4.4 Data Management Subsystem Overview

The Data Management Subsystem (DMS) provides catalog interoperability between the V0 Information Management System (IMS) and the ECS. The Release B.0 Search and Order (B0SOT) Tool communicates with the DMS using the Version 0 protocol. The DMS provides this service by supplying a gateway process and an information manager. The DMS maintains a Data Dictionary of data collection information with metadata, attributes and keywords used by the gateway and information manager in achieving interoperability. The Data Dictionary also contains collection attribute and keyword mappings used to translate requests between the ECS and V0 systems. The collection attribute and keyword mappings buffer the users and programs from the DMS methods used by a particular site to access and obtain the data.

Data Management Subsystem Context

Figure 4.4-1 is the Data Management Subsystem context diagram. The diagram shows the events sent to the Data Management Subsystem and the events the Data Management Subsystem sends to other ECS subsystems. Table 4.4-1 provides descriptions of the interface events shown in the Data Management Subsystem context diagram.

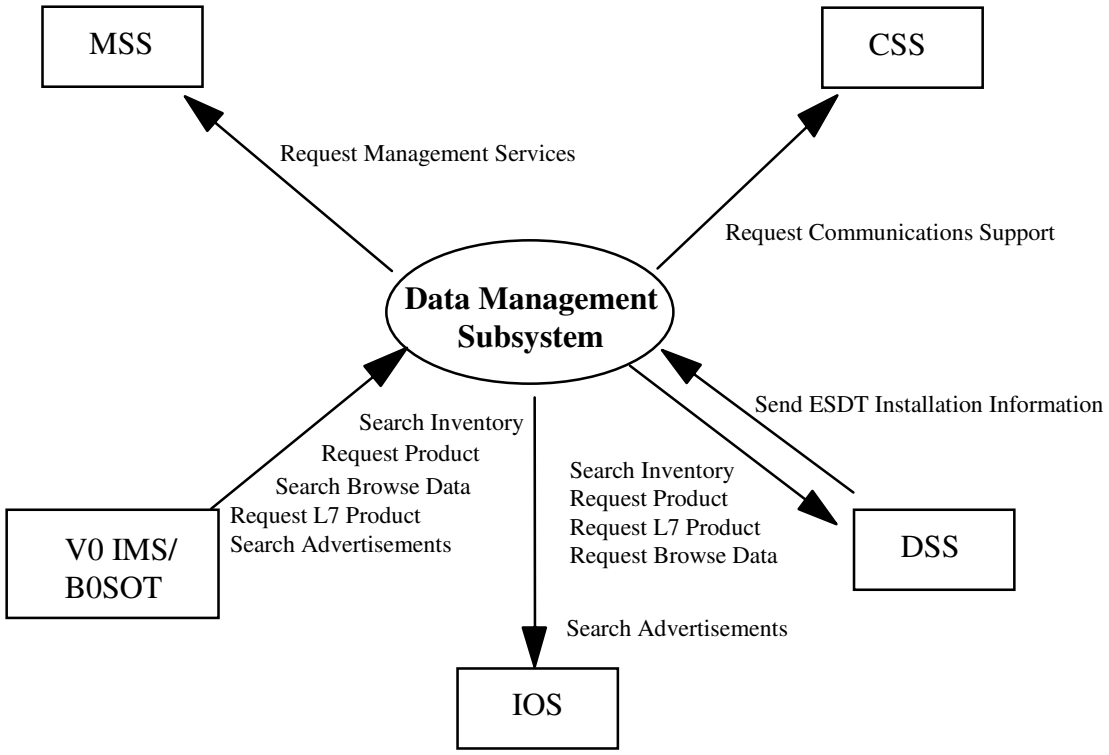

Figure 4.4-1. Data Management Subsystem Context Diagram

Table 4.4-1. Data Management Subsystem Interface Events

Event	Interface Event Description
Request management services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) in which the application should run. The MSS also interfaces with other subsystems to perform the following: • **DMS Order/Request tracking update** - The DMS interfaces with the MSS Order/Request Tracking service to create a user product order. • **User Profile Request** - The MSS provides requesting subsystems with access to User Profile parameters such as e-mail address and shipping address to support their processing activities.
Request Communications Support	The CSS provides a library of services available to each ECS subsystem. The services required to perform the specific subsystem assignments are requested by the subsystem from the CSS. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Send ESDT Installation Information	The DSS inserts new collection level information into the DMS Data Dictionary database as new Earth Science Data Types (ESDTs) are added to the ECS system.
Search Inventory	The DMS receives inventory search requests from the V0 IMS via the B.0 Search and Order (B0SOT) Tool and submits them to the ECS Science Data Server within the DSS. This is a user-initiated event.
Request Product	The B0SOT within the V0 IMS Tool submits product requests via the DMS to the DSS to acquire data granules. This is a user-initiated event.
Request L7 Product	The B0SOT tool within the V0 IMS requests Landsat 7 Products from ECS via the DMS to the DSS. This is a user-initiated event.
Request/Search Browse Data	The DMS receives browse requests from the B0SOT tool within the V0 IMS and submits the browse requests via the DMS to the DSS. This is a user-initiated event.
Search Advertisements	The Interoperability Subsystem (IOS) receives requests to search for subscription event and signature service advertisements from the B0SOT Tool within the V0 IMS via the DMS. A user initiates the search requests. The DMS obtains the proper signatures for acquiring data granules from the DSS (for the insert and update of metadata within the DSS) returns the signatures to B0SOT.

Data Management Subsystem Structure

The DMS is comprised of three CSCIs and two Hardware Configuration Items (HWCIs):

- The Data Dictionary (DDICT) is a software configuration item. DDICT manages the definitions of data collections including the metadata, data domains (valid values), and data location. The Data Dictionary information is stored persistently in a Relational Database Management System (DBMS).
- The Local Information Manager (LIMGR) is a software configuration item. The LIMGR CSCI provides access to the data and services of a site with respect to data made available by the data servers of the site. The LIMGR CSCI accepts requests, such as a search, and produces and executes the corresponding requests required by the site data servers. An operator specifies the accessible objects at the various site data servers.
- The Version 0 Gateway (V0 GTWAY) is a software configuration item. The V0 GTWAY CSCI provides access to data and services between the SDSRV CSCI and the V0 IMS. V0 GTWAY services include inventory searches, browse requests, and product requests.
- The DMS hardware comprises one hardware configuration item Data Management Hardware (DMGHW) CI and one hardware configuration item it shares with the Interoperability Subsystem (IOS), Interoperability Hardware (INTHW) CI. DMGHW and INTHW provide the servers and workstations needed for all data management functions. The DMGHW and INTHW provide processing and storage for the DDICT and V0 GTWAY CSCIs. The DMS hardware also supports the processing requirements of the IOS. The IOS consists of a single hardware configuration item (INTHW) and is described in Section 4.5.2.1.

Use of COTS in the Data Management Subsystem

- RogueWave's Tools.h++

  The Tools.h++ class libraries are used by the DMS to provide basic functions and objects such as strings and collections. The Tools libraries must be installed with the DMS software for any of the DMS processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries are used to interact with the Sybase database Structured Query Language (SQL) server. The use of DBTools buffers the DMS processes from the relational database used. The DBTools libraries must be installed with the DMS for the Data Dictionary Server, Information Managers, and ECS to V0 and V0 to ECS Gateways to run and allow client applications to perform queries of DDICT.

- The ICS Builder Xcessory GUI

  The Builder Xcessory GUI builder tool modifies the displays of the Data Dictionary Maintenance Tool (Mtool). The builder tool also generates the C++ code to produce the

Mtool displays at run time. There is no operational component of Builder Xcessory needed at run-time.

- Sybase Server

  Sybase's SQL server provides access for the Data Dictionary to insert, update, and delete Data Dictionary database information. The Sybase SQL Server must be running during operations for the Data Dictionary Server to execute, search, and update requests on the Data Dictionary database.

4.4.1 Data Dictionary Software Description

4.4.1.1 Data Dictionary Functional Overview

The Data Dictionary (DDICT) CSCI provides access to the Data Dictionary database containing information about science data collections, data attributes, data operations, and the domain(s) of the attributes. The DDICT CSCI describes the data objects accessible through Data Servers, the LIMGR, and the Gateways. The DDICT CSCI provides information support for users to retrieve definitions of the available items and provides infrastructure support to the other CSCIs within the DMS.

The Information contained within the Data Dictionary database includes all collections known within the ECS. Clients (other ECS CSCIs, CSCs, or processes) of the Data Dictionary obtain data collection information by sending queries to the Data Dictionary. Mappings between ECS attributes and keywords, and the V0 IMS attributes and keywords are also maintained within the Data Dictionary. These mappings are used to translate requests between the V0 IMS and ECS.

The location of a data collection within a data server at a particular site is also stored within the data dictionary. This information allows users to perform queries through user software such as the B.0 Search and Order Tool (B0SOT) from any geographical location to forward inventory search, browse, and acquire requests to the appropriate Data Server, LIMGR, or Gateway located at the site where the data is physically stored.

4.4.1.2 Data Dictionary Context

Figure 4.4.1.2-1 is the DDICT CSCI context diagram. The diagram shows the events sent to other CSCIs or CSCs and the events the DDICT CSCI receives from other CSCIs and CSCs. Table 4.4.1.2-1 provides descriptions of the interface events shown in the DDICT CSCI context diagram.

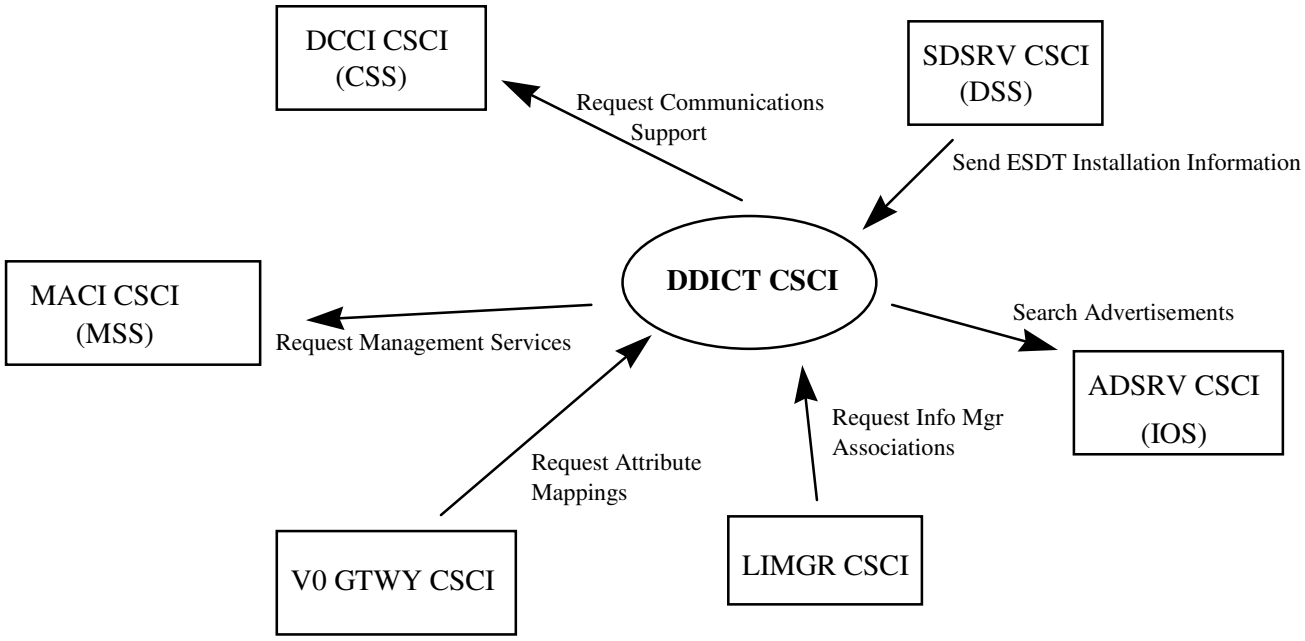

Figure 4.4.1.2-1. Data Dictionary CSCI Context Diagram

Table 4.4.1.2-1. Data Dictionary CSCI Interface Events (1 of 2)

Event	Interface Event Description
Send ESDT Installation Information	The SDSRV CSCI inserts new collection level information into the DMS Data Dictionary database via the EcDmDictServer, as new Earth Science Data Types (ESDTs) are added to the ECS.
Search Advertisements	The ADSRV CSCI receives requests to search for subscription event and signature service advertisements from the DDICT CSCI. A user initiates the search request from the CLS or the B0SOT Tool within the V0 IMS. The DDICT CSCI obtains the proper signatures for acquiring data granules from the SDSRV CSCI (for the insert and update of metadata within the SDSRV inventory) and the signatures are returned to the requester.
Request Info Mgr. Associations	The LIMGR CSCI requests the Information Manager associated with a given data collection to determine where to forward user requests such as browse and product requests. This is a user-initiated event from the CLS or the B0SOT Tool within the V0 IMS.
Request Attribute Mappings	A user, at the B0SOT Tool within the V0 IMS, requests data collection attribute and keyword mappings (via the V0 GTWAY CSCI) from the DDICT CSCI to translate requests from the V0 IMS to the ECS and back again.

Table 4.4.1.2-1. Data Dictionary CSCI Interface Events (2 of 2)

Event	Interface Event Description
Request management services	The MACI CSCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required performing the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.

4.4.1.3 Data Dictionary Architecture

Figure 4.4.1.3-1 is the DDICT CSCI architecture diagram. The diagram shows the events sent to the DDICT CSCI processes and the events the DDICT CSCI processes send to other processes.

The Data Dictionary is comprised of two ECS developed processes, the Data Dictionary Server, EcDmDictServer and the Data Dictionary Maintenance Tool, EcDmMaintenanceTool, along with the COTS process, the Sybase Server. The Data Dictionary Server, Maintenance Tool, and Sybase Server processes reside inside a DAAC and run on the DMGHW. The Data Dictionary uses one data store per DAAC, the EcDmDictServer Database, as shown in Figure 4.4.1.3-1.

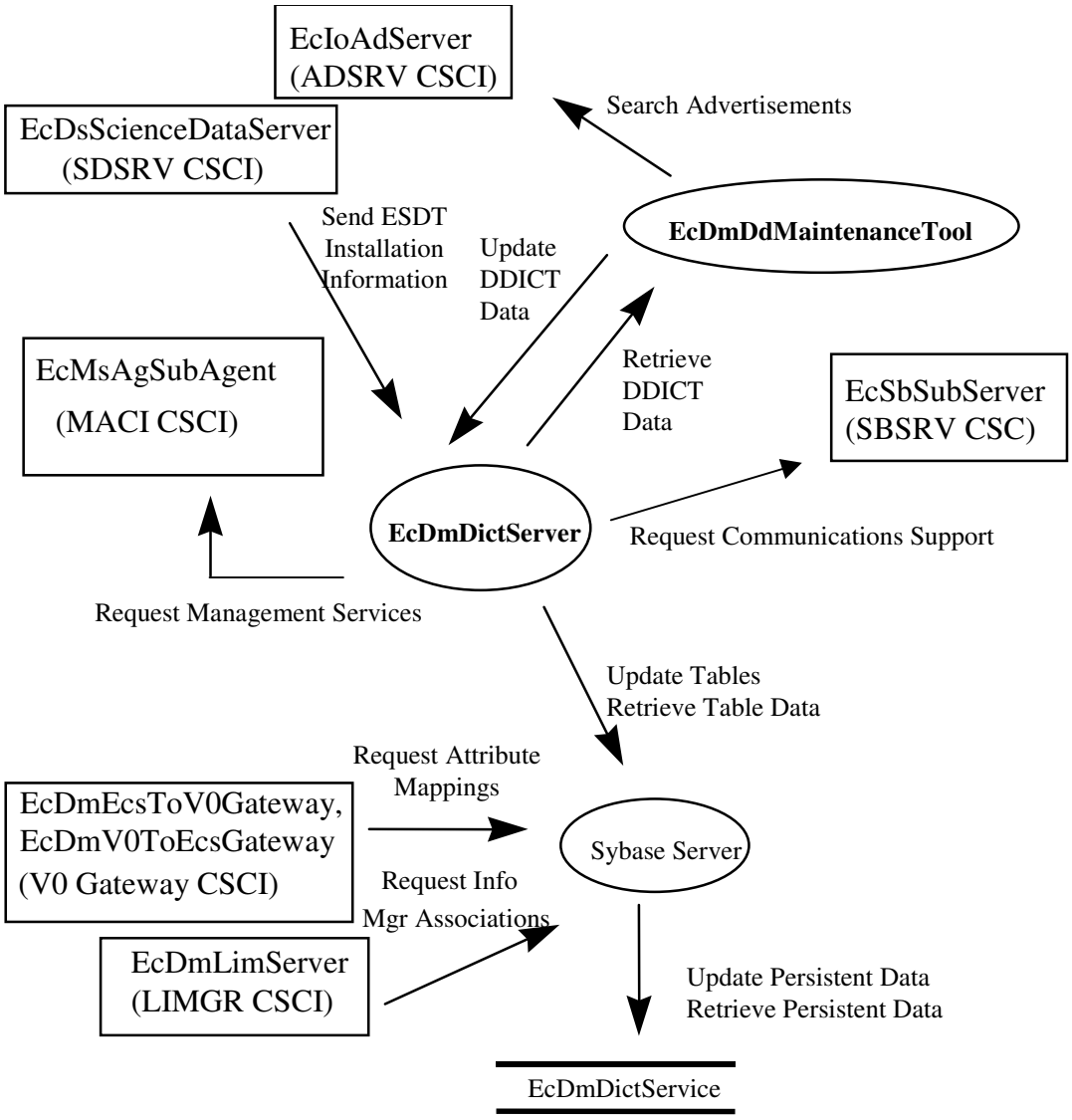

Figure 4.4.1.3-1. Data Dictionary CSCI Architecture Diagram

4.4.1.4 Data Dictionary Process Descriptions

Table 4.4.1.4-1 provides descriptions of the processes shown in the Data Dictionary CSCI architecture diagram.

Table 4.4.1.4-1. Data Dictionary CSCI Processes

Process	Type	COTS/ Developed	Functionality
EcDmDictServer	Server	Developed	The Data Dictionary Server is the primary server interface to collection and collection related information for the DMS and other subsystems. It allows DDICT client processes the capability to perform data searches, insertions, updates, or deletions to the collection information held in the DDICT database. The Data Dictionary offers two basic interfaces DDICT Data Search: The Data Dictionary Server allows a user to specify search requests on the Data Dictionary database using a GlParameterList. DDICT Data Insert and Delete: Provides a client process with the capability to insert and delete data within the Data Dictionary database. The Data Dictionary Server supports: Single requests at a time Synchronous request processing Asynchronous request processing
EcDmDdMaintenanceTool	GUI	Developed	Provides a graphical user interface (GUI) to insert, update, or delete schema information held in the DDICT database, allowing DAAC operations staff to maintain the data stored in the Data Dictionary database. The Data Dictionary Maintenance Tool also provides the following capabilities: Import and Export of Valids: The tool allows DAAC operations staff to import and export data collection attribute valids to and from the ECS and V0 IMS for two-way catalog interoperability. Data Collection Attribute and Keyword Mapping: Allows DAAC operations staff to map data collection attributes and keyword valids between the V0 IMS and ECS. The V0 GTWAY CSCI processes (EcDmEcsToV0Gateway and EcDmV0GatewayToEcs) to translate requests between these two systems use this information.
Sybase Server	Server	COTS	The Sybase Server acts as a SQL server for the Data Dictionary, and is only run at the DAACs by DAAC operations staff. Refer to Sybase documentation for details.

4.4.1.5 Data Dictionary Process Interface Descriptions

Table 4.4.1.5-1 provides descriptions of the interface events shown in the Data Dictionary CSCI architecture diagram.

Table 4.4.1.5-1. Data Dictionary CSCI Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Send ESDT Installation Information	One per new ESDT added to ECS	*Library*: EcDmDdClient *Class*: DmDdClSchema Request	*Process:* EcDsScienceDataServer *Library*: DsDe1 *Class*: DsDeDataDictController	EcDsScienceDataServer inserts new collection information into the DMS Data Dictionary database as new Earth Science Data Types (ESDTs) are added to the ECS system.
Update Persistent Data	One per set number of queries	COTS SW Sybase server	*Process:* Sybase server	The Sybase Server updates data persistently stored on disk(s) based on queries from the Data Dictionary Server.
Retrieve Persistent Data	One per set number of queries	COTS Sybase server	*Process:* Sybase server	The Sybase Server retrieves data persistently stored on disk(s) based on search queries from the Data Dictionary Server.
Update DDICT data	One per table information update	*Library*: EcDmDdClient *Class*: DmDdClSchema Request	*Process*: EcDmDdMaintenanceTool *Classes:* DmLmDbiUtilities, DmDdMtMainWindow	A user, via the EcDmDdMaintenanceTool, updates table information within the DDICT database (via the EcDmDictServer) including mapping collection attributes to keywords and mapping collections to information managers.
Retrieve DDICT data	One per Maintenance Tool search	*Library*: EcDmDdClient *Class*: DmDdClRequest	*Process*: EcDmDdMaintenanceTool *Classes:* DmDdMtDBExtract DmDdMtDBUtilities DmDdMtMainWindow	A user, via the EcDmDdMaintenanceTool, searches for data collection information including collection lists, and collection attributes and keyword valids. This is a user initiated event.
Update Tables	One per database update	COTS SW RWDBTools.h++ classes	*Library*: DmDdReqProc *Classes*: DmDdMapper, DmDdProcMsg	The EcDmDictServer updates data within the Data Dictionary database by inserting and deleting collections and collection metadata, attributes and keywords and attribute and keyword mappings.

Table 4.4.1.5-1. Data Dictionary CSCI Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Retrieve Table data	One per retrieve from the database	COTS SW RWDBTools.h++ classes	*Library*: DmDdServer *Class*: DmDdSearchRequest	The EcDmDictServer retrieves data within the Data Dictionary database such as collections and collection metadata, attributes and keywords and attribute and keyword mappings. This is a user-initiated event.
Search Advertisements	One per request for subscription event advertisement search	*Library:* IoAdSearch: *Class*: IoAdApprovedAdvSearchCommand	*Process*: EcDmDdMaintenanceTool *Class*: DmDdMtDatasetGroup	The EcIoAdServer receives search requests (via users) for subscription event and signature service advertisements from the EcDmDdMaintenanceTool for it to obtain the proper signatures for acquiring data granules from the EcDsScienceDataServer (for the insert and update of metadata within SDSRV archives).
Request Attribute Mappings	One per request from V0 Gateway	COTS RWDBTools.h++ *Classes:* Many DDICT classes	*Library:* Persistent *Class*: DmGwAttributeMap	The EcDmEcsToV0Gateway request data collection attribute and keyword mappings from the Data Dictionary database via the Sybase Server to translate requests from the V0 IMS to the ECS and back again.
Request Info Mgr Associations	One per request from Local Information Manager Server	COTS SW RWDBTools.h++ *Classes*: Many DDICT classes	*Library*: DmLmReqProc *Class*: DmLmParser	The EcDmLimServer requests identification of the server to forward requests to from the server associated with a collection in the Dictionary database. This is a user-initiated event.

Table 4.4.1.5-1. Data Dictionary CSCI Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Request service(s) as required	To simplify the interface table for the Request Communications Support event, refer to the CSS section of Appendix B, Software Libraries, for the libraries and classes used to fulfill the services requested by ECS processes.	*Process:* EcDmDictServer	The CSS Process Framework provides a library of services available to each ECS process. The services required to perform the specific process assignments are requested by the process from the Process Framework. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Request management services	Request service(s) as required		*Process:* EcDmDictServer	The EcMsAgSubAgent provides a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include:
	One per command start or stop network applications	*Script*: EcDmDataDictionary AppStart	*Process*: EcMsAgSubAgent	**Lifecycle commands** - The EcMsAgSubAgent process forwards commands to managed hosts in the network to start and to stop applications. On startup, a parameter is passed identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.

4.4.1.6 Data Dictionary CSCI Data Stores

Table 4.4.1.6-1 provides descriptions of the data stores shown in the Data Dictionary CSCI architecture diagram.

Table 4.4.1.6-1. Data Dictionary CSCI Data Stores

Data Store	Type	Functionality
EcDmDictService	Database	The Data Dictionary database, EcDmDictService is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. Data stores in the Data Dictionary database include: • Collection Types: A list of all the data types within the ECS. • Collection Metadata: Various types of collection metadata including instrument, platform, sensor, topic, keyword, temporal and spatial data. • Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the ECS. • Collection Attribute and Keyword Mappings: Associations between the V0 IMS attributes and valid keywords and the ECS attributes and keywords are maintained.

4.4.2 Information Manager Software Description

4.4.2.1 Information Manager Functional Overview

The Information Manager CSCI provides access to data and services accessible at local and remote sites. Information Managers decompose requests and dispatch the request parts to other servers including other information managers, gateways, or data servers. Information managers, gateways and data servers make themselves known, and therefore accessible, by associating themselves with data collections. This association is maintained as part of the data stored within the Data Dictionary database and can be modified by DAAC operations staff using the Data Dictionary Maintenance Tool.

There is only one type of information manager that exists at this time, the Local Information Manager (LIMGR).

The current plan is to have one LIMGR per DAAC. Local Information Managers only have visibility to data local to the site. This provides good performance for local queries.

The software can be configured at the DAAC to have only a LIMGR accessible to external DAAC servers.

4.4.2.2 Information Manager Context

Figure 4.4.2.2-1 is the Local Information Manager CSCI context diagram. The diagram shows the events sent to other CSCIs or CSCs and the events the Local Information Manager CSCI receives from other CSCIs or CSCs. Table 4.4.2.2-1 provides descriptions of the interface events shown in the Local Information Manager context diagram.

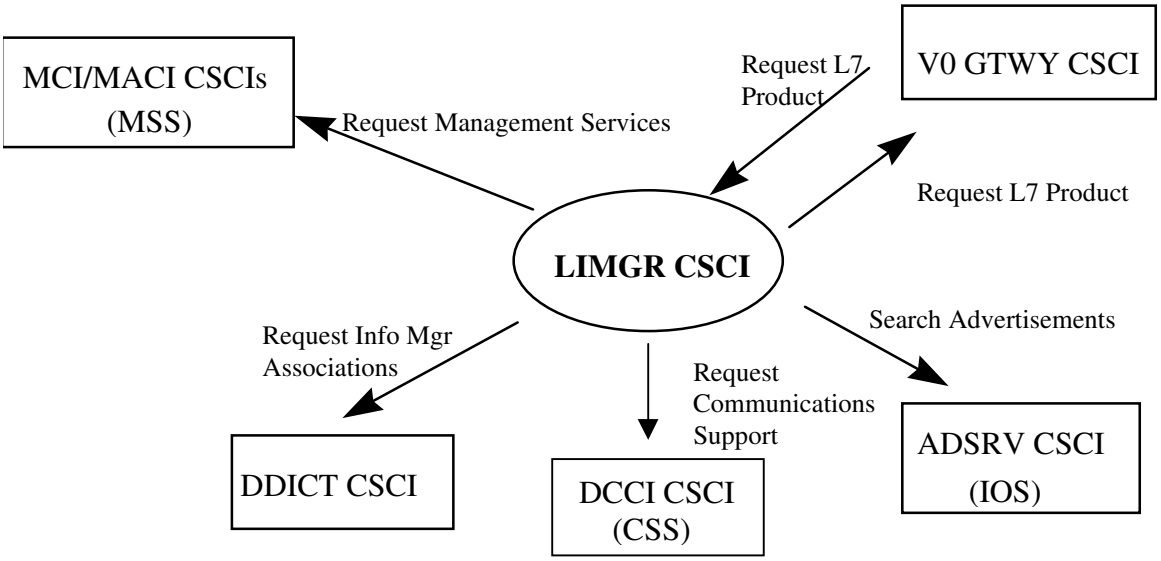

Figure 4.4.2.2-1. Local Information Manager CSCI Context Diagram

Table 4.4.2.2-1. Local Information Manager CSCI Interface Events (1 of 2)

Event	Interface Event Description
Request Info Mgr Associations	The LIMGR CSCI requests the Local Information Manager associated with a given data collection to determine where to forward user requests such as browse and product requests. This is a user-initiated event.
Request L7 Product	The LIMGR CSCI submits Landsat 7 product requests to the V0 IMS to allow the user to be billed by the EDC's Dorran Billing and Accounting System for acquiring Landsat 7data granules. This is a user-initiated event.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Search Advertisements	The ADSRV CSCI receives requests to search for subscription event and signature service advertisements from the LIMGR CSCI. This is a user-initiated event.

Table 4.4.2.2-1. Local Information Manager CSCI Interface Events (2 of 2)

Event	Interface Event Description
Request management services	The MACI and MCI CSCIs provide a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run. The MCI CSCI also interfaces with other subsystems to perform the following: • **DMS Order/Request tracking update** - The LIMGR CSCI interfaces with the MCI CSCI Order/Request Tracking service to create a user product order. • **User Profile Request** - The MCI CSCI provides requesting CSCIs with access to User Profile parameters such as email address and shipping address to support their processing activities.

4.4.2.3 Local Information Manager Architecture

Figure 4.4.2.3-1 is the Local Information Manager architecture diagram. The diagram shows the events sent to the LIMGR CSCI processes and the events the LIMGR CSCI processes send to other processes.

The LIMGR CSCI is comprised of one process, EcDmLimServer, a background server for the DAAC and forwards requests to the V0 gateway processes. Typically a DAAC is configured to run one server as shown in Figure 4.4.2.3-1.

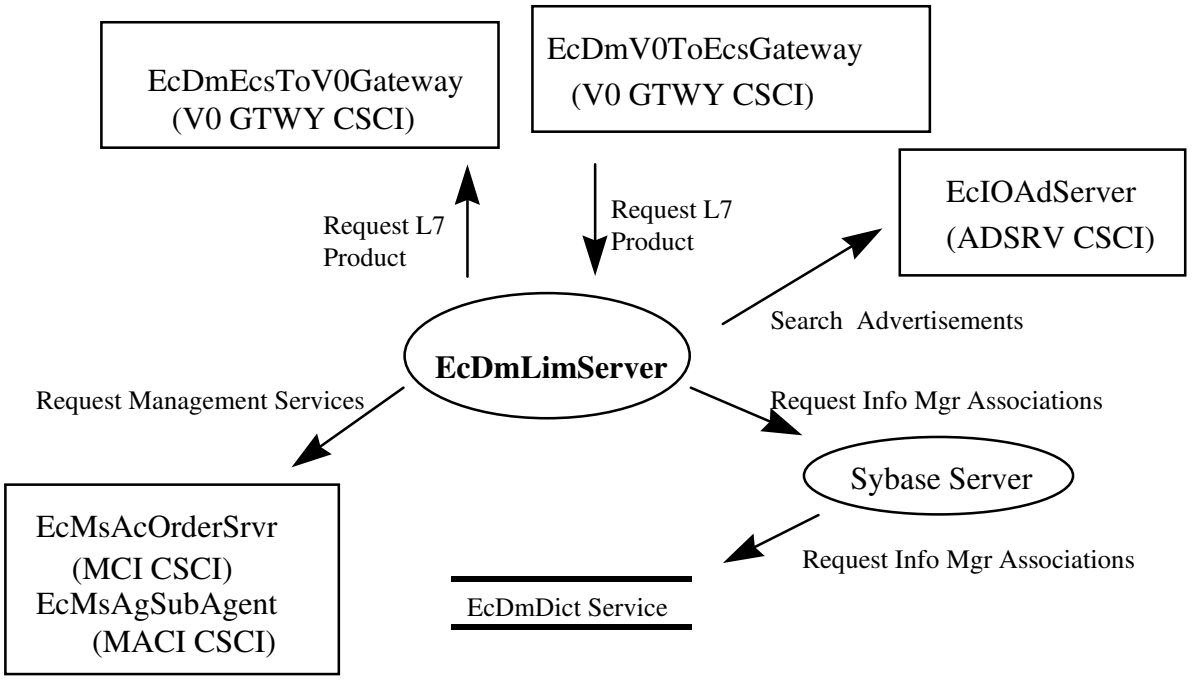

Figure 4.4.2.3-1. Local Information Manager CSCI Architecture Diagram

4.4.2.4 Local Information Manager Process Descriptions

Table 4.4.2.4-1 provides descriptions of the process shown in the Local Information Manager CSCI architecture diagram.

Table 4.4.2.4-1. Local Information Manager CSCI Process

Process	Type	COTS/ Developed	Functionality
EcDmLimServer	Server	Developed	The EcDmLimServer receives acquire requests from the EcDmV0ToEcsGateway, translates it, and forwards it to the local EcDsScienceDataServer or the EcDmEcsToV0Gateway. The EcDmLimServer determines where to send the data types requested and queries the EcDmDictServer to find the server to contact. The results received from the local EcDmV0ToEcsGateway or EcDsScienceDataServer are translated appropriately and returned to the requester. Major Interfaces: Product request: Provides the capability for client processes to submit data acquire requests for obtaining data granules. The EcDmLimServer supports synchronous request processing, asynchronous request processing, and multiple concurrent requests.

4.4.2.5 Local Information Manager Process Interface Descriptions

Table 4.4.2.5-1 provides descriptions of the interface events shown in the Local Information Manager CSCI architecture diagram.

Table 4.4.2.5-1. Local Information Manager Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated By	Event Description
Request L7 Product	One per L7 product request	*Library:* EcDmLmClient *Classes:* DmLmClRequest, DmLmClRequestServer	*Library:* DmLmExIf *Class:* DmLmInfoMgrIF	The EcDmLimServer submits Landsat 7 product requests to the V0 IMS for billing and accounting.
Search Advertisements	One per advertisement search request sent to IOS	*Library:* IoAdSearch *Class:* IoAdApproved AdvSearchCommand	*Library:* DmLmReqProc *Class:* DmLmProductPlan	The EcIoAdServer receives requests to search for subscription event and signature service advertisements from the EcDmLimServer. The EcDmLimServer obtains the proper signatures for acquiring data granules from the EcDsScienceDataServer (for the insert and update of metadata within the SDSRV archives).
Request Info Mgr Associations	One per LIMGR request for user forwarding information	*COTS SW* RWDBTools.h++ *Classes:* Many RWDB Tools classes	*Library:* DmLmReqProc *Class:* DmLmParser	The EcDmLimServer requests the Information Manager associated with a given data collection to determine where to forward user requests such as browse and product requests.

Table 4.4.2.5-1. Local Information Manager Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated By	Event Description
Request management services				The EcMsAgSubAgent and EcMsAcOrderSrvr provide a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include:
	One per command to start or stop network applications	*Script* EcDmLimServerStart	*Process*: EcMsAgSubAgent	**Lifecycle commands** - The EcMsAgSubAgent process forwards commands to managed hosts in the network to start and to stop applications. On startup, a parameter is passed identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
	One per notice received	*Library*: MsAcClnt *Class*: EcAcOrderCMgr	*Library*: DmLmReqProc *Class*: DmLmProductPlan	**DMS Order/Request tracking update** - The EcDmLimServer has an interface with the EcMsAcOrderSrvr (Order/Request Tracking service) to create a user product order.

4.4.2.6 Local Information Manager Data Stores

Table 4.4.2.6-1 provides descriptions of the data stores shown in the Local Information Manager CSCI architecture diagram.

Table 4.4.2.6-1. Local Information Manager Data Stores

Data Store	Type	Functionality
EcDmDictService	Database	The Data Dictionary database, EcDmDictService, is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. The DDICT database is replicated wholly to each DAAC. The data stores in the Data Dictionary database used by the Local Information Managers are: • Collection Types: A list of all the data types within the ECS. • Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the ECS are used by the LIM to validate request attribute parameters. • Collection to Information Manager Mappings: Collections are associated with data servers and Information Managers at a given DAAC site for user requests to be forwarded to the correct data archive for processing.

4.4.3 V0 Gateway Software Description

4.4.3.1 V0 Gateway Functional Overview

The Version 0 Gateway (V0 GTWY) CSCI provides interoperability with the V0 Information Management System for inventory searches, browse requests, and product orders.

The V0 GTWAY CSCI is comprised of two processes, the V0 to ECS Gateway server and the ECS to V0 Gateway server. Both servers combine to provide two-way catalog interoperability between the V0 IMS and the ECS.

Queries are passed between the V0 IMS and the V0 Gateway processes using the Object Description Language (ODL) format. The V0 GTWAY CSCI translates ODL requests used by the V0 Gateway into V0 IMS requests via the ECS Hierarchical Data Format (HDF) since HDF is used by the V0 GTWAY CSCI and ECS servers. The structure of the V0 ODL messages is documented in "Messages and Development Data Dictionary for v5.0 of IMS Client" (IMS V0-PD-SD-002 v1.0.14 950928).

Since the V0 IMS uses different attributes to describe data collections within its data archive, the V0 GTWAY CSCI translates those attributes as defined in the ECS. To perform the translation, the V0 Gateway uses the data collection attribute and valid keyword mapping information contained within the Data Dictionary database to translate the V0 attributes into equivalent ECS attributes.

4.4.3.2 V0 Gateway Context

Figure 4.4.3.2-1 is the V0 GTWAY CSCI context diagram. The diagram shows the events sent to the V0 GTWAY CSCI and the events the V0 GTWAY CSCI sends to other CSCIs.

Table 4.4.3.2-1 provides descriptions of the interface events shown in the V0 GTWAY CSCI context diagram.

