At ECS DAACs, this is rather simplistic since the schemas normally already conform to the ECS metadata standard within the SDSRV. The LIM is still an important piece of ECS software if the DAAC wishes to present any non-standard schemas as part of the ECS one-stop shopping network. At external data providers sites, the LIM is used to map local variations on the ECS standards. When the local schema has been mapped to the ECS metadata standard, it is said to be "integrated." The one-stop shopping scenarios in Section 5 describe the tools provided to the data provider to perform the integration.

3.3.1.2 Schema Federation

In the SDS, the term schema federation was used in relationship to the schema seen by the ECS DIM. The DIM cannot correlate attributes that are not part of the ECS metadata standard, since it does not know whether it is semantically correct to relate these attributes. For example, if one site has a product specific attribute called CloudCover and other side has an attribute named CloudCvr, there is no way for the DIM to determine that these are equivalent attributes. As a result, the DIM will only accept and execute well-formed searches which access either ECS metadata standard attributes or product specific attributes that are matched correctly with the collections (and thus the site) that contain that attribute. The DIM views the system as one schema with multiple collections of the data across the network. In order to provide services available through one-stop shopping, the data provider must become part of the schema federation, either by adopting the ECS metadata standard as its own, or by mapping its schema in the Data Dictionary available to the ECS software at the external data provider's site.

3.3.2 One-Stop Shopping Options

Given that each site that becomes part of the one-stop shopping network must adhere to the ECS metadata standard and the LIM maps between site specific definitions and local definitions, it follows that each external data provider becoming part of one-stop shopping must install and configure a LIM. The following options are available to become part of one-stop shopping:

1. *LIM Accessing a Custom Gateway:* This option gives the data provider two pieces of ECS software that already work together. The LIM is used to control access from the ECS world to the data provider world. The data provider also receives the V0 Gateway which in many cases can be customized to provide a gateway to an existing system at the data provider site. More information on this option is available in Section 5.1.1.
2. *LIM Accessing an LDAS:* This option gives the data provider two pieces of ECS software that work together "out of the box." Customizations would be applied to the LDAS software to provide for local site deviations from the ECS standard design. More information on this is provided in section 5.1.2.
3. *Modification of a LIM:* In this option, the LIM code is modified to talk to the underlying data server in much the same way as the Custom Gateway approach. The main difference in this approach is that there is only one process that must be maintained operationally. The data provider will be given source code for all of the LIM except for the client-server interface to the ECS internal network. For this, object code will only exist, along with documentation fo any public interfaces available via the object library. This allows for a stable interface between ECS and all data providers.

3.4 External User to ECS Options

The options described in the previous sections were for the ECS user to obtain data and access services at an external data provider site. There are other forms of interoperability in which external users wish to gain access to ECS data and services through their current user interfaces. There are two options to achieving this goal:

1. *Use of ECS Public APIs:* In this option, the external system would use the APIs documented in the API IDD. The second version of this document (available Sept. 1996) will describe how to write code using the interfaces to the DIM, LIM, GTWAY, DDICT, and SDSRV. It describes the OTS software requirements of writing to these APIs. Therefore, this option will not be discussed any further in this document.
2. *Customize the ASTER Gateway:* ECS is building a two-way interoperability gateway to the ASTER Ground Data System in Japan. This gateway is an extension of the V0 Gateway. The V0 Gateway provides access from the V0 client to a Science Data Server at an ECS DAAC. The ASTER Gateway will provide access from a modified V0 client to all components of the ECS system in the same manner that a DIM would support. The ASTER Gateway also provides the reverse direction as well; i.e. from an ECS Client to the ASTER GDS system. An external data provider could adapt this gateway to support two-way interoperability with ECS.

4. EOSDIS Advertising Network Scenarios

4.1 Advertising Off-the-Shelf Software Requirements

Table 4.1-1 summarizes the off-the-shelf software required with each Advertising Service option. These options are very low cost to the data provider, but the disadvantages and advantages of each option must be carefully examined by the data provider to make sure that the disadvantages don't outweigh the low cost of the solution.

Table 4.1-1. Advertising Options OTS Software Requirements

Internet Service	Data Provider Supplied Client	LDAS Advertisement
Web browser (public domain ok)	Web browser (public domain ok)	Web browser (public domain ok)
Web server (public domain ok)	FTP service for downloading the client software	Illustra
		OSF/DCE Client
		OODCE App Dev. Toolkit
		RogueWave Tools.h++

4.2 Internet Service Advertisement

The Internet service advertisement is the easiest of the external data provider options. Once the data provider has the Internet service established at their site, the advertisement is as easy as filling out a couple forms on a Web browser. The steps for inserting an advertisement in the ECS Advertising Service at an ECS DAAC are as follows:

1. Connect to Advertising Service WWW server, using published URL.
2. Follow the submissions link.
3. Select the Service link, which will give the user two options. The user should select the MIME type option.
4. Fill in the form. The form requires that the user submit a contact ID and a provider ID. If the data provider has never submitted any services before, a provider form will have to be filled out in addition to the service form. The required provider information includes: a title, a description, a group (EOS or OTHER), a role name, and an organization name. The required service advertisement information for a MIME type includes: title, description, URL to the service, a group, an expiration time, a contact ID, and a provider ID.
5. Select the Submit option. If all the required information is filled in, an ID for the submitted service is returned to the user.

Figure 4.1-1 shows what happens when an advertisement is submitted. The CGI script that is linked in with the Web server, inserts the advertisement in the database. A moderator is notified that a new advertisement has been received. (This is done through subscriptions but is not shown on the diagram since the mechanism is an e-mail message.) Once the moderator approves the advertisement, the advertisement can be seen by users searching the Advertising Service. The picture shows what happens when the data provider selects the Submit button in step #5 above.

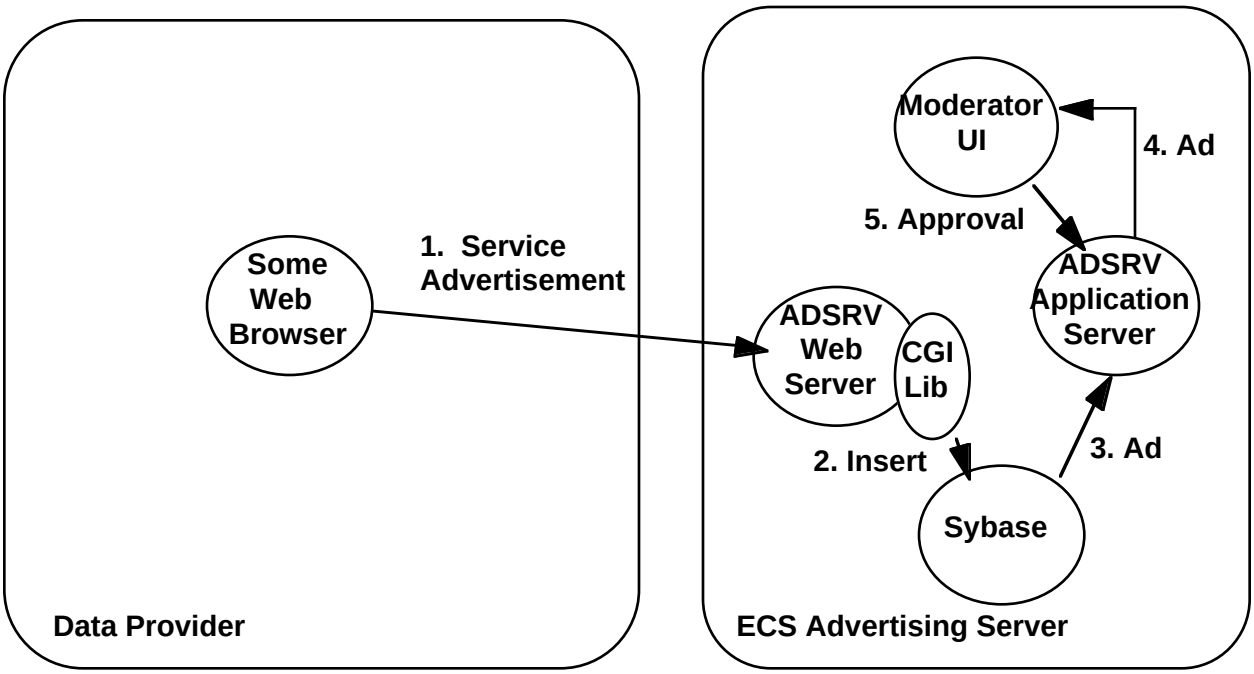

Figure 4.2-1. Advertising a Web Service

Figure 4.2-2 shows what happens when the user searches the Advertising Service and finds the a MIME type service advertisement. When the user selects the install option, the Installer retrieves the service information. It determines that the service is a MIME type and retrieves the URL. A desktop object is created (if desired by the user). The desktop object specifies an action that will start the Web browser with the URL supplied in the advertisement.

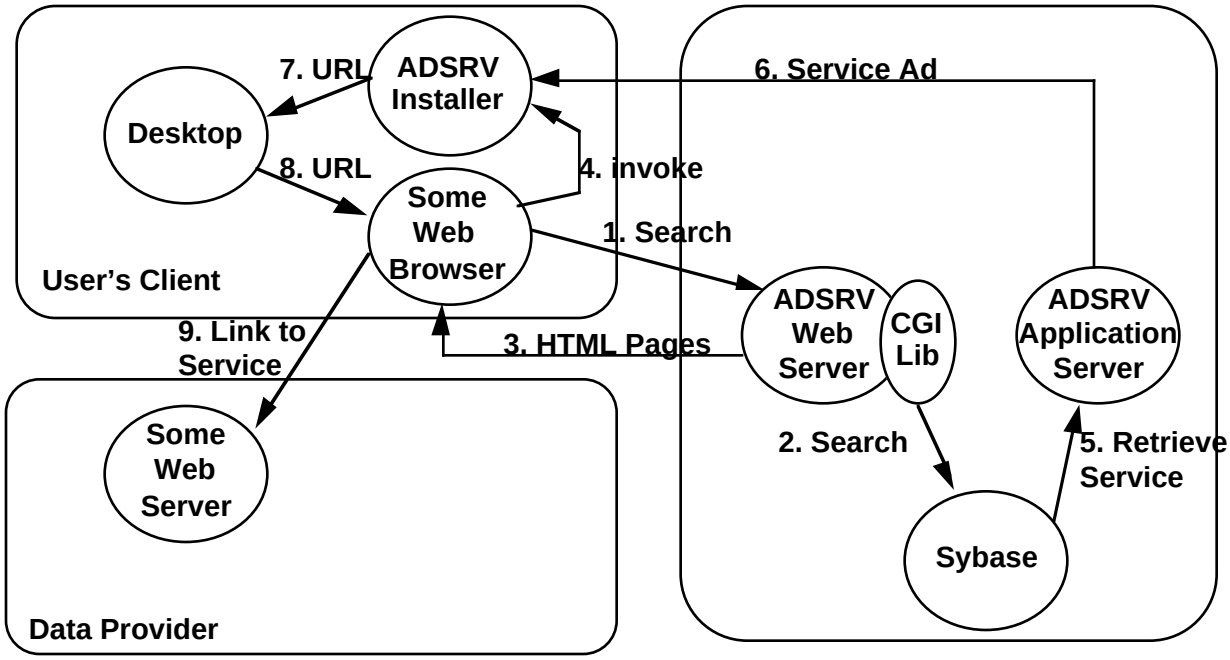

Figure 4.2-2. Installation and Invocation of a Web Service Advertisement

4.2.1 Internet Service Advertisement Features

Table 4.2-1 summarizes the features of the Internet Service Advertisement option and the disadvantages of this approach.

Table 4.2-1. Internet Service Advertisement Features

Feature	Disadvantage	Advantage
Discovery through Advertising Service	No distributed search capability at lower than collection level	Wide visibility: Surfing is common practice on Internet
No metadata standard requirements at local site	Will never be capable of interfacing to DIM in Release B.	Lower development costs if trying to connect existing database.
Does not require ECS infrastucture	No capability of user to track requests using ECS client software.	Lower cost of commercial software relating to ECS
Low level of effort for ECS interface – simply supply advertisement description and URL.		Takes only minutes to set up the advertisement.

The data provider must weigh this low cost solution against the disadvantages for the user who has to learn different interfaces to access different Internet Service providers. Some users may decide to only use the tools provided by ECS that can search, browse, subset, order, etc. data directly from one interface (namely the Earth Science Search Tool (ESST) or the Web version of

this client) rather than look through individual advertisements. An option could be to start with this approach and migrate to another if access to the site through ECS is found to be too limiting. This would be a good choice if the site already has an Internet Service for searching and accessing the data at their site.

4.3 Data Provider Supplied Client

When the Advertising Service installs or invokes a service that is not a MIME-type, a program called the Advertising Service Installer is invoked. The Installer program communicates with the Advertising Service to determine the client software required to execute the service. If the client software is not resident on the client workstation, the Installer will execute a script resident in the Advertising Service database or file system that will perform the following two functions:

1. Initiate a download of the client software to the client workstation (usually via FTP).
2. Configure the software on the client workstation including the specification of any necessary command line arguments.

The installation script and any configuration files are delivered to the Advertising Service by the data provider when submitting the advertisement. Some work on the part of the data provider will be required prior to submitting the advertisement. For example, the data provider will have to figure out how to support to client software installation multiple platform through the script language. The “help” information on how to write these scripts will be available through the Advertising Service itself. Once the installation and configuration script has been executed, the ECS infrastructure is no longer involved in the communications between the client and data provider, unless the data provider happens to be advertising a client that accesses ECS components.

The steps for the data provider in this scenario are:

1. Access the Advertising Service Web page and read about the installation script and configuration file syntax and supported variables.
2. Write an installation and configuration script file, using the information obtained from step #1.
3. Go to Advertising Service Submissions page and select General text link to get to service advertisement form. Input service advertisement information including: service title, service type (signature), service description, provider ID, product ID (optional), and URL to documentation (optional). Input the script information.
4. Select the submit button to submit this advertisement to the Advertising Service.

The service advertisement follows the same steps as shown in Figure 4.2-1 to become approved by a moderator. The installation of a signature type service is slightly different from the MIME type service. Figure 4.3-1 shows what happens when a user finds and installs a signature type service. A user will use their favorite Web browser to search the advertising service. The Advertising Web server will satisfy the search using the Sybase database for advertisements. When the user selects a service to install, the Web browser on the user’s workstation will spawn

the Advertising Service Installer (must already be installed on user's workstation). The Installer will connect to the Advertising Service Application Server and ask for the service information, including the installation script and configuration file. The Advertising Service Application Server will retrieve this information from Sybase and return it to the Installer. The Installer will then run the installation script which will typically include an FTP to get the client software from the external data provider site. Once the software is installed on the user's workstation, the Installer will create the desktop object using the configuration file information (if the user has an ECS desktop). When the user double clicks on the desktop object to initiate the service, the Desktop will spawn the client software just installed. The client software will connect directly to the service at the external data provider site.

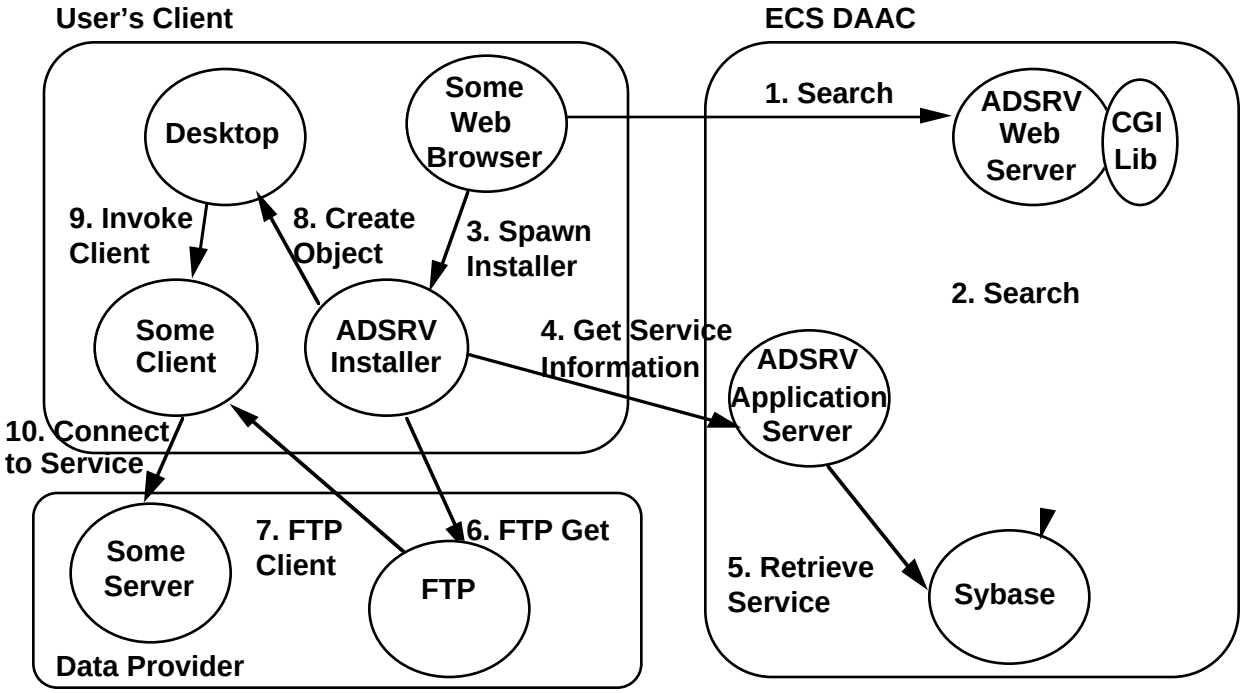

Figure 4.3-1. Installing Client Supplied by External Data Provider

For clients that access ECS services, the same steps would occur, except for the last step. If the provider is supplying a client that accesses ECS services, the provider would have to create an installation script similar to the one for the above scenario. Upon initiation of the client from the Desktop, the client software would then use the ECS infrastructure to connect to the ECS services.