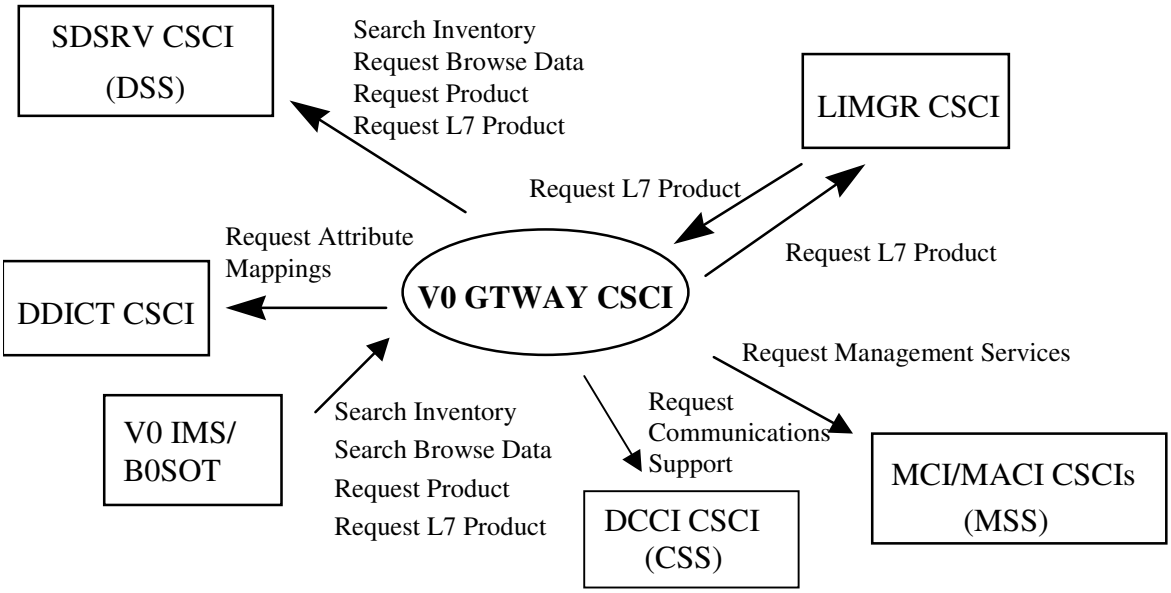

Figure 4.4.3.2-1. V0 GTWAY CSCI Context Diagram

Table 4.4.3.2-1. V0 GTWAY CSCI Interface Events (1 of 2)

Event	Interface Event Description
Search Inventory	The V0 GTWAY CSCI submits inventory search requests to the SDSRV CSCI within the DSS. This is a user-initiated event.
Request Product	The V0 GTWAY CSCI submits product requests to the SDSRV CSCI within the DSS. This is a user-initiated event.
Request L7 Product	The V0 GTWAY CSCI submits Landsat 7 product requests to the V0 IMS for billing and accounting. This is a user-initiated event.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.

Table 4.4.3.2-1. V0 GTWAY CSCI Interface Events (2 of 2)

Event	Interface Event Description
Request management services	The MCI and MACI CSCIs provide a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run. The MCI CSCI also interfaces with other CSCIs to perform the following: • **DMS Order/Request tracking update** - The V0 GTWAY CSCI interfaces with the MCI CSCI Order/Request Tracking service (EcMsAcOrderSrvr) to create a user product order. • **User Profile Request** - The MCI CSCI provides requesting CSCIs with access to User Profile parameters such as e-mail address and shipping address to support their processing activities.
Request/Search Browse Data	The V0 GTWAY CSCI receives browse requests from the V0 IMS or B0SOT and submits the browse requests to the SDSRV CSCI within the DSS. This is a user-initiated event.
Request Attribute Mappings	The V0 GTWAY CSCI requests data collection attribute and keyword mappings from the Data Dictionary database via the Sybase Server to translate requests from the V0 IMS to the ECS protocol and back again. This is a user-initiated event.

4.4.3.3 V0 Gateway Architecture

Figure 4.4.3.3-1 is the V0 GTWAY CSCI architecture diagram. The diagram shows the events sent to the V0 GTWAY CSCI processes and the events the V0 GTWAY CSCI processes send to other processes.

The V0 GTWAY CSCI is comprised of two processes as shown in the V0 GTWAY CSCI architecture diagram.

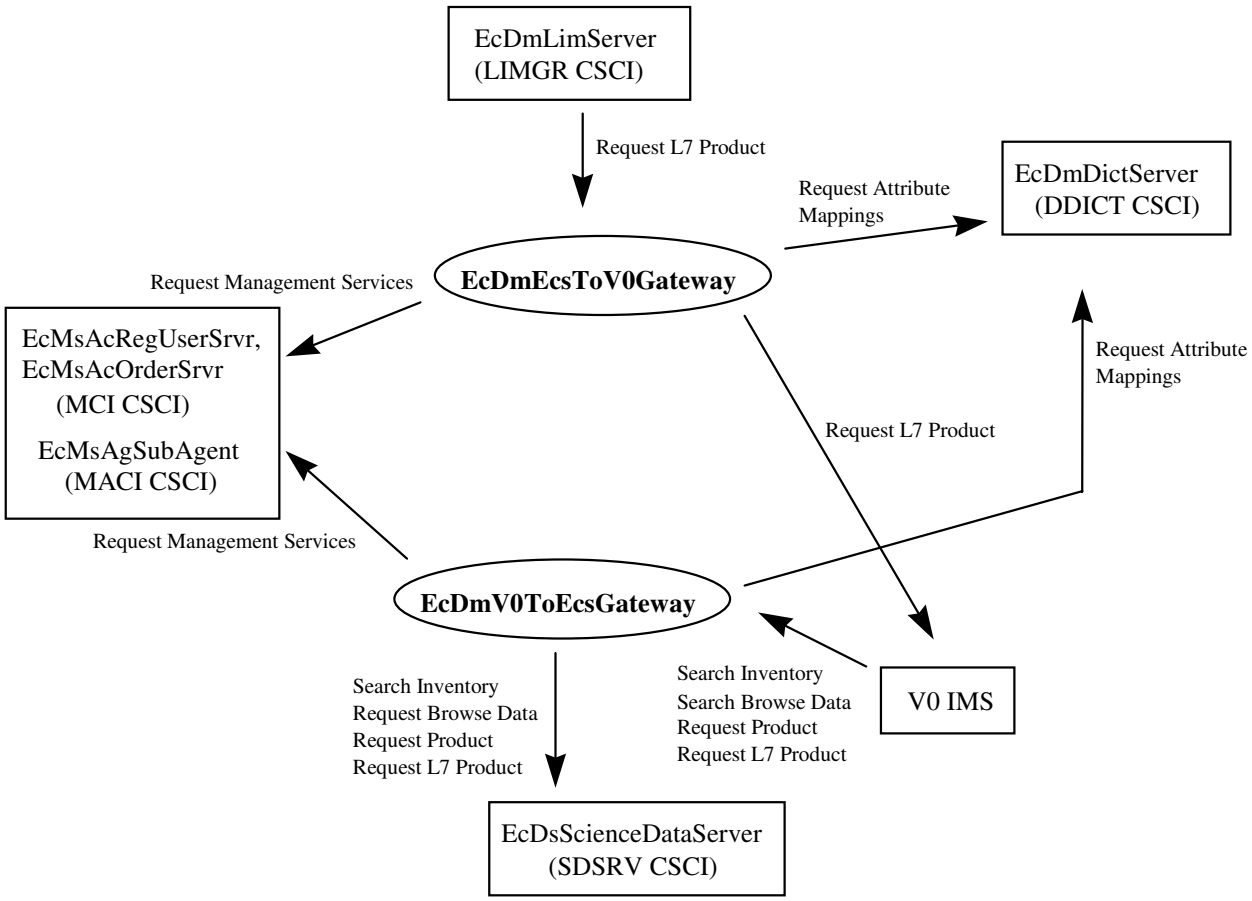

Figure 4.4.3.3-1. V0 GTWAY CSCI Architecture Diagram

4.4.3.4 V0 Gateway Process Descriptions

Table 4.4.3.4-1 provides descriptions of the processes shown in the V0 GTWAY CSCI architecture diagram.

Table 4.4.3.4-1. V0 GTWAY CSCI Processes

Process	Type	COTS/ Developed	Functionality
EcDmEcsToV0Gateway	Server	Developed	The ECS to V0 Gateway server allows users to search for data and request data in the V0 IMS data archive and request services of the V0 IMS. Major Interfaces: • Inventory Search: The ECS to V0 Gateway allows requesters to perform searches for data granules within the V0 IMS archive. • Browse: Allows users to browse data granules previously found during a search. • Acquire: Provides the capability for requesters to submit data acquire requests for obtaining billing and accounting information from the V0 IMS. Server Supports: • Synchronous request processing • Asynchronous request processing • Multiple concurrent requests
EcDmV0ToEcsGateway	Server	Developed	The V0 to ECS Gateway server allows users of the V0 IMS to query on data and services defined within the ECS. Major Interfaces: • Inventory Search: Allows a user to perform searches for data granules within the ECS archive. • Browse: Allows users to browse data granules previously found during a search. • Product request: Provides the capability for users to submit data acquire requests for obtaining data granules from the ECS archive. Server Supports: • Synchronous request processing • Asynchronous request processing • Multiple concurrent requests

4.4.3.5 V0 Gateway Process Interface Descriptions

Table 4.4.3.5-1 provides descriptions of the interface events shown in the V0 GTWAY CSCI architecture diagram.

Table 4.4.3.5-1. V0 GTWAY CSCI Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated By	Event Description
Search Inventory	One per inventory search request	*Library:* DsCl *Classes:* DsClESDTReference Collector, DsClRequest	*Library:* RequestProcessing *Class:* DmGwInvSearchRequest	The EcDmV0ToECSGateway submits inventory search requests to the EcDsScienceDataServer. This is a user-initiated event.
Request/Search Browse Data	One per browse request	*Library:* DsCl *Classes:* DsClESDTReference Collector, DsClRequest	*Library:* RequestProcessing *Class*: DmGwBrowseRequest:	The EcDmV0ToECSGateway submits browse requests to the EcDsScienceDataServer. This is a user-initiated event.
Request Product	One per product request	*Library:* DsCl *Classes*: DsClESDTReference Collector, DsClRequest	*Library:* RequestProcessing *Class:* DmGwAcquireRequest	The EcDmV0ToECSGateway submits product requests to the EcDsScienceDataServer. This is a user-initiated event.
Request L7 Product	One per L7 product request	*COTS SW Library*: ik *COTS Function:* Ik_txODL	*Library:* DmGwV0If *Class*: DmGwEcsServRequest	The EcDmEcsToV0Gateway submits Landsat 7 product requests to the V0 IMS for billing and accounting. This is a user-initiated event.
Request Attribute Mappings	One per data request to DDICT	*COTS SW Library:* RWDBTools.h++ *Class:* Many V0 classes	*Library:* Persistent Class: DmGwAttributeMap	The EcDmEcsToV0Gateway and EcDmV0ToECSGateway request data collection attribute and valid keyword mappings from the EcDmDictServer to translate requests from the V0 IMS to the ECS and back again. This is a user-initiated event.

Table 4.4.3.5-1. V0 GTWAY CSCI Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated By	Event Description
Request management services				The EcMsAgSubAgent and EcMsAcRegUserSrvr provide a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include:
	One per command to start or stop network applications	*Script:* EcDmV0Gateway AppStart	*Process*: EcMsAgSubAgent	**Lifecycle commands** - The EcMsAgSubAgent process forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
	One per profile request	*Library:* MsAcClnt *Class:* EcAcProfileMgr	Process: EcDmV0ToEcsGateway Class: DmGwRequestReceiver	**User Profile Request** - The EcMsAcRegUserSrvr provides requesting processes with access to User Profile parameters such as e-mail address and shipping address to support their processing activities.

4.4.3.6 V0 Gateway Data Stores

Table 4.4.3.6-1 provides descriptions of the data stores shown in the V0 GTWAY CSCI architecture diagram. The V0 to ECS and ECS to V0 Gateway processes access the Data Dictionary data store.

Table 4.4.3.6-1. V0 GTWAY CSCI Data Store

Data Store	Type	Functionality
EcDmDictService	Database	The Data Dictionary database, EcDmDictService is a Sybase relational database that persistently stores the collection and collection related information on a physical disk medium. The DDICT database is replicated wholly to each DAAC. The data stores in the Data Dictionary database used by the V0 GTWAY CSCI are: Collection Types: A list of all the data types within the ECS. Collection Attributes and Keywords: Attributes and keywords associated with collections originating within and outside the ECS are used by the V0 GTWAY CSCI to translate requests between the V0 IMS and the ECS.

4.4.4 Data Management Subsystem Hardware

The primary components which comprise the Data Management Subsystem include two hardware CIs, Data Management Hardware CI (DMGHW) and Interface Hardware CI (INTHW), co-owned by the Interoperability Subsystem, as described below. Custom code and applications are loaded on the internal disks of all hosts to prevent dependencies on specific hosts or peripherals. The general-purpose workstations are standalone hosts without fail-over capability. In the event of a host failure, any of the available workstations could be used to support end user DAAC maintenance.

4.4.4.1 Data Management Hardware CI (DMGHW) Description

The DMGHW CI includes general-purpose low-end SUN and HP workstations, and one mid-range HP Server. These workstations are used as end user workstations in maintenance of each of the respective DAAC sites. The Server is used to support Sybase database replication and backup.

4.4.4.2 Interface Hardware CI (INTHW) Description, as used by the Data Management Subsystem

The INTHW CI includes two Interface Servers. The Interface Servers support the Client Subsystem and a portion of the Communications Subsystem. The servers are SUN class machines with detailed specifications in the site specific hardware design diagram, baseline document number 920-TDx-001. Because of their common configuration, these hosts can be configured interchangeably. DMS software runs on these hosts; DDICT, LIMGR and V0 GTWAY. The Data Dictionary Server (EcDmDictServer) allows authorized users to perform data searches, inserts, updates and deletions to data within the Data Dictionary Database. The LIMGR executes a single process, EcDmLimServer, to enable the Information Manager to access data and services from each DAAC site and accept and process requests. The V0 GTWAY consists of multiple processes to allow access to data and services between the ECS

Data Server and the V0 IMS System. Detailed information can be found in the site-specific hardware/software mapping, baseline document number 920-TDx-002.

A SUN SPARC Storage Array, Model 114, is dual ported between both hosts and provides storage for the Data Dictionary Database and Sybase Replication software. A detailed configuration is specified in baseline document number 920-TDx-009.

The Interface Servers are both "hot" and share the resident RAID device. In the event of a host failure, the operational server assumes total ownership of the RAID and all processes. In this state, the server is recognized to be running in degraded mode until recovery is completed.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.5 Interoperability Subsystem Overview

The Interoperability Subsystem (IOS) or Advertising Service allows ECS servers and non-ECS users to insert and subsequently search for Earth Science related services, advertisement providers, and data.

The Advertising Service provides interfaces for supporting browsing, searching, and retrieving of advertisements. Although there is a single correct format for submitting advertisements to the Advertisement Service, different interfaces support database searching, text searching, and hyper-linked data access and retrieval according to the viewing styles such as plain ASCII text, interactive form, or HTML document.

There are two types of advertisements: service and product. Each type is associated with the provider submitting the advertisement. Each type also has sub-types. For example, the Science Data Server submits signature service advertisements, which are a type of service advertisement.

The SDSRV CSCI and non-ECS users advertise data collections and services with the Advertising Service by adding ESDTs. An advertisement describes the data collection with a set of product attributes. Signature services related to an ESDT are also advertised. Signature services, such as acquire, contain a signature and a server UR needed to retrieve granule data from the Data Server. The signature is parsed by a client application to determine what parameters should be passed to the server for this service. In addition, the Subscription Server advertises Subscription Events such as those registered by the Science Data Server.

Product advertisements include collection level metadata and therefore, the attributes reflected in the Advertising Service include a subset of ECS Core Metadata collection level attributes.

The advertisements are stored in a relational database and the Advertising Server provides a COTs interface to the database.

Interoperability Subsystem Context

Figure 4.5-1 is the Interoperability Subsystem context diagram. The diagram shows the events sent to the Interoperability Subsystem and the events the Interoperability Subsystem sends to other subsystems. Table 4.5-1provides descriptions of the interface events shown in the Interoperability Subsystem context diagram.

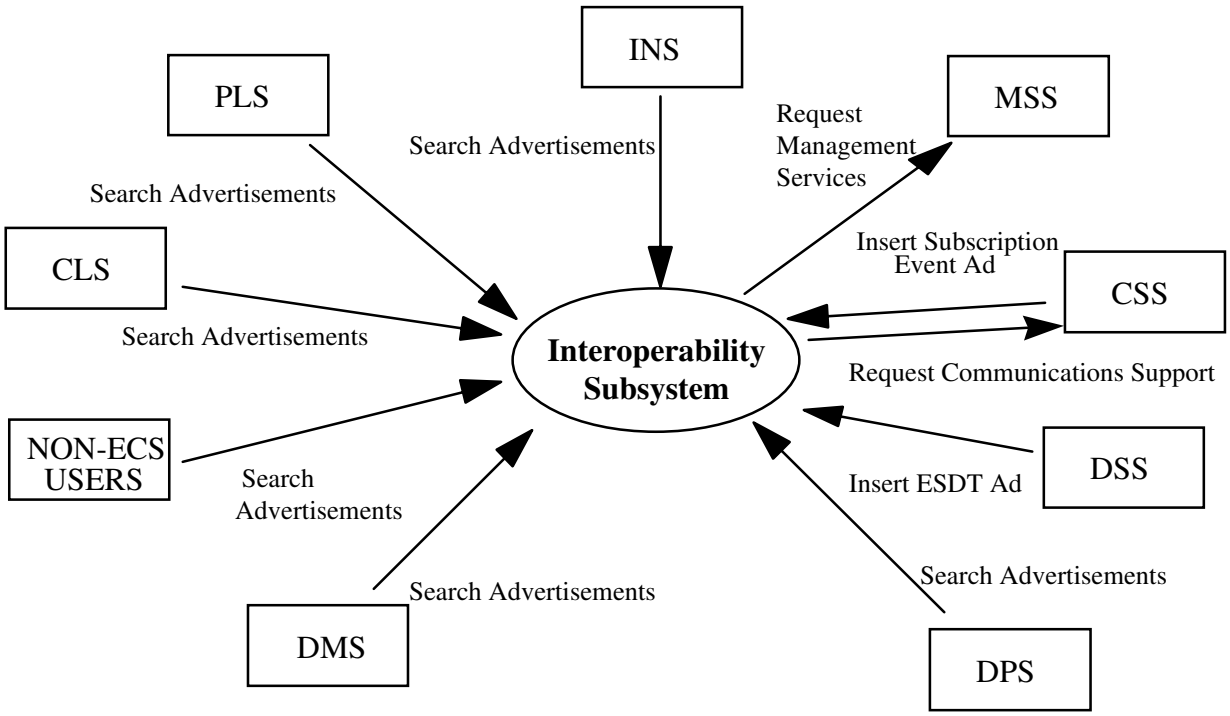

Figure 4.5-1. Interoperability Subsystem Context Diagram

Table 4.5-1. Interoperability Subsystem Interface Events (1 of 2)

Event	Interface Event Description
Insert ESDT Ad	The IOS receives requests to insert advertisements for data types (ESDTs) from the DSS that includes both data product and signature service advertisements.
Request Communications Support	The CSS provides a library of services available to each ECS subsystem. The services required to perform the specific subsystem assignments are requested by the subsystem from the CSS. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User authentication services, and Mode information.
Insert Subscription Event Ad	The IOS receives requests to insert subscription event advertisements from the CSS Subscription Server.
Search Advertisements	The IOS receives requests to search for subscription event and signature service advertisements from the PLS, CLS, DMS, DPS, and INS. These subsystems enter subscriptions with the CSS Subscription Server or obtain the proper signatures for acquiring data granules from the DSS (for the insert and update of metadata within the DSS). Non-ECS users also search for advertisements, which are essentially directory searches for the types of data that exist in the system.

Table 4.5-1. Interoperability Subsystem Interface Events (2 of 2)

Event	Interface Event Description
Request management services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) in which the application should run.

Interoperability Subsystem Structure

The IOS is comprised of one CSCI, ADSRV, and one HWCI, the Interface Hardware CI. The Interface Hardware CI is shared with the Data Management Subsystem

- The Advertising Service (EcIoAdServer) is a software configuration item. The Advertising Service manages Earth Science related advertisements. The advertisement information is stored persistently in a relational Database Management System (DBMS). The Advertising Service data is replicated within each DAAC using Sybase COTS software.
- The Interoperability Subsystem contains one hardware CI, the Interface Hardware (INTHW) co-owned by the Data Management Subsystem hardware. The INTHW CI provides processing and storage for the ADSRV (Advertising Service) software configuration item.

Use of COTS in the Interoperability Subsystem

- RogueWave's Tools.h++

  The Tools.h++ class libraries are used by the IOS to provide basic functions and objects such as strings and collections. These libraries must be installed with the IOS software for any of the IOS processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries are used to interact with the Sybase database SQL server. The use of DBTools buffers the IOS processes from the relational database used. These libraries must be installed with the IOS for the Advertising Server to run and allow client processes to perform queries of Advertising database information.

- Sybase Server

  Sybase's SQL server provides access for the Advertising Service to insert, update and delete advertisement database information. The Sybase SQL Server must be running during operations for the Advertising Server to execute search and update requests on the Advertisement database.

- Netscape Enterprise Server

  Netscape's Enterprise server is used by the Advertising Service CSCI for interpreting Hypertext Transport Protocol (HTTP) allowing users to search, insert, and maintain advertisements. Hypertext Markup Language (HTML) web pages are included as part of the Advertising Service CSCI to allow access via the Netscape Enterprise Servers at the DAACs to make requests.

4.5.1 Advertising Service Software Description

4.5.1.1 Advertising Service Functional Overview

The Advertising Service (ADSRV) CSCI is comprised of two processes, the Advertising Server and the Earth Science On-line Directory (ESOD).

The Advertising Server is a background process that interacts with the Advertising persistent store for searching, inserting and updating advertisements.

The ESOD is a combination of HTML web pages and CGI programs called from the HTML web pages to communicate with the Advertising Server. The web pages provide an interface to allow users to:

- Search for Advertisements: Users can search for Earth Science related data and services through the web interfaces of the Earth Science On-line Directory. Searches are done with specific attributes or with wild card text.

4.5.1.2 Advertising Service Context

Figure 4.5.1.2-1 is the Advertising Service CSCI context diagram. The diagram shows the events sent to the ADSRV CSCI and the events the ADSRV CSCI sends to other CSCIs. Table 4.5.1.2-1 provides descriptions of the interface events shown in the ADSRV CSCI context diagram.

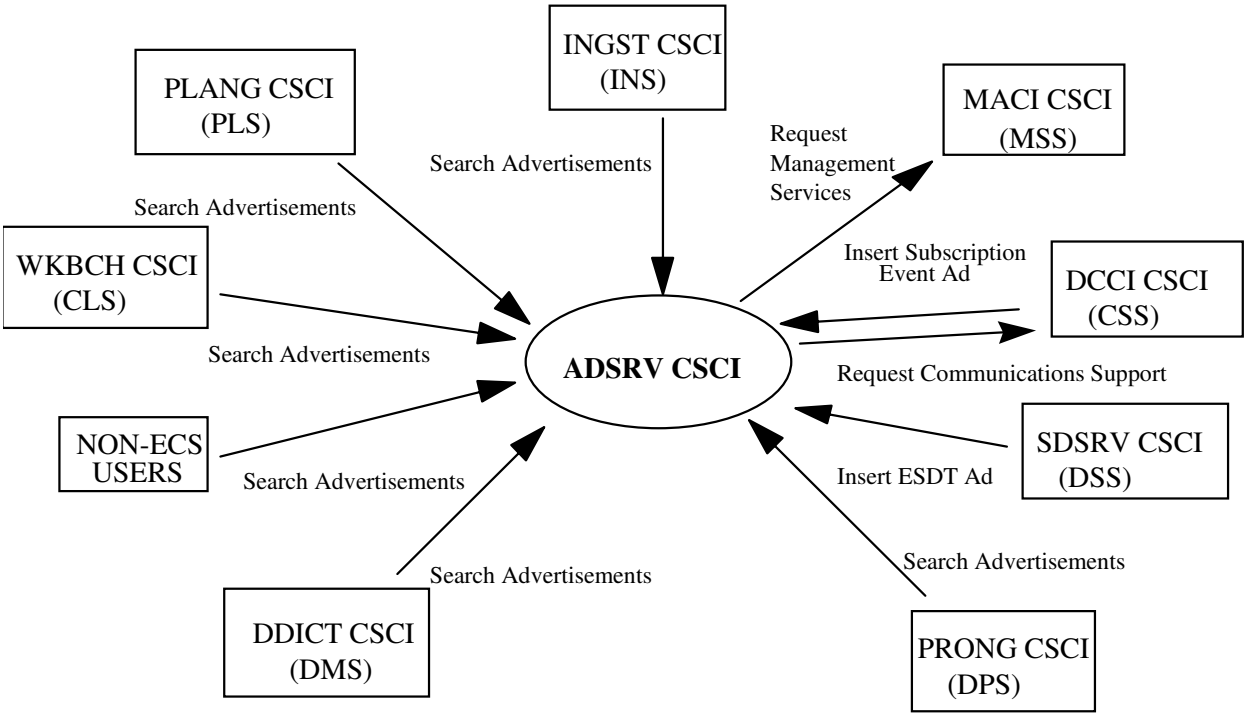

Figure 4.5.1.2-1. Advertising Service CSCI Context Diagram

Table 4.5.1.2-1. Advertising Service CSCI Interface Events (1 of 2)

Event	Interface Event Description
Insert ESDT Ad	The ADSRV CSCI receives requests to insert advertisements for data types (ESDTs) from the SDSRV CSCI that includes both data product and signature service advertisements.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User authentication services, and Mode information.
Insert Subscription Event Ad	The ADSRV CSCI receives requests to insert subscription event advertisements from the DCCI CSCI Subscription Server.
Search Advertisements	The ADSRV CSCI receives requests to search for subscription event and signature service advertisements from the PLANG, WKBCH, DDICT, PRONG, and INGEST CSCIs. These CSCIs enter subscriptions with the DCCI CSCI Subscription Server or obtain the proper signatures for acquiring data granules from the SDSRV CSCI (for the insert and update of metadata within the SDSRV archives). Non-ECS users also search for advertisements, which are essentially directory searches for the types of data that exist in the system.

Table 4.5.1.2-1. Advertising Service CSCI Interface Events (2 of 2)

Event	Interface Event Description
Request management services	The MACI CSCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.

4.5.1.3 Advertising Service Architecture

Figure 4.5.1.3-1 is the ADSRV CSCI architecture diagram. The diagram shows the events sent to the ADSRV CSCI processes and the events the ADSRV CSCI processes send to other processes.

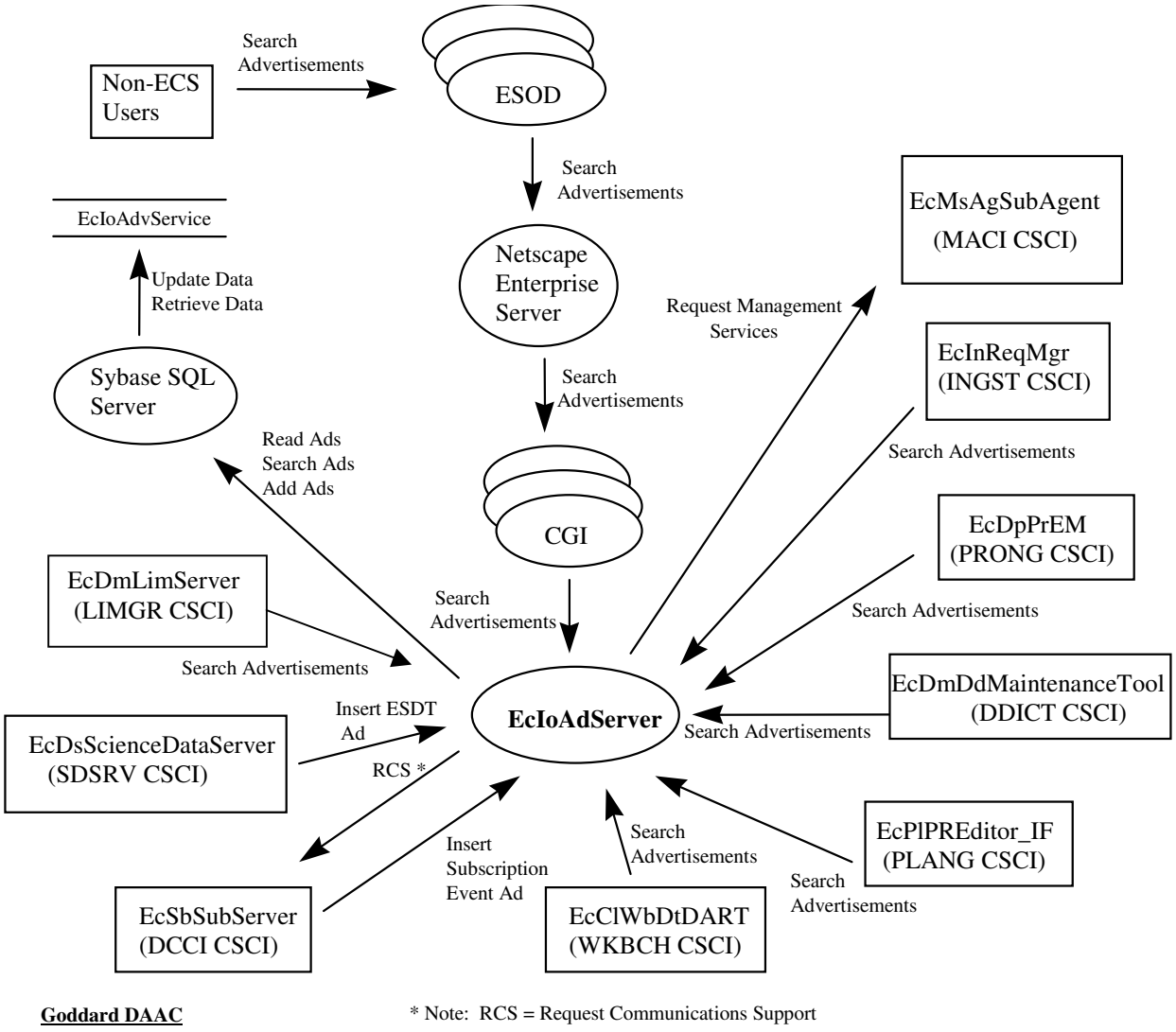

Figure 4.5.1.3-1. Advertising Service Architecture Diagram.

4.5.1.4 Advertising Service Process Description

Table 4.5.1.4-1 provides descriptions of the processes shown in the Advertising Service architecture diagram.

Table 4.5.1.4-1. Advertising Service Processes

Process	Type	COTS/ Developed	Functionality
EcIoAdServer	Server	Developed	The Advertising Server is the only CSCI within the Interoperability subsystem. Users of the Advertising Server use it for searching, inserting, deleting and updating various types of advertisements. The Advertising server uses a relational DBMS server (Sybase) for persistent storage of the advertisements. The Sybase server is shared with the DMS software configuration items. The Advertising Service offers two basic interfaces • Advertising Search: The Advertising Server allows a user to specify search requests on the Advertising database. Searches include searches for data, signature service and subscription event advertisements. • Advertisement Insert and Delete: Provides a user with the capability to insert and delete advertisements within the Advertising database. The Advertising Server supports: • Multiple concurrent requests • Synchronous request processing • Asynchronous request processing
Earth Science On-line Directory (ESOD)	HTML	Developed	The Advertising Service user interface uses generic HTML that is accessible via common web browsers (no JAVA involved). This is the CSC that uses the HTML Framework to build the actual HTML files that are viewed by the users using a Web browser. There are a number of programs associated with the HTML interfaces. See the Common Gateway Interface (CGI) event for details.
Netscape Enterprise Server	Server	COTS	The Netscape Enterprise Server runs at the DAACs and receives and interprets the Hypertext Transport Protocol (HTTP) from the ESOD web pages. Refer to Netscape Server administration documentation for further information.
CGI	CGI	Developed	The Earth Science On-line Directory HTML interface communicates with the Advertising Server through the use of CGI programs. The CGI programs are run on the ADSHW CI after being spawned from the Netscape Enterprise Server. A number of these CGIs exist within the Advertising Service for forwarding requests to the Advertising server and receiving results back. The CGI program process names are IoAdEsodamGroups, IoAdEsodamGroupSearch, IoAdEsodamModeration, IoAdEsodamModerationForm, IoAdEsodamModerationGroups, IoAdEsodamModerationQueue, IoAdEsodamObsoleteReqs, IoAdEsodContributionForm, IoAdEsodContributions, IoAdEsodEntryDetail, IoAdEsodExamples, IoAdEsodGroupDetail, IoAdEsodScienceSearch, IoAdEsodScienceSearchForm, IoAdEsodTextSearch, IoAdEsodTextSearchForm, IoAdEsodUpdateTemplate, and IoAdEsodWhatsNew.
Sybase Server	Server	COTS	The Sybase Server acts as a SQL server for the Advertising Service. Refer to Sybase documentation for details.

4.5.1.5 Advertising Service Interface Descriptions

Table 4.5.1.5-1 provides descriptions of the interface events shown in the Advertising Service architecture diagram.

Table 4.5.1.5-1. Advertising Service Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Search Advertise-ments	One per request to search advertise-ments	Library: IoAdSearch Class: IoAdApprovedAdv Searchcommand CUSTOM libraries: IoAdSearch, IoAdCore, IoAdSubs	*CLS Process*: EcClWbDtDART *CLS Class*: SubscriptionInterface *PLS Library:* PlCore1 *PLS Class:* PlDataType *DMS Process:* EcDmDdMaintenanceTool *DMS Class:* DmDdMtDatasetGroup *DPS Library:* PlCore1 *DPS Class:* PlDataType *DMS Library:* DmLmReqProc *DMS Class:* DmLmProductPlan *INS Library:* InUpdateUR *INS Class:* InUpdateUR	The EcIoAdServer receives requests to search and retrieve advertisements from EcPlPREditor_IF, EcDpPrEM, EcInReqMgr, EcDmDdMaintenanceTool, EcClWbDtDART, and EcSbSubServer; It also receives requests to insert ESDTs from EcDsScienceDataServer, and subscription events from the EcSbSubServer. All inserts are performed at the master site Advertising Server only. Non-ECS users also search for advertisements, which are essentially directory searches for the types of data that exist in the system.
Update Data	One per update request	COTS: Standard SQL Engine	Sybase SQL Server	The Sybase Server updates data persistently stored on disk based on update requests from the Advertising Server.
Retrieve Data	One per search query	COTS: Standard SQL Engine	Sybase SQL Server	The Sybase Server retrieves data persistently stored on disk based on search queries from the Advertising Server.

Table 4.5.1.5-1. Advertising Service Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Search Ads/ Read Ads/ Add Ads	One per request to insert advertise-ment	COTS libraries: RWDBTools.h CUSTOM libraries: IoAdSearch, IoAdCore, IoAdSubs	*Process*: EcIoAdServer Library: IoAdServer	The EcIoAdServer receives requests to search and retrieve advertisements from EcPlPREditor_IF, EcDpPrEM, EcInReqMgr, EcDmDdMaintenanceTool, EcClWbDtDART, and EcSbSubServer; It also receives requests to insert ESDTs from EcDsScienceDataServer, and subscription events from the EcSbSubServer. All inserts are performed at the master site Advertising Server only.
Insert Subscription Event Ad	One per request to insert advertise-ment	*Libraries:* IoAdCore, IoAdSubs *Classes:* IoAdSignatureServiceAdv, IoAdApprovedAdv, IoAdGroup, IoAdProvider	*Library:* EcSbSr *Class:* EcSbEvent	The EcIoAdServer receives requests to insert subscription event advertisements from the EcSbSubServer. All inserts are performed at the master site Advertising Server only.
Insert ESDT Ad	One per data type being inserted	*Libraries:* IoAdcore, IoAdSubs *Classes:* IoAdSignatureServiceAdv, IoAdApprovedAdv, IoAdGroup, IoAdProvider	*Process:* EcDsScienceDataServer *Class:* DsDeIOSController	The EcIoAdServer receives requests to insert ESDT advertisements from the EcDsScienceDataServer including both data product and signature service advertisements. All inserts are performed at the master site Advertising Server only.

Table 4.5.1.5-1. Advertising Service Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	One per service request.	To simplify the interface table for the Request Communications Support event, refer to the CSS section of Appendix B, Software Libraries, for the libraries and classes used as the interfaces to fulfill the services requested by ECS processes.	*Process:* EcIoAdServer	The CSS Process Framework provides a library of services available to each ECS process. The services required to perform the specific process assignments are requested by the process from the Process Framework. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Request Management Services	One per command to start or stop an application	*Script:* EcIoAdServer	*Process:* EcMsAgSubAgent	The EcMsAgSubAgent process provides a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The EcMsAgSubAgent, process forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) in which the application should run.

4.5.1.6 Advertising Service Data Stores

Table 4.5.1.6-1 provides descriptions of the data stores shown in the Advertising Service architecture diagram.

Table 4.5.1.6-1. Advertising Service Data Stores

Data Store	Type	Functionality
EcIoAdService	Database	The Advertising Service database, EcIoAdService, is a Sybase relational database that persistently stores the advertisements and advertisement related information on a physical disk. The types of data stored in the Advertising Service database include: • Data: A list of all the data collections along with their associated metadata within the ECS. • Signature Services: Signature services include the signature required for one server to obtain the services of another server. One example is the acquire signature required for users of the DSS' Science Data Server (EcDsScienceDataServer) to obtain data granules. • Subscription Events: Users or servers within the ECS can subscribe to and be notified of available data.

4.5.2 Interoperability Subsystem Hardware Components

4.5.2.1 Interface Hardware CI (INTHW) Description, as used by the Interoperability Subsystem

The INTHW CI consists of two Interface Servers. In addition, the Interface Servers support the Client Subsystem and a portion of the Communication Subsystem software components. Client and Communication Subsystem related topics are discussed in their respective sections.

The Interface Servers are SUN Server class machines. Detail specifications can be found per the site-specific hardware design diagram, base-line document number 920-TDx-001. Because of their common configuration, these hosts can be configured interchangeably. The ADSRV is the only Interoperability software component that runs on these systems. The Advertising Service provides management of Earth Science related advertisements.

Detailed mappings can be found per the site-specific hardware/software mapping, base line document number 920-TDx-002.

A SUN SPARC Storage Array is dual ported between both hosts and provides storage for the Advertising Database and Sybase Replication components. A detailed configuration is specified per disk partition, base-line document number 922-TDx-009.

In general, custom code and applications are loaded on the internal disks of all hosts. This prevents dependencies on specific hosts or any peripherals.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.6 Planning Subsystem Overview

The Planning Subsystem (PLS) manages the data production activities at ECS sites in support of the operations staff by providing the following capabilities:

- Identifies the data processing tasks (via data processing requests) performed by a site
- Generates the data production plans for scheduling the identified processing tasks
- Coordinates data production with the DSS and the DPS to achieve an automated production system.

Planning Subsystem Context Diagram

Figure 4.6-1 is the context diagram for the PLS. The diagram shows the events sent to other ECS subsystems and the events the PLS receives from other ECS subsystems. Table 4.6-1 provides descriptions of the interface events shown in the Planning Subsystem Context Diagram.