4.3.1 Data Provider Supplied Client Features

Table 4.3-1 summarizes the features of the Data Provider Supplied Client option.

Table 4.3-1. Data Provider Supplied Client Features

Feature	Disadvantage	Advantage
Discovery through Advertising Service	No distributed search capability at lower than collection level	Wide visibility: Surfing is common practice on Internet
No metadata standard requirements at local site	Will never be capable of interfacing to DIM in ECS	Lower development costs if trying to connect existing database.
Does not require ECS infrastucture	No capability of user to track requests using ECS client software.	Lower cost of commercial software relating to ECS
Low level of effort for ECS interface – simply supply advertisement description and client installation instructions.		Takes only minutes to set up the advertisement.
More sophisticated user interfaces can be supplied for services.	May require users to have certain prerequisite COTS which may limit the audience of the client.	Could be used to advertise tools that ECS does not provide such as data analysis and visualization tools.

This option provides a wide variety of opportunities for the data or service provider. The ECS APIs are public and documented in the API IDD. This option can be used to provide an alternative interface to ECS data. For example, the ESST is a client provided by ECS, but others may provide their own, more specialized user interface. It could also be used to advertise a user interface that provides more complete data analysis tools than any of the ECS clients. For example, LinkWinds is a tool developed at JPL which provides many more analysis EOSView, provided by ECS. LinkWinds could be advertised to ECS users; interested users could use the Advertising Service to download the software to their local client. A third option is the client that provides direct access to the data of the data provider, thus freeing the data provider from having to conform or map to the ECS metadata standards.

4.4 ECS Client with LDAS Option

The ECS Client with LDAS option is similar to the Data Provider Supplied Client in that an alternative client is provided to connect to some service at the external provider's site. The difference in this case is that the client is ECS' Earth Science Search Tool and the service includes some ECS provided software for searching metadata (in the LDAS). The external provider reuses server components to reduce server development and does not have to develop a client, because ECS has already developed one.

The chief advantage to this approach is the reuse of ECS software to provide for searching of metadata. Users would discover the external data provider through the Advertising Service and create a direct connection to the LDAS at the external data provider's site. The disadvantage to using this approach is that the LDAS schema is not guaranteed to be integrated into the ECS metadata standard; therefore, the external data provider site cannot be accessed from a DIM and cannot be part of distributed searching or one-stop shopping from the ECS client.

The activities that must be performed by the external data provider in this instance are as follows:

1. Install and configure the COTS software required for this option. LDAS requires OSF/DCE and OODCE. LDAS comes configured to run with the Illustra object-relational DBMS. Other DBMSs can be supported with modifications by the external data provider.

2. Create the database tables and set up the system to ingest the metadata into the DBMS.

3. Establish the ESDTs for the contents of the LDAS. Refer to Release-B SDPS/CSMS Operations Scenarios (605-CD-002-001) for more discussion on how ESDTs are established.

4. Export the ESDT services to the ECS Advertising Service. A Science Data Server tool will be available that performs this function. The ECS Advertising Service will be located at an ECS DAAC, and will not need to be hosted at the external provider site. The contents of the advertisements will be signature type advertisements that describe the services available at the external provider site. The client software associated with the services will be the ECS client, Earth Science Search Tool (ESST). This will tell the advertising service that the ESST is the component that can access the LDAS at the external provider's site. The external provider may use a standard ESST installation script instead of writing one from scratch.

5. Export the ESDT definition to the ECS Data Dictionary Service. A Science Data Server tool will be available that performs this function. The ECS Data Dictionary Service will be located at an ECS DAAC and will not need to be hosted at the external provider site.

Figure 4.4-1 shows a simplistic view of the Advertising of an LDAS option. The LDAS software is installed and the advertisements and data dictionary information are sent to an ECS DAAC. When a user finds the advertisement in the Advertising Service, the ESST configuration information is retrieved from the Advertising Service. When the ESST is initiated with the service information for the LDAS, a direct connection is established between the LDAS and the ESST.

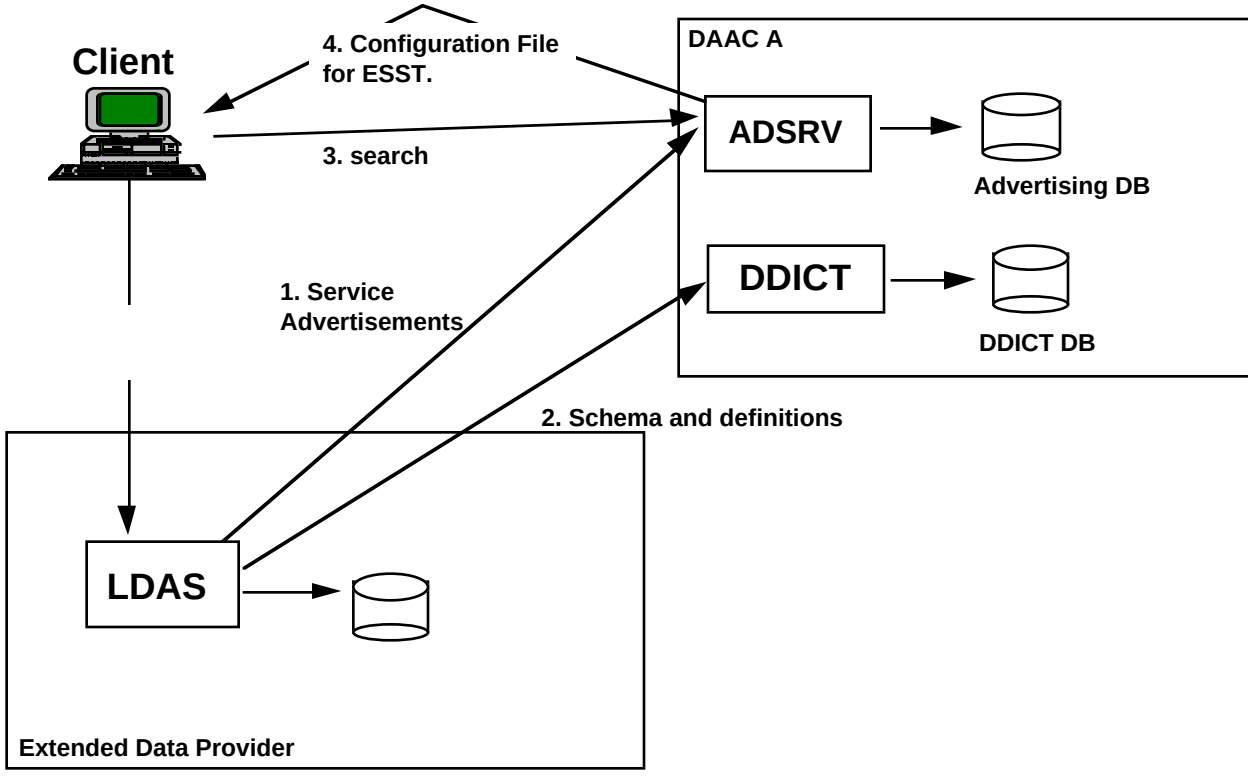

Figure 4.4-1. Advertising an LDAS Option

4.4.1 LDAS with ECS Supplied Client Features

The following table summarizes the costs and benefits of the Internet service advertisement scenario.