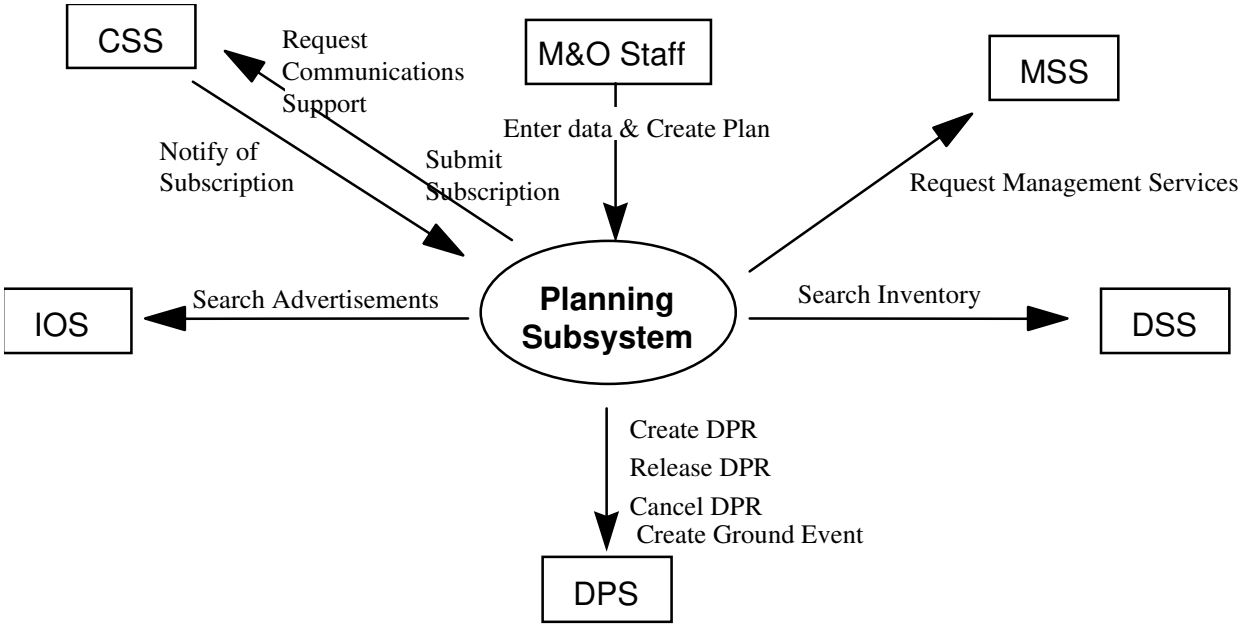

Figure 4.6-1. Planning Subsystem Context Diagram

Table 4.6-1. Planning Subsystem Interface Events

Event	Interface Event Description
Enter data & Create Plan	The M&O staff enter production request data and issues commands to control the creation of a Production Plan.
Search Advertisements	The IOS receives search requests for subscription event and signature service advertisements from the PLS. The PLS enters subscriptions with the Subscription Server within the CSS or obtains the proper signatures for acquiring data granules from the DSS (for the insert and update of metadata within the DSS).
Submit Subscription	The PLS creates a subscription, sent to the CSS, using the advertisement for subscribing to an insert event for an ESDT. In response, PLS receives a corresponding subscription identifier.
Notify of Subscription	A message passing callback in the PLS subscription manager is called, by the CSS, with the UR of the granule inserted into the Data Server as one of the calling parameters.
Search Inventory	The PLS sends inventory search or inspect requests to the DSS to search the ECS inventory/archives (granules). In response, the PLS receives URs for the respective granules satisfying the search.
Create DPR	The PLS sends, to DPS, the Data Processing Request Identification (dprId) and whether the DPR is waiting for external input data.
Release DPR	The PLS sends the dprId to the DPS for DPR release.
Cancel DPR	The PLS sends a request to cancel the dprId to the DPS for the deletion of a DPR.
Request management services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Create Ground Event	The PLS sends the ground event id, resource id, and start time to the DPS.
Request Communications Support	The CSS provides a library of services available to each ECS subsystem. The services required to perform the specific subsystem assignments are requested by the subsystem from the CSS. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User authentication services, and Mode information.

The following paragraphs describe the relationships between the PLS and other ECS subsystems.

DPS Interface

The PLS uses a database link with the DPS Processing CSCI to describe the Product Generation Executives (PGEs) needed to fulfill the production goals. A Data Processing Request (DPR) describes a PGE run to the DPS. A DPR describes the specific input granules, output filenames, and run-time parameters for a PGE, as well as dependencies and predicted run-times. The DPS provides status and processing completion information to the PLS.

DSS Interface

The PLS queries the DSS inventory for data required for processing. If the data exists, the DSS responds to the PLS with granule information (identification, metadata, and location). If the data does not exist, an error message or notification is sent to the PLS.

CSS Interface

The CSS Subscription server provides a notification on the arrival of ECS data. The ECS Advertising service provides the advertisement data required by the PLS to generate subscriptions. The PLS exchanges mode management information with and receives event notifications from the CSS.

MSS Interface

The PLS sends fault management, accounting, security, and performance data to the MSS for logging. The PLS receives configuration management resource information from the MSS for Resource Planning and initializes the PLS database. The PLS receives the actual times of ground events from the MSS log.

Planning Subsystem Structure

The PLS is comprised of one computer software CI, Production Planning (PLANG CSCI) and one hardware CI, Production Planning (PLNHW).

The Planning and Data Processing Subsystems (PDPS) database resides in the PLNHW and serves both planning and scheduling activities.

Use of COTS in the Planning Subsystem

- Hughes- Delphi Scheduling Class Libraries.

  The PLS uses Delphi for scheduling of the Resource Planning Workbench and the Production Planning Workbench. Delphi uses C++ classes to provide user-oriented, integrated, and modular planning and scheduling software utilities.

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide libraries of object strings and collections. These libraries must be installed for the PLS processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries interact with the Sybase database Structured Query Language (SQL) server and buffer the processes from the relational database used.

- ICS' Builder Xcessory

  The Builder Xcessory GUI builder tool modifies displays. The Builder generates the C++ code to produce the Mtool display at run time. There is no operational component of Builder Xcessory needed at run-time.

- Sybase Server

  The Sybase SQL server provides the capabilities to insert, update and delete PDPS database content. The Sybase SQL Server must be operational during the PLS operations.

4.6.1 Production Planning (PLANG) Software Description

4.6.1.1 Production Planning Functional Overview

The PLANG CSCI manages the data production activities at each site by providing the Operations staff with the following capabilities:

- Defining the data processing tasks (via data processing requests) to perform at the site
- Generating data production plans for scheduling processing tasks
- Coordinating data production with the DSS and the DPS to automate the production system.

4.6.1.2 Production Planning Context

Figure 4.6.1.2-1is the PLANG CSCI context diagram. The diagram shows the events sent to the PLANG CSCI and the events the PLANG CSCI sends to other CSCIs and the Operations staff. Table 4.6.1.2-1 provides descriptions of the interface events shown in the PLANG CSCI context diagram.

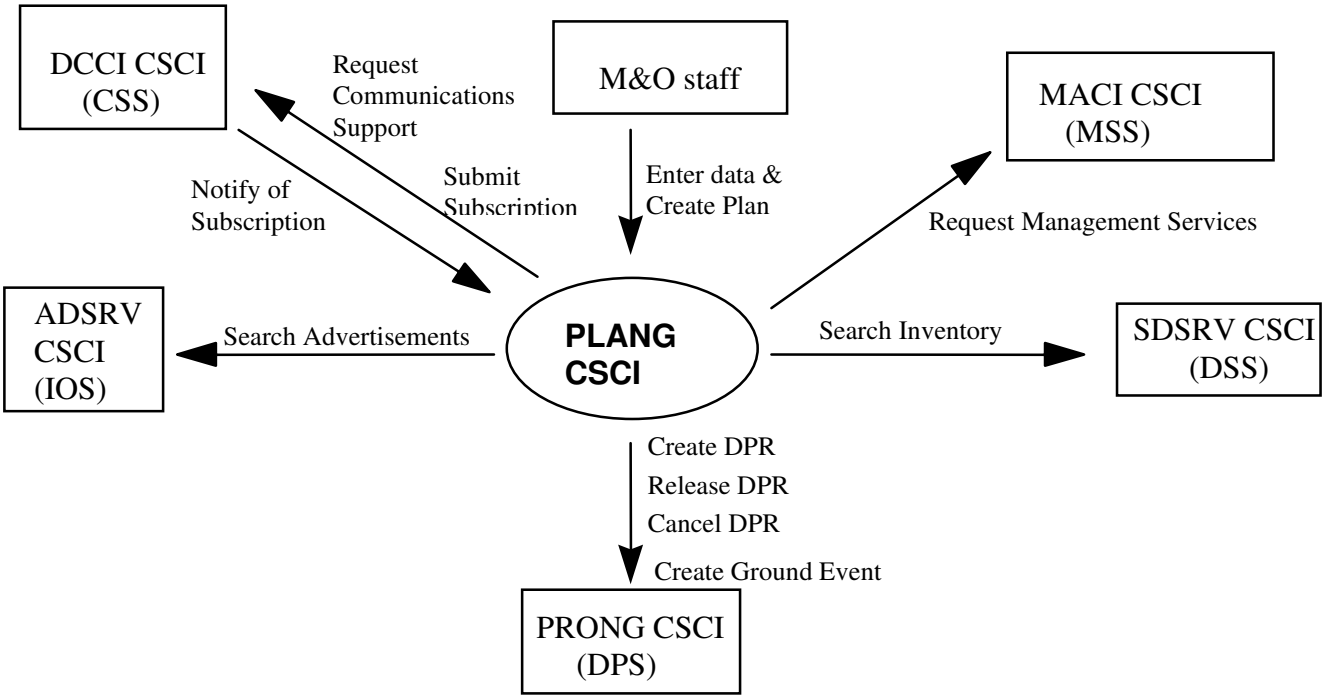

Figure 4.6.1.2-1. PLANG CSCI Context Diagram

Table 4.6.1.2-1. PLANG CSCI Interface Events

Event	Interface Event Description
Enter data & Create Plan	The M&O staff enter production request data and issues commands to control the creation of a Production Plan.
Search Advertisements	The ADSRV CSCI receives search requests for subscription event and signature service advertisements from the PLANG CSCI. The PL ANG CSCI enters subscriptions with the Subscription Server within the CSS or obtains the proper signatures for acquiring data granules from the SDSRV CSCI (for the insert and update of metadata within the SDSRV inventory).
Submit Subscription	The PLANG CSCI creates a subscription using the advertisement for subscribing to an ESDT insert event. In response, the PLANG CSCI receives a subscription identifier.
Notify of Subscription	A message passing callback in the PLANG CSCI subscription manager is called with the granule UR inserted into the SDSRV inventory as a calling parameter.
Search Inventory	The PLANG CSCI sends inventory search or inspect requests to the SDSRV CSCI to search the ECS inventory/archives (granules). In response, the PLANG CSCI receives granule URs satisfying the search.
Create DPR	The PLANG CSCI sends the Data Processing Request Identification (dprId) and whether the DPR is waiting for external input data to the PRONG CSCI.
Release DPR	The PLANG CSCI sends the dprId to the PRONG CSCI.
Cancel DPR	The PLANG CSCI sends a request to cancel the dprId to the PRONG CSCI for the deletion of a DPR.
Request management services	The MACI CSCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Create Ground Event	The PLANG CSCI sends the ground event id, resource id, and start time to the PRONG CSCI.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User authentication services, and Mode information.

PLANG CSCI interfaces include:

PDPS Database Interface (Common database pseudo-interface with DPS)

The PLS retrieves PGE data stored by the DPS Algorithm Integration and Test Tools CSCI. This PGE data includes the PGE executable, the input data type(s) it requires, the output data type(s) it generates, and the resource requirements (e.g., hardware platform, memory, and disk storage). The PLS uses the PGE data to schedule data processing requests with the DPS.

Operator Interface

The Operations staff personnel enter Production Requests into the PLS via the Planning User Interface. Production Requests describe the order for data to be produced by the DPS. Production Requests are used to process new data (Routine Production Requests, also known as standing orders) or for reprocessing data (Reprocessing Production Requests). The PLS uses the PGE profile information from the Production Requests to generate the DPRs needed to fulfill the request for data. The Planning User Interface also issues commands to initiate plan creation, plan activation and plan cancellations, and provide reports and status of plan progress. The Operations staff performs resource planning for the entire DAAC through the Planning User Interface with awareness of the impact of ground events on data processing resources.

4.6.1.3 Production Planning Architecture

Figure 4.6.1.3-1 is the PLANG CSCI architecture. The diagram shows the events sent to the PLANG CSCI processes and the events sent by the PLANG CSCI processes to other processes.

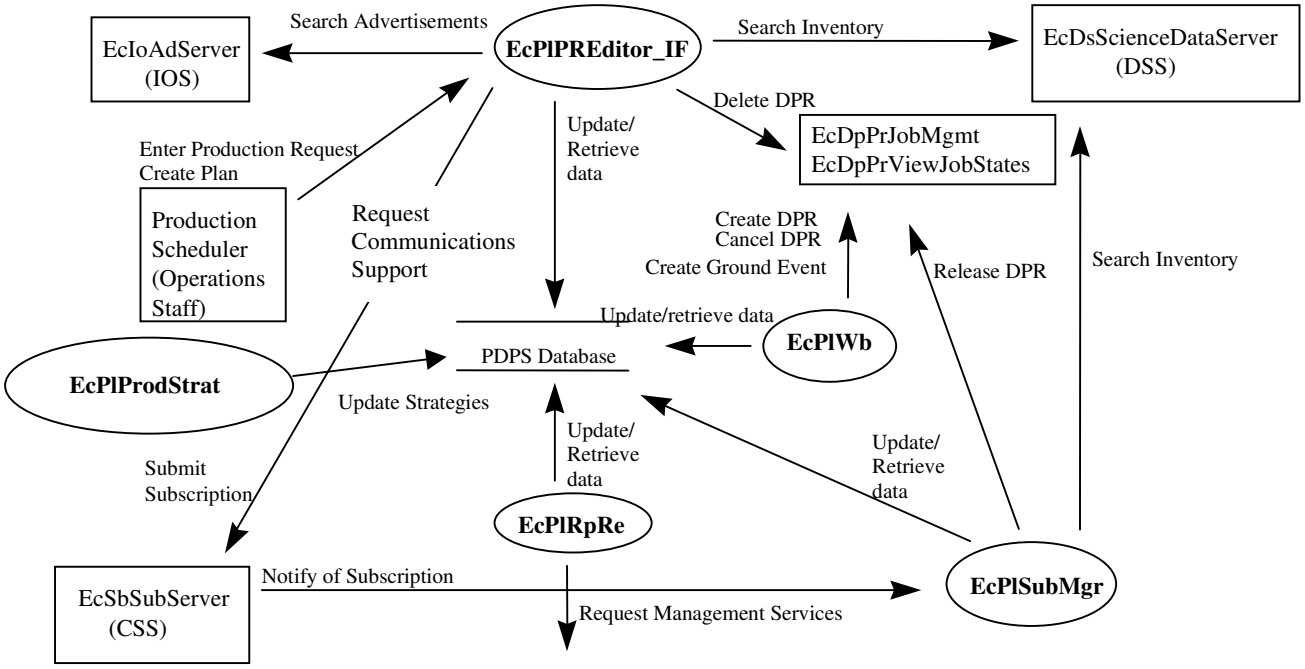

Figure 4.6.1.3-1. PLANG CSCI Architecture Diagram

4.6.1.4 Production Planning Process Descriptions

Table 4.6.1.4-1 provides descriptions of the Production Planning processes shown in the PLANG CSCI architecture diagram.

Table 4.6.1.4-1. PLANG CSCI Processes (1 of 2)

Process	Type	COTS / Developed	Functionality
EcPlRpRe	GUI	Developed code using Delphi Class Libraries.	The Resource Planning Workbench prepares a schedule for the resources at each respective site, and forecasts the start and completion times of the ground events and the impact on the resources used within the schedule. The workbench allows the Operations staff to: • Edit the resources currently available at a site • Associate the resources with production strings (logical groupings of resources used by AutoSys and Data Processing) when allocating resources for a particular PGE. • Create ground events (maintenance, etc.) on the allocated resources
EcPlWb	GUI	Developed code using Delphi Class Libraries.	The Production Planning Workbench prepares a schedule for the production at a site, and forecasts the start and completion times of the activities within the schedule. Specifically, the Workbench allows: • Candidate Plan Creation—from the production requests prepared by the Production Request Editor • Plan Activation—activating a candidate plan • Update of the Active Plan—feedback from the DPS activities are incorporated into the active plan • Cancellation/Modification of the Active Plan. Activating a plan entails rolling a portion of a selected plan into the AutoSys COTS via the DPS. The “schedule” is managed within the DPS. The forecast times generated by the planner are used to set up operator alerts to gross departures from the predicted schedule.
EcPlPREditor_IF	GUI	Developed	The Planning User Interface (Production Request Editor) allows the Operations staff to submit production requests to describing the data products to generate. The production request uses the PGE descriptions (profiles) entered during Algorithm Integration and Test (AI&T) to define the Data Processing Requests. The request adds, modifies, and deletes Production Requests, and reviews and modifies the resulting Data Processing Requests. The user specifies rules for producing the individual DPRs for the reprocessing requests. The production request editor is a distinct application and separate from the workbench because defining a production request is unrelated to the planning of a production request.

Table 4.6.1.4-1. PLANG CSCI Processes (2 of 2)

Process	Type	COTS / Developed	Functionality
EcPlSubMgr	server	Developed	The Subscription Manager receives subscription notifications from the DSS via the CSS. Subscription notification notifies planning of the arrival of required input data. The Subscription Notification is handled through the Infrastructure message passing service and contains URs pointing to the data objects stored in the DSS. The Subscription Manager updates the PDPS database when data is available. When all input data for a DPR is available, the job defined for that DPR is released within the DPS.
EcPlProdStrat	GUI	Developed	The Production Strategies GUI is used to create a set of planning priorities to be applied to each DPR in a plan. This strategy takes user, PGE type, PGE instance, and Production Request priorities into account. This strategy is then saved to the PDPS database.

4.6.1.5 Production Planning Process Interface Descriptions

Table 4.6.1.5-1 provides descriptions of the interface events shown in the PLANG CSCI architecture diagram.

Table 4.6.1.5-1. PLANG CSCI Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Enter Production Request	One per production request	*GUI:* Production Request Editor	Operator	The M&O staff request production by selecting PGE type and the time duration for the PGE to process the input data.
Create Plan	One per created and activated plan	*GUI:* Planning Workbench	Operator	The M&O staff create and activate a plan.
Update/Retrieve data	One per update/retrieve request	*GUI:* Production Request Editor *Process:* EcPlSubMgr	*Process:* EcSbSubServer	Requests to update/retrieve data defining a PGE. Also, allows the PGE to be scheduled and executed. Requests allow updates of granule information (location, size, etc.), processing status, and check pointing.
Update Strategies	One per strategy created.	*GUI:* Production Strategies GUI	Operator	The M&O staff create strategies when certain jobs need to be prioritized over others. The strategy is saved by name and later can be read by the Planning Workbench to prioritize the DPRs in a plan.

Table 4.6.1.5-1. PLANG CSCI Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Search Advertise-ments	One per advertise-ment search request	*Library:* IoAdSearch *Class:* IoAdApprovedAdvSearchCommand	*Library:* PlCore1 *Class:* PlDataType	The EcIoAdServer process receives requests to search for subscription event and signature service advertisements from the EcPlPREditor_IF process. The EcPlPREditor_IF process enters subscriptions via the Subscription Service (within the CSS) or obtains the proper signatures for acquiring data granules from the EcDsScienceDataServer process (within the DSS) for the insert and update of metadata within the DSS inventory.
Submit Subscrip-tion	One per subscription created	*Library:* EcSbSr *Class:* EcSbSubscription	*Library*: PlCore1 *Class*: PlDataType	The EcPlPREditor_IF process creates subscriptions using the advertisement for subscribing to an ESDT insert event.
Notify of Subscrip-tion	One per message passing callback	*Process:* EcPlSubMgr *Class:* PlSubMsgCb	*Process:* EcSbSubServer	The SBSRV process calls a message passing callback in the Subscription Service, with the granule UR inserted into the data server as a calling parameter.
Search Inventory	One per query	*Library:* DsCl *Class:* DsClQuery	*Processes:* EcPlPREditor_IF EcPlSubMgr *Library:* DpPrDssIF *Class:* DpPrDSSInterface	The EcPlPREditor_IF and EcPlSubMgr processes create two types of queries. One type only has the ESDT short name and data start and stop times and the other type also includes spatial coordinates. The EcPlPREditor_IF process queries when the predicted data is available. The EcPlPREditor_IF process creates an ESDT Reference from a UR after receiving an ESDT Reference from a query. The EcDsScienceDataServer returns ESDT References for granules to satisfy the query. The EcPlSubMgr process creates an ESDT Reference from a UR after receiving a subscription notification or receiving an ESDT reference from a query. The EcPlSubMgr process queries when predicted data is not available. The EcDsScienceDataServer returns metadata information about the granule being inspected.
Create DPR	One per list of predecessor DPRs	*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Class:* PlWbScheduler	The EcPlWb process sends the dprId and whether the DPR is waiting for external data to the EcDpPrJobMgmt process.

Table 4.6.1.5-1. PLANG CSCI Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Release DPR	One per dprld send	*Library:* DpPrJM *Class:* DpPrScheduler	*Library:* PlCore1 *Class:* PlDpr	The EcPlSubMgr process Subscription Manager sends the dprId to the EcDpPrJobMgmt process.
Cancel DPR	One per dprld send	*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb *Library:* PlCore1 *Class:* PlDpr	The EcPlWb process sends a request to cancel the dprId to the EcDpPrJobMgmt process for the deletion of a DPR.
Create Ground Event		*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlWb	The EcPlWb process sends the ground event id, resource id, and start time to the EcDpPrJobMgmt process.
Request manage-ment services		*Process:* EcPlSubMgr *Class:* PlSubscriptionManager	*Processes:* EcMsAgSubAgent	The EcMsAgSubAgent provides a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include:
	One per command to start or stop network applications			**Lifecycle commands** - The M&O staff use the OVW Map (COTS SW) to send commands to managed hosts in the network to start and to stop applications. On startup, the OVW Map passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Request Communi-cations Support	One per process request.	To simplify the interface table for the Request Communications Support event, refer to the CSS section of Appendix B, Software Libraries, for the libraries and classes used as the interface to fulfill the services requested by ECS processes.	*Process:* EcPlPREditor	The Process Framework provides a library of services available to each ECS process. The services required to perform the specific process assignments are requested by the process from the Process Framework. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User authentication services, and Mode information.

4.6.1.6 Production Planning Data Stores

Table 4.6.1.6-1 provides descriptions of the production planning data stores shown in the PLANG CSCI architecture diagram.

Table 4.6.1.6-1. PLANG CSCI Data Stores

Data Store	Type	Functionality
PDPS Database	Database	The PDPS database is replicated within each site for fault handling purposes. This PDPS database holds all the persistent data (and facilitates the sharing of this data) including, but not limited to: • resource information entered with the Resource Planning utilities • PGE and data type information entered at SSIT • Production Request, Data Processing Request and Data Granule information entered using the Production Request Editor • plan information entered using the Production Planning Workbench • task recovery information The PDPS database also provides security, fault tolerance, and verifies requests for concurrent access to data.

4.6.2 Planning Subsystem Hardware Components

4.6.2.1 Planning Hardware CI (PLNHW) Description

The PLNHW hardware (PLNHW) consists of an SNMP server with the Sybase database management system (DBMS) and the workstations to support the Operations staff by providing the Planning Workbench.

The PDPS DBMS Server runs on either a four processor SUN Server or a dual-processor Sun workstations (see 920-TDx-001 series of baseline documents) with 64-bit Ultra-SPARC processors. Each PDPS DBMS Server is equipped with 512 MB of memory (see 920-TDx-001 series of baseline documents) required by the workstation processors and the Sybase DBMS.

The internal disks provide swap space and file system space for the operating system and the file space for applications software (see 920-TDx-001 series of baseline documents).

Either a Storage Array or an appropriately sized storage unit configured for the database application provides storage for the PDPS database. These storage units are attached to the host via a fast-wide small computer system (SCSI). Additionally, this storage unit backs up the Queuing Server database (see Section 4.7.3.1: Data Processing Hardware).

A Fiber Distributed Data Interface (FDDI) sub-network is implemented at each site to support the PDPS. Each processing unit of SPRHW (including the Queuing Server) is dual-attached to the PDPS FDDI sub-network (see 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of baseline documents).

The PDPS database on the PDPS DBMS Server is replicated by the MSS Backup Server (see Section 4.9.16: MHWCI Description) to a physical location on a storage unit of the Queuing Server. When a disk or PDPS database failure occurs on a primary database, the backup database is used on the Queuing Server.

The dual-ring FDDI implementation provides a fault tolerance capability. Media failures within the FDDI fabric do not result in a loss of service and do no require a re-configuration. With the inherent fault tolerance of FDDI, multiple physical communications paths to each host are not necessary.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.6.2.2 Planning Workstation Description

The Planning workstation contains and runs the Planning Workbench software. The Planning Workbench is the Production Planning function and the Resource Planning function. One or more workstations are used to run Production Planning and/or Resource Planning at each site based on the site size.

The Planning workstation is a SUN workstation class machine with either a single SPARC or Ultra-SPARC based processor (see 920-TDx-001 series of baseline documents).

The Planning workstation has 384 MB of memory (see 920-TDx-001 series of baseline documents) and each has a fast-wide SCSI controller to attach to a storage subsystem.

The internal disks provide swap space for the operating system and file system space for the operating system and applications (see 920-TDx-001 series of baseline documents). Additional storage for the Planning workstations can be attached to the SCSI controller.

A FDDI sub-network is implemented at each site to support the PDPS. The Planning workstations use a single-attached FDDI interface to connect with the remaining members of the PDPS suite (see 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of baseline documents). The Planning workstations also communicate with other ECS hardware items and the external world via the DAAC FDDI switch.

Sites generally have a minimum of two Planning workstations. If a planning workstation fails, another planning workstation assumes the Planning Workbench functionality from the failed workstation. In cases with one Planning workstation, the Planning workstation tasks are assumed by an available equivalent workstation. Faulty hardware is either repaired or replaced by a certified technician.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "x." The document provides the recovery procedure for each host.

4.7 Data Processing Subsystem Overview

The Data Processing Subsystem (DPS) provides the Data Processing capabilities at each ECS site. The DPS capabilities include:

- A queued processing environment to support data product generation. The DPS executes DPRs on available processing resources as an associated processing job containing all the information needed to accomplish the processing. DPRs are submitted by the PLS and triggered by the arrival of data or triggered internally by the PLS (i.e., reprocessing). PGEs resulting from the integration and test of delivered science algorithms [ref.: ECS White Paper 193-00118] and encapsulated into the ECS with the Science Data Processing (SDP) Toolkit are used by DPRs to process data. User-specified methods are also used for processing specific data types
- The Operational interfaces required to monitor the execution of the science software (PGEs).
- Support for science algorithm execution via the SDP Toolkit. The SDP Toolkit is a set of tools to provide a common interface for encapsulating each science algorithm into the SDPS environment. (See the SDP Toolkit Users Guide for the ECS Project (333-CD-003-002)) and PGS Toolkit Requirements Specification for the ECS Project (193-801-SD4-001, a.k.a. GSFC 423-06-02) for guidance on the roles and responsibilities of the SDP Toolkit to support the execution of science software.
- Support for the preliminary format processing of data sets (L0 data products) required by the science algorithms
- Providing an Algorithm Integration and Test (AI&T) environment to integrate new science algorithms, new versions of existing science algorithms, and user methods into the SDPS environment. The system acquires the algorithm or method via an ingest process reflecting local site policies for acceptance of software for integration into the environment. (See Section 4.7.2 “Algorithm Integration and Test Tools (AITTL) CSCI Description).
- The DAAC Quality Assurance (QA) procedures and conditions to verify each data product by the scientific personnel at each DAAC. All data products, both those generated by and input to a submitted job, are available for examination by DAAC scientific personnel to verify data content to be in accordance with quality standards set by the DAAC.

Data Processing Subsystem Context

Figure 4.7-1 is the Data Processing Subsystem Context Diagram. The diagram shows the events sent to the DPS and the events the DPS sends to other ECS subsystems and the Operations staff.

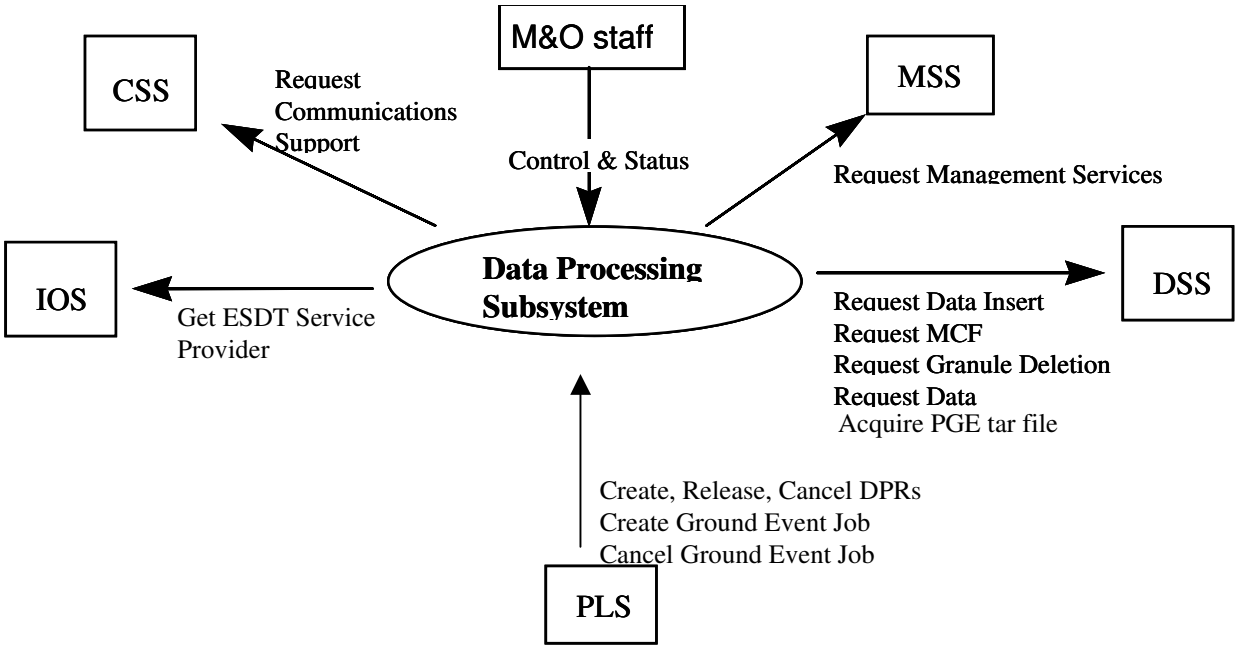

Figure 4.7-1. Data Processing Subsystem Context Diagram

Table 4.7-1 provides descriptions of the interface events shown in the Data Processing Subsystem context diagram.

Table 4.7-1. Data Processing Subsystem Interface Events (1 of 2)

Event	Interface Event Description
Control & Status	The M&O staff provide Data processing control and supports DPR status activities.
Request Communications Support	The CSS provides a library of services available to each ECS subsystem. The services required to perform the specific subsystem assignments are requested by the subsystem from the CSS. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Get ESDT Service Provider	The IOS receives requests to search for signature service advertisements from the DPS. The DPS obtains the proper signatures for communicating with the DSS.

Table 4.7-1. Data Processing Subsystem Interface Events (2 of 2)

Event	Interface Event Description
Create DPR	The DPS uses the dprId to insert a job box for a DPR into AutoSys if all the required input data is available. If the input data is not available, information used to construct the job in AutoSys is queued by the Job Management CSC to not adversely impact the performance of AutoSys.
Release DPR	The DPS uses the dprId to release jobs currently waiting for external data in the Job Management queue into AutoSys.
Cancel DPR	The DPS uses the dprId to delete jobs in AutoSys or from the queue.
Create Ground Event Job	The DPS uses the ground event Id to create a ground event job in AutoSys.
Cancel Ground Event Job	The DPS uses the ground event Id to delete a ground event job.
Request Data Insert	The DPS sends requests to the DSS to insert a particular file or files into the archive, and the associated metadata cataloged into the SDSRV inventory, as a granule of a particular ESDT short name and version. These files can be processing output, static files received with PGEs, PGE tar files, APs, SSAPs or DAPs, failed PGE tar files, or production history files.
Request MCF	The INS and DPS request the MCF template, from the DSS, prior to a data insert request.
Acquire PGE tar file	The DPS acquires a tar file for any PGE not currently local to the science processor from the DSS. The tar file is removed from the tape archive and used during PGE execution.
Request Data	The DPS sends requests to the DSS for a particular data granule to be FTP pushed onto the DPS science processor and used as input for data processing or for SSIT work.
Request Granule Deletion	The DPS sends delete requests, to the DSS, for particular granules (interim data) in the archive and the associated metadata to be deleted from the SDSRV inventory.
Request management services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.

The DPS has an internal interface to the COTS software product AutoSys. The DPS creates, starts, and deletes job boxes in AutoSys via this interface.

The PLS determines the processing activities required to generate data products specified by the Operations staff in a Production Request. Each processing activity is called a DPR. The PLS creates, releases or deletes DPRs in AutoSys via the DPS.

The DSS accesses the data archives via authorized user requests. The DPS requests the required input data for a PGE and Metadata Configuration Files (MCFs) from the DSS. The DPS also inserts PGE generated products, provides product production histories, and provides failed PGE information for debugging purposes. The DPS uses the DSS as a permanent repository for PGE

tar files, Algorithm packages (APs), Science Software Archive Packages (SSAPs) and Delivered Algorithm Packages (DAPs).

Data Processing Subsystem Structure

The DPS is comprised of three CSCIs:

- The Processing (PRONG) CSCI manages and monitors the Science Data Processing (SDP) environment to execute Science Software and algorithms (called PGEs) and generates data products.
- The Algorithm Integration and Test Tools (AITTL) CSCI is a set of tools for test and integration of new science software, new versions of existing science software, and user methods in the SDP operational environment. AITTL combines custom developed code with COTS software starting from a central application called the SSIT Manager.
- The SDP Toolkit (SDPTK) CSCI provides a set of software libraries to integrate Science Software into the ECS environment. By promoting the POSIX standard, these libraries allow the SDP environment to support the generation of data products in a heterogeneous computer hardware environment. (See SDP Toolkit Design Specification (455-TP-001-001) for the SDPTK architecture).

Use of COTS in the Data Processing Subsystem

- **Platinum Technology's AutoSys** is a job scheduling software application to automate operations in a distributed UNIX environment. AutoSys performs automated job control functions for scheduling, monitoring, and reporting on the jobs residing on any Unix machine attached to an ECS network on the Science Data Processing hardware. AutoSys provides job-scheduling support with an Operator Console for monitoring and human intervention in the job stream. The Operator console allows the Operations staff to restart failed jobs and to view the status of events related to the job's execution. The Operator console includes an alarm manager, set in the job definition, to assist the Operations staff when responding to fault situations.
- **Platinum Technology's AutoXpert** is a GUI providing different methods of viewing a job schedule progress. Noting color changes on the JOBSCAPE GUI can monitor the progression of DPR execution. Failed jobs can be detected and restarted if the job has failed due to the unavailability of an external resource. The HostScape GUI can be used to view the status of the science processors.
- **Sybase Server**

  The Sybase SQL server provides the capabilities to insert, update and delete PDPS database content. The Sybase SQL Server must be operational during the DPS operations.

The DPS provides the hardware resources for science software execution, queuing, dispatching, and managing in a distributed environment of computing platforms. The DPS hardware comprises three hardware CIs.

- Science Processing - The Science Processing HWCI (SPRHW) contains processing resources (central processing units, memory, disk storage, and input/output subsystems) necessary to perform first-time processing, reprocessing, and Algorithm Integration and Test (AI&T). Also, SPRHW provides the hardware resources (a Queuing Server) to support management of the science processes.
- Algorithm Quality Assurance - The Algorithm Quality Assurance HWCI (AQAHW) supports the DAAC Operations staff in performing the planned science and non-science product data quality validation procedures.
- Algorithm Integration and Test - The AI&T HWCI (AITHW) resources provide the operating system and support for the integration and test of science software at each DAAC. AITHW is the workstations and hardware tools required for software integration and test. AITHW does not, in this case, provide the computer capacity required for science software test (SPRHW provides the test capacity).

4.7.1 Processing Software Description

4.7.1.1 Processing Functional Overview

The Processing (PRONG) CSCI initiates, monitors, and manages the execution of science software algorithms (referred to as PGEs). The PRONG CSCI is informed of the required execution of a PGE through a DPR received from the PLS. When all necessary input data becomes available, PRONG initiates the execution of the PGE. (N.B.: Some or all input data can reside in a Data Server not at the DAAC site.)

The PRONG CSCI has the following capabilities:

- Manages execution of science software algorithms
- Manages SDP computer hardware resources
- Manages the data flow required to execute a science software algorithm
- Manages the data flow generated by the execution of a science software algorithm
- Monitors processing status, and allows manual intervention, when necessary, in the SDP operations environment, including processing queue control
- Supports validation of product data quality

4.7.1.2 Processing Context

Figure 4.7.1.2-1 is the PRONG CSCI context diagram. The diagram shows the events sent to the PRONG CSCI and the events the PRONG CSCI sends to other CSCIs and the Operations staff.

Table 4.7.1.2-1 provides descriptions of the interface events shown in the PRONG CSCI context diagram.