Table 4.4-1. ECS Client with LDAS Costs and Benefits

Feature	Disadvantage	Advantage
Discovery through Advertising Service	No distributed search capability at lower than collection level	Wide visibility: Surfing is common practice on Internet
No metadata standard requirements at local site	Will never be capable of interfacing to DIM in Release B	Existing schema does not have to be modified to conform to ECS standards.
Requires some ECS infrastructure components, such as DCE.	Higher cost of commercial software to support ECS functions.	Provides security mechanisms that may be important for restricting access to certain data types.
Interfaces with ECS client.	Costs more in development for ECS interaction.	Client software is provided by ECS to interpret advertisements and send search, browse, and product requests.

5. EOSDIS One-Stop Shopping Connection Scenarios

5.1 LIM/Gateway Off-the-Shelf Software Requirements

For each of the one-stop shopping scenarios a LIM is required to support the hiding of the underlying physical data and/or transaction architecture of the site. Therefore, there are some basic requirements for each data provider site that wants to support one-stop shopping. A Data Dictionary Service has to be installed as part of configuration of the LIM. Table 5.1-1 describes the hardware and software requirements of both the Data Dictionary Service and the LIM. These also apply to the Gateway, if a custom gateway is being built by the data provider.

Table 5.1-1. Data Management COTS HW/SW Needs

HW/SW Requirement	Version of SW or OS	Comment
Sun / HP workstation or server class machine	Solaris 2.5 & HP/UX 10	Arrangements can be made for other platforms and operating system versions to be made available.
OSF Distributed Computing Environment (DCE)	1.1	Host of the LIM must have a DCE 1.1 client configured in an EOSDIS DCE cell or a trusted counterpart cell.
Relational DBMS	Sybase 11	The LIM will have been tested with Sybase, but can support other similar RDBMSs as supported by RW Dbtools.h++.
X-Windows server	X11R5	This is required for using the Data Dictionary Maintenance Tool. This is a Motif interface that must be displayed on an X11 server.
Builder Xcessory and Epak Widget Set	3.5.1	Required only for recompilation of Data Dictionary Maintenance Tool. Not required if binary can be used.
Illustra	3.2	***This is only required for the LDAS option and can be swapped out for another DBMS.***

For customization of either the LIM or the V0 Gateway, the following software is required for the LIM or Gateway to be compiled without any code changes. The code can be changed to replace some of these products as noted in the table.

Table 5.1-2 COTS SW Requirements for Customizations

SW Requirement	Version of SW or OS	Comment
OODCE - Object-oriented DCE[4]	1.0.3a	Library required only for modification of a LIM or building a custom gateway based on ECS code. There is no run-time component of this.
C++ & C compilers		Public domain gnu compiler or the compiler from the HW vendor.
RogueWave Tools.h++	6.1 or greater	This is a base class library that provides many fundamentals such as linked lists, strings, etc.
RogueWave DbTools.h++	1.x or greater	This is a class library that provides a generic access to the RDBMS. It supports the use of Sybase, Oracle, Informix and Ingres. Each of these DBMSs can be supported without even recompiling, according to the vendor.

5.2 Local Information Manager Installation and Configuration

The LIM and Gateway software will be delivered to the data provider complete with makefiles and instructions for building the software on different hardware platforms. Once the software has been built and installed in the place most appropriate to the data provider, the configuration of the LIM includes a few steps:

1. The assignment of a name for the LIM and Data Dictionary Service. When a server initializes, it registers a name with the Cell Directory Service (CDS) of DCE. The CDS name is ultimately what the ECS software uses to find the LIM to initiate a connection. At the ECS sites, the name consists of the site short name plus the acronym LIM. For example, the Goddard Space Flight Center LIM has a CDS name of GSFC-LIM. For external providers, the Data Dictionary Service is used only within the local site for configuration of the LIM using the Data Dictionary Maintenance Tool. However, the LIM will require a CDS name in an EOSDIS cell or trusted counterpart cell.

2. The Data Dictionary Maintenance Tool is used to configure the Data Dictionary database and set the scope of the LIM services. The LIM will read the local Data Dictionary database to retrieve this information upon startup. The local Data Dictionary Service must be running to perform these tasks. This tool also submits advertisements to the ECS Advertising Service for the services that are available at the LIM.

Once these configuration items are set up, using the LIM configuration tool, the LIM can be started. The following sections discuss the modifications that are supported with each option and the specific customization that must take place with each option.

---

[4] OODCE for HP/UX is an HP product. For other available platforms, contact ECS Contracts Office c/o Hughes ITS/Civil Systems, 1616 McCormick Drive, Upper Marlboro MD, 20774-5372.

5.1.1 LIM Accessing a Custom Gateway

In the option requiring both the LIM and V0 Gateway (potentially customized), the LIM is installed with no software modifications. The software can be built, configured, and executed as supplied. If the Gateway is to be customized, the code will have to be modified to support new protocols or messages. The Gateway consists of C++ classes that are grouped into several components. These components are discussed and documented in the Data Management Subsystem Design Specification (305-CD-023-002). Some classes are missing from the current release of this volume but will be updated before the software becomes available. Figure 5.1-1 shows all the components of the V0 gateway.

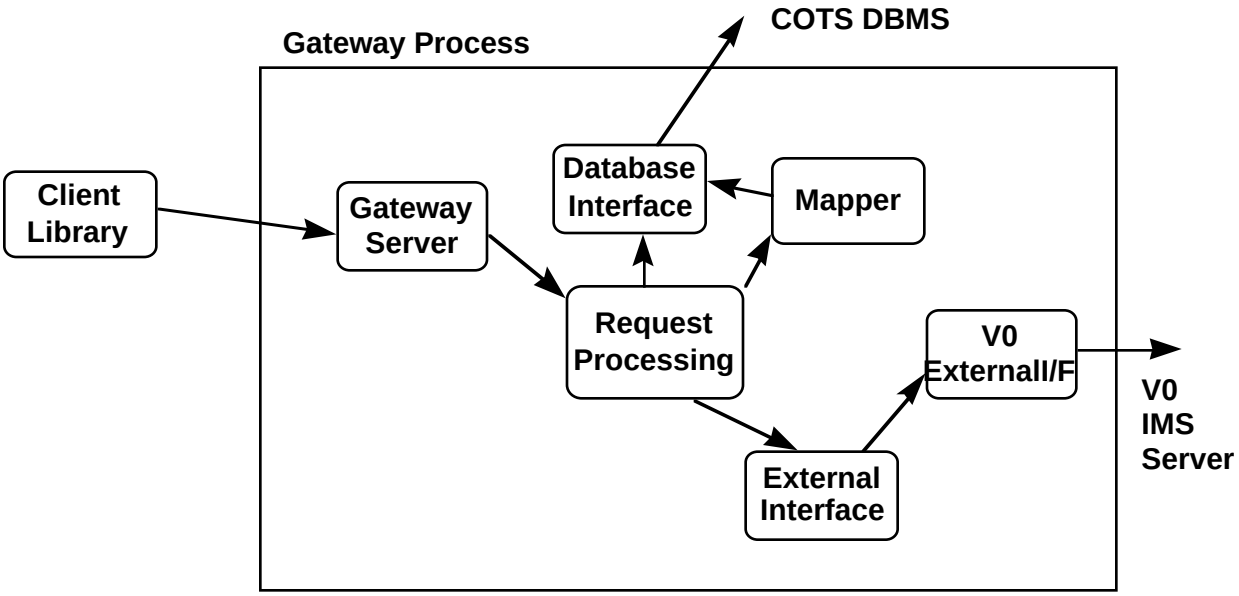

Figure 5.1-1. Gateway Component Interaction Diagram

The components that would typically be modified include the following:

1. Request Processing - The request processing component is a set of classes that parse a request into its component parts and pass it to the appropriate external interface type (i.e., SDSRV, LIM, V0, etc.) One method in one class will have to be modified to use the correct external interface when the external component is called.

2. External Interface - The external interface component is a set of classes used to talk to the other components in the system such as a SDSRV or V0 IMS server. If the Gateway is to interface to a V0-like system, the developer could modify the existing V0 interface to provide new message types and protocols. Otherwise, it might be simpler to create a new subclass that handles the protocol and language of the system to which the gateway will talk. The parent class is an abstract base class such that the new classes will have some mandatory methods that must be provided, such as OpenConnection, SubmitRequest, etc.

3. Mapper - The mapper component performs the mapping of ECS attributes to local attributes. The only style of mapping provided by the current LIM and Gateway is table based. Attribute names are located in a Data Dictionary table, and internal (to ECS) and external names are related to each other. For string types, the values can also be mapped using a Data Dictionary table. The mapper component supports this type of mapping without code modifications. This is available through the Data Dictionary configuration. If attributes in the existing system cannot be mapped through simply name and value pairs, then the mapper component would have to be modified to support more complex mappings. This might be required for cases such as units conversion, date type conversions, or integer to character conversions.