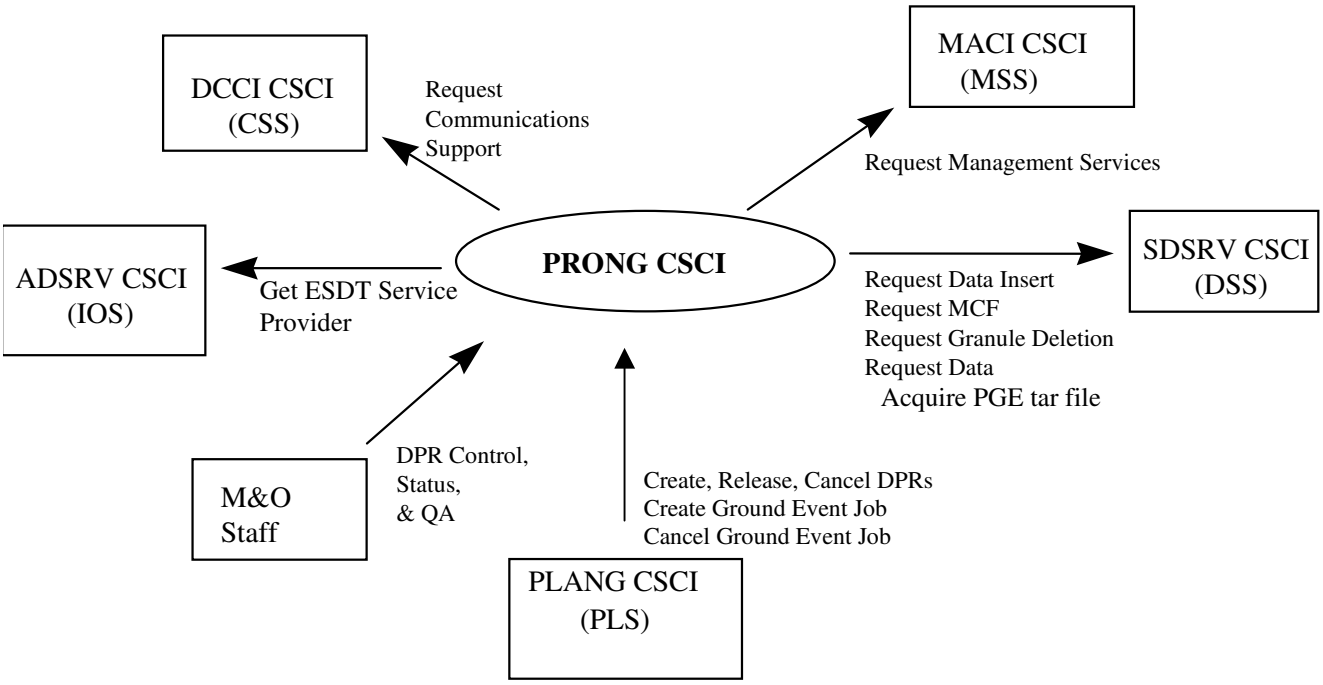

Figure 4.7.1.2-1. PRONG CSCI Context Diagram

Table 4.7.1.2-1. PRONG CSCI Interface Events (1 of 2)

Event	Interface Event Description
DPR Control, Status & Q/A	The M&O staff controls DPR activity with the capability to cancel, suspend, resume, and modify a DPR. The M&O staff supports status collecting, PRONG hardware resource monitoring and Quality Assurance validating processes.
Request Communications Support	The DCCI CSCI provides a library of services available to each ECS CSCI. The services required to perform the specific CSCI assignments are requested by the CSCI from the DCCI CSCI. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Get ESDT Service Provider	The ADSRV CSCI receives search requests for signature service advertisements from the PRONG CSCI. The PRONG CSCI obtains the proper signatures for communicating with the SDSRV CSCI.

Table 4.7.1.2-1. PRONG CSCI Interface Events (2 of 2)

Event	Interface Event Description
Create DPR	The DPS uses the dprId to insert a job box for a DPR into AutoSys if all the input data that is required for the DPR is available. If the input data is not available, information used to construct the job in AutoSys is queued by the Job Management CSC to not adversely impact the performance of AutoSys.
Release DPR	The DPS uses the dprId to release jobs currently waiting for external data in the Job Management queue into AutoSys.
Cancel DPR	The DPS uses the dprId to delete jobs in AutoSys or from the queue.
Create Ground Event Job	The DPS uses the ground event Id to create a ground event job in AutoSys.
Cancel Ground Event Job	The DPS uses the ground event Id to delete a ground event job.
Request Data	The PRONG CSCI sends requests to the SDSRV CSCI for a particular data granule to be FTP pushed onto the DPS science processor and used as input for data processing or for SSIT work.
Request Granule Deletion	The PRONG CSCI sends delete requests to the SDSRV CSCI for particular granules (interim data) in the archive and associated metadata to be deleted from the SDSRV inventory.
Request MCF	The PRONG CSCI requests a MCF template for each output data type for specific PGEs from the SDSRV CSCI and the MCF template is populated with metadata from the output granule.
Acquire PGE tar file	The PRONG CSCI acquires a tar file for any PGE not currently local to the science processor from the SDSRV CSCI. The tar file is removed from the tape archive and used during PGE execution.
Request Data Insert	The DPS sends requests to the DSS to insert a particular file or files into the archive, and the associated metadata cataloged into the SDSRV inventory, as a granule of a particular ESDT short name and version. These files can be processing output, static files received with PGEs, PGE tar files, APs, SSAPs or DAPs, failed PGE tar files, or production history files.
Request management services	The MACI CSCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.

4.7.1.3 Processing Architecture

Figure 4.7.1.3-1 is the PRONG CSCI architecture diagram. The diagram shows the events sent to the PRONG CSCI processes and the events the PRONG CSCI processes send to other processes. The PRONG CSCI consists of COTS software and ECS developed processes.

The PRONG CSCI has interfaces with:

- Planning Subsystem – The PLS creates a production plan executed by the PRONG CSCI through the use of DPRs. Each DPR represents one processing job performed by a DPS computer resource. The PRONG CSCI provides DPR status information to the PLS to assist in production management activities.
- Data Server Subsystem – The PRONG CSCI supports ECS data generation by requesting and receiving data (Data Staging) from a Data Server maintaining the raw data and generated products. Also, the PRONG CSCI transfers data (Data de-staging) to a Data Server to archive generated data products.
- SDP Toolkit – The PRONG CSCI provides the location of input data and the location for the generated output data products. While a PGE is executing, the PRONG CSCI monitors the execution and provides current status to the Operations staff. Status includes current processing event history (e.g., data staging, and execution). Process monitoring includes checking resource usage by the PGE. At PGE execution completion, the PRONG CSCI informs the PLS and as needed initiates the transfer of the generated data product to the respective Data Server.
- System Management Subsystem – The PRONG CSCI relies on the MSS services for resource management and thus provides system management information including fault, accounting, configuration, security, performance, and accountability to the MSS.
- Operations – Supports PGE execution management and monitoring and the generation of SDPS Data Products via a Human Machine Interface (HMI). The HMI supports status information collection for a DPR, controlling DPR executions, and monitoring the status of the DPS hardware resources. The HMI also supports manual quality assurance activities performed at the DAAC.

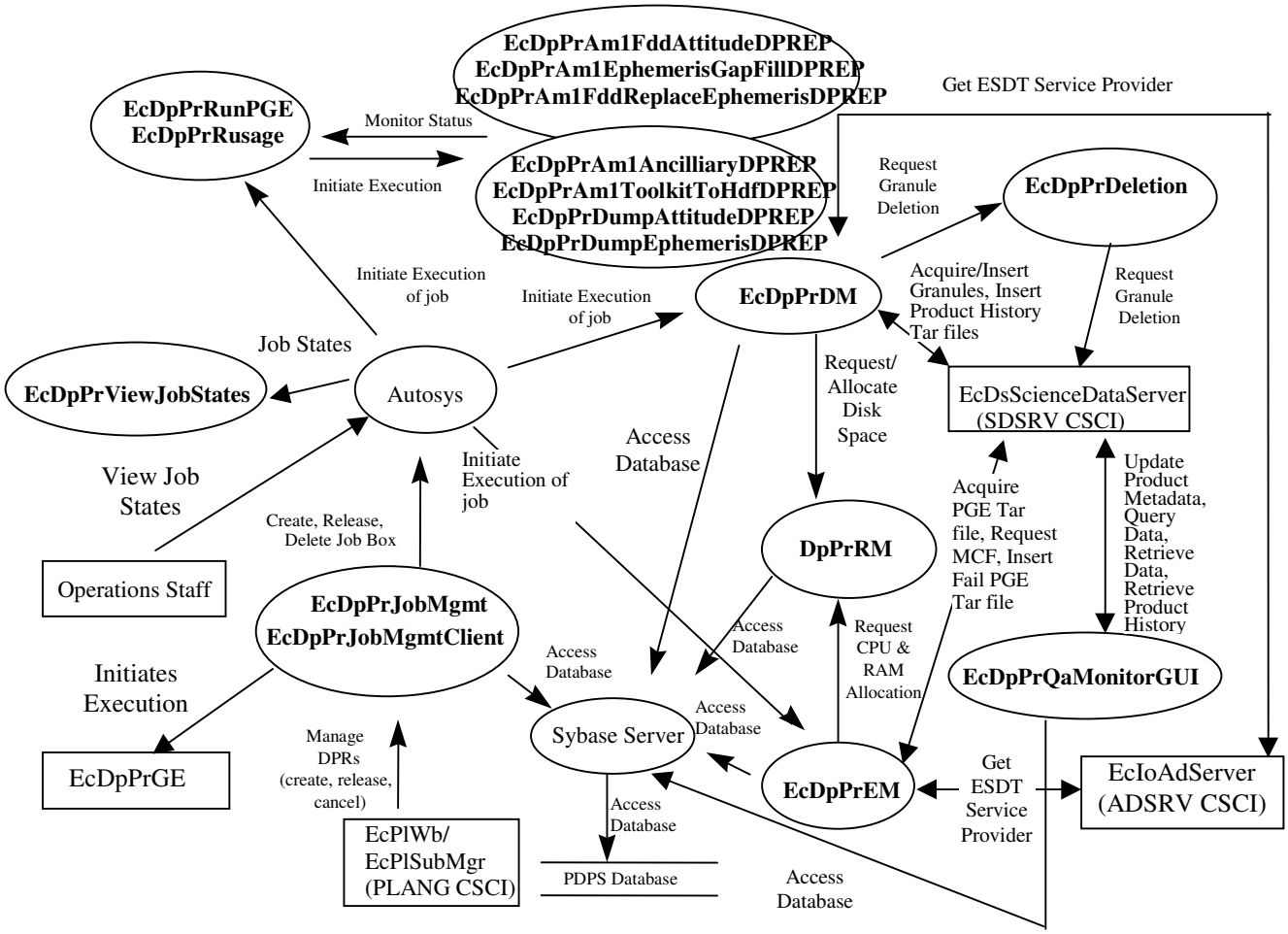

Figure 4.7.1.3-1. PRONG CSCI Architecture Diagram

4.7.1.4 Processing Process Descriptions

Table 4.7.1.4-1 provides descriptions of the processes shown in the PRONG CSCI architecture diagram.

Table 4.7.1.4-1. PRONG CSCI Processes (1 of 3)

Process	Type	COTS/ Developed	Functionality
EcDpPrDM	Other	Developed	The Data Management process manages the flow of science data to and from science processing resources including communication mechanisms to interface with the EcDsScienceDataServer. Data Management manages data retention on science processing resources to support PGE executions.
EcDpPrEM	Other	Developed	The Execution Management process initiates the execution of PGEs (via the COTS product AutoSys). EcDpPrEM supports the preparation activities prior to the execution of PGEs and subsequent activities to the execution of PGEs.
EcDpPrRunPGE EcDpPrRusage	Other	Developed	The PGE Execution Manager process controls and monitors PGE executions including Process Control File creation and output product storage growth. EcDpPrRusage measures the actual resources used by the PGE and reports to AutoSys unexpected resource usage.
EcDpPrDeletion	Server	Developed	This DCE Server notifies the EcDsScienceDataServer to remove interim granules via the data management process (EcDpPrDM) when they are no longer needed. The interim products are removed after the last PGE in the chain has used them or a pre set time has expired after the last use of the interim product.
AutoSys	GUI	COTS	AutoSys is a job scheduling software application used to automate operations in a distributed UNIX environment. AutoSys executes jobs to automate support for PGE execution. AutoSys creates job boxes consisting of a series of related jobs, and manages job dependencies. AutoSys provides graphical depictions of completed jobs and jobs being processed. It includes the Operator Console GUI to allow human intervention into monitoring and altering the AutoSys job stream. The daily job schedule is submitted to the Job Management server at the start of the processing day. Jobs, which have data available, are released into AutoSys. To support job executions, AutoSys requires additional help for: • Allocation of sufficient resources (e.g., disk space) to support executions. The EcDpPrDM provides the capabilities to manage disk space and monitor resources • Managing remote host data acquisition, data retention on the DPS processing host, and data distribution from the DPS processing host • Initialization and PGE executions.

Table 4.7.1.4-1. PRONG CSCI Processes (2 of 3)

Process	Type	COTS/ Developed	Functionality
EcDpPrJobMgmt EcDpPrJobMgmtClient EcDpPrViewJobStates	Server	Developed	The Job Management process uses the AutoSys COTS product to create and initiate execution of PRONG administrative jobs for managing SPRHW assets and for PGE execution. This work is performed by seven Unix processes bundled together into an AutoSys job box. Job Management is responsible for efficient AutoSys management so the maximum number of jobs possible can be continuously run using the product. This involves controlling the flow of jobs through AutoSys by only allowing jobs ready to run into the product and by removing jobs as they complete. Job Management also creates and starts execution of Ground Event jobs in AutoSys. The Job Management Client process is used by programs that need access to the Job Management Server services to modify jobs in AutoSys to change the priority of the jobs. The various events this process provides are: CreateDPR: A data processing request identified is then translated into seven standard process steps (one, the PGE execution, the remaining performing support activities). If the science data is available, the job box containing the seven individual jobs is released into AutoSys. ReleaseDPR: A previously created data processing request waiting for the availability of science data is released into AutoSys to begin execution. CancelDPR: This provides the capability to cancel/terminate a data processing request. CreateGEvntJob: Create a Ground Event Job in AutoSys. CancelGEvntJob: Cancel a Ground Event Job. The View Job States process allows the Operations Staff to view jobs in the queue to determine the jobs that have completed, the jobs executing, and the jobs awaiting execution.

Table 4.7.1.4-1. PRONG CSCI Processes (3 of 3)

Process	Type	COTS/ Developed	Functionality
EcDpPrQaMonitorGUI	GUI	Developed	This process provides the capability to transfer science data from the archives, browse data images, and examine and update science metadata. It is an automated tool for performing data analysis in support of DAAC Quality Assurance activities.
EcDpPrAm1AncilliaryDPREP, EcDpPrAm1AncillaryDPREP, EcDpPrAm1EphemerisGapFillDPREP, EcDpPrAm1FddAttitudeDPREP, EcDpPrAm1FddReplaceEphemerisDPREP, EcDpPrAm1ToolkitToHdfDPREP, EcDpPrDumpAttitudeDPREP, EcDpPrDumpEphemerisDPREP	Other	Developed	Data Preprocessing manages L0 attitude and ephemeris ancillary data preprocessing for inputs to PGEs. Data preprocessing is the preliminary processing or application of an operation on a data set that does not alter or modify scientific content of the data set. Preprocessing includes data set format changes by reordering the lower level byte structure, data set reorganization (ordering data items within and between physical files), and preparing additional metadata based on lower level metadata.
DpPrRm	Other	Developed	Resource Management is a library to support efficient use of computer resources required for science production processing based upon mappings of logical to physical resources. It is used for the allocation of resources (i.e., disk space, memory, and Central Processing Unit (CPU)) to execute PGEs. Prior to PGE execution, all required resources must be satisfied. After PGE execution completes, the CPU and memory allocated are available for other PGE executions. Files on the production disk are removed as space is needed.
Sybase Server	Server	COTS	The Sybase Server acts as a SQL server for the PDPS database.
EcDpPrGE	Other	Developed	The EcDpPrJobMgmt server initiates the EcDpPrGE when the server gets a ground event request. The ground event process starts at a specified time and runs a specified duration. During the time the ground event process runs, it sets a computer resource (cpu, ram, etc.) off-line and the computer resource is not available for PGEs.

4.7.1.5 Processing Process Interface Descriptions

Table 4.7.1.5-1 provides descriptions of the interface events shown in the PRONG CSCI architecture diagram.

Table 4.7.1.5-1. PRONG CSCI Process Interface Events (1 of 5)

Event	Event Frequency	Interface	Initiated By	Event Process Description
Initiate execution of job (control)	One per PGE job execution	*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcDpPrJobMgmt	AutoSys initiates the execution of the EcDpPrDM, the EcDpPrEM and the EcDpPrRunPGE processes to control the preparation (data staging), execution, archiving higher level products, which are produced, and cleanup of each PGE run.
Initiate execution (control)	One per PGE job execution	AutoSys Job Dependency	Successful completion of Preprocessing job	The EcDpPrRunPGE provides a buffer between AutoSys and the PGE. This serves as a wrapper to the PGE process, initiates the PGE execution and captures the PGE's exit status. The EcDpPrJobMgmt process initiates the EcDpPrGE process when ground events occur.
Monitor status (status)	One per PGE job execution	AutoSys Job Dependency	Successful completion of Preprocessing job	The EcDpPrRunPGE process, apart from initiating the PGE process, also monitors the PGE's computer resources. If the PGE's computer resources exceed its expected usage an alarm is sent to the AutoSys. This wrapper also captures the PGE's resource usage and its exit code.
Request/Allocate Disk Space	One per disk request	*Library:* DpPrRM *Class:* DpPrResourceManager	*Process:* EcDpPrDM	The EcDpPrDM requests disk space via use of the DpPrRm library software for each input granule that needs to be staged to the local processing disk and output granule needed by a PGE.
Request CPU and RAM Allocation	One per PGE job execution	*Library:* DpPrRM *Class:* DpPrResourceManager	*Process:* EcDpPrEM	EcDpPrEM requests CPU and RAM allocations from ResourceMgmt for each PGE based on values entered at SSIT.
Acquire/Insert Granules (Request data, Request Data Insert)	One per acquire granule	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrDM	The EcDpPrDM acquires the input granules needed by a PGE not currently local on a science processor from the EcDsScienceDataServer. After the PGE has successfully completed executing, the EcDpPrDM sends insert requests for the EcDsScienceDataServer to store the output granules into the SDSRV inventory/archives.

Table 4.7.1.5-1. PRONG CSCI Process Interface Events (2 of 5)

Event	Event Frequency	Interface	Initiated By	Event Process Description
Insert Production History Tar files	One per successful PGE execution	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrDM	After the PGE has successfully completed executing and archiving the resulting outputs, EcDpPrDM requests the PGE Production History Tar file be inserted into the Science Data Server for permanent archive.
Insert Fail PGE Tar file	One per unsuccessful PGE execution	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrEM	After an unsuccessful execution of PGE, EcDpPrEM requests the Tar file containing the PGE log files, core dump (if any), PCF and other files, be inserted into the Science Data Server for permanent archive.
Acquire PGE Tar file	One per tar file acquire	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrEM	The EcDpPrEM acquires a tar file for any PGE not currently local to the science processor from the EcDsScienceDataServer. The tar file is removed from the tape archive and used during PGE execution.
Request MCF	One per MCF request	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrEM	The EcDpPrEM requests a MCF template for each output data type for specific PGEs from the EcDsScienceDataServer and the MCF template is populated with metadata from the output granule.
Create Job Box	One per execution of job box creation	*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcDpPrJobMgmt	The EcPlWb (of the PLANG CSCI) uses this interface to inform the EcDpPrJobMgmt process that a DPR is ready for execution, provided all input data is available in the archive. The EcDpPrJobMgmt, which acts as an interface between EcPlWb and AutoSys, places the DPR in AutoSys when all input data is available using Job Interface Language (JIL). If all the input data is not available, the EcDpPrJobMgmt stores the DPR in a priority-based queue.
Release Job Box	One per execution of a DPR in the jobmgmt queue	*Library:* DpPrJM *Class:* DpPrScheduler	Process: EcPlSubMgr	When all input data is available to execute a DPR, EcPlSubMgr (of the PLANG CSCI) notifies the Jobmgmt server that a DPR is ready for execution. The EcDpPrJobMgmt uses the Job Interface Language (JIL) to place the DPR in AutoSys and begins execution.

Table 4.7.1.5-1. PRONG CSCI Process Interface Events (3 of 5)

Event	Event Frequency	Interface	Initiated By	Event Process Description
Delete Job Box	One per job box deletion	*Library:* DpPrJM *Class:* DpPrScheduler	*Process:* EcPlPREditor	The EcDpPrJobMgmt deletes a job box in AutoSys and performs any cleanup when an operator requests a DPR to be canceled.
Access Database	One per update/retrieve data request	Sybase Database	Most PRONG processes	The EcDpPrJobMgmt, EcDpPrDM, EcDpPrEM, and DpPrRM processes request update and retrieval of data in the database that defines a PGE and allows the PGE to be scheduled and executed by AutoSys. The processes also allow updates for granule information (location, size, etc.), processing status, and check-pointing information stored in the database.
Query Data	One per query	*DPS Library:* DpPrQaMonitor *DPS Class:* DpPrQaDataGranule *DSS Library:* DsCl *DSS Class:* DsClESDTReferenceCollector	*Process:* EcDpPrQaMonitorGUI	The QA Monitor submits requests of this type to the Science data Server. It searches the archive for granules that match the user-supplied selection criteria: data type and begin/end date. Results are displayed to the user.
Retrieve Data	One per request	*DPS Library:* DpPrQaMonitor *DPS Class:* DpPrQaMonitor *DSS Library:* DsCl *DSS Class:* DsClAcquireCommand	*Process:* EcDpPrQaMonitorGUI	The QA Monitor submits requests of this type to the Science data Server. It transfers a granule from the Science Data archive to the user's host machine.
Retrieve Product History	One per request	*DPS Library:* DpPrQaMonitor *DPS Class:* DpPrQaMonitor *DSS Library:* DsCl *DSS Class:* DsClAcquireCommand	*Process:* EcDpPrQaMonitorGUI	The QA Monitor submits requests of this type to the Science data Server. It transfers the Production History tar file from the Science Data archive to the user's host machine.

Table 4.7.1.5-1. PRONG CSCI Process Interface Events (4 of 5)

Event	Event Frequency	Interface	Initiated By	Event Process Description
Manage DPRs	One per control request	*Library:* DpPrJM *Class:* DpPrScheduler	*Processes:* EcPlWb, EcPlSubMgr, EcPlPREditor	The EcPlWb process sends requests to the EcDpPrJobMgmt to create and cancel (delete) DPR jobs in AutoSys. The EcPlSubMgr process sends requests to the EcDpPrJobMgmt to release DPR jobs in AutoSys.
Update Product Metadata	One per metadata product update	*DPS Library:* DpPrQaMonitor *DPS Class:* DpPrQAGranuleQaFlags *DSS Library:* DsCl *DSS Classes:* DsClCommand, DsClRequest, DsClESDTReferenceCollector	*Process:* EcDpPrQaMonitorGUI	The EcDpPrQaMonitorGUI provides the operator with capabilities to update product metadata.
Get ESDT Service provider	One per Advertising search	*Library:* PlCore1 *Class:* PlDataType	*Processes:* EcDpPrEM, EcDpPrDM	The EcDpPrDM and EcDpPrEM send requests to the EcIoAdServer, using the Universal Reference obtained from the EcDsScienceDataServer, for a particular ESDT.
Request Granule Deletion	One per granule delete request	*Library:* DpPrDssIF *Class:* DpPrDSSInterface	*Process:* EcDpPrDeletion *Class:* DpDeletion	The EcDpPrDM sends requests to the EcDpPrDeletion process to delete interim granules a PGE had used in processing or after a defined storage period has elapsed.
Job States	Per AutoSys Status update.	*Class:* DpPrListJobs	*Process:* EcDpPrViewJobStates	The AutoSys provides the job state (completed, executing, or queued to be executed) to the EcDpPrViewJobStates process.
View Job States	Per Operations Staff request.	Operations Staff terminal	Operations Staff	The Operations staff can view the job state via the EcDpPrViewJobStates process, as an aid in scheduling jobs.

Table 4.7.1.5-1. PRONG CSCI Process Interface Events (5 of 5)

Event	Event Frequency	Interface	Initiated By	Event Process Description
Request Management Services	Per applications need.	Library: EcAgInstrm Class: EcAgManager	Processes: All	The EcMsAgSubAgent process provides a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** – The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.
Request Communications Support	One service per request.	To simplify the interface table for the Request for Communications Support event, refer to the CSS section of Appendix B, Software Libraries, for the libraries and classes used as the interfaces to fulfill the services requested by ECS processes.	*Process:* *All*	The CSS Process Framework provides a library of services available to each ECS process. The services required to perform the specific process assignments are requested by the process from the Process Framework. These services include: DCE support, file transfer services, Network & Distributed File Services, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.

4.7.1.6 Processing Data Stores

Table 4.7.1.6-1 provides descriptions of the data stores shown in the PRONG CSCI architecture diagram.

Table 4.7.1.6-1. PRONG CSCI Data Stores

Data Store	Type	Functionality
PDPS database	Database	The PDPS database is replicated within the same site and holds the persistent data for PDPS. The persistent data includes, but is not limited to, resource information entered with the Resource Planning utilities, PGE and data type information entered at SSIT, Production Requests, Data Processing Requests and Data Granule information entered using the Production Request Editor and plan information entered using the Production Planning Workbench.

4.7.2 Algorithm Integration and Test Tools Software Description

4.7.2.1 Functional Overview

The Algorithm Integration and Test Tools (AITTL) are used by the DAAC Integration and Test (I&T) team to:

- Retrieve science software and submit it for configuration control
- Compile and link the delivered source files
- Execute test cases
- Provide error diagnosis using interactive debuggers, and data viewers
- Collect resource metrics of CPU time, memory, and disk space to build the PGE Profile and thus enable the PLANG and PRONG CSCIs to execute the science software
- Update the system databases after the science software completes acceptance testing

The AITTL tools are in the following categories:

- Compilers, linkers, debuggers, and other development and operating system tools
- Tools for viewing science software documentation
- Tools for checking compliance of science software to Earth Science Data and Information System (ESDIS)-specified coding standards
- Code analysis tools (e.g., Sparc Works, CASEVision)
- Data viewing tools (e.g., EOSView).
- Tools for comparing HDF files
- Tools for comparing Binary files
- Tools for providing executable profiles (to get a PGE performance profile)
- Tools to register the science software with the Planning and Data Processing Subsystems
- Tools to add and update Science Software Archive Packages (SSAPs) in the Data Server
- Tools for writing reports and maintaining the I&T logs
- Tools for checking Process Control Files and for prohibited functions
- Tools to display product metadata

For information on science software integration and test procedures, see Science User's Guide and Operations Procedures Handbook for the ECS Project (205-CD-002-001) Part 4, and Science Software Integration and Test (JU9403V1). For information on the ESDIS science software coding standards and guidelines, see Data Production Software and Science Computing Facility (SCF) Standards and Guidelines (423-16-01).

4.7.2.2 Algorithm Integration and Test Tools Context

Figure 4.7.2.2-1 is the AITTL CSCI context diagram. The diagram shows the events sent to the AITTL CSCI and the events the AITTL CSCI sends to other ECS subsystems.

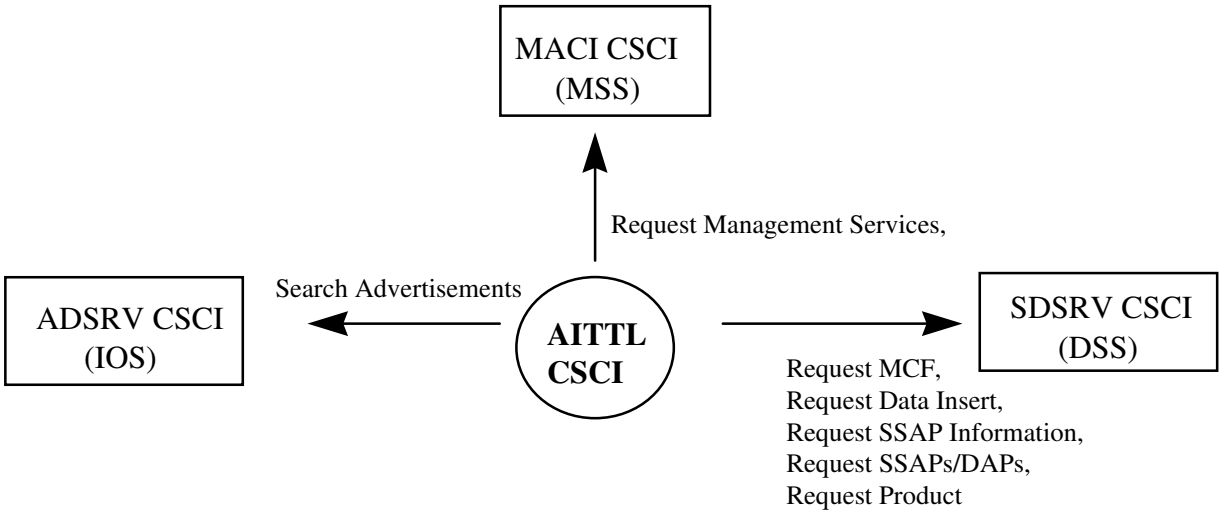

Figure 4.7.2.2-1. AITTL Context Diagram

Table 4.7.2.2-1 provides descriptions of the interface events shown in the AITTL Context Diagram.

Table 4.7.2.2-1. AITTL Interface Events (1 of 2)

Event	Interface Event Description
Request Data Insert	The AITTL CSCI puts various types of data into the SDSRV inventory, from SSAP information to Static files and PGE executables. In response, the AITTL CSCI gets the results of the insert and the UR to perform acquires requests.
Request SSAPs/DAPs	The AITTL CSCI requests DAPs and SSAPs based on URs from the SDSRV CSCI. The DAPs and SSAPs are placed on a local AITTL disk.
Search Advertisements	The ADSRV CSCI receives search requests for subscription event and signature service advertisements from the AITTL CSCI. The AITTL CSCI obtains the proper signatures for acquiring and inserting data granules from/to the SDSRV CSCI.

Table 4.7.2.2-1. AITTL Interface Events (2 of 2)

Event	Interface Event Description
Request SSAP Information	Requests for SSAP information, including names of existing SSAPs and the information associated with a specific SSAP. In response, the SDSRV CSCI sends lists of SSAPs and related information.
Request MCF	The AITTL CSCI requests the MCF template for use during SSI&T. The PRONG CSCI also requests the MCF template from the SDSRV CSCI prior to a data insert request.
Request Product	The AITTL CSCI sends requests, to the SDSRV CSCI, for particular data granules to be FTP pushed onto the DPS science processor as input for data processing or for SSIT work.
Request Management Services	The MACI CSCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the CSS Process Framework. The basic management library of services include: • **Lifecycle commands** - The MACI CSCI forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.

4.7.2.3 Algorithm Integration and Test Tools Architecture

Figure 4.7.2.3-1 is the AITTL CSCI architecture diagram. The diagram shows the events that launch the AITTL CSCI processes and the events the AITTL CSCI processes send to processes in other CSCIs.

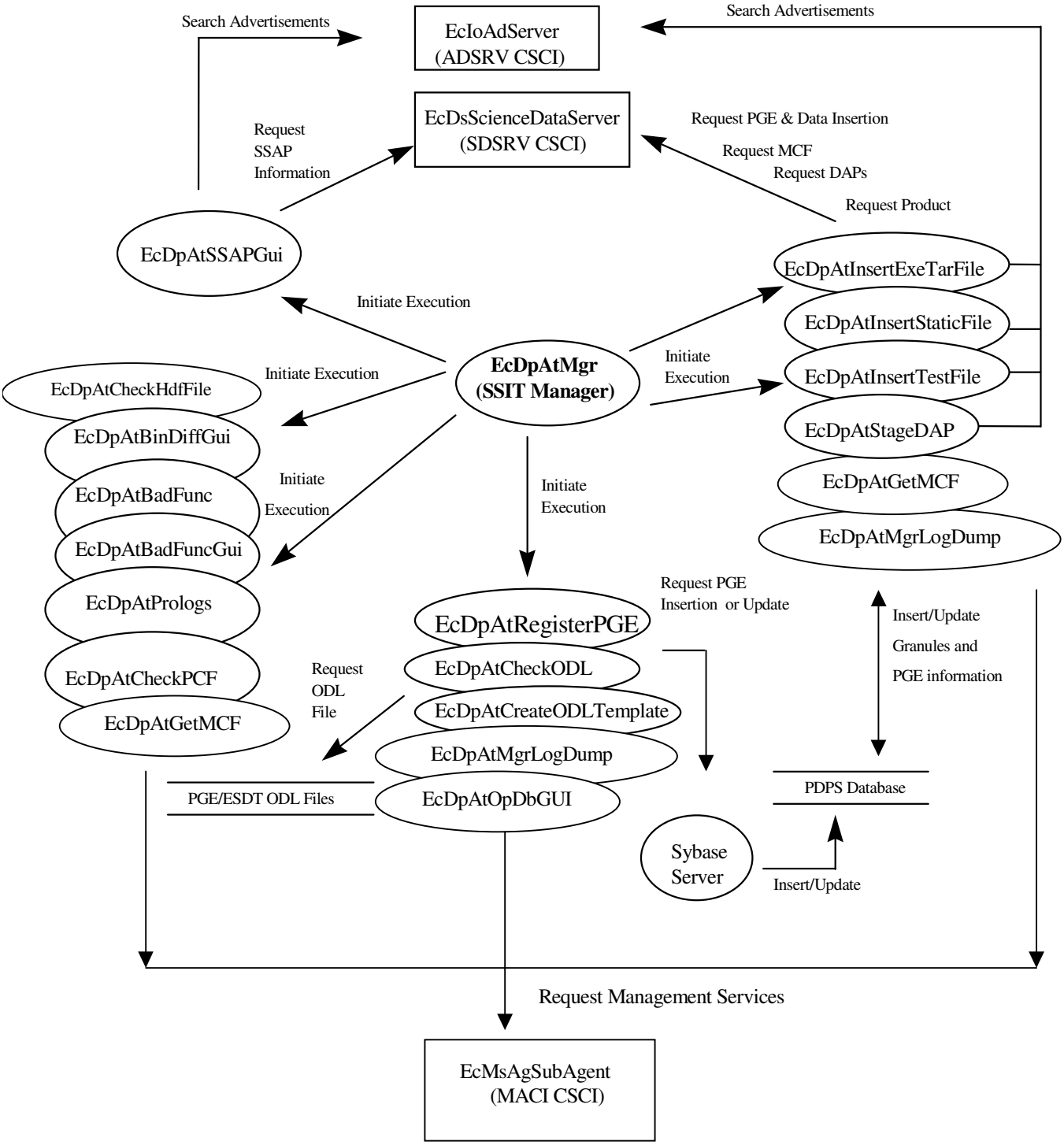

Figure 4.7.2.3-1. AITTL CSCI Architecture Diagram

4.7.2.4 Algorithm and Test Tools Process Description

Table 4.7.2.4-1 provides descriptions of the processes shown in the AITTL CSCI architecture diagram.

Table 4.7.2.4-1. AITTL Processes (1 of 2)

Process	Type	COTS / Developed	Functionality
EcDpAtSSAPGui	GUI	Developed	This GUI allows operators to create, update and delete SSAPs and to acquire information about a SSAP for modification or testing (such as test plans).
EcDpAtMgr	GUI	Developed	This application provides menus to launch other SSIT applications and provides a checklist to users for marking each SSIT function as completed.
ECDpAtRegisterPGE, EcDpAtCheckODL, EcDpAtCreateODLTemplate, EcDpAtOpDbGui	GUI and cmd line interface (I/F)	Developed	This application group allows a PGE to be defined in the PDPS database. ODL is read and checked by the tools and translated into the fields defining a PGE in the PDPS database. If the ODL files are valid, each row already existing in the PDPS database is updated and non-existent rows are inserted. Performance information is input by the SSIT personnel via a GUI.
EcDpAtBinDiffGui, hdiff, EcDpAtCheckhdfFile, EcDpAtMgrXdiff	GUI and cmd line I/F	Developed and COTS	This application group supports data file viewing and comparisons. The group includes EOSView, the COTS language IDL, and tools to compare binary and HDF files. SSIT personnel to compare actual PGE outputs versus expected outputs use these tools.
EcDpAtBadFunc, EcDpAtBadFuncGui, EcDpAtPrologs, EcDpAtCheckPCF	GUI	Developed and COTS	This applications group checks the source code for PGEs and PGE PCFs for errors or prohibited functions. The Sparc Works COTS product is included for editing and debugging functions, and a checker is provided for use during testing. Also provided is a checker to monitor the software for any prohibited calls
EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, EcDpAtInsertTestFile, EcDpAtStageDAP	cmd line I/F	Developed	This applications group provides mechanisms to insert and acquire data items from the EcDsScienceDataServer in the SDSRV CSCI. Static Files and PGE executables are inserted into the SDSRV archives by these tools, and the respective PDPS database tables are updated with the results. The Delivery Archive Package (DAP) and MCFs are acquired via these tools for command line testing the PGE.

Table 4.7.2.4-1. AITTL Processes (2 of 2)

Process	Type	COTS / Developed	Functionality
EcDpAtGetMCF	Cmd line I/F	Developed	Command line interface operations staff tool invoked from the SSIT Manager GUI. The EcDpAtGetMCF accepts the ESDT short name, version and a directory location from the operations staff and acquires the MCF for the ESDT from the EcDsScienceDataServer. EcDpAtGetMCF writes the MCF to a predetermined specified location.
EcDpAtMgrLogDump	Cmd line I/F	Developed	Command line interface to dump the SSIT checks list database to a file that can be sent to the printer.

4.7.2.5 Algorithm and Test Tools Process Interface Descriptions

Table 4.7.2.5-1 provides descriptions of the interface events shown in the AITTL CSCI architecture diagram.

Table 4.7.2.5-1. AITTL Process Interface Events (1 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request SSAP Information	One per SSAP information request	*Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDpAtSSAGui *Class:* DpAtSSAPManager	The EcDpAtSSAGui sends requests to the EcDsScienceDataServer for information about SSAPs, including names of existing SSAPs and the components associated with a specific SSAP.
Initiate Execution	One per tool initialization	UNIX system calls	*Process:* EcDpAtMgr	The EcDpAtMgr initiates the tools and GUI interface from a menu.
Request ODL File	One per ODL file request	*Library:* DpAtMetadata *Class:* DpAtScienceMd	*Processes:* EcDpAtCheckODL, EcDpAtRegisterPGE *Classes:* DpAtDatabase DpAtCheckOdl	In response to a request for an ODL File, the EcDpAtCheckODL and EcDpAtRegisterPGE processes receive data in “parameter = value” format about a PGE, its inputs and outputs, and scheduling information.
Request PGE Insertion or Update	One per insert/update request	*Library:* PlCore2 *Class:* PlResourceRequirement	*Process:* EcDpAtInsertExeTarFile *Class:* DpAtDsrv	The EcDpAtInsertExeTarFile sends PGE and data insert/update requests to the EcDsScienceDataServer for data that defines a PGE and allows it to be scheduled and executed.

Table 4.7.2.5-1. AITTL Process Interface Events (2 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Insert/ Update	One per insert/update request	*Library:* PlCore1 *Classes:* PlOutputYield PlPge PlDataType PlTileSchema PlSnapshotScheduled PlDataSource PlDataTypeReq PlTileScheduled PlTimeScheduled PlDataScheduled *Library:* PlCore1IF *Classes:* PlDataSource PlDataTypeReq PlTileScheduled PlTimeScheduled PlDataScheduled *Library:* PlCore2 *Classes:* PlRoutineArrival PlNonRoutineArrival PlUserParameters *Library:* PlSSIT *Classes:* PlDataTypeFiles PlExitMessage *Library:* PlUtil *Classes:* PlOrbitModel PlPathModel	*Process:* EcDpAtRegisterPGE *Class:* DpAtDatabase	The EcDpAtRegisterPGE sends requests to the Sybase Server to insert/update data (inputs/outputs and scheduling information) in the database that defines a PGE.
Insert/ Update Granules and PGE information	One per insert/update of granule information	*Library:* PlCore1 *Class:* PlDataGranule	*Process:* EcDpAtInsertStaticFile *Class:* DpAtDsrv	Insert/update granule information received from an insert request or with data about modified existing PGEs.