Once the Gateway has been customized and built, the configuration of the LIM and Gateway can begin. As above, there are two steps: (1) DCE name configuration and (2) Data Dictionary maintenance. The DCE name configuration is very simple and should take very little time depending on whether Access Control Lists (ACLs) are needed. ACLs are a mechanism for defining data and service access "rules" for users or groups of users. Please refer to the Communications Subsystem Design Specification and the Management Subsystem Design Specification for more information on DCE configuration requirements and discussions of ACLs.

The Data Dictionary maintenance involves running a Motif user interface to define the attributes accessible from the site. The Data Dictionary Service must be started to be able to configure the local data dictionary database. The Data Dictionary Maintenance tool will communicate to the DDICT to capture the configuration of the local site. There are several inputs that will be made to the maintenance tool:

1. *Collection Descriptions.* The textual description and keyword associations will be stored in the local data dictionary as well as being exported to an Advertising Service and Data Dictionary Service at an ECS DAAC. This will allow the rest of the ECS community to be aware of the collections available at the data provider sites.

2. *Attribute Definitions*. The definition of the attributes accessible through the Gateway and LIM and their relationship to each collection will be entered into the tool. For standard ECS attributes, the tool will provide a definition and the operator can simply relate the ECS attribute to the collection rather than having to define it again. Site specific attributes will have to be defined in terms of their attribute name (as seen to the ECS client), the description, the data type (numeric, date, spatial, string), and domain. The domain specification can be specified as explicit values for string types or as ranges for numeric and date types.

3. *Attribute Mappings*. The site specific attributes will be mapped to their ECS standard counterparts if this can be table driven. For example, in the V0 gateway, the ECS term InstrumentShortName would be mapped to SENSOR_NAME in the V0 system. If the mappings can not be described as simple definitions of relationships of one attribute to another, then the Mapper component will have to be modified with code to perform this application defined mapping.

4. *Service Descriptions*. The services available through the LIM and Gateway will be described and related to the data collections available. The service descriptions will be exported to the ECS Advertising Service. Any service can be described as long as the External Interface component has been modified to support non-ECS services. For example, the V0 interface only supports the V0 services, search, browse, and product request. The SDSRV interface is very flexible and can pass on many types of interfaces. The external interface provided in the custom gateway, must map the services supported to the methods used to invoke them.

5. *Information Manager - Collection Connections*. The Information Manager descriptions (the description of the LIM and Gateway), must be connected to the collections that they can access.

5.1.2 LIM Accessing a LDAS

In the option requiring both the LIM and LDAS (potentially customized), the LIM is installed with no software modifications. The software can be built directly, configured, and executed. The LDAS might be modified to support a different DBMS, rather than the Illustra software it comes configured with.

The LDAS is the Science Data Server (SDSRV) of the Data Server Subsystem, configured at a local level. The SDSRV source code will be provided with the appropriate makefiles. There are some basic steps to setting up a SDSRV that must be performed before the LDAS can become active. These steps will be described in detail in a future paper on ESDT definition, due out in the fall of 1996.

1. *Creation of ESDT (Earth Science Data Type) code.* There is a class that must be written which handles the creation and deletion of the ESDT in the SDSRV.

2. *Creation of the Illustra database.* The Illustra database must be created with the ECS schema.

3. *ESDT Descriptor generation.* An ESDT descriptor file must be generated that describes the ESDT and the services that can be invoked on this ESDT.

4. *Exportation of the Data Dictionary Information and Advertisements.* The information described and contained at the collection level of the LDAS must be extracted and sent to an ECS Data Dictionary Service and Advertising Service. An operator tool or command will perform this function.

Once the LDAS has been customized and built, the configuration of the LIM can begin. The configuration of the LIM takes on the same steps as described in section 5.1.1.

5.1.3 Modification of a LIM

In the option in which the LIM is modified rather than the Gateway or LDAS, the software for the LIM is provided with source code for everything but the ECS server interface. This allows for the external provider to customize the internals of the LIM while keeping the interface from ECS to the external provider constant. The LIM consists of C++ classes that are grouped into

several components. These components are discussed and documented in the Data Management Subsystem Design Specification (305-CD-023-002). Some classes are missing from the current release of this volume but will be updated before the software becomes available. Figure 5.1-2 shows the components of the LIM.

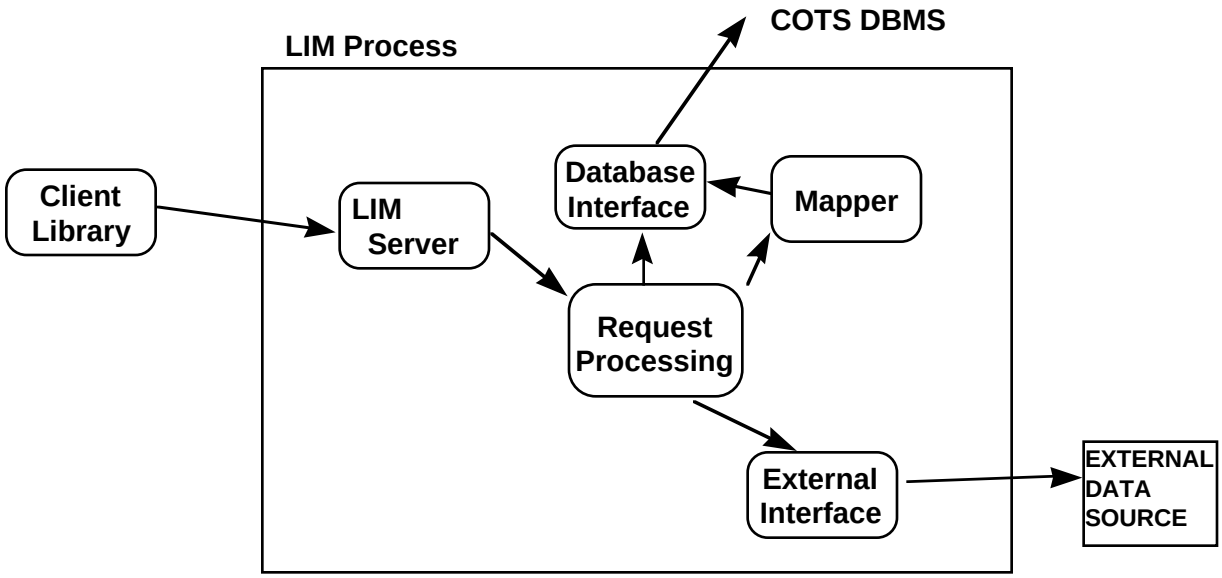

Figure 5.1-2. LIM Component Interaction Diagram

The modifications of the LIM would typically correspond to those described in Section 5.1.1 in the customized Gateway scenario. The advantage to the LIM modification approach is that there is one process that needs to be operated rather than two. The disadvantage is that the Gateway cannot serve as an example of the customization of the external interface to support socket level protocols. The decision to customize the LIM or the Gateway might be based on the similarity of the site application an network protocols to the V0 protocols.

5.2 EOSDIS One-Stop Shopping Test Case

This section provides a sample implementation of the EOSDIS One-Stop Shopping Connection Option. The example utilizes the Catalog Interoperability Protocol (CIP) as developed by the Committee on Earth Observation Satellites (CEOS). There are three sub-sections: 1) an overview of the CEOS-CIP and anticipated architecture for international catalog interoperability, 2) ECS searching of CEOS collections using the CIP, and 3) CEOS searching of ECS collections using the CIP.

5.2.1 CEOS CIP Overview

The CIP is the result of an initiative to develop a generic Earth observation data retrieval protocol that international Earth observation data providers can utilize to make their data

available in a coherent manner to the users of that data. The CIP is being developed in an incremental fashion by member agencies working under the auspices of the CEOS Protocol Task Team (PTT). Release A of the CIP has been ratified and work on Release B has begun (*Reference: CIP Specification - Release A, Version 1.2, available at ftp://styx.esrin.esa.it in /pub/od/CIP/cip_release_a*). One agency, CEO, is requiring implementation to the CIP-A as part of a funded development. Agencies participating in the development of the CIP include CEO, NASA, ESA, CCRS, CNES, NASDA, DLR, and others.