Table 4.7.2.5-1. AITTL Process Interface Events (3 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request DAPs	One per DAPs request	*Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDpAtAcquireDAP *Class:* DpAtDsrv	The EcDpAtStageDAP requests DAPs from the SDSRV Archives based on the UR. In response, the DAPs are returned and stored on the local AITTL disk.
Request MCF	One per MCF request	*Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDpAtGetMCF *Class:* DpAtDsrv	The EcDpAtGetMCF application processes a request for a MCF template. In response, the MCF template is returned and populated.
Request PGE & Data Insertion	One per data put into the archive.	*Library:* DpAtDsrv *Class:* DpAtDsrv	*Processes:* EcDpAtInsertTestFile, EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile *Class:* DpAtDsrv	The EcDpAtInsertStaticFile, EcDpAtInsertTestFile and EcDpAtInsertExeTarFile processes put various types of data into the archive, from SSAP information to Static files and PGE executables. In response, The EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, and EcDpAtInsertTestFile processes get the results of the inserts, including the UR, for future acquires requests.
Search Advertise-ments	One the first time each ESDT is used.	*Libraries:* PlCore1 PlCore1IF *Class:* PlDataType	*Processes:* EcDpAtInsertTestFile, EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile EcDpAtSSAPGui *Class:* DpAtDsrv	The EcIoAdServer receives search requests for subscription event and signature service advertisements from the EcDpAtInsertExeTarFile, EcDpAtInsertStaticFile, and the EcDpAtInsertTestFile processes or the EcDpAtSSAPGui. The same processes or GUI obtains the proper signatures for acquiring data granules from the EcDsScienceDataServer (for the insert and update of metadata within the SDSRV archives).
Request Product	One per user request.	*Library:* DpAtDsrv *Class:* DpAtDsrv	*Process:* EcDpAtStageDAP	The EcDpAtStageDAP sends requests to the EcDsScienceDataServer for particular data granules to be FTP pushed onto the DPS science processor as input for data processing or for SSIT work.

Table 4.7.2.5-1. AITTL Process Interface Events (4 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Manage-ment Services	Per process request.	*Library:* EcAgInstrm *Class:* EcAgManager	*Processes:* All AITTL Processes	The EcMsAgSubAgent provides a basic management library of services to the processes, implemented as client or server applications, using the CSS Process Framework. The basic library management of services include: • **Lifecycle commands** - The EcMsAgSubAgent forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the mode (e.g., OPS, SHARED, test, training) for the application to run.

4.7.2.6 Algorithm and Test Tools Data Stores

Table 4.7.2.6-1 provides descriptions of the data stores shown in the AITTL CSCI architecture diagram.

Table 4.7.2.6-1. AITTL Data Stores

Data Store	Type	Functionality
PGE/ESDT ODL files	Files	These files are written in *parameter = value* formats to define the inputs and outputs of a PGE and any relevant scheduling information (including Production Rules), and are created by the Instrument Teams and the SSIT personnel.
PDPS database	Database	The PDPS database is replicated at each site for fault handling and recording purposes. The PDPS database holds all the persistent data including: • Resource information entered with the Resource Planning utilities • PGE and data type information entered at SSIT • Production Request, Data Processing Request and Data Granule information entered using the Production Request Editor • Plan information entered using the Production Planning Workbench

4.7.3 Data Processing Hardware Components

4.7.3.1 Science Processor Hardware CI (SPRHW) Description

Science Processor hardware (SPRHW) consists of the Science Processor Hardware and the Queuing Server Hardware.

The Science Processor Hardware features Redundant Arrays of Inexpensive Disks (RAID) devices set at RAID Level 3. The Queuing Server Hardware features attached disk packs for additional storage. X-terminals are also provided as part of the SPRHW for additional user access to ECS.

Science Processor

The Science Processor is based on a 64-bit SGI machine. Each Science Processor consists of 12 or 16 processors (See 920-TDx-001 series of base-line documents to see how the configuration is determined). Each Science Processor has one to six GB of memory with eight-way interleaving to improve the input/output (I/O) performance between the processors and memory (see 920-TDx-001 series of base-line documents). The SGI architecture is configured with I/O subsystems attached to the back plane and referred to as PowerChannel2 or IO4 cards. Each IO4 provides serial and parallel connections, two fast-wide differential SCSI-2 channels, and space for two High Input Output (HIO) controller cards. HIO controller card includes a HIPPI card, a FDDI card, and a card to support three SCSI-2 channels.

The number of IO cards specified for each Science Processor is determined by allocating HIO slots to the FDDI and HIPPI interfaces, and counting the number of SCSI-2 interfaces required. The number of internal and external SCSI-2 devices supported by the system determines the required number of SCSI-2 interfaces. The first SCSI-2 channel is delegated to internal devices, i.e., CD-ROMs, floppy disk drives, and tape drives. Internal disks ranging in aggregate size from eight GB to 12 GB are allocated to the second SCSI-2 channel. External disk arrays are allocated to subsequent SCSI-2 channels; the number of channels is based on the required throughput of the external file systems (see 920-TDx-001 and 922-TDx-015 series of base-line documents).

The internal disks of the Science Processor are only used to provide swap space for the operating system and to provide file system space for the operating system and applications (see 920-TDx-001 and 922-TDx-015 series of base-line documents).

A FDDI sub-network is implemented at each site to support the Planning and Data Processing Subsystems (PDPS). Each processing unit of Science Processor (including the Queuing Server) is dual-attached to the PDPS FDDI sub-network (see 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents).

Data transfer requirements between the Science Processor and the DSS are met with a switched HIPPI network implemented via a central HIPPI switch with switched 800 Mbps interface ports. The Science Processor connects directly to the DSS hosts (see 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents).

Because the Science Processor does not provide long term, secure storage, data backup and recovery are not provided. The Science Processor storage is used to hold ancillary data files and data granules for short periods of time. If a file system failure occurs within a Science Processor, the algorithms, ancillary data files, and data granules are recoverable from the Data Server.

The dual-ring implementation of FDDI provides fault tolerance. Media failures within the FDDI fabric do not result in any loss of service and does not require re-configuration. Neither is it required to have multiple physical communications paths to each host. Hosts within the Science Processor use dual-attached station cards.

Failure recovery for the HIPPI switch used is supplied by stocking spare Line Replaceable Units of the switch power supplies, interface cards, fan. If an individual interface card fails, a host is re-configured to a hot spare interface card by moving two cables and sending the activating software commands to the switch. If the control module fails, it is replaced with a spare module and the switch is re-configured. In the event of a failure of the entire switch, the switch is either replaced or repaired.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

Queuing Server

The Queuing Server with a Sybase database, directs AutoSys to load and execute the daily production schedule of a DAAC.

The Queuing Server is based on the SUN Server or the SUN workstation depending on the DAAC site capacity requirements. With a load requirement on the AutoSys and Sybase database for a 24-hour production run of 187,200 jobs, the Queuing Server uses four Ultra-SPARC processors. DAAC sites with smaller production runs are equipped with a Queuing Server based on a dual-processor, SPARC-based workstation (see 920-TDx-001 series of baseline documents).

Each Queuing Server is equipped with a minimum of 384 MB of memory to meet the AutoSys and Sybase database processing requirements (see 920-TDx-001 series of baseline documents).

The internal disks on a Queuing Server are only used to provide swap space for the operating system and to provide file system space for the operating system and applications (see 920-TDx-001 and 922-TDx-014 series of baseline documents).

Additional storage needed to support the Sybase database and to back-up the database from the Planning and Data Processing Subsystems (PDPS) Database Management System Server is via a SCSI-2 interface. To support failure recovery of the Sybase databases, two times the normal operating storage is available (see 920-TDx-001 and 922-TDx-014 series of baseline documents).

A FDDI sub-network is implemented at each site to support the Planning and Data Processing Subsystems (PDPS). Each processing unit of a Science Processor and the Queuing Server is dual-attached to the PDPS FDDI sub-network (see 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of baseline documents).

The AutoSys database located on the Queuing Server is replicated by the MSS Backup Server (See Section 4.9.16: MHWCI Description) to a physical location on the PDPS Database Management System Server. When a disk or database failure occurs on the primary database, AutoSys continues to operate using the backup database on the PDPS Database Management System Server.

The dual-ring implementation of the FDDI provides fault tolerance. Most media failures within the FDDI fabric do not result in any loss of service or require re-configuration of the hardware. Given the fault tolerance of FDDI, it is not required to have multiple physical communications paths to each host in the SPRHW Each host within the SPRHW uses dual-attached station cards.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.7.3.2 Algorithm Quality Assurance Hardware CI Description

Algorithm Quality Assurance Hardware (AQAHW) used to validate the quality of ECS products include non-science QA, in-line QA, and SCF-based QA. Non-science QA, is specified by the DAAC Operations staff and includes data integrity checks on the data products and the metadata. In-line QA is a form of science QA validating product content using science algorithms. The ECS provides support for SCF-based QA by providing archive and communications capacity for the SCFs to sample and validate the contents of the products.

The AQAHW is an AQA workstation and a Disk/RAID Driver.

AQA Workstation

The AQA workstation provides a software execution environment equivalent to the AI&T software execution environment in order to facilitate the use of the AQA workstation for AI&T when necessary. Also, the AQA supports complex data viewing techniques.

The AQA workstation is a 64-bit SGI machine. For information on the processors used, see the 920-TDx-001 series of base-line documents. The AQA workstation is equipped with a minimum of 128 MB of memory. The AQA workstation is equipped with four EISA slots. These EISA slots have a transfer rate of 33 MB per second. Additionally, the AQA workstation is equipped with two fast SCSI-2 connections. The FDDI interface card and the graphics subsystem use two EISA slots.

The internal disk provides swap space for the operating system and file system space for the operating system and applications (see 920-TDx-001 series of base-line documents). There are no external storage arrays.

A FDDI sub-network is implemented at each site to support the Planning and Data Processing Subsystems (PDPS). The AQAHW uses a single-attached FDDI interface to connect with the remaining members of the PDPS suite (See 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents). The AQAHW can also communicate with other ECS hardware items and the external world via the DAAC FDDI switch.

The function of the FDDI is not critical to AQAHW data processing and in the event of a failure, the faulty hardware (including the FDDI interface) is either repaired or replaced by a certified technician.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

AQA Disk/RAID Driver

The AQA Disk/RAID Driver supports the AQA Workstation by providing storage for the QA and SCF-based QA activities.

The AQA Disk/RAID Driver is a 64-bit SGI machine. For information on the processors, see the 920-TDx-001 series of base-line documents. The AQA Workstation is equipped with a minimum of 128 MB of memory.

The internal disk provides swap space for the operating system and file system space for the operating system and applications (See 920-TDx-001 and 922-TDx-003 series of base-line documents).

SGI storage units referred to as "Vaults" are attached to the AQA Disk/RAID Driver, via a SCSI-2 interface, to provide the additional storage space (See 920-TDx-001 and 922-TDx-003 series of base-line documents) to support QA.

A FDDI sub-network is implemented at each site to support the Planning and Data Processing Subsystems (PDPS). The AQAHW uses a single-attached FDDI interface to connect with the remaining members of the PDPS suite. The AQAHW also communicates with other ECS units and the external world via the DAAC FDDI switch (See 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents). The function of the FDDI is not critical to AQAHW data processing and in the event of a failure, the faulty hardware (including the FDDI interface) is either repaired or replaced by a certified technician.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.7.3.3 Algorithm Integration and Test Hardware CI Description

The Algorithm Integration and Test Hardware (AITHW) Configuration Item is the hardware to support the system level software validation, integration, and test and the integration and test of science software at a DAAC.

AITHW contains an AIT workstation and an AIT/Sybase Server with a laser printer and X-terminals to provide additional user access.

AIT Workstation

The AIT workstation is a 64-bit SUN workstation class machine with 128 MB of memory (See 920-TDx-001 series of base-line documents). The AIT workstation is for building and testing software in the AIT environment.

The AIT/Sybase Server internal disk provides swap space for the operating system and file system space for the operating system and applications (See 920-TDx-001 series of base-line documents). An external disk pack is attached via the SCSI port to provide additional storage.

A FDDI sub-network is implemented at each site to support the Planning and Data Processing Subsystems (PDPS). The AITHW uses a single-attached FDDI interface to connect with the remaining members of the PDPS suite. The AITHW also communicates with other ECS units and the external world via the DAAC FDDI switch (See 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents).

The function of this component is not critical to data processing. In the event of a component failure, the faulty component (including the FDDI interface) is either replaced or repaired by a certified SUN technician.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

AIT/Sybase Server

The AIT/Sybase Server is the hardware tools and database support for AIT.

The AIT/Sybase Server is a 64-bit SUN workstation class machine. For information on the processor utilized in this workstation see the 920-TDx-001 series of base-line documents. The AIT/Sybase Server is equipped with a minimum of 256 MB of memory (See 920-TDx-001 series of base-line documents).

The AIT internal disk provides swap space for the operating system and file system space for the operating system and applications (See 920-TDx-001 series of base-line documents) with external multi-packs attached via the SCSI port to provide additional storage.

A FDDI sub-network is implemented at each site to support the PDPS. The AITHW use either a single or dual-attached FDDI interface to connect with the other members of the PDPS suite. The AITHW communicates with other ECS units and the external world via the DAAC FDDI switch (See 920-TDx-001, 921-TDx-002, 921-TDx-003, and 921-TDx-004 series of base-line documents).

The function of the FDDI is not critical to AITHW data processing. In the event of a failure, the faulty hardware is either repaired or replaced by a certified technician.

The dual-ring implementation of FDDI provides a fault tolerant capability. Media failures within the FDDI fabric do not result in the loss of service and do not require hardware re-configuration.

With the inherent fault tolerance of FDDI, multiple physical communications paths to each host are not necessary.

Recovery/Fail-over for Hardware CIs is described in the 920-TDx-014 series of documents. There is a version for each DAAC, indicated by the letter appearing in place of the "**x.**" The document provides the recovery procedure for each host.

4.8 Communications Subsystem Overview

The Communications Subsystem (CSS) provides the capability to:

- Transfer information internal to the Earth Observing System Data and Information System (EOSDIS) Core System (ECS)
- Transfer information between the ECS sites
- Provide connections between the ECS users and service providers
- Manage the ECS communications functions
- Provide services requested to support System Management Subsystem (MSS) operations

CSS Subsystem Context Diagram

Figure 4.8-1 is the Communications Subsystem context diagram and Table 4.8-1 provides descriptions of the interface events shown in the Communications Subsystem context diagram. **NOTE**: In Table 4.8-1, Request Communications Support is shown as a single event to simplify the table and provide a list of services available from CSS to the other ECS subsystems.

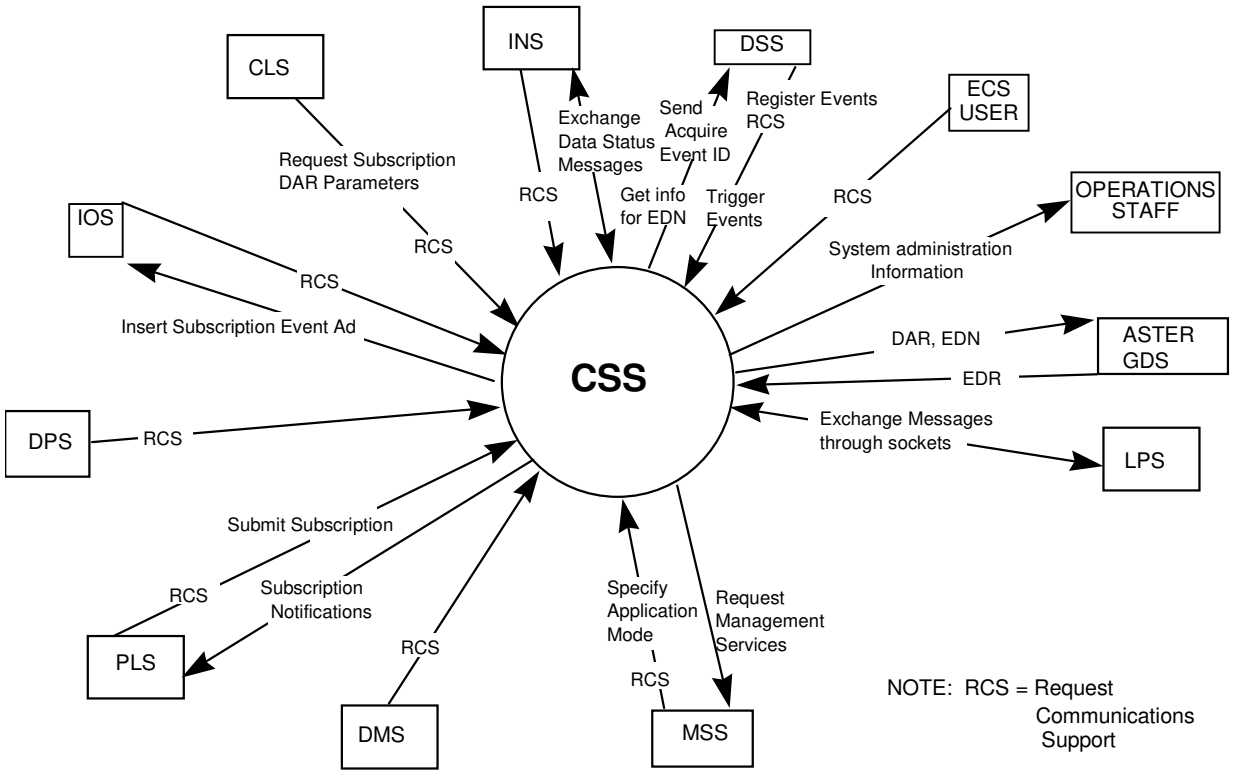

Figure 4.8-1. CSS Subsystem Context Diagram

Table 4.8-1. CSS Subsystem Interface Events (1 of 2)

Event	Interface Event Description
Request Communications Support (RCS)	The CSS provides a library of services available to each ECS subsystem. The services required to perform the specific subsystem assignments are requested by the subsystem from CSS. These services include: DCE support, file transfer servers, Network and Distributed File Servers, Bulk Data transfer services, file copying services, name/address services, password services, Server Request Framework (SRF), UR, Error/Event logging, message passing, Fault Handling services, User Authentication services, and Mode information.
Send Acquire	The E-mail Parser Gateway creates an acquire using the received Expedited Data Set Request (EDR) and sends it to the DSS via RPC. Also, the Subscription Server sends an acquire command to the DSS when an acquire action is specified in a subscription.
Get info for EDN	Information for the Expedited Data Set Notification (EDN) is obtained from the DSS and sent to a MSS e-mail script to be sent to the ASTER Ground Data System (GDS).
Event ID	The CSS sends Event IDs to the DSS when ESDTs are installed.
Register Events	The DSS sends the subscription events for an Earth Science Data Type to the CSS Subscription Server (SBSRV) when an ESDT is installed into the system.
Trigger Events	The DSS notifies the CSS (via an event trigger) when a subscription event has occurred for an ESDT.
Exchange Data Status Messages	Data status messages are sent to and from the CSS Gateways via a Remote Procedure Call (RPC). A Data Availability Notice (DAN) is sent to the INS and additional data status messages are exchanged with the INS.
Request Subscription	A request for notification of a specific event occurring within the system. For example: subscribing to the insert of a particular granule type through the CLS.
DAR Parameters	The CLS sends the parameters required for submittal of Data Acquisition Requests for ASTER instrument data. In response, a DAR Identifier is sent back to the CLS.
Insert Subscription Event Ad	The IOS receives requests to insert subscription event service advertisements from the CSS Subscription Server.
Submit Subscription	The PLS creates a subscription for an ESDT insert event, in response, the PLS receives a subscription ID.
Subscription Notifications	In response to a subscription request, a message is sent to the PLS manager containing the UR of the granule inserted into the DSS.
Request Management Services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the application mode (e.g., OPS, SHARED, test, training) for the application to run. • **User Profile Request** - The MSS provides requesting subsystems with access to User Profile parameters such as E-mail address and shipping address to support their processing activities.

Table 4.8-1. CSS Subsystem Interface Events (2 of 2)

Event	Interface Event Description
Specify Application Mode	Upon startup of any managed applications within a subsystem, the MSS provides the mode of operation to the CSS. Managed applications use the application interface PfGetMode to obtain the operational mode from the CSS.
Exchange Messages through sockets	The Landsat 7 Gateway sends and receives data status messages from the LPS through sockets.
DAR	The ASTER DAR Gateway sends the DAR to the ASTER GDS Storage Server.
EDN	The E-mail Parser Gateway stores the EDN messages with URs, time range, etc., and sends the EDN to the MSS to forward to the ASTER GDS.
EDR	The ASTER GDS personnel select the EDN as needed and send an EDR to the CSS to forward to the E-mail Parser Gateway.
System administration information	The Operations staff requests and receives information on system administration including application administration, fault metrics, performance metrics and system alarms.

CSS Subsystem Structure

The CSS is composed of one CSCI, the Distributed Computing Configuration Item (DCCI) and one HWCI. The CSCI is composed of 16 computer software components (CSCs) briefly described here as processes followed by a description of the HWCI.

- The Subscription Server (SBSRV) supports the detection of a defined event(s) and performs a specified action(s) for the client subscribed to the event. Event examples are science granule insertion, metadata update, new advertisement, and new schema exports to DDICT.
- The ASTER DAR Communications Gateway provides interoperability between the CLS DAR Tool GUI and the DAR API with an interface to the ASTER GDS servers.
- The ASTER E-Mail Parser Gateway supports the automated delivery of ASTER Expedited Data Sets (EDS) from the ECS to the ASTER GDS.
- The Landsat 7 Gateway provides user access to data collected by the Enhanced Thematic Mapper Plus (ETM+) instrument on the Landsat 7 satellite.
- The DCE service group is a COTS software set of Name, Security and Time Services.
    - The Name Service provides a link between distributed applications and associate services with names. The Name Service enables a client to locate a server.
    - The Security Service provides secure transfer of data on local and wide area networks. The Security Service provides authentication of users who try to access ECS data or services.
    - The Time Service keeps the host's system clocks synchronized within the ECS computer network by monitoring and adjusting the operating system clock for each individual host machine in the network.

The remote file access group provides the capability to transfer and manage files using the following five functions: FTP, FTP Notification, Bulk Data Server (BDS), Network File System (NFS), and Filecopy.

- FTP is an Internet standard application for file transfers. FTP enables a user to retrieve one or more files from a remote server and to send one or more files to a remote server. FTP also provides an insecure password protection scheme for authentication.
- FTP Notification provides successful completion notifications for FTP (get) data pulls and (put) data pushes.
- BDS is a fast file transfer utility to move large data files over high-speed networks such as the High Performance Parallel Interface (HIPPI) communications lines.
- The NFS provides a file sharing system among computers.
- The Filecopy utility copies files from a specified source location to a specified destination location with options available for data compression.
- The mail group provides electronic mail and bulletin board services.
    - E-mail is a standard Internet feature for asynchronous data transfers. The CSS E-mail service provides an interactive interface and an object-oriented application program interface (API) to send E-mail messages.
    - The Bulletin Board, another standard Internet feature, sends messages to each member in a specified group.
- Virtual Terminal provides the capability for the Operations staff on an ECS platform to remotely log onto another ECS machine.
- Cryptographic Management Interface (CMI) provides processes for obtaining random passwords and gaining access to non-DCE services like Sybase.
- The Domain Name Service (DNS) provides host names and addresses to a specified network by querying and answering queries. DNS provides naming services between the hosts on the local administrative domain and also across domain boundaries.
- The Infrastructure Library provides a set of services including the following.
    - Process Framework (PF): The PF provides a flexible mechanism (encapsulation) for the ECS Client and Server applications to transparently include specific ECS infrastructure features from a library of services. (Library services include: process configuration and initialization, mode management and event handling, life cycle services (server start-up and shut-down), communications services (message passing, FTP, underlying transport protocol, number of simultaneous threads), naming and directory services (OODCE naming), and set-up of security parameters.) The PF process is the encapsulation of an object with ECS infrastructure features and therefore the encapsulated object is fully equipped with the attributes needed to perform the activities assigned to it. The PF was developed for the ECS custom

developed applications and is not meant for use by any COTS software applications. The PF ensures design and implementation consistency between the ECS Client and Server applications through encapsulation of the implementation details of the ECS infrastructure services. Encapsulation therefore removes, for example, the task of each programmer repeatedly writing common initialization code. The PF is built by first developing a process classification for the ECS project from the client/server perspective. Then the required capabilities are allocated for each respective process level and type.

- Server Request Framework (SRF): The SRF infrastructure provides the standard for ECS synchronous and asynchronous communications between ECS applications. SRF provides enhanced OODCE RPC, message passing and persistent storage as a CSS support capability with the described features available by subsystem request.
- Message Passing: Message Passing provides peer-to-peer asynchronous communications service notifying clients of specific event triggers. Provided by subsystem request from the CSS.
- Universal References: An UR Provider object from C++ objects generates Universal References (URs) during their run time in virtual memory. The UR is a representation of the original object. URs can be transformed from an object to an ASCII representation and again returned to an object. URs are objects the users and applications use with full capabilities. Once the UR is obtained, the original object can be discarded and later reconstituted and used. URs can refer to objects local or remote to an address space. Therefore, the object does not have to remain in memory, and can, as appropriate, be written to a secondary storage system like a database.
- Event Logging: Event logging is the capability of recording events into files and provides a convenient way to generate and report detailed events.
- Server Locator: The Server Locator is a class that enables servers to register their location without referring to its physical location and be uniquely identified and located in the ECS. Client applications use the Server Locator to find any registered server.
- Failure Recovery Framework: The Failure Recovery Framework provides a general purpose fault recovery routine enabling client applications to reconnect with servers after the initial connection is lost.
- EcPo Connections: A suite of classes that provide basic database connection management methods and error handling mechanism for database users.

- CSS software is executed on multiple hardware hosts throughout the ECS system to provide communication functions, processing capability, and storage. The software and hardware relationships are discussed in the CSS Hardware CI description.

Use of COTS in the CSS Subsystem

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide basic functions and objects such as strings and collections. These class libraries must be installed with the CSS software to enable the CSS processes to run.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries provide interaction, in an object-oriented manner, to the Sybase database SQL server. The DBTools provide a buffer between the CSS processes and the relational database used. These class libraries must be installed with the CSS software to enable the Subscription Server to run and enable the clients to perform queries of subscription server database information.

- RogueWave's Net.h++

  The Net.h++ C++ class libraries provide functions and templates that facilitate writing applications, which communicate with other applications. These class libraries must be installed with the CSS software to enable the Landsat 7 Gateway to run.

- ICS' Builder Xcessory

  The Builder Xcessory GUI builder tool provides the capability to modify the displays of the Subscription Server Operator GUI. The tool also generates the C++ code producing the Operator GUI display at run time. There is no operational part of Builder Xcessory used at CSS run-time.

- Sybase Server

  The Sybase SQL server provides access for the Subscription Server to insert, update, and delete Subscription Server database information. The Sybase SQL Server must be running during CSS operations for the Subscription Server to execute database requests.

- Distributed Computing Environment

  The Distributed Computing Environment (DCE) provides a basis for building manageable, secure, distributed, interoperable, and portable applications across heterogeneous platforms. DCE offers APIs for application developers and includes commands sets for administrator and user application generation. DCE provisions include security, distributed file, cell directory, distributed time, and thread services. Using Remote Procedure Calls (RPCs), a feature of DCE, resources and files on a distributed network can be accessed.

- UNIX Network Services

  UNIX Network Services contain DNS, NFS, E-mail and Bulletin Board services, FTP, and TCP/IP capabilities.

4.8.1 Subscription Server Software Description

4.8.1.1 Subscription Server Functional Overview

The Subscription Server (SBSRV) provides the capability to register events, submit subscriptions, and process subscriptions upon event notification. Events and subscriptions are stored persistently in the SBSRV Database. During registration, events are made available through the Interoperability Subsystem's (IOS) Advertisement Service. Subscriptions are submitted for an advertised event. The subscriptions can be qualified and can also include information specifying an action to be performed on behalf of the subscriber (e.g., acquire a data granule). Subscriptions can also be updated or deleted from the database. Upon event notification, all subscriptions for the event are extracted from persistent storage and associated actions are performed. Additionally, subscribers receive notification the event was triggered, via E-mail or through message passing (i.e., a message from a process). The SBSRV also includes an Operator GUI for entering, updating, and deleting subscriptions interactively.

4.8.1.2 Subscription Server Context

Figure 4.8.1.2-1 is the Subscription Server context diagram and Table 4.8.1.2-1 provides descriptions of the interface events in the Subscription Server context diagram.

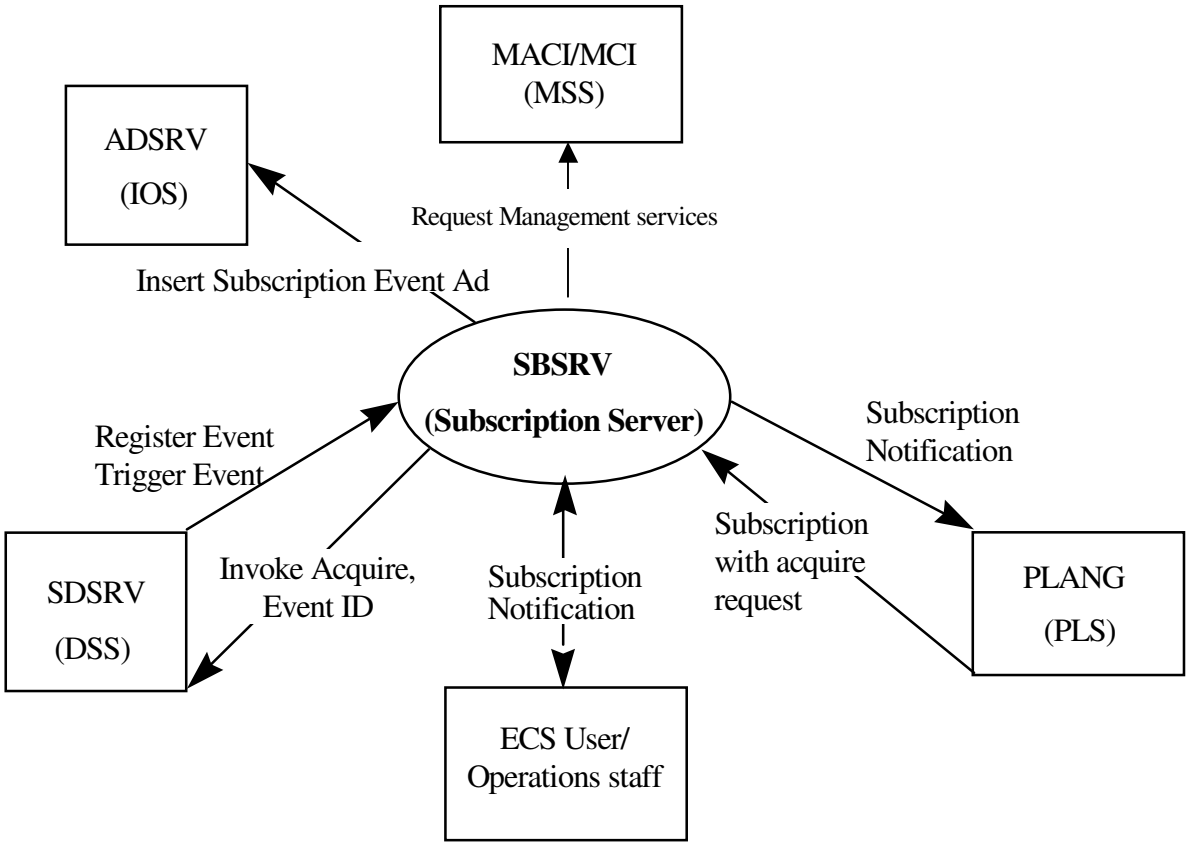

Figure 4.8.1.2-1. Subscription Server Context Diagram

Table 4.8.1.2-1. Subscription Server Interface Events

Event	Interface Event Description
Insert Subscription Event Ad	The ADSRV CSCI receives requests to insert subscription event advertisements from the SBSRV CSC.
Register Event	An event producer (in this version of ECS, only the SDSRV CSCI produces events) tells the SBSRV about the events it can trigger. This information enables the SBSRV CSC to present the events available for subscription to the ADSRV CSCI.
Trigger Event	The event producer notifies the SBSRV CSC of the event occurrence and triggers the SBSRV to notify and (optionally) perform the specified action(s) on behalf of the subscriber.
Invoke Acquire	The SBSRV CSC invokes acquire to the SDSRV CSCI when specified as an action in a subscription.
Event ID	The SBSRV CSC sends Event Ids to the SDSRV CSCI when ESDTs are installed.
Subscription Notification	The SBSRV CSC sends notification (E-mail or inter-process) to the subscriber when the event subscribed to occurs.
Subscription with acquire request	A subscriber (optionally) sends information with the subscription, specifying an action(s) (e.g., acquire) to be taken when the subscribed event occurs.
Request Management Services	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the application mode (e.g., OPS, SHARED, test, training) for the application to run. • **User Profile Request** - The MSS provides requesting subsystems with access to User Profile parameters such as E-mail address and shipping address to support their processing activities.

4.8.1.3 Subscription Server Architecture Diagram

Figure 4.8.1.3-1 is the Subscription Server architecture diagram. The diagram shows the events sent to the Subscription Server process and the events the Subscription Server process sends to other processes.

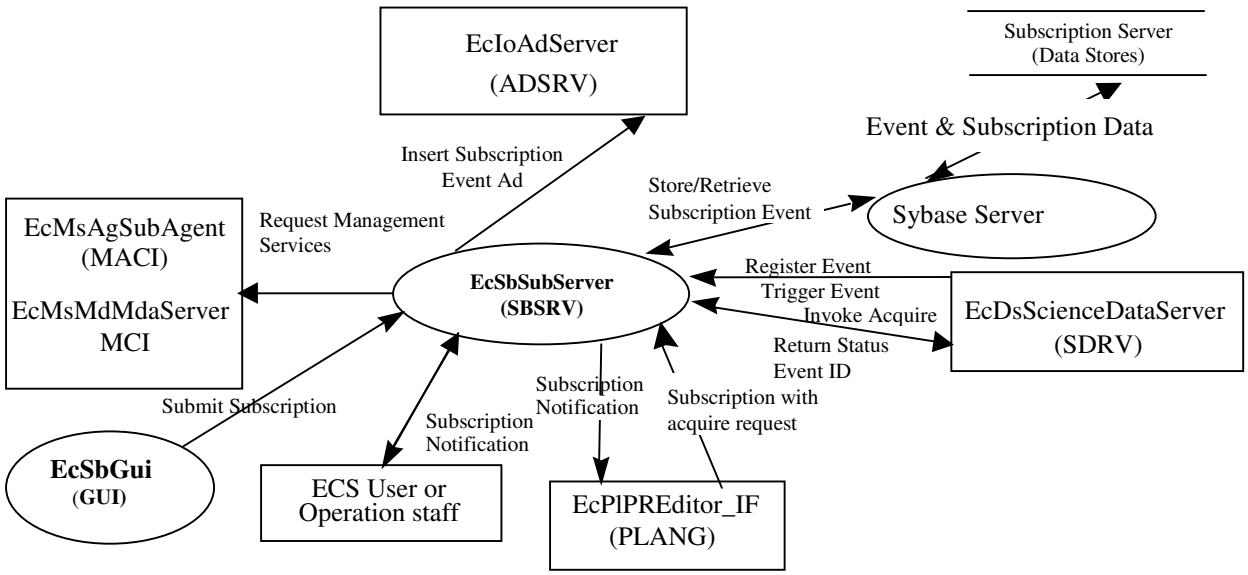

Figure 4.8.1.3-1. Subscription Server Architecture Diagram

4.8.1.4 Subscription Server Process Descriptions

Table 4.8.1.4-1 provides descriptions of the processes shown in the Subscription Server architecture diagram.

Table 4.8.1.4-1. Subscription Server Processes

Process	Type	COTS/ Developed	Functionality
EcSbSubServer	Server	Developed	The Subscription Server enables an event producer to register and trigger events. A subscriber can submit subscriptions for an event. Events and subscriptions can also be updated and deleted.
EcSbGui	GUI	Developed	The Subscription GUI provides an operator interface for submitting, updating and deleting subscriptions.
Sybase Server	Server	COTS	The Sybase Server is the SQL Server for the Subscription Server and is only run by the DAAC Operations staff.

4.8.1.5 Subscription Server Process Interface Descriptions

Table 4.8.1.5-1 provides descriptions of the interface events shown in the Subscription Server architecture diagram.

Table 4.8.1.5-1. Subscription Server Process Interfaces (1 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Insert Subscrip-tion Event Ad	One per event registration	*Libraries:* IoAdCore, IoAdSubs *Classes*: IoAdSignatureServiceAdv, IoAdApprovedAd, IoAdGroup, IoAdProvider	*Library*: EcSbSr *Class*: EcSbEvent	Upon event registration, the EcSbSubServer sends the event information along to the Advertiser to post the event is valid for subscriptions.
Submit subscrip-tion	One per subscription submitted	*Library:* EcSbSr, EcSbCl *Classes*: EcClSubscription, EcSbSubmitSubRequest, EcSbSubscription	PDPS, Operator GUI CLS, EcSbSubscriptionDispatcher, EcSbSubmitSubRequest	Internal: (GUI): submit subscription for an event. External: (client server): submit subscription for an event.
Subscrip-tion notification	One per subscription submitted	*Library:* EcSbSr *Class:* EcSbNotification	*Library:* EcSbSr *Class:* EcSbSubscription	The EcSbSubServer sends E-mail or inter-process notification (when message-passing framework is used) to the client/address specified in the subscription.
Invoke acquire	One per request	*Library:* EcSbSr *Class:* EcSbAcquireAction	*Library* EcSbSr *Class:* EcSbSubscription	Request is sent to action provider to carry out the specified action (e.g., acquire). This is an asynchronous call to the action provider with no completion status made available to the EcSbSubServer.
Return status	One per request	*Library:* EcUt *Class:* EcUtStatus	Most CSS classes	Status returned by the action provider to simply indicate that the request was received, not that the action succeeded.
Event ID	One per ESDT install	*Library* EcSbCl *Class:* EcClEvent	*Library*: DsDelSh *Class*: DsDeDataDictController	The EcSbSubServer sends Event Ids to the EcDsScienceDataServer when ESDTs are installed.