Based on analysis of requirements for the protocol, the CIP was specified as a profile based on Version 3.0 of the Z39.50 protocol (reference: *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*, ANSI/NISO Z39.50-1995 , Official Text, July 1995, Z39.50 Maintenance Agency available *at http://lcweb.loc.gov/z3950/agency/*). Z39.50 is an existing protocol for catalogue search and retrieval. Basing the CIP on Z39.50 provides a stable design baseline and supports reuse of existing Z39.50 software components.

Figure 5.2-1 shows a possible architecture for a distributed, interoperable catalogue system that operates using the CIP. It is only illustrative; it only shows three agencies or data providers as part of the complete catalogue system. This could be expanded to more agencies and facilities providing that they are designed (or are adapted) to support the CIP. Note also that it shows a long-term view of the CIP as the Order service is not included in the CIP Release A Specification.

Figure 5.2-1 represents only grossly the functional architecture of EOSDIS; the key feature for the implementation of CIP is the element labeled a Retrieval Manager. The CIP Retrieval Manager is envisioned as a module common to all CIP implementations. A Retrieval Manager interoperates with other Retrieval Managers to distribute and accept queries and product orders.

The EOSDIS External data provider Gateway is the ECS component analogous to the CIP Retrieval Manager. To implement interoperability between International Partners (IPs) and ECS, NASA would install a LIM with an underlying gateway. The gateway would contain Retrieval Manager functionality and interoperate with IPs using CIP (See Figure 5.2-2).

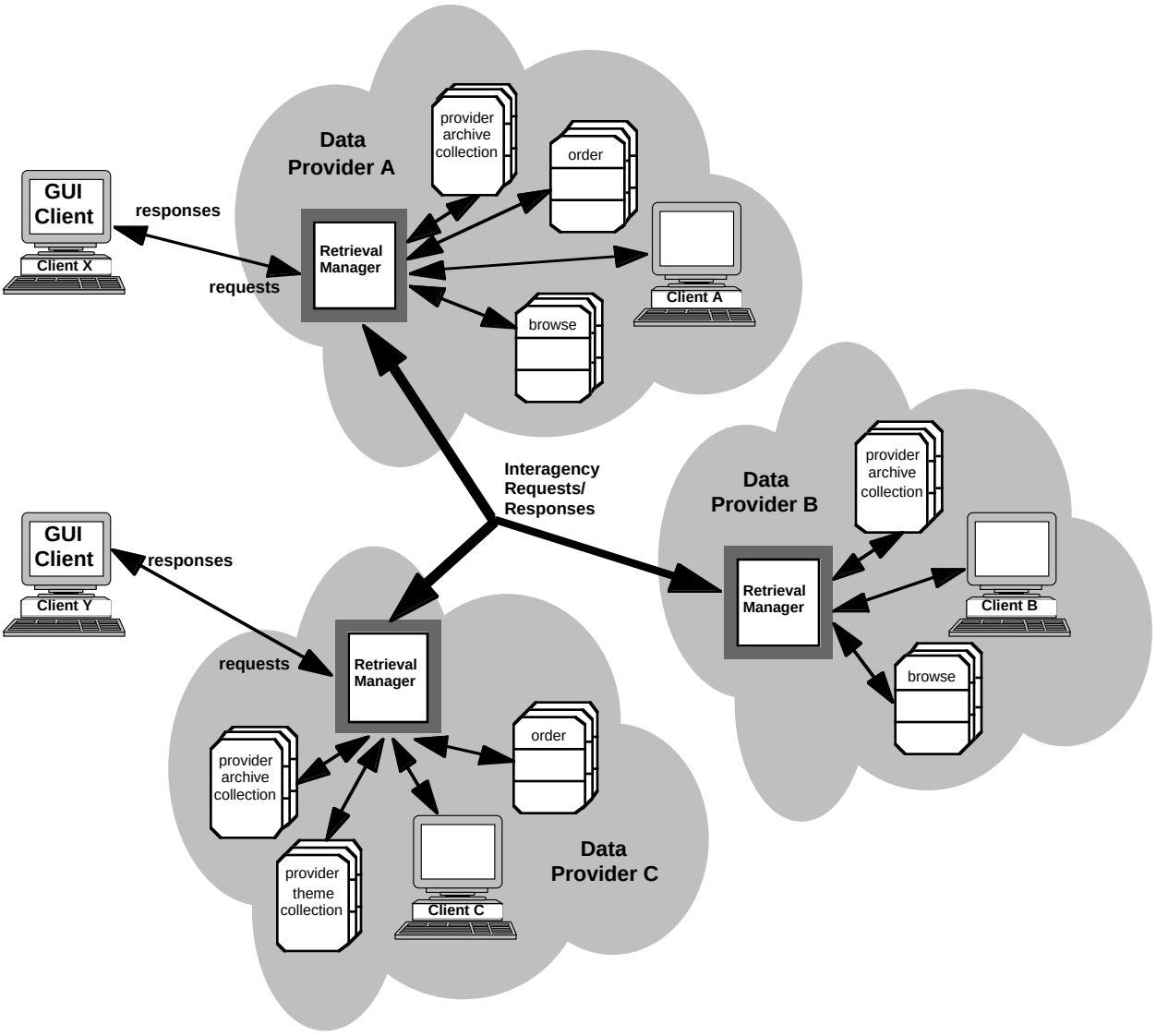

Figure 5.2-1. Potential Physical CIP Architecture

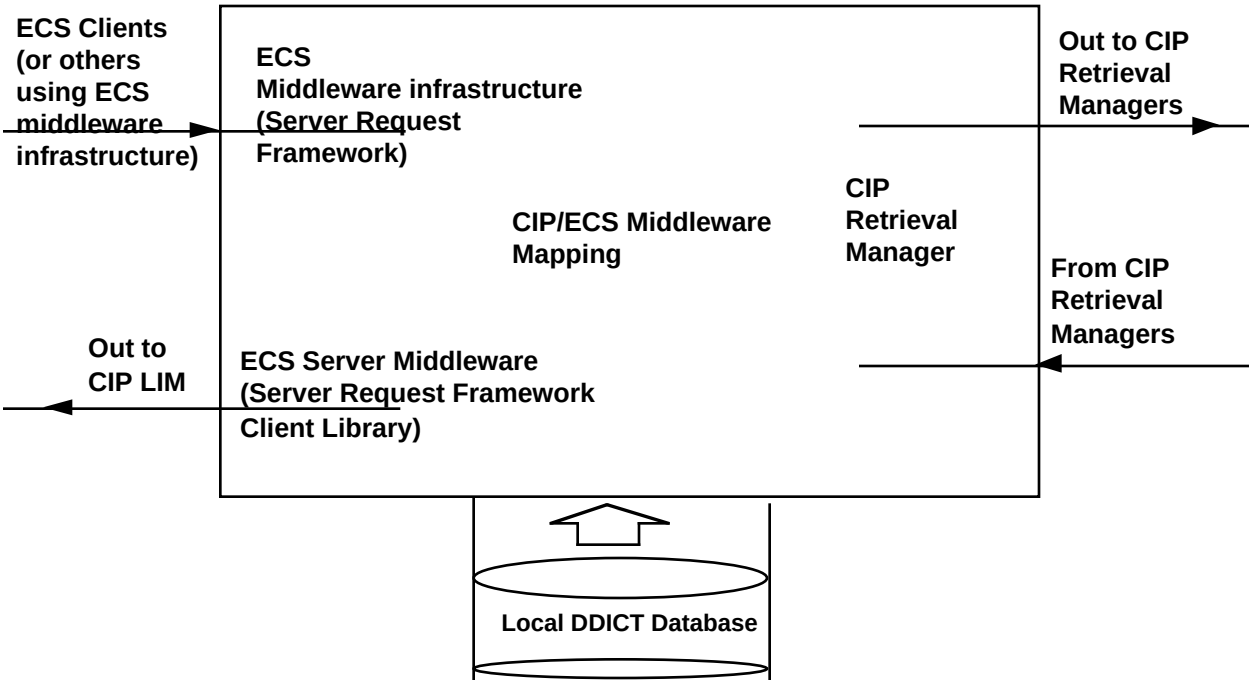

Figure 5.2-2. CIP Gateway

To understand what will be required of the CIP Gateway it is critical to understand the differences and similarities between the CIP and the application and middleware protocols for ECS Version 1. Table 5.2-1 provides such a comparison.

Table 5.2-1. Comparison of ECS Version 1 and CIP

ECS Version 1 Protocol	CIP
Protocol Services: security, search and retrieve, ordering	Protocol Services: security, search and retrieve, ordering
Session oriented, stateful connection	Session oriented, stateful connection
ECS Earth Science Data Model with single and multiple type collections	Data model similar to ECS with hierarchical collections, mixed collections allowed
Common ECS mandatory metadata across collections. Product specific attributes in replicated DDICT	Common CIP attributes across collections. Local attributes held locally in Retrieval Manager
Valid values in replicated DDICT	Search and present, no dependent valids locally
Query routing through interaction with DDICT. SQL query language	Query routing via hierarchical collection structure. RPN query language for CIP Release A. (Additional query languages in later releases of CIP)
OODCE/DCE and Distributed Object Framework with custom object interaction model	CIP profile for Earth observation domain, on top of ANSI standard Z39.50 protocol for search and retrieval
DCE security for authentication and authorization, requires individual principal authentication, with group level authorizations.	Two-way authentication between Retrieval Managers with group authorization

5.2.2 ECS search of CEOS Holdings

This section considers ECS searching of IP collections using the CIP.

All preconditions for the Custom Gateway Scenario (see section 5.1.1) apply to this scenario as well:

1. CIP LIM has exported the data dictionary information conforming to ECS data model to DDICT server using data dictionary tool.
2. DIM, ECS LIM, CIP LIM have been configured in the DDICT database.
3. DIM, ECS LIM, ECS Data Server, CIP LIM, CIP gateway, and DDICT are up and running. (SYBASE SQL server, OODCE server are running)

Populating the ECS data dictionary with CIP collection information will differ from the operational model within ECS. Currently in the CIP architecture there is no single location which contains all of the CIP collection information. Global access to CIP collection information is the subject of a CEOS PTT Technical Note on Collections (Interoperable Catalogues System (ICS) Collection Technical Note (CTN) – 170-WP-006-003).

The scenario is depicted in Figure 5.2-3. The steps in the scenario are as follows:

0. Human user initiates ECS client
1. As user is developing query, ECS Client interacts with DDICT.
2. Based on the contents of the query and information from the DDICT, the query is sent to a DIM at DAAC A.
3. DAAC A DIM builds a query plan and subqueries, which, based on information from DDICT, are sent to separate LIMs or SDSRVs.
4. One sub-query is sent from DAAC A DIM to the Data Server at DAAC B.
5. Another sub-query is sent from DAAC A DIM to the CIP LIM/Gateway
6. CIP LIM interacts with DDICT to determine query plan
7. CIP LIM sends query to Gateway
8. CIP Gateway interacts with DDICT to determine which IPs can satisfy query
9. CIP Gateway acts as a CIP Retrieval Manager and sends queries to other Agency Retrieval Managers using CIP. Remote Retrieval Managers pose queries to local collections and return results to NASA CIP Gateway, which in turn are returned to the DAAC A DIM via CIP LIM.

Once the DIM has received the result sets from all sub-queries, the user receives an indication that the query has been completed.

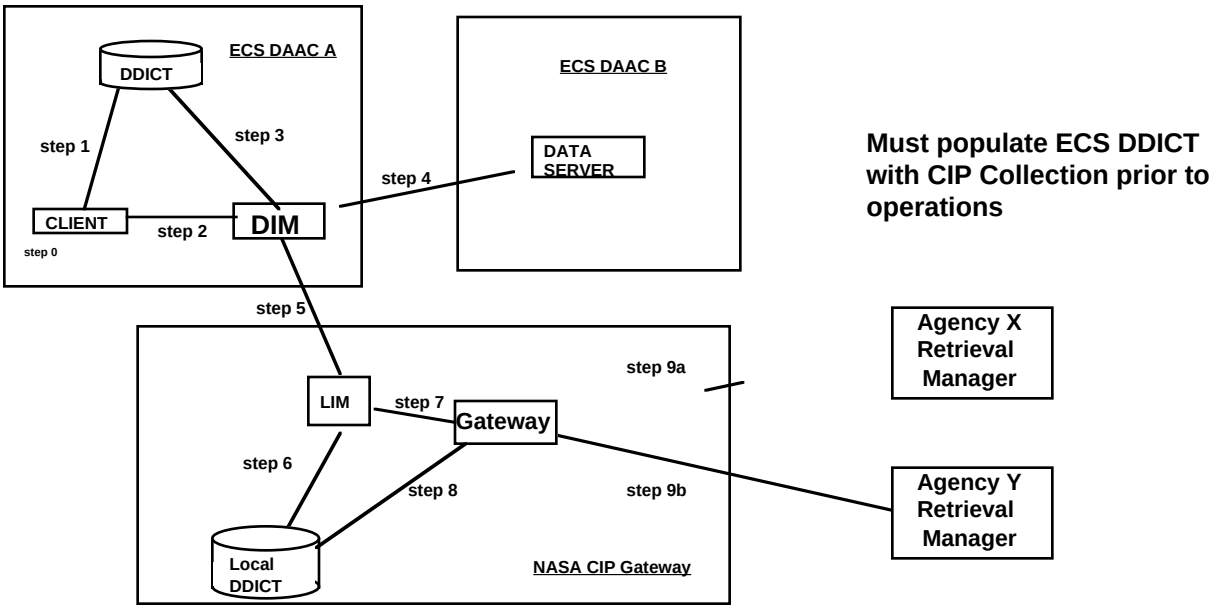

Figure 5.2-3. ECS searching of IP collections

5.3 External Data Provider Access to ECS Services

There are two methods to providing access from external provider clients to ECS data and services.

1. *Use ECS public APIs to create direct access to ECS services.* The public APIs are documented in the API IDD. These APIs can be used to search advertisements, data servers, DIMs, LIMs, and the Data Dictionary service. Rather than using the APIs in the same architecture as ECS does, the external provider can use them to provide their own unique access to ECS. The API IDD should be read to determine the capabilities that can be used.
2. *Create a two-way interoperability gateway.* ECS has two different gateways in the current Release B baseline, the V0 Gateway and the ASTER Gateway. Either one of these can be used by an external provider to create a bi-directional gateway to their system.

5.3.1 Two-way Interoperability Gateway

As discussed, there are two Gateways currently under development for Release B. They can be described as follows:

1. *V0 Gateway.* The V0 Gateway can translate requests from the V0 Client to one SDSRV. The V0 Client addresses queries and service requests directly to each DAAC. The Gateway then translates from the V0 ODL into the ECS SDSRV protocol. In Release B, the V0 Gateway will be extended to accept requests from an ECS Release B Client and translate it into requests to multiple V0 IMS Servers.

2. *ASTER Gateway*. The ASTER Gateway is slightly different from the V0 Gateway. The ASTER Client modifies the V0 ODL to support other requests such as Price Estimates and Product Request Status messages. The ASTER Client (being built by the ASTER Ground Data System in Japan), will communicate to just one ASTER Gateway. The ASTER Gateway will then act like an ECS DIM and figure out which SDSRVs, LIMs, or other Gateways to route the requests to. This Gateway will serve as a model for other two-way interoperability gateways.

Currently, the ASTER Gateway will be hosted at the EROS Data Center (EDC). As such it can share the DDICT database already residing at EDC. For external providers, the steps for setting up the two-way interoperability gateway are similar to those specified in 5.2.1 for the Custom Gateway. The difference in the scenario, is that with the two-way interoperability gateway, the external provider must have the full DDICT database as held at an ECS DAAC (provided the external provider wants access to all ECS data).

Figure 5.3-1 shows how consistency of the ECS DDICT and the external provider can be maintained. One configuration option for the Gateway is to set up subscriptions to obtain notification of inserts, updates, and deletions of collection information to the DDICT. This would have to be set up to each DAAC that the external provider site was interested in. The notification can be an e-mail notification to an operator or a process-to-process notification to the Gateway itself. Figure 5.3-1 shows the process to process notification. In this case, when something new was added to the DDICT at DAAC A, the external provider gateway would receive a notification with the UR of the newly inserted collection. The Gateway would then retrieve all relevant information about that collection and send an insert command to its local DDICT service.