Table 4.8.1.5-1. Subscription Server Process Interfaces (2 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Trigger Event	One per event trigger	*Library:* EcSbCl *Class:* EcClEvent, EcClTriggerEventCb	*Library:* DsBtSh *Class* DsBtSbsrvNotifier	The event producer notifies the EcSbSubServer when an event has occurred. Upon notification, the EcSbSubServer identifies subscriptions for the event, taking qualifiers into account, perform any specified actions via the action provider, and notifies the subscriber that the event has been triggered.
Register Event	One per event	*Library:* EcSbCl *Class:* EcClEvent, EcClRegisterEventCb	*Library:* DsDelsh *Class:* DsDeEventCustomizer	The event producer sends event information to the EcSbSubServer for persistent storage.
Store/ Retrieve Subscription Event	One per store and retrieve event	*Library:* EcSbSr *Class:* EcSbEventStore, EcSbSubscriptionStore	*Library:* EcSbSr *Class:* CSS classes from client or server side	The EcSbSubServer stores and retrieves subscription information and events from the EcSbSubServer Data Stores.
Request Management Services	One per service request	*Library:* EcAgInstrm *Classes:* EcAgManager, MsAgRegistry	*Process:* EcSbSubServer	The MSS provides a basic management library of services to the subsystems, implemented as client or server applications, using the CSS Process Framework. The basic management library of services includes: • **Lifecycle commands** - The MSS forwards commands to managed hosts in the network to start and to stop applications. On startup, it passes a parameter identifying the application mode (e.g., OPS, SHARED, test, training) for the application to run. **User Profile Request** - The MSS provides requesting subsystems with access to User Profile parameters such as E-mail address and shipping address to support their processing activities.
Subscription with acquire request	One per subscription with acquire request	*Library:* EcSbCl *Class:* EcClSubscription	*Process:* EcPlPREditor_IF	A subscriber (optionally) sends information with the subscription, specifying an action(s) (e.g., acquire) to be taken when the subscribed event occurs.

4.8.1.6 Subscription Server Data Stores

Subscription Server uses the COTS software Sybase database for its persistent storage. The following is a brief description of the types of data contained in the database:

- **event data**: includes event type, user id, qualified metadata attribute names, and other information describing an event
- **subscription data**: includes a link to the event data, user id, start and expiration dates, qualified metadata values (optional), and action information (optional) for what to do when an event occurs.

Table 4.8.1.6-1 provides descriptions of the information in the four separate Sybase data stores that are used by the Subscription Server. More detail on these data stores can be found in the Subscription Server Database Design and Schema Specifications for the ECS Project (Refer to CDRL 311).

Table 4.8.1.6-1. Subscription Server Data Stores

Data Store	Type	Functionality
EcSbEvent	Sybase	Contains the list of events to which a user or another subsystem can subscribe.
EcSbNewEventID	Sybase	This data store contains the next available ID for the EcSbEvent table.
EcSbNewSubID	Sybase	This data store contains the next available ID for the EcSbSubscription table.
EcSbSubscription	Sybase	This data store lists all the user and subsystem subscriptions. Each event can have many subscriptions. Each user can have many subscriptions. The same user can subscribe to the same event with different constraints. It is also possible that a user could subscribe to the same event with the same constraints.

4.8.2 ASTER DAR Gateway Software Description

4.8.2.1 ASTER DAR Gateway Functional Overview

The ASTER DAR Gateway provides interoperability between the CLS DAR Tool and the DAR API with an interface to the ASTER GDS servers. The CLS DAR Tool is DCE compatible and based on X/Motif, while the DAR API is based on TCP/IP sockets. The DAR Communications Gateway receives calls from the CLS DAR Tool and dispatches them to the appropriate DAR API operations for remote operations.

ECS users can login to their workstations via the CLS DAR Tool. The CLS DAR Tool communicates with the DAR Gateway via RPCs. The DAR Gateway forwards the requests to the DAR Server via DAR API library calls, which communicates over EBnet using TCP/IP sockets. The DAR Server returns results to the DAR Tool in a reverse sequence. The DAR Gateway supports concurrent requests in both synchronous and asynchronous modes.

The DAR API provides the functionality to transmit data concerning the DAR between the DAR Gateway and the DAR Server and makes the DAR Server database information available to ECS users. The functionality is provided to support seven DAR APIs: SubmitDAR, ModifyDAR,

getxARStatus, getSubxARStatus, getxARContents, querytxARSummary, and queryxARScenes. The ASTER DAR Communications Gateway supports the interoperability between the ECS CLS DAR Tool and the DAR APIs so the CLS DAR Tool only needs to access the gateway distributed objects.

DAR Communications are part of the ECS and ASTER GDS interface, where ground support for mission operations and science data processing are provided for the ASTER instrument on-board the EOS AM-1 spacecraft. The DAR Server is located in Japan and transparently interacts with ASTER Operations Segment (AOS) xAR Server and xAR Database to provide data to its clients. The DAR Server provides ECS users access to DAR database information via an API. DAR-related communication between ECS and the ASTER GDS is through ASTER GDS provided APIs, integrated into the DAR Communications Gateway. (The DAR Communications Gateway server is located at the EROS Data Center (EDC).)

4.8.2.2 ASTER DAR Gateway Context

Figure 4.8.2.2-1 is the ASTER DAR Gateway context diagram and Table 4.8.2.2-1 provides descriptions of the interface events shown in the ASTER DAR Gateway context diagram. The information contained in the context diagram and the interface events table is, respectively, applicable to each of the ASTER DAR Gateway functions: SubmitDAR, ModifyDAR, getxARStatus, getSubxARStatus, getxARContents, querytxARSummary, and queryxARScenes.

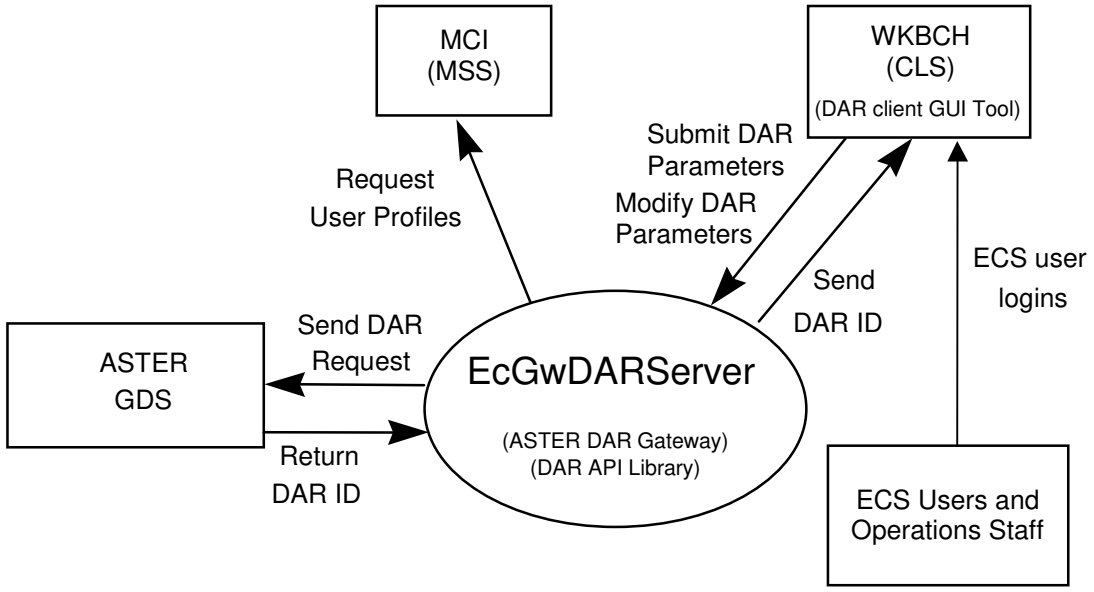

Figure 4.8.2.2-1. ASTER DAR Gateway Context Diagram

Table 4.8.2.2-1. ASTER DAR Gateway Interface Events

Event	Interface Event Description
ECS User Logins	An ECS user logs on to the ASTER DAR Tool.
Submit DAR Parameters	The ECS user submits DARs, via the ASTER DAR tool, to the ECS ASTER DAR Gateway via DCE/RPCs.
Modify DAR Parameters	The ECS user modifies DARs, via the ASTER DAR tool, and sends the modified requests to the ECS ASTER DAR Gateway via DCE/RPCs.
Request User Profiles	The ECS user requests user profiles via asterCatalog. The asterCatalog is a field in the user profile used by the ASTER DAR to check for user authorization.
Send DAR Request	The ECS user sends the request to the ASTER GDS Storage Server by DAR API library calls which communicates via TCP/IP sockets over EBnet.
Return DAR ID	The ASTER GDS returns a DAR ID to the ASTER DAR Gateway server at the ECS.
Send DAR ID	The ASTER DAR Gateway extracts the returned DAR ID and sends it to the ASTER DAR tool.

4.8.2.3 ASTER DAR Gateway Architecture

Figure 4.8.2.3-1 is the ASTER DAR Gateway architecture diagram. The diagram shows the events sent to the ASTER DAR Gateway server process and the events the ASTER DAR Gateway server process sends to other processes.

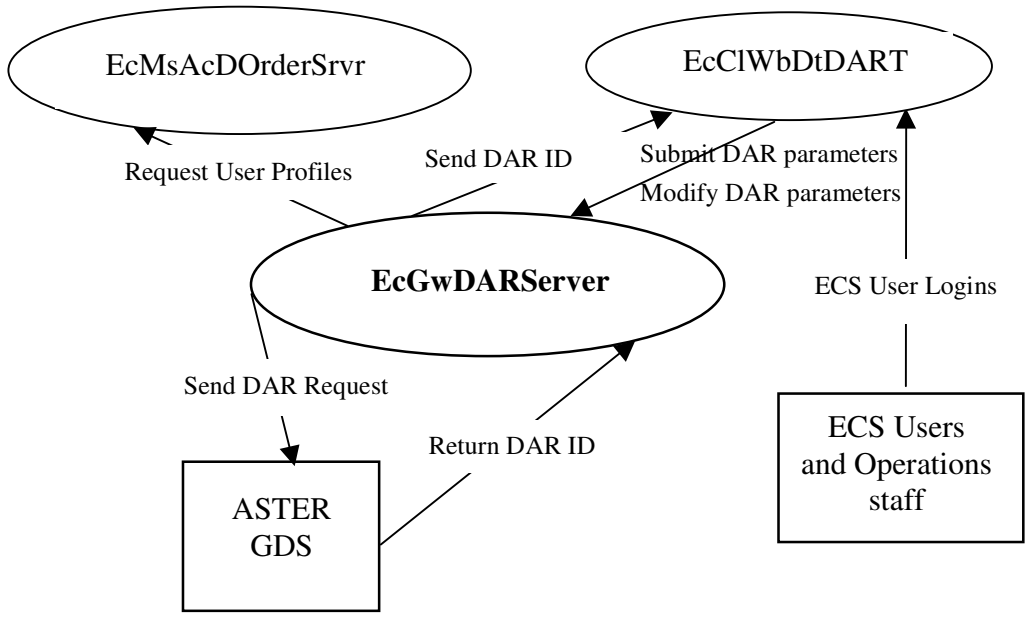

Figure 4.8.2.3-1. ASTER DAR Gateway Architecture Diagram

4.8.2.4 ASTER DAR Gateway Process Descriptions

Table 4.8.2.4-1 provides descriptions of the processes in the ASTER DAR Gateway architecture diagram.

Table 4.8.2.4-1. ASTER DAR Gateway Processes

Process	Type	COTS/ Developed	Functionality
EcGwDARServer	Server	COTS	The ASTER DAR Gateway provides seven functions. • Submit DAR function: Registered users use the DAR GUI tool to create a DAR. The DAR client is called by the GUI requesting the DAR be submitted in the default synchronous mode. The DAR client sends the DAR to the server and gets back a DAR ID. When the DAR GUI tool requesting the DAR be submitted in the asynchronous mode calls a DCE client, the DCE client makes an RPC call to the DCE server based on SRF. • Modify DAR function: A Modified DAR is sent from the DAR client to the server and gets a DAR ID back. Registered users use the DAR tool to “Modify” an existing request and “Submit” the modified request in the default synchronous mode. • GetxARStatus: Gets xARstatus by matching a search condition. A user creates a search condition by filling in the required fields of searchStream and submitting it in the default asynchronous mode. • GetSubxARStatus: Gets SubxARstatus by matching a search condition. A registered user creates a search condition by filling in the required fields of searchSubxarStream and submitting it in the default asynchronous mode. • QueryxARContents: Gets xAR contents by matching xarID. A registered user changes the default mode to synchronous and submits the modified DAR. • QueryxARSummary: Gets a subxAR status from the AOS xAR DB by matching xAR IDs. This function responds to multiple subxAR statuses for one request. • QueryxARScenes: Gets multiple xAR summaries from the AOS xAR DB by matching the search condition. This function responds to multiple xAR summaries for one request. Synchronous request processing is supported. Asynchronous request processing is supported. Multiple concurrent requests are supported.

4.8.2.5 ASTER DAR Gateway Process Interface Descriptions

Table 4.8.2.5-1 provides the descriptions of the interface events shown in the ASTER DAR Gateway architecture diagram.

Table 4.8.2.5-1. ASTER DAR Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
ECS user logins	One per ECS user login	*Process:* EcClWbDtDART	User	ECS Users login to ASTER DAR Client GUI tool.
Submit DAR Parame-ters	One per DAR	*Library:* EcGwDAR *Process:* EcGwDARSubmit DarRequest_C	*Process:* EcClWbDtDART *Class*: submitDarSrfRequest	User submits DAR to ECS ASTER DAR Gateway via DCE RPCs.
Modify DAR Parame-ters	One per modified DAR	*Library:* EcGwDAR *Process:* EcGwDARModifyDar Request_C	*Process:* EcClWbDtDART *Class*: modifyDarSrfRequest	User modifies DAR and sends the modified request to the ECS ASTER DAR Gateway via DCE RPCs.
Request User Profiles	One per ASTER DAR Gateway request	*Library:* MsAcClntSh *Class:* EcAcProfileMgr	*Process*: EcGwDARServer *Class*: EcGwDARGatewayR equest_S	ASTER DAR Gateway requests User Profiles via asterCat. The Category is a field in the user profile used by ASTER DAR to check for user authorization, asterCategory
Send DAR Request	One per ASTER DAR send	*Library:* IcDarApi	*Process*: EcGwDARServer *Class*: EcGwDARGatewayR equest_S	ASTER DAR Gateway sends the request to the ASTER GDS Storage Server via DAR API library, which communicates via TCP/IP sockets over EBnet.
Return DAR ID	One per DAR ID return	*Process:* EcGwDARServer *Class:* EcGwDARGatewayR equest_S	*Process*: EcGwDARServer	ASTER GDS returns a DAR ID to the ASTER DAR Gateway server at the ECS.
Send DAR ID	One per DAR ID sent	*Process:* EcClWbDtDART *Classes:* SubmitDarSrfResult, EcDaSubmissionInfo	*Process:* EcGuiDARServer *Class:* EcGwDARGatewayR equest_S	ASTER DAR Gateway extracts returned DAR ID and sends it to the ECS DAR tool.

4.8.2.6 ASTER DAR Gateway Data Stores

None

4.8.3 E-mail Parser Gateway Software Description

4.8.3.1 E-mail Parser Gateway Functional Overview

Expedited Data Sets (EDS) are raw satellite telemetry data processed into time-ordered instrument packets with packets separated into files for a given downlink contact. The ECS provides EDS to the ASTER GDS to use in evaluating the operation of the instrument. Level 0 EDS produced at the DAAC are staged for up to 48 hours before delivery to investigators at the Science Computing Facilities.

The E-mail Parser Gateway forwards notifications to the ASTER GDS when a EDS is received (the notification is called an EDN) and processes E-mail messages from the ASTER GDS requesting delivery of a EDS (the messages are EDRs). To facilitate this, EDS subscriptions are placed at the GSFC DAAC by user services personnel on behalf of the ASTER GDS. Each time the GSFC DAAC receives a EDS from EDOS, the subscription is triggered and an E-mail message is sent to the ASTER GDS. The subscription notifications are sent to the E-mail Parser Gateway to turn them into properly formatted EDN mail messages and sends them to the ASTER GDS via the MSS ASTER E-mail header handler to have the appropriate mail header information added. When ASTER orders the EDS, an E-mail message is sent via the MSS ASTER E-mail header handler to the E-mail Parser Gateway. The E-mail Parser Gateway formulates and submits the corresponding acquire request to the DSS SDSRV CSCI for an FTP push distribution of the EDS to ASTER.

4.8.3.2 E-mail Parser Gateway Context Diagram

Figure 4.8.3.2-1 is the E-mail Parser Gateway context diagram. Table 4.8.3.2-1 provides descriptions of the interface events shown in the E-mail Parser Gateway context diagram.

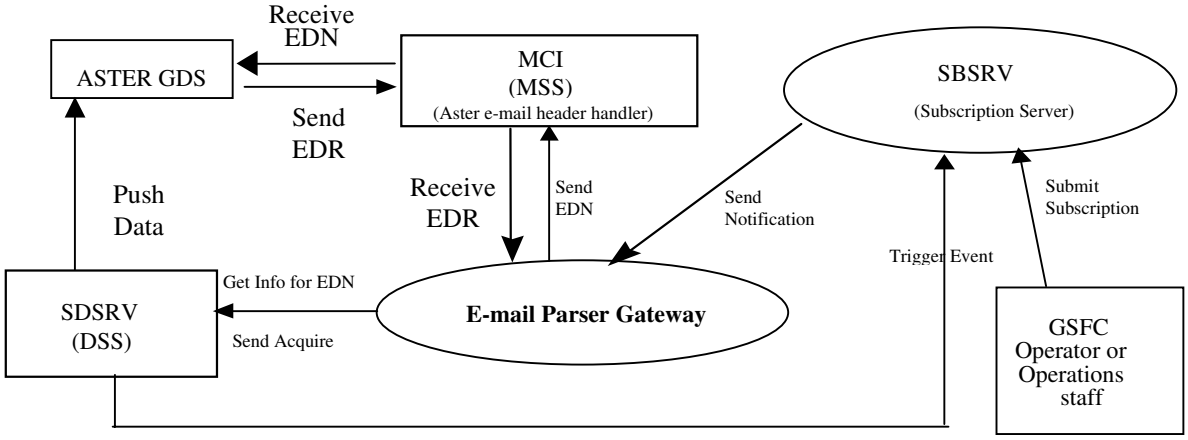

Figure 4.8.3.2-1. E-mail Parser Gateway Context Diagram

Table 4.8.3.2-1. E-mail Parser Gateway Interface Events

Event	Interface Event Description
Submit Subscription	GSFC DAAC Operations staff place a subscription with the subscription server (SBSRV CSC), on behalf of the ASTER GDS, once in the beginning of the mission and/or once at a time defined in an Operations Agreement between the ASTER GDS and the ECS.
Trigger Event	The SDSRV sends a trigger to notify the SBSRV of ASTER Expedited Data Sets.
Send Notification	The subscription server sends a notification via E-mail to the E-mail Parser Gateway to notify the ASTER GDS of the arrival of EDS from EDOS to the ECS.
Get Info for EDN	Information for the Expedited Data Set Notification (EDN) is obtained from the DSS and sent to a MSS e-mail script to be sent to the ASTER Ground Data System (GDS).
Send EDN	The E-mail Parser Gateway stores the EDN messages with URs, time range, etc., and sends the EDN to the MCI CSCI.
Receive EDN	The MCI CSCI E-mail header handler adds a header to the EDN and is received by the ASTER GDS via E-mail over EBnet.
Send EDR	ASTER GDS personnel select the EDN needed and send an EDR to the MCI CSCI.
Receive EDR	The MCI CSCI E-mail header handler strips the EDR header and is received by the E-mail Parser Gateway CSC.
Send acquire	The E-mail Parser Gateway CSC creates an acquire using the received EDR and sends it to the SDSRV CSCI via an RPC.
Push Data	The SDSRV CSCI receives an acquire command and FTP pushes the requested data (via the DDIST and STMGT CSCIs) and a signal file to the destination specified in the acquire command. (This interface is shown for completeness.)

4.8.3.3 E-mail Parser Gateway Architecture

Figure 4.8.3.3-1 is the E-mail Parser Gateway architecture diagram. The diagram shows the events sent to the E-mail Parser Gateway server process and the events the E-mail Parser Gateway server process sends to other processes.

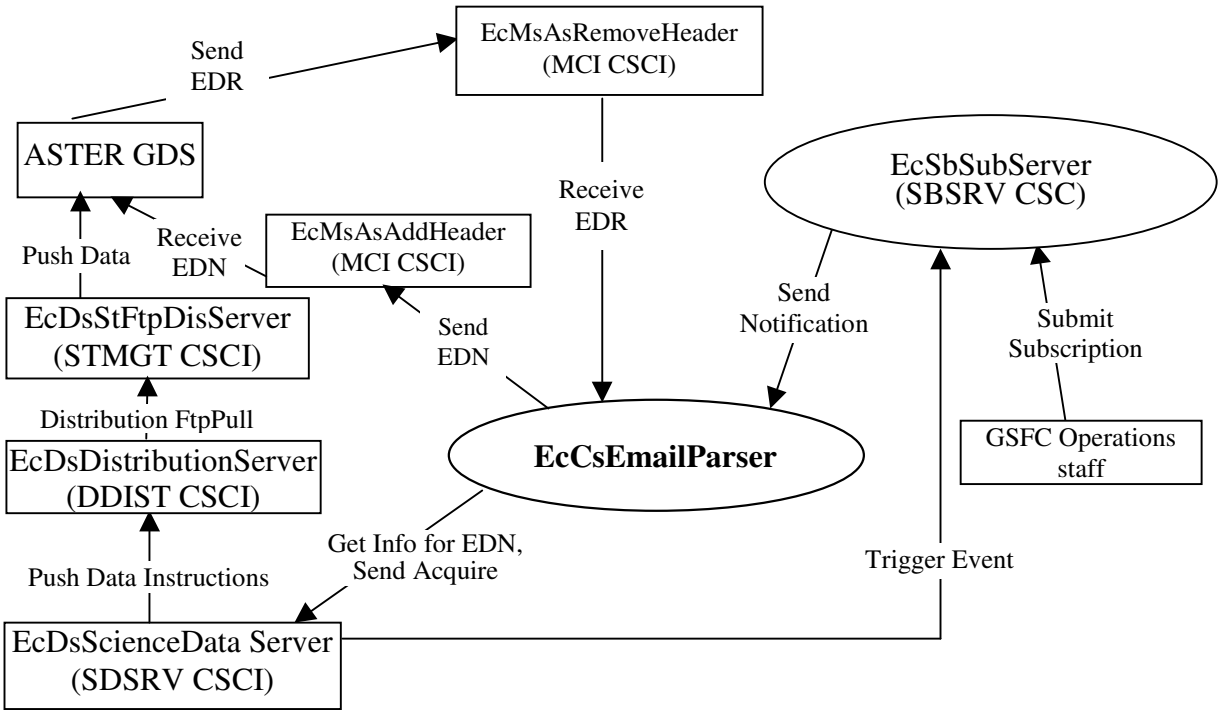

Figure 4.8.3.3-1. E-mail Parser Gateway Architecture diagram

4.8.3.4 E-mail Parser Gateway Process Descriptions

Table 4.8.3.4-1 provides a description of the process shown in the E-mail Parser Gateway architecture diagram

Table 4.8.3.4-1. E-mail Parser Gateway Processes

Process	Type	Cots/Developed	Functionality
EcCsEmailParser	Server	Developed	• Get UR from the Subscription Notification and use this UR to get the information for EDN from the EcDsScienceDataServer and send it to the EcMsAsAddHeader for notifying ASTER GDS. • Get EDR from the EcMsAsRemoveHeader. • Store and parse subscriptions and EDR in /EDN, /EDR directory • Send an Acquire request to the EcDsScienceDataServer via an RPC for an EDR request.

4.8.3.5 E-mail Parser Gateway Process Interface Descriptions

Table 4.8.3.5-1 provides descriptions of the interface events shown in the E-mail Parser Gateway architecture diagram.

Table 4.8.3.5-1. E-mail Parser Gateway Process Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Submit Subscrip-tion	One per subscription request	*GUI*	*User*	GSFC personnel subscribe to EDS.
Trigger Event (RPC)	One per trigger event	*Process:* EcDsScienceDataServer	*Process:* EcDsScienceData Server	The EcDsScienceDataServer triggers the EDS event to the EcSbSubServer.
Send Notification	One per send of EDN	*Process:* EcCsEmailParser	*Process:* EcSbSubServer	The EcSbSubServer sends EDN via E-mail to EcCsEmailParser.
Get Info for EDN	One per notification of the ASTER GDS	*Library:* DsCl *Class:* DsClESDTReference	*Process:* EcCsEmailParser	Uses the UR to get the EDN information from the EcDsScienceDataServer before sending the EDN to EcMsAsAddHeader.
Send EDN	One per E-mail send	*Process:* EcCsEmailParser	*Process:* EcCsEmailParser	The EcCsEmailParser sends the Send EDN to the EcMsAsAddHeader to have a header added.
Send EDR	One per EDR send	*Script:* EcMsAsRemoveHeader	*ASTER GDS*	After selecting the EDN, the ASTER GDS personnel send an EDR to the EcMsAsRemoveHeader to have the header removed.
Send Acquire (RPC)	One per acquire created	*Library:* DsCl *Class:* DsClRequest	*Process:* EcCsEmailParser	The EcCsEmailParser creates an acquire using the received EDR and sends it via an RPC to the EcDsScienceDataServer.
Push Data	One per file transfer of data via FTP	*Process:* EcDsStFtpDisServer *Library:* DsStDisFtp	*Process*: EcDsStFtpDis Server	The EcDsStFtpDisServer FTP pushes the requested data and a signal file to the destination specified in the acquire command. The signal file is also sent to the same directory immediately after completion of the FTP to denote the completion of the file transfer.
Push Data Instruction s	One per Push Data Instruction	*Process:* EcDsScienceDataServer	*Process:* EcDsScienceData Server	The EcDsScienceDataServer receives the Get Info for EDN with the send Acquire.
Distribution FtpPull	One per Distribution FtpPull	*Process:* EcDsDistributionServer	*Process:* EcDsDistribution Server	The EcDsDistributionServer receives Push Data Instruction from the EcDsScienceDataServer.

4.8.3.6 E-mail Parser Gateway Data Stores

None

4.8.4 Landsat 7 Gateway Server Software Description

4.8.4.1 Landsat 7 Gateway Server Functional Overview

The ECS user interface provides access to the Landsat 7 Processing System (LPS) data collected with the Enhanced Thematic Mapper Plus (ETM+) instrument on the Landsat 7 satellite. The Landsat 7 Project reformats the raw instrument data into Level 0R data and provides the data to ECS for ingest, archive and distribution. All ECS registered users are permitted access to Level 0R data, metadata, and browse data archived by the ECS.

Because LPS is not DCE compatible, the Landsat 7 Gateway Server is used as an interface between the LPS and DCE/Object Oriented DCE (OODCE) based ECS services. This gateway provides the capabilities for the following activities:

- Automated data transfer from LPS to ECS requiring transmission of control messages to provide the file information and handshaking required to complete the data transfer
- LPS sends Level 0R data, associated inventory metadata, and browse data to ECS
- ECS sends an acknowledgment to LPS, after archiving the Landsat 7 data
- LPS sends data to ECS for ingesting, storing, and distributing for Pre-launch checkout of instruments and development of initial calibration information
- ECS interface testing, operations testing, and acceptance testing activities with LPS
- ECS ingesting, archiving, and acknowledging receipt of Level 0R data from LPS for the previous 12-hour period, within 8 hours of the receipt of the data availability notice (DAN) from the LPS.

4.8.4.2 Landsat 7 Gateway Server Context

Figure 4.8.4.2-1 is the Landsat 7 Gateway Server context diagram and Table 4.8.4.2-1 provides descriptions of the interface events shown in the diagram.

Using sockets for exchanging messages, the Landsat 7 Gateway Server receives the Data Availability Notice (DAN) from LPS when it has data for ingest. After the data is ingested and delivered, LPS sends a Data Delivery Acknowledgment (DDA) back to the Landsat 7 Gateway Server. Also using sockets, the gateway forwards the Data Availability Acknowledgment (DAA) and the Data Delivery Notice (DDN) from INS to LPS.

Using an RPC to exchange data status, the Landsat 7 Gateway Server receives DAA and DDN from INS. The Landsat 7 Gateway forwards the DAN and the DDA to INS.

Figure 4.8.4.2-1. Landsat 7 Gateway Server Context Diagram

Table 4.8.4.2-1. Landsat 7 Gateway Server Interface Events

Event	Interface Event Description
DAN	The DAN originated by LPS is sent to Landsat 7 Gateway via sockets and is forwarded to INS by the Landsat 7 Gateway using RPC.
DAA	The DAA originated by INS is sent to Landsat 7 Gateway via RPC and is forwarded to LPS by the Landsat 7 Gateway using sockets.
DDN	The DDN originated by INS is sent to Landsat 7 Gateway via RPC and is forwarded to LPS by the Landsat 7 Gateway using sockets.
DDA	The DDA originated by LPS is sent to Landsat 7 Gateway via sockets and is forwarded to INS by the Landsat 7 Gateway using RPC.

4.8.4.3 Landsat 7 Gateway Server Architecture

The Landsat 7 Gateway Server is one ECS developed process, EcCsLandsat7Gateway. The Landsat 7 Gateway Server is managed via the Process Framework mechanism and Figure 4.8.4.3-1 is the architecture diagram of the Landsat 7 Gateway Server.

Figure 4.8.4.3-1. Landsat 7 Gateway Server Architecture Diagram

4.8.4.4 Landsat 7 Gateway Server Process Descriptions

Table 4.8.4.4-1 provides a description of the process shown in the Landsat 7 Gateway Server architecture diagram.

Table 4.8.4.4-1. Landsat 7 Gateway Server Process

Process	Type	COTS/ Developed	Description
EcCsLandsat7 Gateway	Server/ Client	Developed	The Landsat 7 Gateway Server is the interface for automated data transfers between the LPS and the ECS. It enables LPS to send DANs and DDAs to the ECS and enables the ECS to send DAAs and DDNs to the LPS. The Landsat 7 Gateway Server provides two basic interfaces: • Socket interface: used by the gateway and LPS to exchange messages. The gateway acts as both socket client and socket server. • RPC interface: used by the gateway and EcInAuto and EcInReqMgr to exchange messages. The gateway acts as both RPC client and RPC server. The Landsat 7 Gateway Server supports: • Multiple processes at a time • Asynchronous request processing.

4.8.4.5 Landsat 7 Gateway Server Process Interface Descriptions

Table 4.8.4.5-1 provides descriptions of the interface events shown in the Landsat 7 Gateway Server architecture diagram.

Table 4.8.4.5-1. Landsat 7 Gateway Server Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
DAN	One per message exchange with LPS	*RogueWaveNet library* *Socket interface*	LPS	LPS sends a DAN to EcInAuto through the Landsat 7 Gateway Server when it has data for ingest
DDA	One per message exchange with LPS	*RogueWaveNet library* *Socket interface*	LPS	LPS sends a DDA to EcInReqMgr through the Landsat 7 Gateway Server to acknowledge a delivery of ingested data
DDN	One per status exchange with EcInReqMgr	*Library: EcCsIDLLIB* *Classes: CsGwLEG, CsGwELG*	EcInReqMgr	EcInReqMgr sends a DDN to LPS through the Landsat 7 Gateway Server to provide information about a delivery of ingested data
DAA	One per status exchange with EcInAuto	*Library: EcCsIDLLIB* *Classes: CsGwLEG, CsGwELG*	EcInAuto	EcInAuto sends a DAA to LPS through the Landsat 7 Gateway Server to acknowledge the availability of data to be ingested

4.8.4.6 Landsat 7 Gateway Server Data Stores

None

4.8.5 Distributed Computing Environment Support Group Description

The Distributed Computing Environment (DCE) support group consists of the Security Authentication service, the Time Service, and the Server Locator functions.

4.8.5.1 Distributed Computing Environment Functional Overview

The Security Authentication of the CSS Security Service enables the server processes to obtain a valid login context for DCE. Security Authentication uses an Access Control List (ACL) to determine the access rights of a client for an object. All servers are required to use Security Authentication to login to DCE and perform the DCE operations for normal execution.

The CSS Time Service uses the DCE Distributed Time Service (DTS) to synchronize the system clocks on the ECS hosts by directly adjusting the operating system time on each host as needed. When a host clock needs to be advanced, the time adjustment is made in transparent increments until the correction is complete. When a host clock is found to be ahead of the actual time, the host clock is slowed down transparently in increments until the correction is complete. (A host clock is never adjusted in a backward direction). The CSS Time Service provides time within a millisecond resolution. ECS applications use the APIs provided by the CSS Time Service to obtain time in various formats. Applications needing to simulate a time other than the current time apply a specified delta time to the current time. Time classes enable applications to obtain the current time in various formats with or without a predetermined delta time.

The Server Locator of the CSS enables clients to locate and communicate with the various ECS servers. The ECS servers register their location information into the Cell Directory Service (CDS) of the Server Locator independent of the server's physical location. Servers registering in the CDS are advertising the services of the server in the same process. Clients use the Server Locator and the ECS operating mode to find the server of interest.

4.8.5.2 Distributed Computing Environment Context

Figure 4.8.5.2-1 is the Distributed Computing Environment context diagram. Table 4.8.5.2-1 provides descriptions of the interface events shown in the Distributed Computing Environment context diagram.

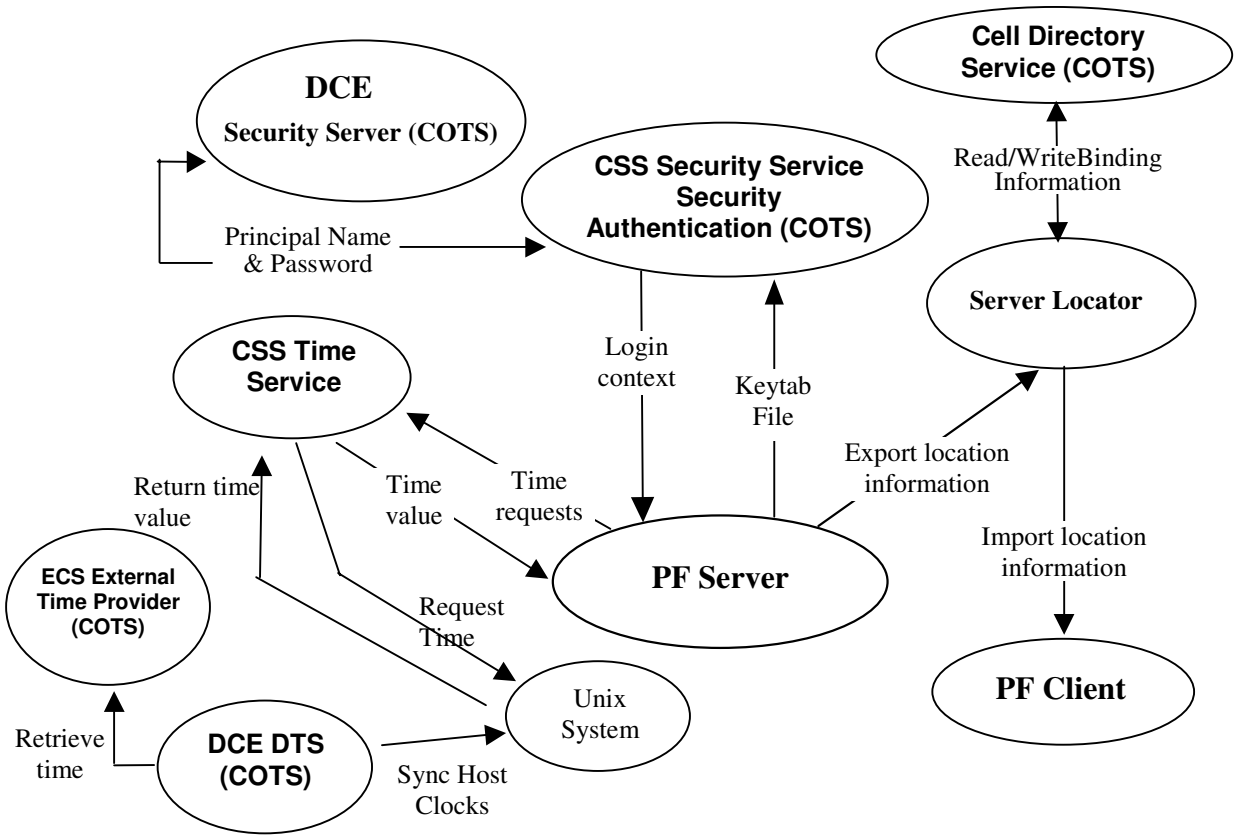

Figure 4.8.5.2-1. Distributed Computing Environment Context Diagram

Table 4.8.5.2-1. Distributed Computing Environment Interface Events (1 of 2)

Event	Interface Event Description
Keytab File	The Process Framework (PF) Server provides a keytab file to the Security Authentication to obtain a login key (password).
Principal Name & Password	The Security Authentication communicates with the DCE Security Server to verify the server's principal login name and password.
Login Context	The Security Authentication creates a login context on behalf of the server. The login context is returned to the server upon request.
Time Requests	Applications submit time requests to the CSS Time Service.
Time Value	The CSS Time Service returns a time value to the applications.
Retrieve Time	The DCE DTS retrieves a timestamp from the ECS external time provider.
Request time	The CSS Time Service submits requests to the Unix Operating System for current time values.
Return time	The Unix Operating System returns time values to the CSS Time Service.
Export location information (binding information)	The Server places physical and logical location information in CDS via the Server Locator.

Table 4.8.5.2-1. Distributed Computing Environment Interface Events (2 of 2)

Event	Interface Event Description
Import location information (binding information)	A Client retrieves server location information from CDS via the Server Locator.
Sync Host Clocks	DCE DTS adjusts the host's operating system clock, as needed, to maintain synchronization of host clocks.
Read/Write Binding Information	The Server Locator communicates with CDS to read and write server location information.

4.8.5.3 Distributed Computing Environment Architecture

Figure 4.8.5.3-1 is the Distributed Computing Environment support group architecture diagram. The diagram shows the events sent to the DCE processes and the events the DCE processes send to other processes.