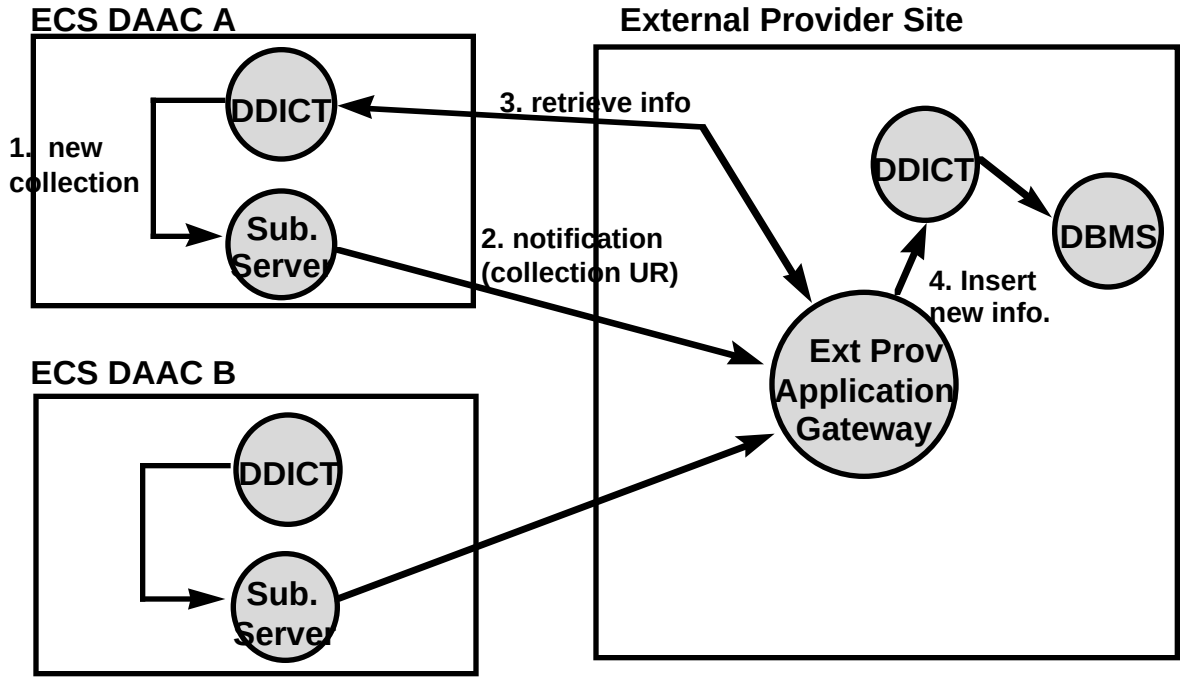

Figure 5.3-1. Two-way Interoperability Data Dictionary Maintenance

The e-mail notification option can be used to manually update the local DDICT using the Data Dictionary Maintenance Tool described in the previous options. This would be used to have a person filter the ECS collections that are of interest to the external provider's users.

To implement the two-way interoperability gateway, the steps are basically the same as the Custom Gateway one-way interoperability except that the external provider must be able to translate the messages coming in from its clients and use the Gateway's Request Processing objects to construct a request to ECS. The COTS for this option are the same as the other LIM options. The DDICT service must be running for the automated data dictionary maintenance, but this requires no new COTS products. This option would primarily be used to provide one location that translates ECS formats and message to and from the local formats and messages. One database (the DDICT) is used in both directions.

5.4 System Management Issues

The ECS system and its components have a complex infrastructure to support the ECS system-wide services including: security, request tracking, monitoring and management of services, and accounting and billing. These services provide the ECS DAAC operators with a common framework to work with ECS application servers such as the SDSRV, LIM, and DIM. Due to the need of these services at an ECS level, these capabilities are difficult to extract from the baseline Release B components. Therefore, these services will be available and operating in the software supplied to external data providers. External data providers will have the option to use these services or simply ignore them. Regardless, there is some added COTS products that must be available to support these management services. The following sections describe the COTS requirements and capabilities of the ECS infrastructure subsystems: Communications Subsystem (CSS) and Management Subsystem (MSS).

5.4.1 DCE Cell Management

Options available to data providers when becoming part of the EOSDIS provider network include:

1. *Become a client to an existing ECS cell (such as at a DAAC)*. The main functionality that is needed by a LIM and LDAS is the ability to register the application names in the Cell Directory Service (CDS). In order to do this in another site's cell, at least one user name within the external data provider's site would have to have permissions to write to the CDS at the DCE server site. This would also mean that all the external data provider's users would have to be created as DCE users in the DCE server site.

2. *Install a cell that cooperates with an existing ECS cell*. Currently, the ECS DAACs have cooperating cells that allow for foreign user access to the CDS. This same capability could be extended to external data providers, if the external provider is defined as "trusted" under the terms of the EOSDIS Security Policy. The external data provider would have DCE server capabilities, including its own local users and CDS. The LIM and LDAS would register their application names with the local CDS. Other ECS sites could read the external provider's CDS in order to find these LIM and LDAS servers.

Given these two options the COTS requirements for DCE at the external data provider sites are as follows:

Table 5.4-1. DCE COTS Requirements

SW Requirement	Version of SW or OS	Comment
DCE Client Services	OSF v. 1.1	Each hardware vendor may have a different version number. The importance is in the OSF version that the hardware vendor's version complies to. ECS Release B baselines is OSF version 1.1.
DCE Server	OSF v. 1.1	See note above. This is only pertinent to option number 2 above.

5.4.2 Event Tracking / System Management

The LIM, GTWAY, and LDAS all log important events an errors to MSS. Some events are considered application events that require no intervention by MSS. These are written to a file on disk and require no special COTS software. Other events such as faults or errors are sent to the system management framework. The management framework provides insight to operators so they can assess the health of the system. These types of events require COTS products to trap and monitor them. Table 5.4-2 shows the COTS products that are required by MSS and thus would be required to run the LIM, GTWAY, and SDSRV.

Table 5.4-2 MSS COTS Requirements

HW/SW Requirement	Version of SW or OS	Comment
Tivoli Management Platform: Tivoly Admin, Sentry, Courier, Enterprise Console	2.5	Some of these may not be required if the external provider is not going to monitor the events being generated by the LIM, GTWAY, or LDAS or is part of an ECS DAAC cell rather than having its own cell.
Tivoli Tool-kits: Build Tivoli/Plus Modules Build Tivoli Event Adaptors	2.5	
HP Openview	4.1	This is only required if the external provider wants to be able to monitor the status of the applications and do point-and-click shutdown and startup of the applications.
PATROL SNMP tool kit (by Peer Networks)	2.2	

6. Summary

There are various ways of providing services to or obtaining services from ECS. This paper describes some of those ways. Each external provider's situation is unique and may require some modification of the scenarios described here. Each external provider may need different pieces of the ECS system to meet their specific needs. This paper does not intend to be all inclusive but focuses on some of the more common configurations of external providers.

Different scenarios have been presented for interoperability between external providers and ECS. These scenarios fall in different ranges of cost (in terms of COTS products required and custom code and effort) and benefits (in terms of visibility of the external provider to ECS users). Table 6-1 summarizes the options in these terms.

Table 6-1. External Provider Summary

Option	Cost	COTS	Benefit
Internet Service	Low	Web browser	Lower Visibility
Data Provider Supplied Client	Low	Web browser	Lower Visibility to Locate the Service
ECS Supplied Client with LDAS	High	Web browser, Illustra, OSF/DCE, OODCE, RW Tools.h++, Tivoli, HP OpenView	Lower Visibility to Locate Service
LIM with Custom Gateway	High	OSF/DCE, OODCE, RW Tools.h++, RW Dbtools.h++, Sybase 11, Tivoli, HP OpenView	High Visibility - One-stop shopping
LIM with LDAS	High	OSF/DCE, OODCE, RW Tools.h++, RW Dbtools.h++, Sybase 11, Tivoli, HP OpenView	High Visibility - One-stop shopping
LIM modification	High	OSF/DCE, OODCE, RW Tools.h++, RW Dbtools.h++, Sybase 11, Tivoli, HP OpenView	High Visibility - One-stop shopping

This page intentionally left blank.

7. Future Directions

In order to validate the reusability of ECS components, several prototype efforts have been planned to reuse ECS components to implement interoperability with external data providers. These prototypes are designed to cover a range of interoperability scenarios including:

- reuse of ECS components to enable an external data producer to offer full processing and archive, services;
- reuse of the ECS LIM, Data Dictionary, Data Server, and Gateway components to implement data access interoperability, including support for one stop shopping; and
- reuse of the ECS Advertising service to support loosely coupled interoperability with an external data provider that has its own client interface and search protocols.

Lessons learned from these prototypes will be used to develop refinements to ECS components that enable easier and more cost effective support for external data provider interoperability. Likely enhancements will include a reduction in the number of high end COTS products required to reuse ECS components and more automated support for the system administration activities required to configure components for reuse.

In addition, ECS is exploring emerging infrastructure technologies that may provide new and cost effective implementations of distributed searching and interoperability. Promising technologies include distributed objects, active agents, and web enterprise technologies. Such technologies may be incorporated into ECS as part of ongoing evolutionary enhancements.

This page intentionally left blank.