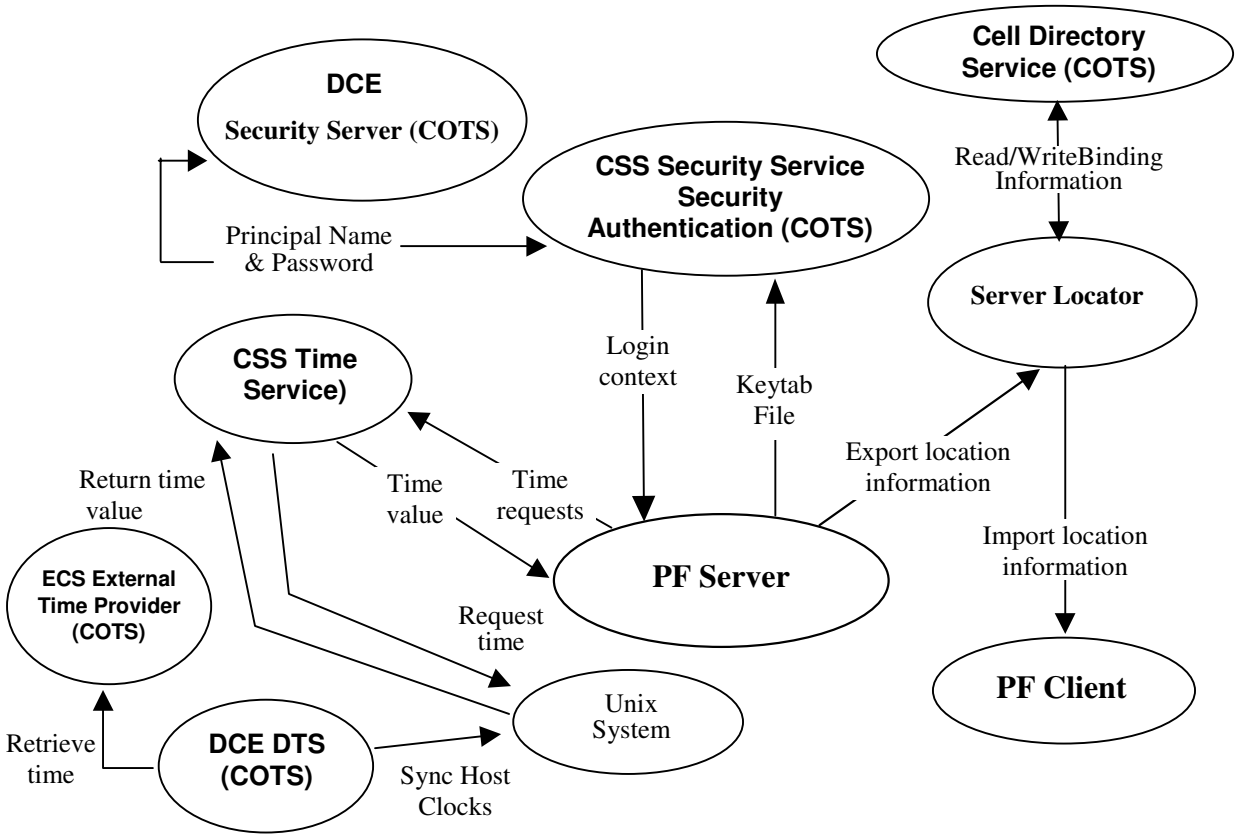

Figure 4.8.5.3-1. Distributed Computing Environment Architecture Diagram

4.8.5.4 Distributed Computing Environment Process Descriptions

Table 4.8.5.4-1 provides descriptions of the processes shown in the Distributed Computing Environment architecture diagram.

Table 4.8.5.4-1. Distributed Computing Environment Processes

Process	Type	COTS/ Developed	Functionality
DCE Security Server	Server	COTS	Stores registry information and verifies login names and passwords.
CSS Security Service Security Authentication	Server	COTS	Receives Keytab files and returns a Login context for valid users. Receives and returns Principal Name and Password information to the Security Server for login validation.
Server Locator	Server	Developed	Stores and retrieves server location data on the CDS.
PF Server	Internal	Developed	Provides Keytab files to Security Authentication and receives the returned Login context. Provides export location information to the Server locator. Requests Time from the Time Service and receives a Time value as the result. Provides Host clocks information to Distributed Time Service and receives Sync Host Clocks information for clock synchronization.
PF Client	Internal	Developed	Uses the Server Locator to retrieve server location information from CDS.
Cell Directory Service	Internal	COTS	Stores server location information and provides interfaces for storing and retrieving the location information.
DCE DTS	Server	COTS	Receives the delta time from an external time from the CSS Time Service and adjusts the UNIX clock as needed.
CSS Time Service	Server	Developed	Retrieves the current time from an external time provider and provides precise time in server requested formats
ECS External Time Provider	Server	COTS	Provides accurate time to synchronize the DCE cell.

4.8.5.5 Distributed Computing Environment Process Interface Descriptions

Table 4.8.5.5-1 provides descriptions of the interface events shown in the Distributed Computing Environment architecture diagram.

Table 4.8.5.5-1. Distributed Computing Environment Interface Events (1 of 2)

Event	Interface Event Description
Keytab File	The Process Framework (PF) Server provides a keytab file to the Security Authentication to obtain a login key (password).
Principal Name & Password	The Security Authentication communicates with the DCE Security Server to verify the server's principal login name and password.
Login Context	The Security Authentication creates a login context on behalf of the server. The login context is returned to the server upon request.
Time Requests	Applications submit time requests to the CSS Time Service.
Time Value	The CSS Time Service returns a time value to the applications.

Table 4.8.5.5-1. Distributed Computing Environment Interface Events (2 of 2)

Event	Interface Event Description
Retrieve Time	The DCE DTS retrieves a timestamp from the ECS external time provider.
Request time	The CSS Time Service submits requests to the Unix Operating System for current time values.
Return time	The Unix Operating System returns time values to the CSS Time Service.
Export location information (binding information)	The Server places physical and logical location information in CDS via the Server Locator.
Import location information (binding information)	A Client retrieves server location information from CDS via the Server Locator.
Sync Host Clocks	DCE DTS adjusts the host's operating system clock, as needed to maintain host clock synchronization.
Read/Write Binding Information	The Server Locator communicates with CDS to read and write server location information.

4.8.5.6 Distributed Computing Environment Data Stores

Table 4.8.5.6-1 provides a description of the data store shown in the Distributed Computing Environment architecture diagram.

Table 4.8.5.6-1. Distributed Computing Environment Data Stores

Data Store	Type	Functionality
ServerUR.map	Other	A flat file for the Server Locator classes to map short, logical service names to CDS entry names.

4.8.6 Remote File Access Group - File Transfer Protocol Description

The remote file access group consists of five functional processes described separately. The five processes are File Transfer Protocol (FTP), File Transfer Protocol Notification (FTPN), Network File System (NFS), Bulk Data Server (BDS), and Filecopy.

4.8.6.1 File Transfer Protocol Functional Overview

FTP is a user interface to the Internet standard File Transfer Protocol. With FTP a user is able to transfer files to and from remote network sites. FTP is client-server software where the user starts the client program first while the FTP daemon is the server started on the target machine.

4.8.6.2 File Transfer Protocol Context

Figure 4.8.6.2-1 is the FTP context diagram. Table 4.8.6.2 provides descriptions of the interface events in the FTP context diagram.

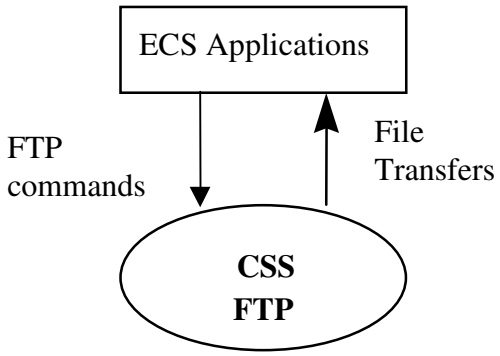

Figure 4.8.6.2-1. File Transfer Protocol Server Context Diagram

Table 4.8.6.2-1. File Transfer Protocol Interface Events

Event	Interface Event Description
FTP commands	ECS applications send FTP commands like login, dir, put, or get
File transfers	The FTP server and FTP client program transfers the requested file(s) from source to destination.

4.8.6.3 File Transfer Protocol Architecture

Figure 4.8.6.3-1 is the File Transfer Protocol architecture diagram. The ECS FTP is the standard UNIX utility with the CSS wrapper classes applied to provide additional ECS-developed capabilities. The CSS wrapper classes also provide APIs for more control and easier access to other applications.

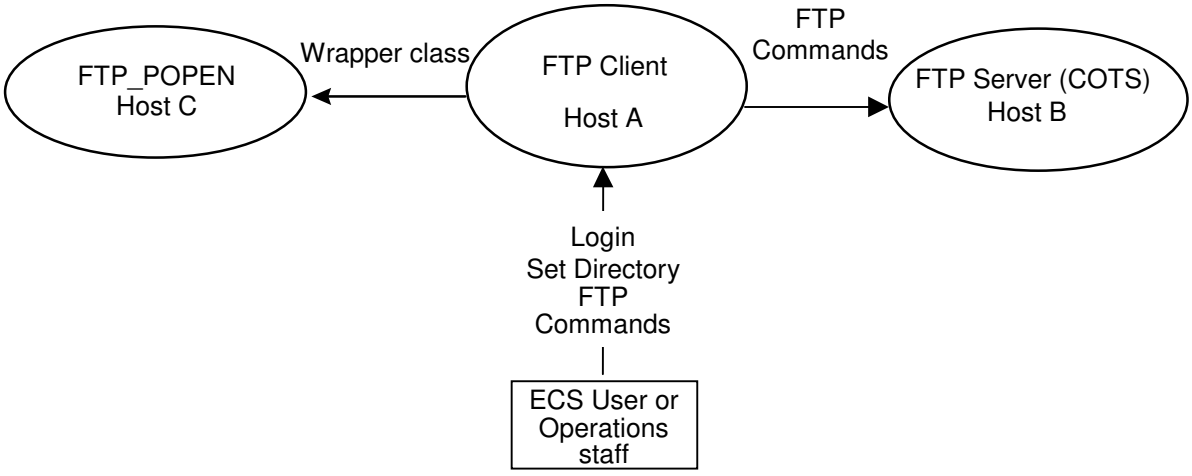

Figure 4.8.6.3-1. File Transfer Protocol Architecture Diagram

4.8.6.4 File Transfer Protocol Process Descriptions

Table 4.8.6.4-1 provides descriptions of the processes shown in the File Transfer Protocol architecture diagram.

Table 4.8.6.4-1. File Transfer Protocol Processes

Process	Type	COTS/ Developed	Functionality
FTP Server	Server	COTS	Provides basic FTP capabilities.
FTP Client w/ ECS FTP	API	Developed	Provides ECS specific additions for improved access by ECS applications to FTP servers.
User Process (FTP_POPEN)	Client Application	Developed	ECS application processes use the Wrapper class whenever FTP is used with other class methods.

4.8.6.5 File Transfer Protocol Process Interface Description

Table 4.8.6.5-1 provides descriptions of the interface events shown in the File Transfer Protocol architecture diagram.

Table 4.8.6.5-1. File Transfer Protocol Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Login	One per connection to FTP	*Library: CsFtFTPRelA Class: CsFtFTPRelA*	*Classes:* DsStFtpWrappers, EcMjEcsFtpProxy, MsMdHost, InMediaIngestRPUhl, ImMessage, DpPrDataManager	An application establishes the connection with the File Transfer Protocol Daemon server for FTP.
SetDirectory	One per directory set	*Library: CsFtFTPRelA Class: CsFtFTPRelA*	*Classes:* DsStFtpWrappers, EcMjEcsFtpProxy, MsMdHost, InMediaIngestRPUhl, ImMessage, DpPrDataManager	Command used for setting the (working) directory (on the client and server).
FTP Commands	One per file transfer	*Library: CsFtFTPRelA Class: CsFtFTPRelA*	*Classes:* DsStFtpWrappers, EcMjEcsFtpProxy, MsMdHost, InMediaIngestRPUhl, ImMessage, DpPrDataManager	Commands used to transfer files from host to host.
Wrapper class	One per FTP	*Library: CsFtFTPRelA Class: CsFtFTPRelA*	*Classes:* DsStFtpWrappers, EcMjEcsFtpProxy, MsMdHost, InMediaIngestRPUhl, ImMessage, DpPrDataManager	Provides wrapper functions to carry out FTP between two hosts.

4.8.6.6 File Transfer Protocol Data Stores

None

4.8.7 Remote File Access Group - File Transfer Protocol Notification

4.8.7.1 File Transfer Protocol Notification Functional Overview

The CSS provides an FTP Notification to the Pull Monitor task (in the DSS STMGT) upon completion of FTP pulls from the pull disk area of the DSS STMGT. The CsFtFTPNotify Class provides a method invoked by the Pull Monitor at predefined time intervals. The CsFtFTPNotify extracts the successful FTP information from the SYSLOG file FTPD Debug messages and sends the involved directory and file names to the Pull Monitor.

4.8.7.2 File Transfer Protocol Notification Context

Figure 4.8.7.2-1 is the File Transfer Protocol Notification context diagram. Table 4.8.7.2-1 provides descriptions of the interface events shown in the File Transfer Protocol Notification context diagram.

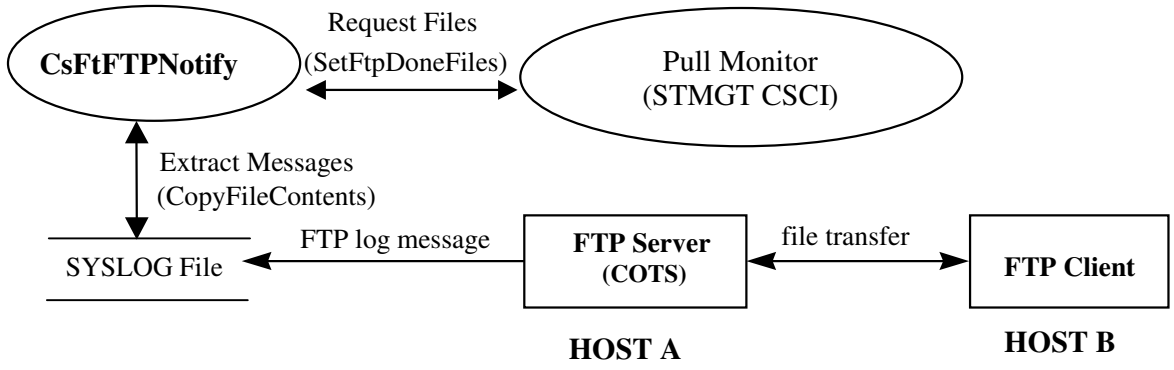

Figure 4.8.7.2-1. File Transfer Protocol Notification Context Diagram

Table 4.8.7.2-1. File Transfer Protocol Notification Interface Events

Event	Interface Event Description
Request Files	An ECS application (Pull Monitor) requests a transfer of files via the FTP service by the SetFtpDone function. After the transfer, FTPNotify extracts information from the syslog file and sends the file and directory names to Pull Monitor.
Extract messages	The “CopyFileContents” f application copies the log files.
FTP log message	The FTP server logs all FTP events to the system log (syslog) file.
File Transfer	The file transfer of specified files between the client and server.

4.8.7.3 File Transfer Protocol Notification Architecture

Figure 4.8.7.3-1 is the File Transfer Protocol Notification architecture diagram. The diagram shows the events sent to the CsFtFTPNotify class and the events the CsFtFTPNotify class sends to other processes. The Class method SetFtpDoneFiles reads the SYSLOG file and extracts the file and directory names involved in the completed transfer.

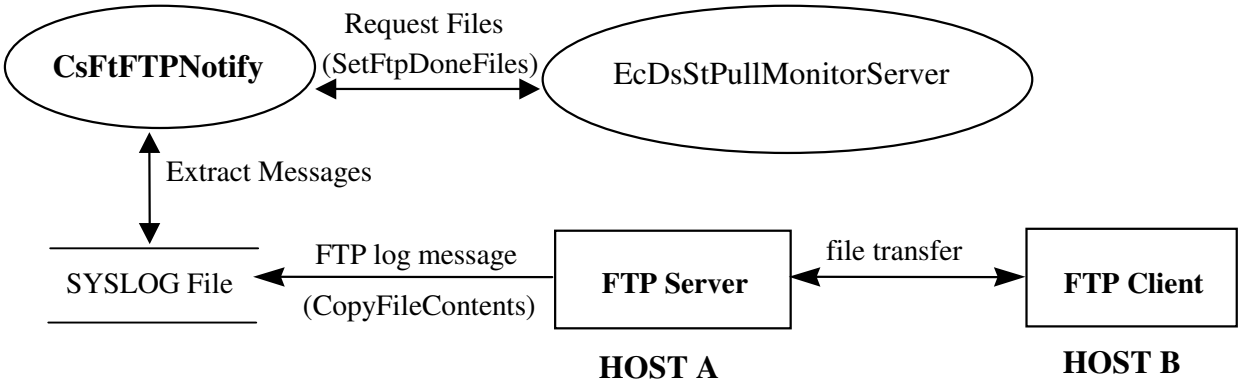

Figure 4.8.7.3-1. File Transfer Protocol Notification Architecture Diagram

4.8.7.4 File Transfer Protocol Notification Process Descriptions

Table 4.8.7.4-1 provides descriptions of the processes shown in the File Transfer Protocol Notification architecture diagram. The CsFtFTPNotify class extracts the file name and location of the files transferred from the SYSLOG file. The information is written to a file supplied by the caller of the class.

Table 4.8.7.4-1. File Transfer Protocol Notification Processes

Process	Type	COTS/ Developed	Functionality
FTP Server	Server FTP	COTS	The FTP server running on HOST logs the FTP messages to the SYSLOG file. A copy of the logs is made by the CsFtFTPNotify class
FTP Client	Client FTP	COTS	Initiates the FTP commands and gets or puts the specified file from or to a remote host and copies it to the local host initiating the FTP.

4.8.7.5 File Transfer Protocol Notification Process Interface Descriptions

Table 4.8.7.5-1 provides descriptions of the interface events shown in the File Transfer Protocol Notification architecture diagram.

Table 4.8.7.5-1. File Transfer Protocol Notification Interfaces

Event	Event Frequency	Interface	Initiated by	Event Description
Request Files	One per FTP request	*Library:* CsFtFTPNotify *Class:* CsFtFTPNotify	*Class:* DsStPullFtpNotify	The ECS process EcDsStPullMonitorServer (Pull Monitor) requests a transfer of files via the FTP service by the SetFtpDone files function. After the file transfer, FTPNotify extracts information from the syslog file and gives the file and directory name to the Pull Monitor.
Extract messages	One per log file copy	*Library:* CsFtFTPNotify *Class:* CsFtFTPNotify	*Class:* CsFtFTPNotify	The CopyFileContents function is used to copy log files.
FTP log message	One per system log file entry	syslog	FTP	The FTP server logs all FTP events to the system log (syslog) file.
File Transfer	One per file transfer	FTP	FTPclient Application	The specified file transfer takes place between the client and server.

4.8.7.6 File Transfer Protocol Notification Data Stores

Table 4.8.7.6-1 provides descriptions of the information in the SYSLOG File data store. More detail on these data stores can be found in the Subscription Server Database Design and Schema Specifications for the ECS Project (Refer to CDRL 311).

Table 4.8.7.6-1. File Transfer Protocol Data Stores

Data Store	Type	Functionality
SYSLOG File	File	Storage for details of successful or failed file transfers.

4.8.8 Remote File Access Group - Bulk Data Server Description

4.8.8.1 Bulk Data Server Functional Overview

The Bulk Data Server (BDS) is a non-standard extension to NFS, implemented as an enhancement on the client system and a daemon process on the server for transferring large (100 Megabytes and larger) files over high-speed networks. Figure 4.8.8.1-1 is a comparison with the Network File System (NFS)/Transmission Control Protocol/Internet Protocol (TCP/IP) protocols in the International Standards Organization (ISO) Open Systems Interconnect (OSI) 7-layer model. BDS exploits the data access speed of the Extended File System (XFS) and data transfer rates of network media, such as HIPPI and fiberchannel, to accelerate standard NFS performance. The BDS protocol, XBDS, modifies NFS functions and reduces the time needed to transfer large files over a network connection. BDSpro is the Silicon Graphics implementation of XBDS.

BDSpro is run on SGI machines with IRIX 6.2 (or later versions) and connected to a high-speed network (such as HIPPI or fiberchannel) running the TCP/IP suite.

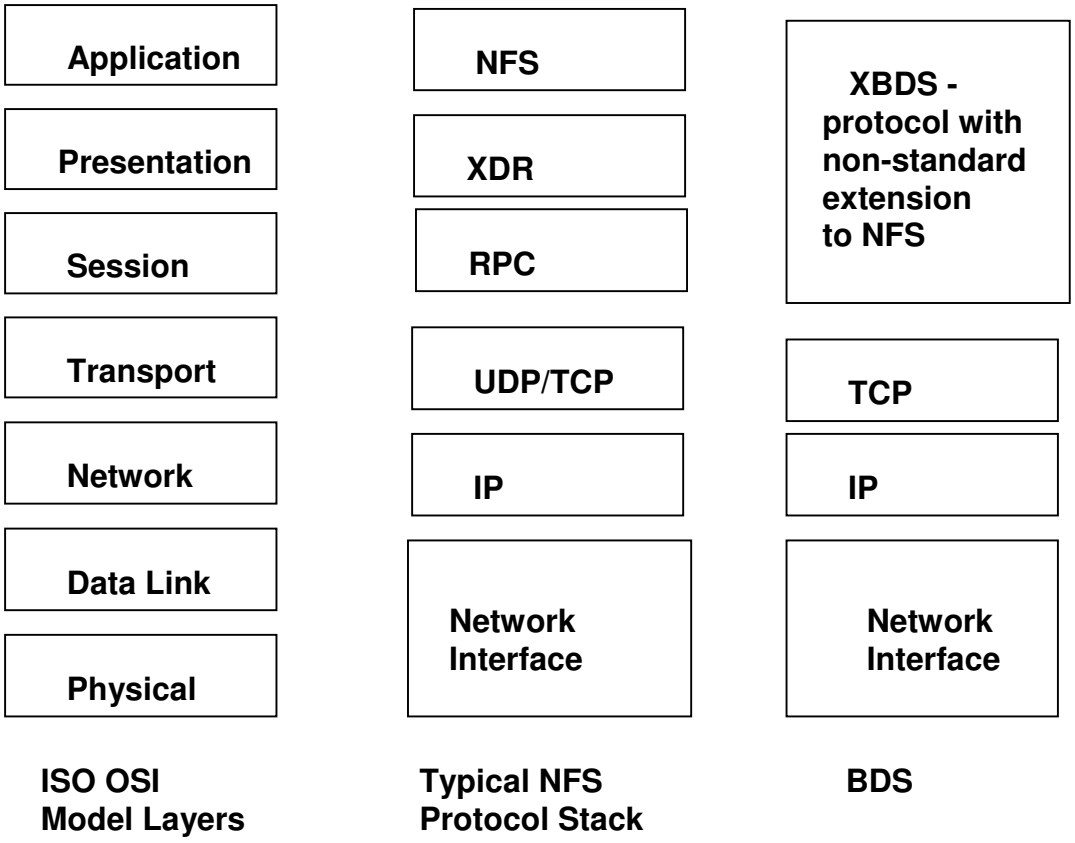

Figure 4.8.8.1-1. Bulk Data Server Protocol compared with ONC Protocols

4.8.8.2 Bulk Data Server Context

BDS is a file transfer utility to move large data files over the HIPPI communications lines. Figure 4.8.8.2-1 is the Bulk Data Server context diagram shown with an ECS application. The BDS applications within the ECS are in the DsStArchiveReal module of the Data Server Storage Management Archive software. The storage location calculation takes the vector of the data file as parameters with the location of the file, the unique file name, the original file name, the size of the file, and a checksum. BDS transfers data files produced by PDPS to archive. The data files are transferred via BDS over HIPPI and stored on AMASS.

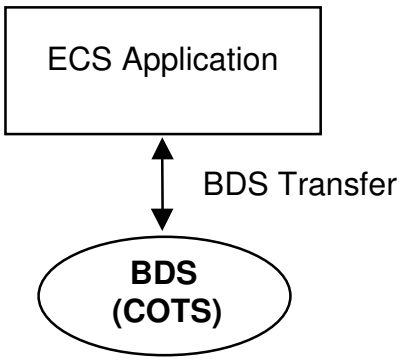

Figure 4.8.8.2-1. Bulk Data Server Context Diagram

Table 4.8.8.2-1 provides a description of the interface event shown in the Bulk Data Server context diagram.

Table 4.8.8.2-1. Bulk Data Server Interface Events

Event	Interface Event Description
BDS transfer	The BDS transfers large ECS application data files over the HIPPI interface.

4.8.8.3 Bulk Data Server Architecture

BDS is implemented as an enhancement to the NFS on the client system and as a daemon process on the server. Figure 4.8.8.3-1 is the BDS architecture diagram shown over HIPPI on an SGI client-server model.

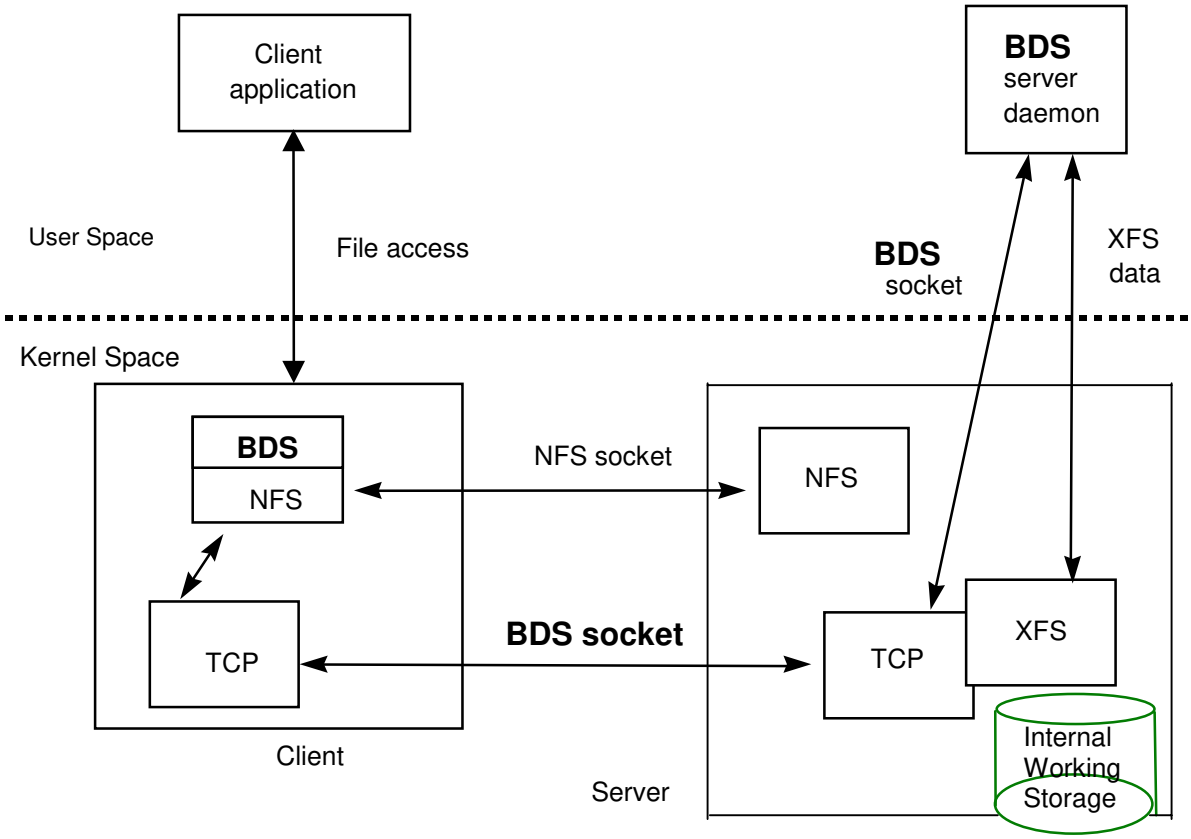

Figure 4.8.8.3-1. Bulk Data Server Architecture Diagram

4.8.8.4 Bulk Data Server Process Descriptions

Table 4.8.8.4-1 provides descriptions of the processes shown in the Bulk Data Server architecture diagram.

Table 4.8.8.4-1. Bulk Data Server Processes

Process	Type	COTS/ Developed	Functionality
Client application	Client	COTS	Provides XBDS protocol for client functions implemented as enhanced NFS/ External Data Representation (XDR)/RPC protocols.
BDS server daemon	Server	COTS	Provides XBDS protocol server functions, implemented as enhanced protocols for NFS/XDR/RPC protocols.

4.8.8.5 Bulk Data Server Process Interface Descriptions

Table 4.8.8.5-1 provides descriptions of the interface event shown in the Bulk Data Server architecture diagram. Pull Monitor Process, EcDsStPullMonitorServer.

Table 4.8.8.5-1. Bulk Data Server Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
File access	One per file transfer	COTS BDS	STMGT Applications	The application uses the file transfer via the BDS interface.

4.8.8.6 Bulk Data Server Data Stores

None

4.8.9 Remote File Access Group - Network File System Description

4.8.9.1 Network File System Functional Overview

The Network File System (NFS) provides a file sharing system between computers. NFS consists of a mounting protocol with a server, a file locking protocol with a server, and daemons to coordinate the file services provided. A server exports (or shares) a system of files by providing file system access to other hosts on a common network. An NFS client must explicitly mount the file system of interest before the file system is made accessible.

4.8.9.2 Network File System Context

Figure 4.8.9.2-1 is the Network File System context diagram. The NFS mounted directories reside on mount points made accessible for the use of other hosts machines. Table 4.8.9.2-1 provides descriptions of the interface event shown in the context diagram.

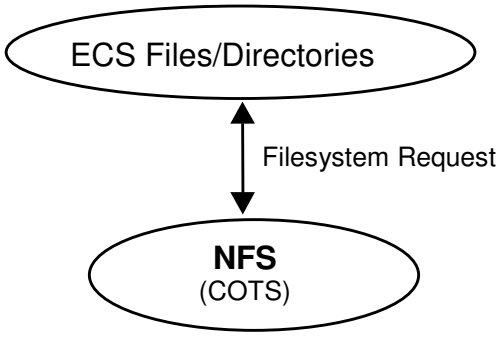

Figure 4.8.9.2-1. Network File System Context Diagram

Table 4.8.9.2-1. Network File System Interface Events

Event	Interface Event Description
Filesystem Request	The NFS clients request files or directories via an established mount point. The NFS Server makes the storage device(s) and its data accessible for use by the clients.

4.8.9.3 Network File System Architecture

Figure 4.8.9.3-1 is the Network File System architecture diagram. The diagram shows the file requests are via system calls from the Virtual File Server (VFS). The NFS client uses the XDR/RPC and networking.

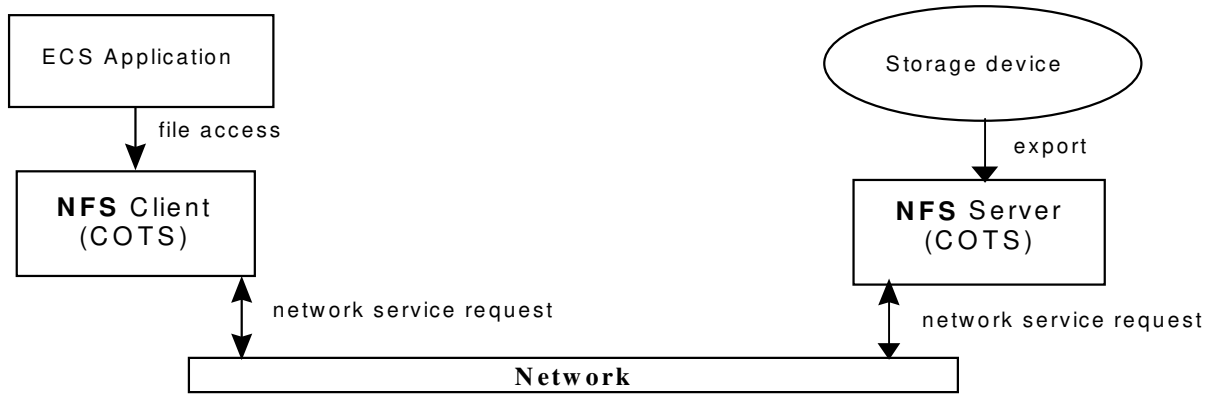

Figure 4.8.9.3-1. Network File System Architecture Diagram

4.8.9.4 Network File System Process Descriptions

Table 4.8.9.4-1 provides descriptions of the processes shown in the Network File System architecture diagram.

Table 4.8.9.4-1. Network File System Processes

Process	Type	COTS/ Developed	Functionality
NFS client	Client	COTS	The Target host providing the mounts.
NFS server	Server	COTS	The Source host exporting the data.

4.8.9.5 Network File System Process Interface Descriptions

Table 4.8.9.5-1 provides the descriptions of the interface events shown in the NFS architecture diagram.

Table 4.8.9.5-1. Network File System Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
File Access	One per file access	*COTS NFS*	*Most ECS applications*	The application uses the file accessed via the NFS client.
Export	One per server export	*COTS NFS*	*Most ECS applications*	The NFS server exports the details of storage devices to verify clients.

4.8.9.6 Network File System Data Stores

None

4.8.10 Remote File Access Group - Filecopy Description

4.8.10.1 Filecopy Functional Overview

Filecopy is the utility to copy large files from a specified source location to a specified destination location with the option of compression and decompression. The utility uses the gzip option to reduce the file size using the Lampel-Ziv coding (LZ77) technique. For Decompression, it uses the gunzip option to return the file to its original size. The EcUtCopyExec utility uses Unix read/write commands to actually copy the large file and in the event of NFS time errors, the utility retries ten times with a five-second-time delay in between retries.

4.8.10.2 Filecopy Context

The Filecopy utility is used by the STMGT and SDSRV CSCIs to copy files from the INGEST staging disk to the archive and from the archives to the Read Only Cache (RAID Disk Array). Also, the MCF Files are copied from the SDSRV to the DDIST staging Disk using Filecopy. Figure 4.8.10.2-1 is the Filecopy context diagram.

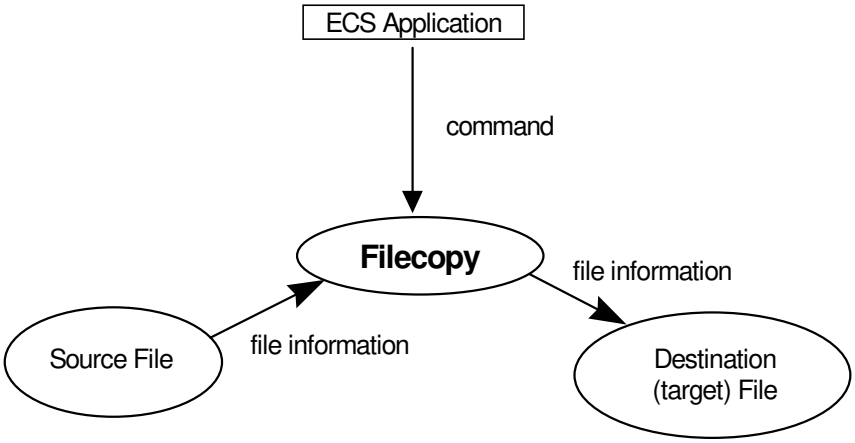

Figure 4.8.10.2-1. Filecopy Context Diagram

Table 4.8.10.2-1 provides descriptions of the interface events shown in the Filecopy context diagram.

Table 4.8.10.2-1. Filecopy Interface Events

Event	Interface Event Description
command	The ECS applications issue commands to a copy file(s) from the source to the destination.
file information	File information includes source location, destination location, and file size.

4.8.10.3 Filecopy Architecture

The Filecopy utility uses options to provide copy features of file compression, decompression, or the standard copy. Figure 4.8.10.3-1 is the Filecopy architecture diagram. The diagram shows the events sent to the FileCopy class and the events the FileCopy class sends to update the directories.

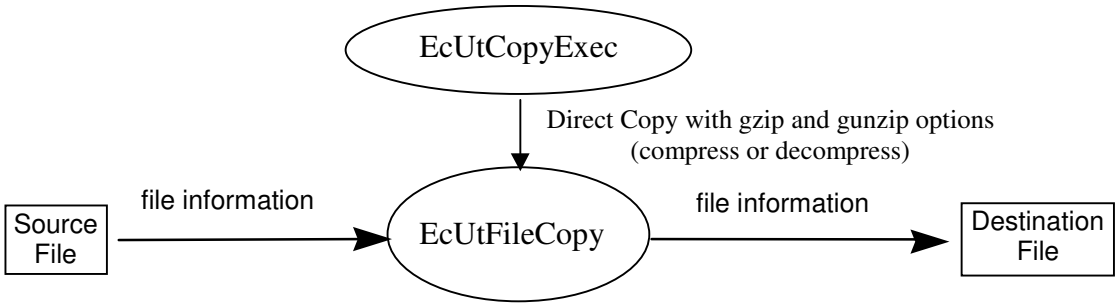

Figure 4.8.10.3-1. Filecopy Architecture Diagram

4.8.10.4 Filecopy Process Descriptions

Table 4.8.10.4-1 provides descriptions of the processes shown in the Filecopy architecture diagram. The EcUtFileCopy utility class copies files, with copy options, from one specified location to another.

Table 4.8.10.4-1. Filecopy Process

Process	Type	COTS/ Developed	Functionality
EcUtCopyExec	Utility	Developed	Used for direct copy of files with timeout checks and retry features.

4.8.10.5 Filecopy Process Interface Descriptions

Table 4.8.10.5-1 provides descriptions of the interface events shown in the FileCopy architecture diagram.

Table 4.8.10.5-1. Filecopy Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
DirectCopy	One per direct copy	*binary:* EcUtFilecopy main function	*Process:* EcUtCopyExec *Classes:* DsStArchiveReal, DsStStagingDisk, DsStStagingMonitorReal, DpPrDataManager	EcUtCopyExec is used for making direct copy. Checks for NFS timeout and retries when an error is encountered. Also checks for compression type specified with the compress/decompress option.
File Information	One per file copy	*Library:* EcUtMisc *Class:* EcUtFilecopy	*Process:* EcUtCopyExec	File information contains data file location information
gzip/gnuzip options	One per copy	*Library:* EcUtMisc *Class:* EcUtFilecopy	*Class:* EcUtFileCopy	Used for copy with compression or decompression

4.8.10.6 Filecopy Data Stores

None

4.8.11 Mail Support Group Description

4.8.11.1 E-mail and Bulletin Board Servers Functional Overview

The E-mail and Bulletin Board servers provide an interactive and a development interface for managing the electronic mail and the bulletin board functions. The interactive interface is implemented with COTS products and provides send, receive, and read message functionality. The development interfaces, or Application Programming Interfaces (APIs), are limited to sending messages.

4.8.11.2 E-mail and Bulletin Board Servers Context

Figure 4.8.11.2-1 is the E-mail and Bulletin Board Servers context diagram. Table 4.8.11.2-1 provides descriptions of the interface events shown in the E-mail and Bulletin Board Servers context diagram.

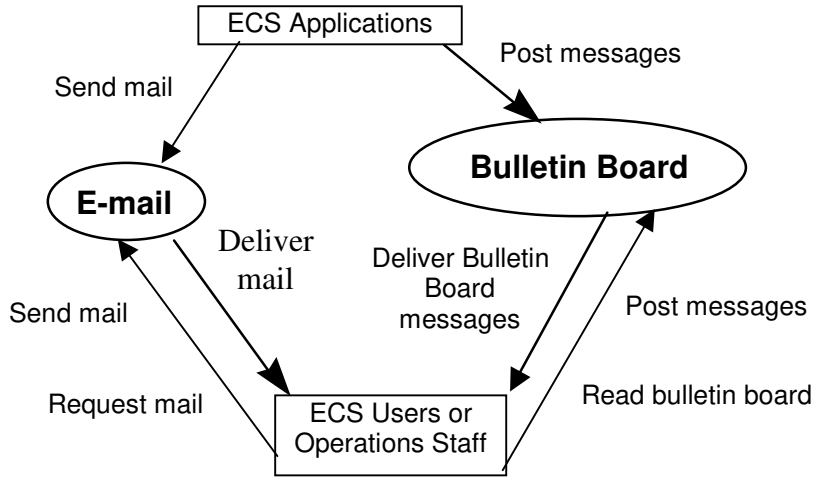

Figure 4.8.11.2-1. E-mail and Bulletin Board Servers Context Diagram

Table 4.8.11.2-1. E-mail and Bulletin Board Servers Interface Events

Event	Interface Event Description
Send mail	The ECS applications use the development interface to send mail and the API spawns a *sendmail* process to deliver the message. Interactive users, use the COTS software product, which delivers the mail message.
Post messages	The ECS applications post messages and a *nnpost* process is spawned to post the message to the bulletin board host. For interactive users, the COTS software product used posts the message.
Request mail	The COTS software product sends a request to the mail server to deliver all mail addressed to the user initiating the request.
Read bulletin board	The COTS software product sends a request to the bulletin board server for messages posted on the bulletin board
Deliver mail	The mail server delivers the mail to the addressed user.
Deliver Bulletin Board messages	The bulletin board server delivers messages posted on the bulletin board server to the requesting user.

4.8.11.3 E-mail and Bulletin Board Servers Architecture

The E-mail and bulletin board servers are COTS software products. Figure 4.8.11.3-1 is the E-mail and Bulletin Board architecture diagram.

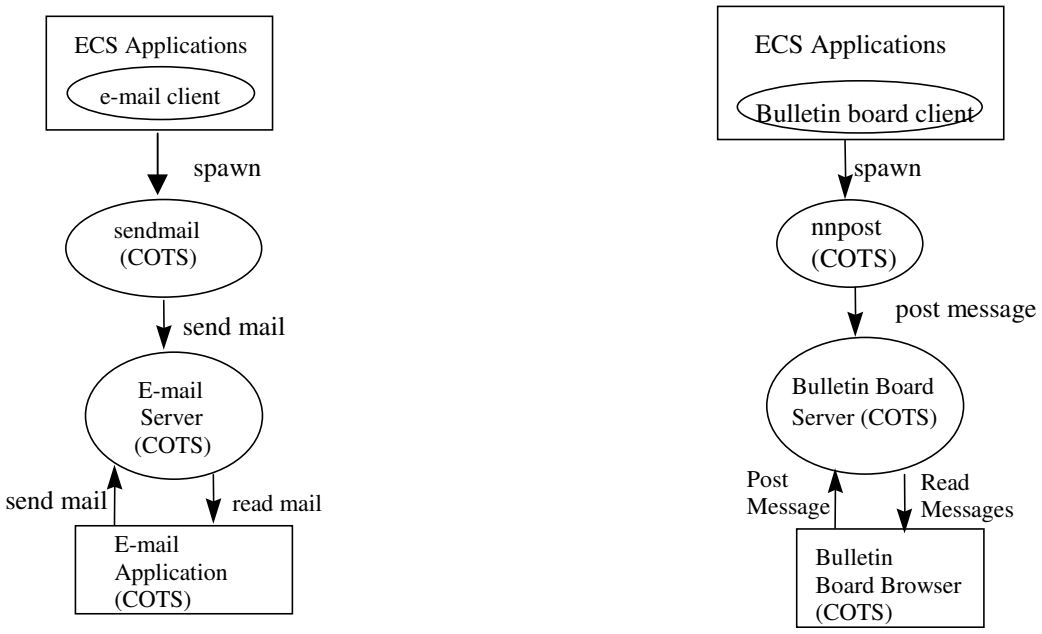

Figure 4.8.11.3-1. E-mail and Bulletin Board Servers Architecture Diagram

4.8.11.4 E-mail and Bulletin Board Servers Process Descriptions

Table 4.8.11.4-1 provides descriptions of the processes shown in the E-mail and Bulletin Board architecture diagram.

Table 4.8.11.4-1. E-mail and Bulletin Board Servers Processes (1 of 2)

Process	Type	COTS/ Developed	Functionality
E-mail Client	Other	Developed	The E-mail client is a library used by ECS applications to send electronic mail. The E-mail client provides APIs for creating E-mail messages and spawns a *sendmail* process to deliver the mail to the mail server.
Sendmail	Other	COTS	Sendmail is a COTS software product spawned by the E-mail client when E-mail is ready to send. The SMTP protocol is used to send the E-mail to the E-mail server.
E-mail Server	Server	COTS	A COTS software E-mail server product.
E-mail Application	Other	COTS	A COTS software product for sending, receiving, and reading E-mail.

Table 4.8.11.4-1. E-mail and Bulletin Board Servers Processes (2 of 2)

Process	Type	COTS/ Developed	Functionality
Bulletin Board Client	Other	Developed	The bulletin board client is a library used by ECS applications to post messages on a bulletin board. It provides APIs for creating messages and spawns the *nnpost* process to deliver the message to the bulletin board server.
nnpost	Other	COTS	A COTS software product spawned by the bulletin board client to post a message using the Network News Transfer Protocol (NNTP) on the bulletin board server.
Bulletin Board Server	Server	COTS	A COTS software product for reading and posting messages.
Bulletin Board Browser	Other	COTS	A COTS software product for searching and reading bulletin board (or newsroom) messages.

4.8.11.5 E-mail and Bulletin Board Servers Process Interface Descriptions

Table 4.8.11.5-1 provides descriptions of the interface events shown in the E-mail and Bulletin Board architecture diagram.

Table 4.8.11.5-1. E-mail and Bulletin Board Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Spawn	One per process spawned	*Library:* libc API: System ( )	*Library:* CsEmMailRelA *Class:* CsEmMailRelA CsBBMailRelA	To invoke a process to send E-mail or post a bulletin board message.
Send mail	One per E-mail send	*COTS Software:* Sendmail	*COTS Software:* Sendmail	A command from the sendmail process to send electronic mail routed via the E-mail server.
Read mail	One per e-mail read	*COTS Software:* E-mail application	*COTS Software:* E-mail Application	E-mail is read from the COTS application and routed to another user via the E-mail server.
Send mail	One per E-mail send	*COTS Software:* E-mail application	*COTS Software:* E-mail application	E-mail is received from an E-mail application via the E-mail server and sent to another user.
Post message	One per message posted	*COTS Software:* nnpost	*COTS Software:* nnpost	A command from the nnpost process to post messages on a bulletin board.

Table 4.8.11.5-1. E-mail and Bulletin Board Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Read message	One per read from bulletin board	*COTS Software:* bulletin board application	*COTS Software:* bulletin board application	An electronic message is read from a bulletin board.
Post message	One per message posted	*COTS Software:* bulletin board application	*COTS Software:* bulletin board application	An electronic message is written to a bulletin board for reading to be read by other parties.

4.8.11.6 E-mail and Bulletin Board Servers Data Stores

None

4.8.12 Virtual Terminal Description

4.8.12.1 Virtual Terminal Functional Overview

The Virtual Terminal (VT) effectively hides the terminal characteristics and data handling conventions from both the server host and Operations staff, and enables the Operations staff to remotely log on to other ECS machines. The CSS provides the kerberized telnet and the telnetd on available systems and common capability support for the ECS dial-up service.

4.8.12.2 Virtual Terminal Context

The CSS provides the kerberized telnet and the telnetd to the ECS systems. Telnet and telnetd (non-kerberized) are distributed as part of the operating system provided. The dial-up service provides users with access to the ECS character-based user interface (CHUI) search and order tool. Figure 4.8.12.2-1 is the Virtual Terminal context diagram and Table 4.8.12.2-1 provides the descriptions of the interface events shown in the Virtual Terminal context diagram.

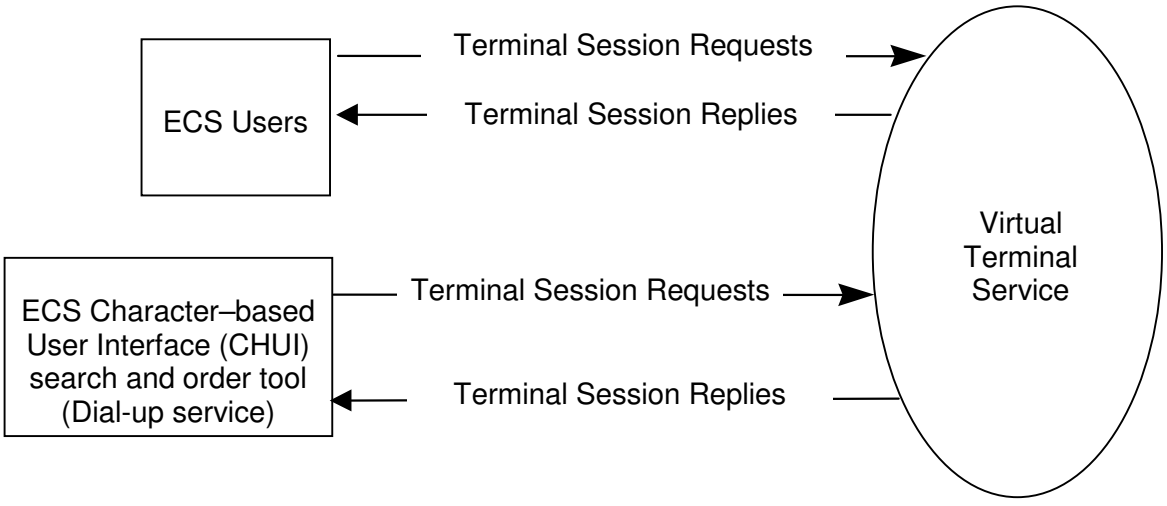

Figure 4.8.12.2-1. Virtual Terminal Context Diagram

Table 4.8.12.2-1. Virtual Terminal Interface Events

Event	Interface Event Description
Terminal Session Requests (Dial-up Users)	ECS users request a connection to a specified host via dial-up service.
Terminal Session Requests (ECS Users)	ECS users request a telnet session with a specified ECS host.
Terminal Session Replies (from the VT Server to ECS or other Dial-Up Users)	The VT Server residing on the ECS host responds to the terminal session requests and interacts via the successful connection.

4.8.12.3 Virtual Terminal Architecture

Figure 4.8.12-3-1 is the Virtual Terminal architecture diagram. The diagram shows the event traffic between the Telnet with ECS Users and Telnet with Dial-up users.

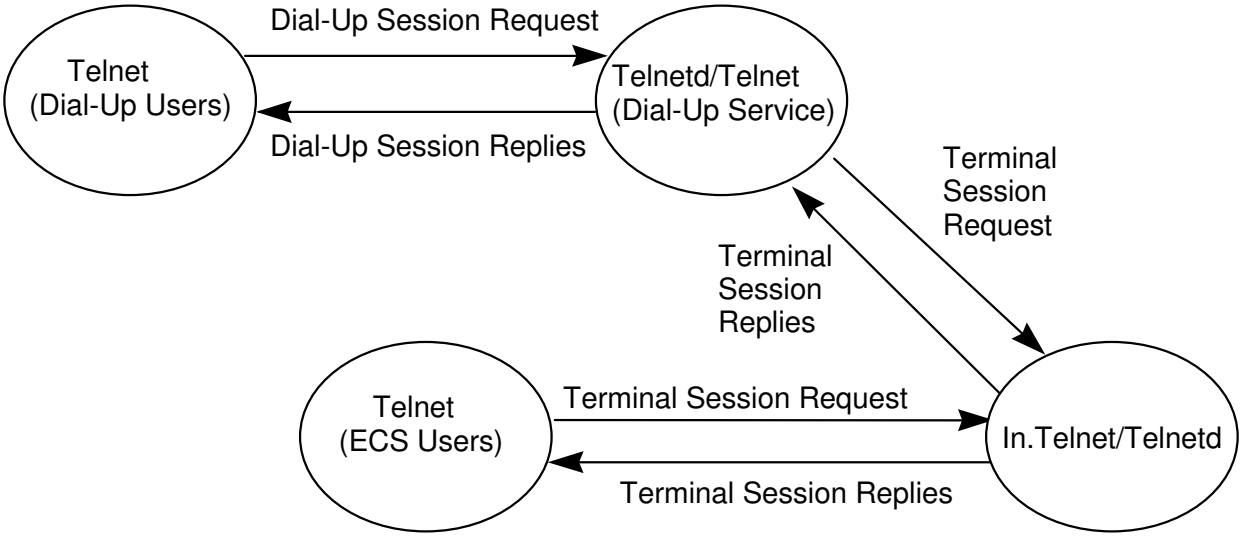

Figure 4.8.12.3-1. Virtual Terminal Architecture Diagram

4.8.12.4 Virtual Terminal Process Descriptions

Table 4.8.12.4-1 provides the descriptions of the processes shown in the Virtual terminal architecture diagram.

Table 4.8.12.4-1. Virtual Terminal Processes (1 of 2)

Process	Type	COTS/ Developed	Functionality
Telnet (Dial-up Users)	Client	COTS	Provides the dial-up terminal session as requested on the client-side via dial-up service.

Table 4.8.12.4-1. Virtual Terminal Processes (2 of 2)

Process	Type	COTS/ Developed	Functionality
Telnet (ECS Users)	Client	COTS	Provides the user interface to a remote system using the TELNET protocol.
Telnetd/Telnet (Dial-up Service)	Server/Client	COTS	Enables users to interact with the host through a dial-up service.
Telnetd or in.telnetd	Server	COTS	Function provides servers supporting TELNET with virtual terminal protocol.

4.8.12.5 Virtual Terminal Process Interface Descriptions

Table 4.8.12.5-1 provides the descriptions of the interface events shown in the Virtual Terminal architecture diagram.

Table 4.8.12.5-1. Virtual Terminal Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Dial-up Session Request	One per connection request	*COTS Cherokees Telnet from Cygnus*	Any ECS user requiring a logon to another machine from the current machine	Users request to establish connection to a specified host via dial-up.
Dial-up Session Replies	One per session reply	*COTS Kerberos Telnet from Cygnus*	Any ECS user requiring a logon to another machine from the current machine	The Dial-up service provides users a dial-up session to request a terminal session to the host's telnetd.
Terminal Session Request (Telnet kerberized request)	One per request to establish a session	*COTS Kerberos Telnet from Cygnus*	Any ECS user requiring a logon to another machine from the current machine	Either the user or dial-up service requests to establish a telnet session with the specified host.
Terminal Session Replies (Kerberos Telnetds)	One per connection request	*COTS Kerberos Telnet from Cygnus*	Any ECS user requiring a logon to another machine from the current machine	The Host Virtual Terminal Process, Kerberos telnetd, responds to the connection requests and establishes or maintains the sessions.

4.8.12.6 Virtual Terminal Data Stores

None

4.8.13 Cryptographic Management Interface Software Description

4.8.13.1 Cryptographic Management Interface Functional Overview

The Cryptographic Management Interface (CMI) classes provide the requesting process with a server account and a randomly generated password so the server can access non-DCE services (i.e., Sybase). These passwords (and optionally login names) are generated dynamically based on a psuedo-random number used as the seed for the password.

4.8.13.2 Cryptographic Management Interface Context

Figure 4.8.13.2-1 is the Cryptographic Management Interface context diagram. Servers (PF or non-PF) use the CMI with a need for access to non-DCE services. Table 4.8.13.2-1 provides descriptions of the interface events shown in the Cryptographic Management Interface context diagram.

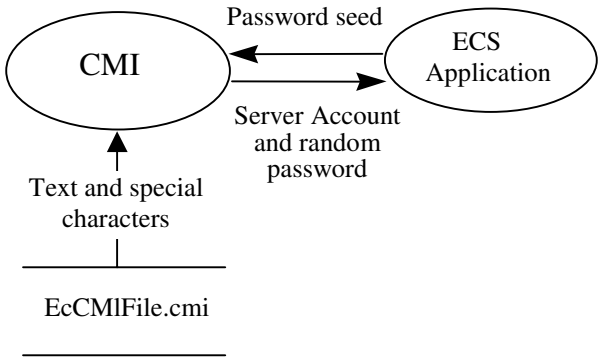

Figure 4.8.13.2-1. Cryptographic Management Interface Context Diagram

Table 4.8.13.2-1. Cryptographic Management Interface Events

Event	Interface Event Description
Password seed	The ECS application requests an account and provides a password seed to CMI.
Text and special characters	Text and special characters read from a file for password generation.
Server account and random password	Account with random passwords created for the server is passed back to the server.

4.8.13.3 Cryptographic Management Interface Architecture

The Cryptographic Management Interface architecture diagram is the same as the context diagram and is not duplicated.

4.8.13.4 Cryptographic Management Interface Process Descriptions

Table 4.8.13.4-1 provides descriptions of the processes shown in the Cryptographic Management Interface context diagram.

Table 4.8.13.4-1. Cryptographic Management Interface Processes

Process	Type	COTS/ Developed	Functionality
ECS application	Server	Developed	Requests account with random password for access to non-DCE services.
CMI	Other	Developed	A server account and randomly generated password are returned to the requesting server.

4.8.13.5 Cryptographic Management Interface Process Interface Descriptions

Table 4.8.13.5-1 provides the descriptions of the interface events shown in the Cryptographic Management Interface architecture diagram.

Table 4.8.13.5-1. Cryptographic Management Interface Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Password seed	One per password seed	*Library:* EcSeCmi *Class:* EcSeCmi	*Any server program*	The server provides a unique number as a seed for generating a password.
Server Account and random password	One per account and password	*Library:* EcSeCmi *Class:* EcSeCmi	*Any server program*	CMI generates a random password for the account based on the seed.

4.8.13.6 Cryptographic Management Interface Data Stores

Table 4.8.13.6-1 provides descriptions of the data store shown in the Cryptographic Management Interface architecture diagram.

Table 4.8.13.6-1. Cryptographic Management Interface Data Stores

Data Store	Type	Functionality
EcCMIFile.cmi	Other	This is a flat file of textual and special characters used by the CMI password generation algorithm to create passwords.

4.8.14 Domain Name Server Software Description

4.8.14.1 Domain Name Server Functional Overview

Domain Name Server (DNS) performs name-to-address and address-to-name resolution between hosts within the local administrative domain and across domain boundaries. DNS is COTS software implemented as server by running a daemon called "in.named." Servers running the in.named daemon are referred to as name servers.

The server is implemented through a resolver instead of a daemon from the client side. The function of in.named is to resolve user queries for device names or addresses (DNS requires the address of at least one name server to be in the file /etc/resolv.conf). The name server, when queried for a name or an address, returns the answer to the query or a referral to another name server to query for the answers.

Each domain uses at least two kinds of DNS servers (primary and secondary) to maintain the name and address data corresponding to the domain. The primary server keeps the master copy of the data when it starts up in the "in.named," daemon and delegates authority to other servers both inside and outside of its domain. A secondary server maintains a copy of the name and address data for the domain. When secondary server boots in.named, it requests the data for a given domain from the primary server. The secondary server then checks with the primary server periodically and requests updates to the daemon data so the secondary server is kept up to date with the primary.

DNS namespace is hierarchically organized, with nested domains, like directories. The DNS namespace consists of a tree of domains. Figure 4.8.14.1-1 is an Internet domain hierarchy diagram. The top-level domains are edu, arpa, com, gov, net, and for simplicity, not showing org, mil, and int, at the root level. The second level domain is nasa for gov. The third level domain is ecs for the ECS project for nasa.gov.

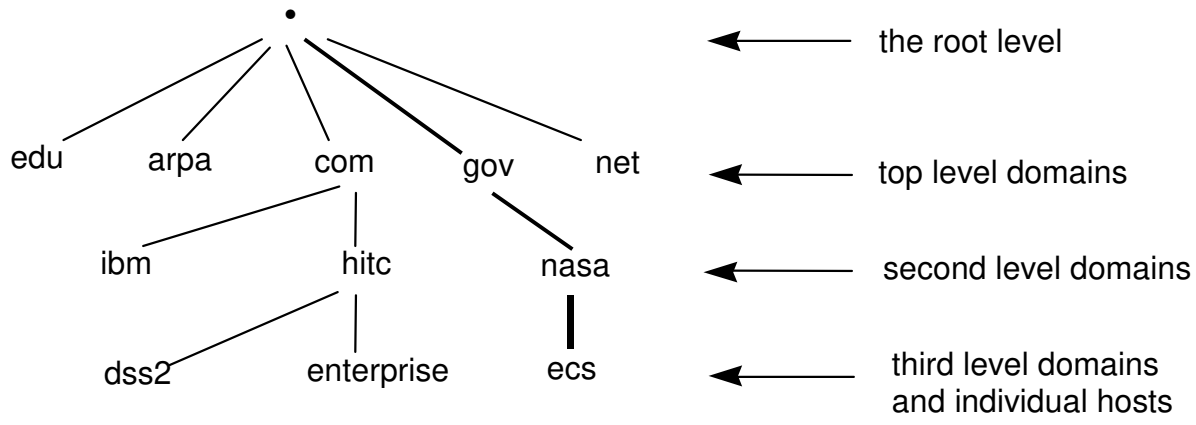

Figure 4.8.14.1-1. Domains Hierarchy Diagram

The fourth level domains in the ECS project include domains of DAACs: gsfcb, gsfcmo, and etc. Figure 4.8.14.1-2 is the hierarchy diagram of the fourth level domains in the ECS project. The

DAAC and M&O domains are part of the overall DNS. The top-level domain is ecs.nasa.gov and the two lower level domains for the DAACs, for example, gsfcb.ecs.nasa.gov and gsfcmo.ecs.nasa.gov for the GSFC DAAC. The former is for the Version 2. 0-production network and the latter are for the GSFC M&O network.

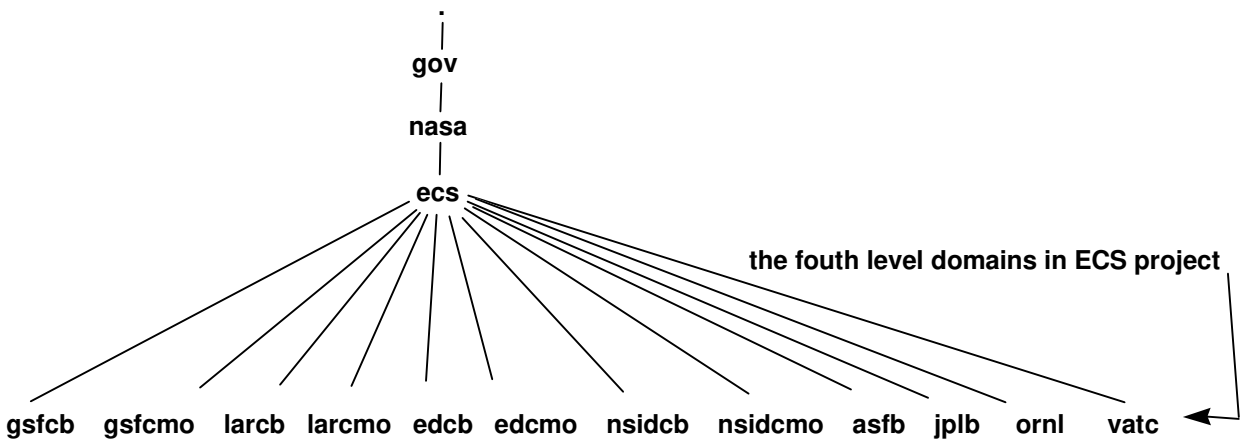

Figure 4.8.14.1-2. DNS Domains of the ECS Project Diagram

Figure 4.8.14.1-3 is the ECS topology domain diagram.

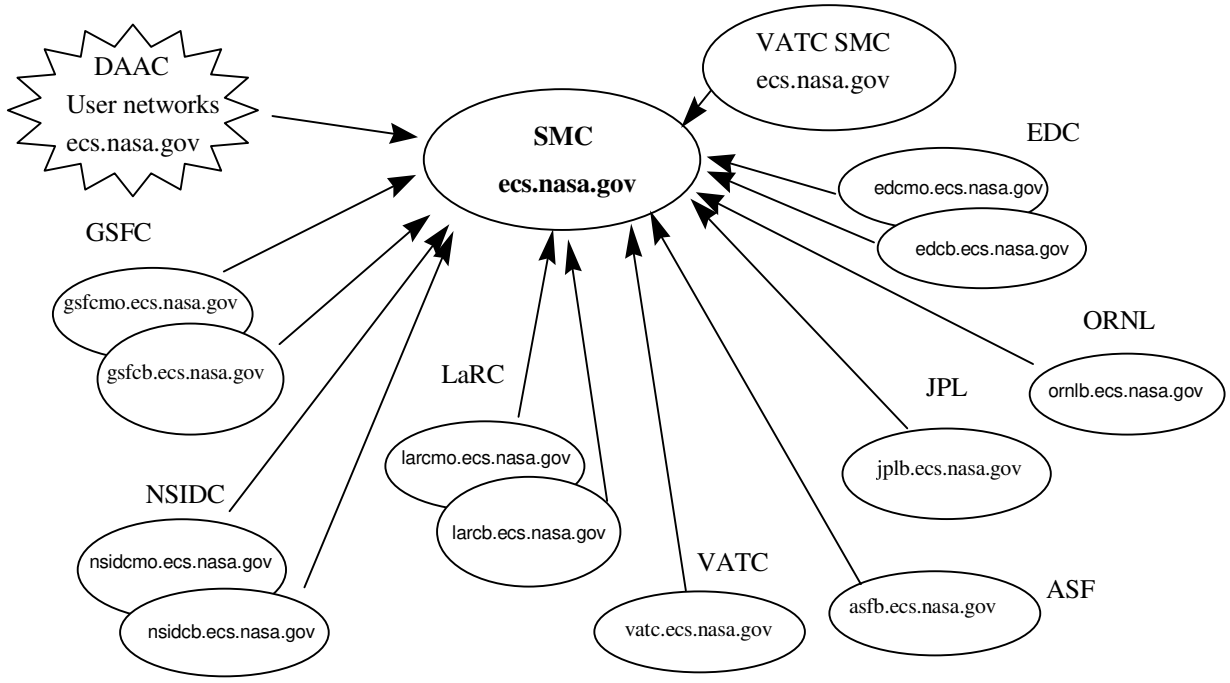

Figure 4.8.14.1-3. ECS Topology Domains Diagram

4.8.14.2 Domain Name Server Context

Figure 4.8.14.2-1 is the Domain Name Server context diagram.

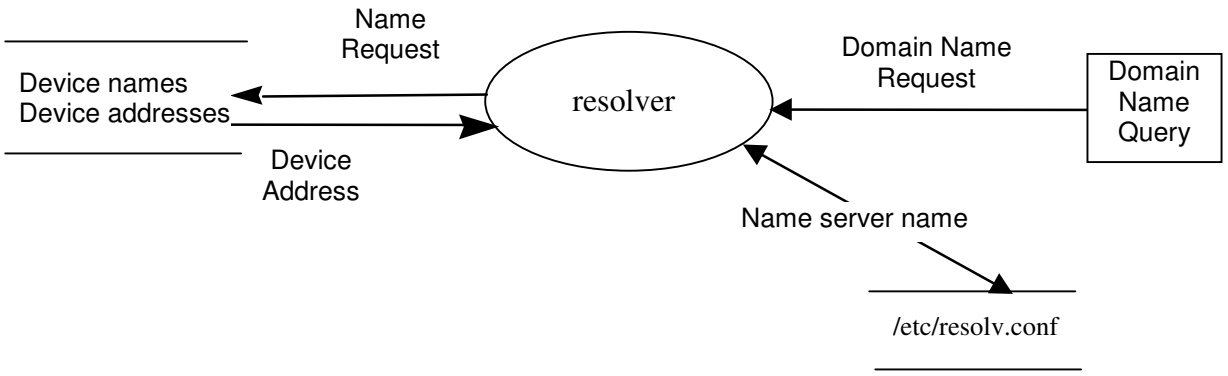

Figure 4.8.14.2-1. Domain Name Server Context Diagram

4.8.14.3 Domain Name Server Architecture

The Domain Name Server architecture diagram is the same as the context diagram and is not duplicated here. When the DNS client has a request for data, it searches the servers listed in the /etc/resolv.conf file in the order the servers were added to the file. When the first server does not contain the information of interest for the client, the second server in the list is searched and the search continues until the information is found.

4.8.14.4 Domain Name Server Process Descriptions

Table 4.8.14.4-1 provides descriptions of the Domain Name Server processes shown in the Domain Name Server context diagram.

Table 4.8.14.4-1. Domain Name Server Processes

Process	Type	COTS/ Developed	Functionality
resolver	Client	COTS	Searches data store of device names and devices addresses for information requested in the Domain Name Request. First entry in the file /etc/resolv.conf is used as the place to start searching.

4.8.14.5 Domain Name Server Process Interface Descriptions

Table 4.8.14.5-1 provides descriptions of the interface events shown in the Domain Name Server architecture diagram.

Table 4.8.14.5-1. Domain Name Process Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Name Request	One per domain name request	*COTS Software:* name server	*COTS Software:* resolver	Requests the domain name from the name server
Device Address	One per resolved address	*COTS Software:* resolver	*COTS Software:* name server	Returns the resolved address to the domain name requester
Domain Name Request	One per user request	*User*	*User*	A DNS user requests data
Name server name	One per search directory change	*COTS Software:* resolver	*COTS Software:* resolver	Pathname for the directory to search for the user requested data is read from /etc/resolv.conf. New file names are added to the list in the order they were stored.

4.8.14.6 Domain Name Server Data Stores

Table 4.8.14.6-1 provides descriptions of the data store shown in the Domain Name Server architecture diagram.

Table 4.8.14.6-1. Domain Name Server Data Stores

Data Store	Type	Functionality
/etc/resolv.conf	Other	Stores the primary and secondary server names.

4.8.15 Infrastructure Libraries Group Description

4.8.15.1 Infrastructure Libraries Group Functional Overview

The Infrastructure Library Group (ILG) is a library of reusable software frameworks and infrastructures used by ECS servers configured as a distributed client-server system. Table 4.8.15.1-1 provides descriptions of the infrastructures in the ILG.

Table 4.8.15.1-1. Infrastructure Libraries (1 of 2)

Library	Description
Process Framework (PF)	Provides an extensible mechanism for transparent incorporation of features, from a library of infrastructure features, such as mode management, error and event logging, life-cycle services, and OODCE directory and security services into ECS application processes.
Server Request Framework (SRF)	Provides enhancements to OODCE RPC functionality by providing synchronous and asynchronous message passing capabilities and with persistent storage on request.

Table 4.8.15.1-1. Infrastructure Libraries (2 of 2)

Library	Description
Universal References (UR)	Provides a mechanism, usable system wide, to uniquely identify the address space of ECS data and service objects (locally and remote).
Error/Event Logging	Provides the mechanism for logging application errors and system events for MSS application monitoring.
Message Passing (MP)	Provides peer-to-peer synchronous and asynchronous communications with store, forward, and persistence features.
ServerUR	Provides unique (Universal Reference) identification for data and service objects in ECS.
Fault Handling (FH)	Provides fault recovery capabilities by enabling clients to reconnect with a server after a prior connection is lost.
DBWrapper directory	The DBWrapper directory is the DBMS Interface Infrastructure Library. Sybase SQL servers operating under the DBMS, operate by ECS defined guidelines for mode management, thread safety, error handling, error recovery, security, configuration management, and performance of database connections.

4.8.15.2 Infrastructure Libraries Group Context

A context diagram is not applicable to the Infrastructure Libraries Group.

4.8.15.3 Infrastructure Libraries Group Architecture

An architecture diagram is not applicable to the Infrastructure Libraries Group.

4.8.15.4 Infrastructure Libraries Group Process Descriptions

Descriptions of the individual processes in the Infrastructure Libraries Group are not applicable.

4.8.15.5 Infrastructure Libraries Group Interface Descriptions

Table 4.8.15.5-1 provides descriptions of the interfaces the Infrastructure Libraries Group.

Table 4.8.15.5-1. Infrastructure Library Group Interfaces (1 of 2)

Library	Interface	Initiated by	Library Description
Process Framework (PF)	*Library:* EcPf *Class:* EcPfManagedSever, EcPfclient	Any application intending to be a server or client	Provides flexible mechanism (encapsulation) for ECS client and server applications to transparently include ECS infrastructure features from a library of services. The PF process is the encapsulation of an object with ECS infrastructure features and therefore the encapsulated object is fully equipped with the attributes needed to perform the activities assigned. Features and services include: • Mode management • Error and event logging • Life-cycle services • OODCE directory and security services

Table 4.8.15.5-1. Infrastructure Library Group Interfaces (2 of 2)

Library	Interface	Initiated by	Library Description
Server Request Framework (SRF)	*Library:* Srf *Class:* EcSrRequestServer_C, EcSrAsynchRequest_C	Any application intending to use asynchronous messaging capabilities	Provides enhancements to OODCE RPC functionality by providing a synchronous and asynchronous message passing capability. Message requests can be persistently stored.
Universal References (UR)	*Library*: EcUr	Object Origination	Provides a system wide unique identification for ECS data and service objects.
Error/Event Logging	*Library:* Event *Class:* EcLgErrormsg	Any application requiring logging	Provides a mechanism for logging application errors and system events for MSS application monitoring.
Message Passing (MP)	*Library:* EcDcMsgPsng1	Any application intending to use asynchronous/ synchronous messaging capabilities	Provides a peer-to-peer synchronous and asynchronous communications with store, forward, and persistence storage.
ServerUR	*Library:* EcUr *Class:* EcUrServerUR	*Processes:* EcDmDdMaintenanceTool *Classes:* EcClWbDtDART EcNsServiceloc, EcGwMojoGateway, EcSbGui, EcCiSubscriptionclient, EcSbSubscriptionRserver, *DSS Libraries:* DsBt, DdsDe1, DsGe	Provides a mechanism for a server resource to be uniquely identified in the ECS.
Fault Handling (FH)	*Library: EcFh Class: EcFhExecutor*	*All ECS applications that instantiate client proxies and make RPCs to servers*	Provides fault recovery capabilities by enabling clients to reconnect with servers after losing a prior connection.
DBWrapper directory	*Library:* EcPoDbRW, *EcPoDb Class:* EcPoConnectionsRW	STMGT, IOS, DDICT, LIM, ECSToV0Gateway, PDPS	This is the DBMS Interface Infrastructure Library. Sybase SQL servers operating under the DBMS, implements ECS defined guidelines for mode management, thread safety, error handling, error recovery, security, configuration, and performance of database connections.

4.8.16.1 Communications Subsystem Hardware CI Description

The Communications Subsystem CI is a SUN SPARC 20/712 CSS Server workstation with an external disk. Detailed specifications can be found per the site-specific hardware design diagram, baseline document number 920-TDx-001. Three DCCI software programs run on this host including the DCE Master, Domain Name Server (DNS) and Mail Server. The DCE Master consists of name, security and timeservers. The name service enables distributed applications the capability to associate information with names. The DCE Time Service (DTS) provides synchronization of all system clocks. The security service provides secure transfer of data over the network. DNS enables host names to be distinguished based on their host name and IP address relationship. The Mail Server provides standard electronic mail capability.

Detailed mappings can be found for the site-specific hardware/software is in baseline document number 920-TDx-002.

A SUN standard multi-pack is used for external storage of all components described above. A detailed configuration is specified per disk partition in baseline document number 922-TDx-005.

Other hosts and various hardware configuration items are used by the Communications Subsystem DCCI.

The Subscription Server (SBSRV), ASTER DAR Communications Gateway, ASTER E-Mail Parser Gateway and Landsat 7 Gateway run on the DMS, INTHW HWCI, Interface Server pair. (Detail specifications can be found per the site-specific hardware design diagram, baseline document number 920-TDx-001.) The SBSRV detects previously defined events. The ASTER DAR Communications Gateway provides interoperability between the DAR Client GUI tool and the DAR API. The ASTER E-Mail Parser Gateway supports automated delivery of ASTER Expedited Data Sets. Finally, the Landsat 7 Gateway provides ECS users the capability to access Landsat 7 satellite data.

A Bulletin Board Service is available at the SMC DAAC only and supported by the Bulletin Board Server. Bulletin Board messages are sent to member in a specific group.

The SMC provides two Sun servers, one primary and one (cold) secondary, to receive data from external (non-EBnet) data providers. These servers, SPARC 20/50 class systems with 18 GB of external storage, are the IGS FTP Servers.

Because the CSS software runs on multiple hosts, hardware fail-over depends on the application host.

The Subscription Server (SBSRV), ASTER DAR Communications Gateway, ASTER E-mail Parser Gateway and Landsat 7 Gateway are stored to local disk on the DMS Interface Server pair. One of the hosts is designated as the primary server as specified per the DAAC Site Host Mapping, baseline document number 910-TDA-005. In the event of failure to the primary host, a new session is initiated on the secondary host.

The CSS Server is a stand-alone host and intrinsically does not have fail-over capability. The DCE Master is replicated on a separate host to enable DCE operations to continue. DNS and DTS are loaded on multiple hosts designated as secondary. Any one of these hosts can operate as

primary servers for the DNS or DTS services in the event of non-recoverable hardware failure of the primary host.

The MSS File Server RAID is dual ported between the MSS File Server and the CM Server. In the event of a MSS File Server failure, the CM Server assumes total ownership of the RAID and all processes.

Specific primary and secondary host designations are specified per the DAAC Site Host Mapping, baseline document number 910-TDA-005.

The Bulletin Board Server is considered a non-critical function and does not have fail-over capability.